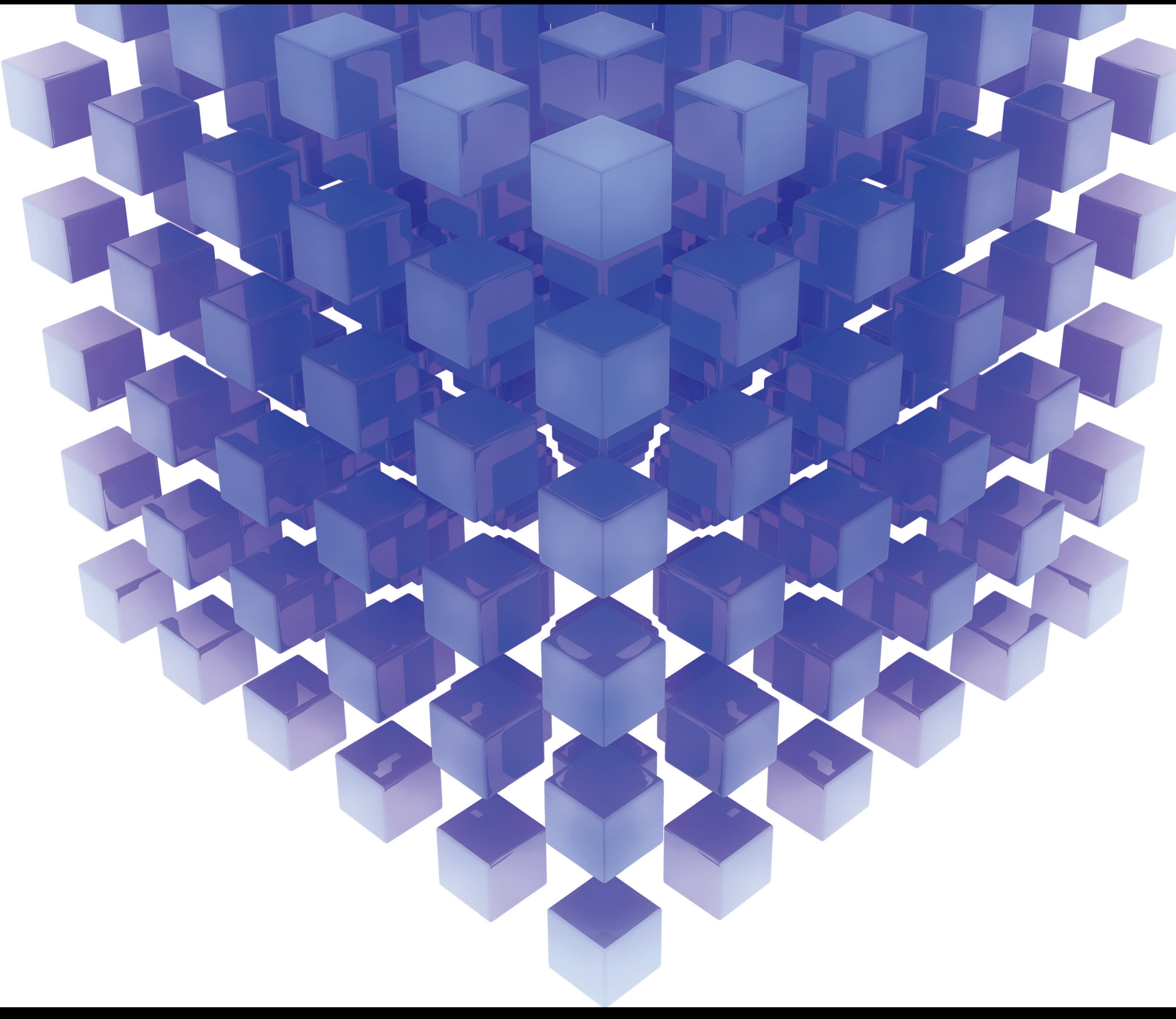
Mathematical Problems in Engineering

Recent Trends in Advanced Robotic Systems

Lead Guest Editor: Bhawani Shankar Chowdhry
Guest Editors: Zain Anwar Ali and Dao B. Wang



Recent Trends in Advanced Robotic Systems

Mathematical Problems in Engineering

Recent Trends in Advanced Robotic Systems

Lead Guest Editor: Bhawani Shankar Chowdhry
Guest Editors: Zain Anwar Ali and Dao B. Wang

Chief Editor

Academic Editors

José António Fonseca De Oliveira Correia, Portugal
Jose Renato De Sousa, Brazil
Michael Defoort, France
Alessandro Della Corte, Italy
Laurent Dewasme, Belgium
Sanku Dey, India
Gianpaolo Di Bona, Italy
Roberta Di Pace, Italy
Francesca Di Puccio, Italy
Ramón I. Diego, Spain
Yannis Dimakopoulos, Greece
Hasan Dinçer, Turkey
José M. Domínguez, Spain
Georgios Dounias, Greece
Bo Du, China
Emil Dumic, Croatia
Madalina Dumitriu, United Kingdom
Premraj Durairaj, India
Saeed Eftekhar Azam, USA
Said El Kafhali, Morocco
Antonio Elipe, Spain
R. Emre Erkmen, Canada
John Escobar, Colombia
Leandro F. F. Miguel, Brazil
FRANCESCO FOTI, Italy
Andrea L. Facci, Italy
Shahla Faisal, Pakistan
Giovanni Falsone, Italy
Hua Fan, China
Jianguang Fang, Australia
Nicholas Fantuzzi, Italy
Muhammad Shahid Farid, Pakistan
Hamed Faroqi, Iran
Yann Favennec, France
Fiorenzo A. Fazzolari, United Kingdom
Giuseppe Fedele, Italy
Roberto Fedele, Italy
Baowei Feng, China
Mohammad Ferdows, Bangladesh
Arturo J. Fernández, Spain
Jesus M. Fernandez Oro, Spain
Francesco Ferrise, Italy
Eric Feulvarch, France
Thierry Floquet, France
Eric Florentin, France
Gerardo Flores, Mexico
Antonio Forcina, Italy
Alessandro Formisano, Italy
Francesco Franco, Italy
Elisa Francomano, Italy
Juan Frausto-Solis, Mexico
Shujun Fu, China
Juan C. G. Prada, Spain
HECTOR GOMEZ, Chile
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, USA
Jaime Gallardo-Alvarado, Mexico
Mosè Gallo, Italy
Akemi Gálvez, Spain
Maria L. Gandarias, Spain
Hao Gao, Hong Kong
Xingbao Gao, China
Yan Gao, China
Zhiwei Gao, United Kingdom
Giovanni Garcea, Italy
José García, Chile
Harish Garg, India
Alessandro Gasparetto, Italy
Stylianos Georgantzinos, Greece
Fotios Georgiades, India
Parviz Ghadimi, Iran
Ştefan Cristian Gherghina, Romania
Georgios I. Giannopoulos, Greece
Agathoklis Giaralis, United Kingdom
Anna M. Gil-Lafuente, Spain
Ivan Giorgio, Italy
Gaetano Giunta, Luxembourg
Jefferson L.M.A. Gomes, United Kingdom
Emilio Gómez-Déniz, Spain
Antonio M. Gonçalves de Lima, Brazil
Qunxi Gong, China
Chris Goodrich, USA
Rama S. R. Gorla, USA
Veena Goswami, India
Xunjie Gou, Spain
Jakub Grabski, Poland

Antoine Grall, France
George A. Gravvanis, Greece
Fabrizio Greco, Italy
David Greiner, Spain
Jason Gu, Canada
Federico Guarracino, Italy
Michele Guida, Italy
Muhammet Gul, Turkey
Dong-Sheng Guo, China
Hu Guo, China
Zhaoxia Guo, China
Yusuf Gurefe, Turkey
Salim HEDDAM, Algeria
ABID HUSSANAN, China
Quang Phuc Ha, Australia
Li Haitao, China
Petr Hájek, Czech Republic
Mohamed Hamdy, Egypt
Muhammad Hamid, United Kingdom
Renke Han, United Kingdom
Weimin Han, USA
Xingsi Han, China
Zhen-Lai Han, China
Thomas Hanne, Switzerland
Xinan Hao, China
Mohammad A. Hariri-Ardebili, USA
Khalid Hattaf, Morocco
Defeng He, China
Xiao-Qiao He, China
Yanchao He, China
Yu-Ling He, China
Ramdane Hedjar, Saudi Arabia
Jude Hemanth, India
Reza Hemmati, Iran
Nicolae Herisanu, Romania
Alfredo G. Hernández-Diaz, Spain
M.I. Herreros, Spain
Eckhard Hitzer, Japan
Paul Honeine, France
Jaromir Horacek, Czech Republic
Lei Hou, China
Yingkun Hou, China
Yu-Chen Hu, Taiwan
Yunfeng Hu, China
Can Huang, China
Gordon Huang, Canada
Linsheng Huo, China
Sajid Hussain, Canada
Asier Ibeas, Spain
Orest V. Iftime, The Netherlands
Przemyslaw Ignaciuk, Poland
Giacomo Innocenti, Italy
Emilio Insfran Pelozo, Spain
Azeem Irshad, Pakistan
Alessio Ishizaka, France
Benjamin Ivorra, Spain
Breno Jacob, Brazil
Reema Jain, India
Tushar Jain, India
Amin Jajarmi, Iran
Chiranjibe Jana, India
Łukasz Jankowski, Poland
Samuel N. Jator, USA
Juan Carlos Jáuregui-Correa, Mexico
Kandasamy Jayakrishna, India
Reza Jazar, Australia
Khalide Jbilou, France
Isabel S. Jesus, Portugal
Chao Ji, China
Qing-Chao Jiang, China
Peng-fei Jiao, China
Ricardo Fabricio Escobar Jiménez, Mexico
Emilio Jiménez Macías, Spain
Maolin Jin, Republic of Korea
Zhuo Jin, Australia
Ramash Kumar K, India
BHABEN KALITA, USA
MOHAMMAD REZA KHEDMATI, Iran
Viacheslav Kalashnikov, Mexico
Mathiyalagan Kalidass, India
Tamas Kalmar-Nagy, Hungary
Rajesh Kaluri, India
Jyotheeswara Reddy Kalvakurthi, India
Zhao Kang, China
Ramani Kannan, Malaysia
Tomasz Kapitaniak, Poland
Julius Kaplunov, United Kingdom
Konstantinos Karamanos, Belgium
Michal Kawulok, Poland

Irfan Kaymaz, Turkey
Vahid Kayvanfar, Qatar
Krzysztof Kecik, Poland
Mohamed Khader, Egypt
Chaudry M. Khalique, South Africa
Mukhtaj Khan, Pakistan
Shahid Khan, Pakistan
Nam-Il Kim, Republic of Korea
Philipp V. Kiryukhantsev-Korneev, Russia
P.V.V Kishore, India
Jan Koci, Czech Republic
Ioannis Kostavelis, Greece
Sotiris B. Kotsiantis, Greece
Frederic Kratz, France
Vamsi Krishna, India
Edyta Kucharska, Poland
Krzysztof S. Kulpa, Poland
Kamal Kumar, India
Prof. Ashwani Kumar, India
Michal Kunicki, Poland
Cedrick A. K. Kwuimy, USA
Kyandoghere Kyamakya, Austria
Ivan Kyrchei, Ukraine
Márcio J. Lacerda, Brazil
Eduardo Lalla, The Netherlands
Giovanni Lancioni, Italy
Jaroslaw Latalski, Poland
Hervé Laurent, France
Agostino Lauria, Italy
Aimé Lay-Ekuakille, Italy
Nicolas J. Leconte, France
Kun-Chou Lee, Taiwan
Dimitri Lefebvre, France
Eric Lefevre, France
Marek Lefik, Poland
Yaguo Lei, China
Kauko Leiviskä, Finland
Ervin Lenzi, Brazil
ChenFeng Li, China
Jian Li, USA
Jun Li, China
Yueyang Li, China
Zhao Li, China
Zhen Li, China
En-Qiang Lin, USA
Jian Lin, China
Qibin Lin, China
Yao-Jin Lin, China
Zhiyun Lin, China
Bin Liu, China
Bo Liu, China
Heng Liu, China
Jianxu Liu, Thailand
Lei Liu, China
Sixin Liu, China
Wanquan Liu, China
Yu Liu, China
Yuanchang Liu, United Kingdom
Bonifacio Llamazares, Spain
Alessandro Lo Schiavo, Italy
Jean Jacques Loiseau, France
Francesco Lolli, Italy
Paolo Lonetti, Italy
António M. Lopes, Portugal
Sebastian López, Spain
Luis M. López-Ochoa, Spain
Vassilios C. Loukopoulos, Greece
Gabriele Maria Lozito, Italy
Zhiguo Luo, China
Gabriel Luque, Spain
Valentin Lychagin, Norway
YUE MEI, China
Junwei Ma, China
Xuanlong Ma, China
Antonio Madeo, Italy
Alessandro Magnani, Belgium
Toqeer Mahmood, Pakistan
Fazal M. Mahomed, South Africa
Arunava Majumder, India
Sarfraz Nawaz Malik, Pakistan
Paolo Manfredi, Italy
Adnan Maqsood, Pakistan
Muazzam Maqsood, Pakistan
Giuseppe Carlo Marano, Italy
Damijan Markovic, France
Filipe J. Marques, Portugal
Luca Martinelli, Italy
Denizar Cruz Martins, Brazil

George A. Papakostas, Greece
Xosé M. Pardo, Spain
You-Jin Park, Taiwan
Manuel Pastor, Spain
Pubudu N. Pathirana, Australia
Surajit Kumar Paul, India
Luis Payá, Spain
Igor Pažanin, Croatia
Libor Pekař, Czech Republic
Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Jian Peng, China
Mingshu Peng, China
Xiang Peng, China
Xindong Peng, China
Yuexing Peng, China
Marzio Pennisi, Italy
Maria Patrizia Pera, Italy
Matjaz Perc, Slovenia
A. M. Bastos Pereira, Portugal
Wesley Peres, Brazil
F. Javier Pérez-Pinal, Mexico
Michele Perrella, Italy
Francesco Pesavento, Italy
Francesco Petrini, Italy
Hoang Vu Phan, Republic of Korea
Lukasz Pieczonka, Poland
Dario Piga, Switzerland
Marco Pizzarelli, Italy
Javier Plaza, Spain
Goutam Pohit, India
Dragan Poljak, Croatia
Jorge Pomares, Spain
Hiram Ponce, Mexico
Sébastien Poncet, Canada
Volodymyr Ponomaryov, Mexico
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Sivakumar Poruran, India
Francesc Pozo, Spain
Aditya Rio Prabowo, Indonesia
Anchasa Pramuanjaroenkij, Thailand
Leonardo Primavera, Italy
B Rajanarayan Prusty, India
Krzysztof Puszynski, Poland
Chuan Qin, China
Dongdong Qin, China
Jianlong Qiu, China
Giuseppe Quaranta, Italy
DR. RITU RAJ, India
Vitomir Racic, Italy
Carlo Rainieri, Italy
Kumbakonam Ramamani Rajagopal, USA
Ali Ramazani, USA
Angel Manuel Ramos, Spain
Higinio Ramos, Spain
Muhammad Afzal Rana, Pakistan
Muhammad Rashid, Saudi Arabia
Manoj Rastogi, India
Alessandro Rasulo, Italy
S.S. Ravindran, USA
Abdolrahman Razani, Iran
Alessandro Reali, Italy
Jose A. Reinoso, Spain
Oscar Reinoso, Spain
Haijun Ren, China
Carlo Renno, Italy
Fabrizio Renno, Italy
Shahram Rezapour, Iran
Ricardo Riaza, Spain
Francesco Riganti-Fulginei, Italy
Gerasimos Rigatos, Greece
Francesco Ripamonti, Italy
Jorge Rivera, Mexico
Eugenio Roanes-Lozano, Spain
Ana Maria A. C. Rocha, Portugal
Luigi Rodino, Italy
Francisco Rodríguez, Spain
Rosana Rodríguez López, Spain
Francisco Rossomando, Argentina
Jose de Jesus Rubio, Mexico
Weiguo Rui, China
Rubén Ruiz, Spain
Ivan D. Rukhlenko, Australia
Dr. Eswaramoorthi S., India
Weichao SHI, United Kingdom
Chaman Lal Sabharwal, USA
Andrés Sáez, Spain

Bekir Sahin, Turkey
Laxminarayan Sahoo, India
John S. Sakellariou, Greece
Michael Sakellariou, Greece
Salvatore Salamone, USA
Jose Vicente Salcedo, Spain
Alejandro Salcido, Mexico
Alejandro Salcido, Mexico
Nunzio Salerno, Italy
Rohit Salgotra, India
Miguel A. Salido, Spain
Sinan Salih, Iraq
Alessandro Salvini, Italy
Abdus Samad, India
Sovan Samanta, India
Nikolaos Samaras, Greece
Ramon Sancibrian, Spain
Giuseppe Sanfilippo, Italy
Omar-Jacobo Santos, Mexico
J Santos-Reyes, Mexico
José A. Sanz-Herrera, Spain
Musavarah Sarwar, Pakistan
Shahzad Sarwar, Saudi Arabia
Marcelo A. Savi, Brazil
Andrey V. Savkin, Australia
Tadeusz Sawik, Poland
Roberta Sburlati, Italy
Gustavo Scaglia, Argentina
Thomas Schuster, Germany
Hamid M. Sedighi, Iran
Mijanur Rahaman Seikh, India
Tapan Senapati, China
Lotfi Senhadji, France
Junwon Seo, USA
Michele Serpilli, Italy
Silvestar Šesnić, Croatia
Gerardo Severino, Italy
Ruben Sevilla, United Kingdom
Stefano Sfarra, Italy
Dr. Ismail Shah, Pakistan
Leonid Shaikhet, Israel
Vimal Shanmuganathan, India
Prayas Sharma, India
Bo Shen, Germany
Hang Shen, China
Xin Pu Shen, China
Dimitri O. Shepelsky, Ukraine
Jian Shi, China
Amin Shokrollahi, Australia
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, Canada
Angelo Sifaleras, Greece
Nuno Simões, Portugal
Mehakpreet Singh, Ireland
Piyush Pratap Singh, India
Rajiv Singh, India
Seralathan Sivamani, India
S. Sivasankaran, Malaysia
Christos H. Skiadas, Greece
Konstantina Skouri, Greece
Neale R. Smith, Mexico
Bogdan Smolka, Poland
Delfim Soares Jr., Brazil
Alba Sofi, Italy
Francesco Soldovieri, Italy
Raffaele Solimene, Italy
Yang Song, Norway
Jussi Sopanen, Finland
Marco Spadini, Italy
Paolo Spagnolo, Italy
Ruben Specogna, Italy
Vasilios Spitas, Greece
Ivanka Stamova, USA
Rafał Stanisławski, Poland
Miladin Stefanović, Serbia
Salvatore Strano, Italy
Yakov Strelniker, Israel
Kangkang Sun, China
Qiuqin Sun, China
Shuaishuai Sun, Australia
Yanchao Sun, China
Zong-Yao Sun, China
Kumarasamy Suresh, India
Sergey A. Suslov, Australia
D.L. Suthar, Ethiopia
D.L. Suthar, Ethiopia
Andrzej Swierniak, Poland
Andras Szekrenyes, Hungary
Kumar K. Tamma, USA

Yong (Aaron) Tan, United Kingdom
Marco Antonio Taneco-Hernández, Mexico
Lu Tang, China
Tianyou Tao, China
Hafez Tari, USA
Alessandro Tasora, Italy
Sergio Teggi, Italy
Adriana del Carmen Téllez-Anguiano, Mexico
Ana C. Teodoro, Portugal
Efstathios E. Theotokoglou, Greece
Jing-Feng Tian, China
Alexander Timokha, Norway
Stefania Tomasiello, Italy
Gisella Tomasini, Italy
Isabella Torcicollo, Italy
Francesco Tornabene, Italy
Mariano Torrisi, Italy
Thang nguyen Trung, Vietnam
George Tsiatas, Greece
Le Anh Tuan, Vietnam
Nerio Tullini, Italy
Emilio Turco, Italy
Ilhan Tuzcu, USA
Efstratios Tzirtzilakis, Greece
FRANCISCO UREÑA, Spain
Filippo Ubertini, Italy
Mohammad Uddin, Australia
Mohammad Safi Ullah, Bangladesh
Serdar Ulubeyli, Turkey
Mati Ur Rahman, Pakistan
Panayiotis Vafeas, Greece
Giuseppe Vairo, Italy
Jesus Valdez-Resendiz, Mexico
Eusebio Valero, Spain
Stefano Valvano, Italy
Carlos-Renato Vázquez, Mexico
Martin Velasco Villa, Mexico
Franck J. Vernerey, USA
Georgios Veronis, USA
Vincenzo Vespri, Italy
Renato Vidoni, Italy
Venkatesh Vijayaraghavan, Australia
Anna Vila, Spain
Francisco R. Villatoro, Spain
Francesca Vipiana, Italy
Stanislav Vítek, Czech Republic
Jan Vorel, Czech Republic
Michael Vynnycky, Sweden
Mohammad W. Alomari, Jordan
Roman Wan-Wendner, Austria
Bingchang Wang, China
C. H. Wang, Taiwan
Dagang Wang, China
Guoqiang Wang, China
Huaiyu Wang, China
Hui Wang, China
J.G. Wang, China
Ji Wang, China
Kang-Jia Wang, China
Lei Wang, China
Qiang Wang, China
Qingling Wang, China
Weiwei Wang, China
Xinyu Wang, China
Yong Wang, China
Yung-Chung Wang, Taiwan
Zhenbo Wang, USA
Zhibo Wang, China
Waldemar T. Wójcik, Poland
Chi Wu, Australia
Qiuhong Wu, China
Yuqiang Wu, China
Zhibin Wu, China
Zhizheng Wu, China
Michalis Xenos, Greece
Hao Xiao, China
Xiao Ping Xie, China
Qingzheng Xu, China
Binghan Xue, China
Yi Xue, China
Joseph J. Yame, France
Chuanliang Yan, China
Xinggang Yan, United Kingdom
Hongtai Yang, China
Jixiang Yang, China
Mijia Yang, USA
Ray-Yeng Yang, Taiwan

Zaoli Yang, China
Jun Ye, China
Min Ye, China
Luis J. Yebra, Spain
Peng-Yeng Yin, Taiwan
Muhammad Haroon Yousaf, Pakistan
Yuan Yuan, United Kingdom
Qin Yuming, China
Elena Zaitseva, Slovakia
Arkadiusz Zak, Poland
Mohammad Zakwan, India
Ernesto Zambrano-Serrano, Mexico
Francesco Zammori, Italy
Jessica Zangari, Italy
Rafal Zdunek, Poland
Ibrahim Zeid, USA
Nianyin Zeng, China
Junyong Zhai, China
Hao Zhang, China
Haopeng Zhang, USA
Jian Zhang, China
Kai Zhang, China
Lingfan Zhang, China
Mingjie Zhang, Norway
Qian Zhang, China
Tianwei Zhang, China
Tongqian Zhang, China
Wenyu Zhang, China
Xianming Zhang, Australia
Xuping Zhang, Denmark
Yinyan Zhang, China
Yifan Zhao, United Kingdom
Debao Zhou, USA
Heng Zhou, China
Jian G. Zhou, United Kingdom
Junyong Zhou, China
Xueqian Zhou, United Kingdom
Zhe Zhou, China
Wu-Le Zhu, China
Gaetano Zizzo, Italy
Mingcheng Zuo, China

Contents

Hindawi
Mathematical Problems in Engineering
Volume 2023, Article ID 9825281, 1 page
https://doi.org/10.1155/2023/9825281

Retraction

Retracted: Internet of Things (IoT)-Enabled Unmanned Aerial Vehicles for the Inspection of Construction Sites: A Vision and Future Directions

Mathematical Problems in Engineering

Received 13 September 2023; Accepted 13 September 2023; Published 14 September 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. Israr, G. E. M. Abro, M. Sadiq Ali Khan, M. Farhan, and S. u. A. Bin Mohd Zulkifli, "Internet of Things (IoT)-Enabled Unmanned Aerial Vehicles for the Inspection of Construction Sites: A Vision and Future Directions," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9931112, 15 pages, 2021.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9912574, 13 pages
https://doi.org/10.1155/2021/9912574

Research Article

Real-Time Air-to-Ground Data Communication Technology of Aeroengine Health Management System with Adaptive Rate in the Whole Airspace

Qiuying Yan,[1] Wei Li,[1] Jiacheng Li,[2] Jie Zhang,[2] Shengyi Liu,[2] Zhe Wang,[2] Tong Liu,[2] Qian Chen,[2] and Hanlin Sheng[2]

[1]*AECC Hunan Aviation Powerplant Research Institute, Zhuzhou, Hunan 412002, China*
[2]*Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China*

Correspondence should be addressed to Wei Li; buaa608@sina.com and Hanlin Sheng; dreamshl@qq.com

Received 22 March 2021; Accepted 26 June 2021; Published 31 July 2021

Academic Editor: Zain Anwar Ali

To overcome the problem of data transmission of the aeroengine health management system, a multilink communication system combining ultrahigh-frequency communication link, 4G cellular mobile communication link, and BeiDou satellite communication link was proposed. This system can realize the functions such as data receiving and sending, data encryption, and resuming transfer from the break point based on multiple links. When the flight altitude is not high, the communication distance is short, so the UHF digital transmission radio communication link is adopted, which is highly efficient and stable. When the communication distance is long, the 4G cellular mobile communication link can ensure both the communication distance and the communication rate. In the area where 4G signal cannot be covered in extreme terrain environment, BeiDou satellite communication link is used for data transmission. Besides, in order to ensure the communication rate of the link, a multilink adaptive switching technology was also developed. The test verified that the system can perform adaptive switching among multiple links, realize air-ground data communication in the whole airspace, and achieve a good communication rate, which has significative value of engineering application.

1. Introduction

With the gradual expansion of domestic aviation platforms and the rapid growth of the transportation industry, the requirements for real-time, safe, and reliable air-to-ground communication technology have continuously improved. The air-to-ground data link system for engine health management, as the main method of real-time monitoring of engine status, has been extensively applied in civil aviation owing to its fast transmission, strong anti-interference ability, low bit error rate, and high reliability [1, 2].

The failure strategy of engine gas circuit performance monitoring, fault diagnosis, and response is an important part of engine health management [3, 4]. In the aviation industry, flight accidents are often caused by engine failures. Therefore, it is extremely important to strengthen real-time monitoring of engine operating status. Although health management technology is the most effective method to ensure the safe operation of the aeroengine and reduce maintenance costs, the timeliness and accuracy of data sources used by it will directly affect the quality of health assessment. In terms of algorithm theory research on aeroengine health management, typical algorithms include methods based on physical models [5, 6], methods based on data [7, 8], methods based on experience, and fusion methods based on intelligent algorithms [9–11].

During the flight, the air-to-ground data link system realizes real-time monitoring of engine operating status by continuously monitoring the operating status of the aircraft, sending the on-board information to the ground terminal in real time, and using the powerful ground network control processing center to analyze and process it, so as to provide constructive decision-making reference for crew members, facilitate the preparation for maintenance before the aircraft landing, shorten maintenance time, and guarantee efficient engine operation [12, 13]. The acquisition of potential engine failure information is the prerequisite for enhancing maintenance efficiency, saving costs, and guaranteeing flight safety. Therefore, it is particularly essential to study the communication technology of air-to-ground data on health management of aeroengine [14, 15].

In the 1940s, the communication in civil aviation mainly relied on the very high-frequency (VHF) communication system [16], which was gradually used as a global aviation mobile service; thus, the VHF communication in aviation has been fully realized [17, 18]. However, in recent years, with the gradual opening of low-altitude airspace, it has become a hot trend to use aerial vehicle for low-altitude flight. Restricted by the flying height and small speed of aircraft, high-frequency aviation communication is difficult to be effectively covered at low altitude, leading to insufficient communication safety guarantee for low-altitude flight. The 4G cellular mobile communication system is suitable for small low-flying aerial vehicles with small flight speed; coupled with a large number of existing base stations and the low construction cost, it can provide a certain degree of communication guarantee [19, 20]. However, mobile base stations cannot be fully covered in remote areas such as the sea, deserts, mountainous areas, and forests, which restricts the development of network communication to a large extent. Therefore, the satellite communication system with long communication distance, wide working frequency, stable and reliable communication, and high communication quality has been widely applied in the field of aviation communication [21, 22].

To meet the requirements for air-to-ground data communication of the aeroengine health management system in the whole airspace, a communication technology that integrates three air-to-ground data links was proposed in this paper, which can realize communication in different flight scenarios. Besides, based on the respective advantages of the three data links, a multilink adaptive switching technology was developed to ensure the quality of air-to-ground data communication. In order to ensure the integrity and security of the data in the process of data transfer, the technologies of resuming transfer from the break point and data encryption were used, which enhanced the accuracy and reliability of the whole airspace air-to-ground data communication of the aeroengine health management system.

In this paper, the first part mainly introduces the real-time communication scheme of air-to-ground data on health management and studies related key technologies to lay a good foundation for the research path of system design, mainly including air-ground segment, and ground-ground segment real-time communication technology, key information encryption technology, key information resuming transfer from the break point technology, and the technology of air-to-ground data communication with adaptative rate. The second part mainly describes the construction of a prototype system for real-time communication of air-to-ground data of health management, which mainly includes the construction of a two-way transmission architecture for air-ground segment, ground-ground segment data, and the implementation of each component of the prototype system for real-time air-to-ground data communication. The third part mainly presents a simulated flight test on real-time communication and transmission of air-to-ground data of health management and verifies the communication of these three communication links and the adaptive switching technology of the system. It was confirmed that the system can realize the adaptive communication among the three links and achieve a good communication rate.

2. The Scheme of Real-Time Air-to-Ground Data Communication of Health Management

2.1. Research on Real-Time Air-Ground Segment, Ground-Ground Segment, and Relaying Communication Technologies. The health management system of aeroengine mainly monitors the operating status of aircraft engine and sends relevant health management data in real time to the ground control center, which, after processing and analysis, distributes it to relevant departments to take corresponding measures [23]. This system involves the communication of two processes, namely, the air-ground segment and ground-ground segment. Therefore, the scheme of air-to-ground data communication of health management requires separate research on these two processes.

2.1.1. Research on Air-Ground Segment Communication Technology. At present, China has formed an air-to-ground communication system dominated by VHF data communication and assisted by satellite communication. In this context, the air-ground segment communication of the health management system for aeroengine needs to meet the requirements for communication in different flight airspace. First, within a limited distance, the communication of the ultrahigh-frequency (UHF) digital radio station link is the safest and fastest. Characterized by fast data update and small communication delay, it can provide reliable air-to-ground communication. Under nonhorizon communication, the 4G cellular mobile communication link is significantly superior to the communication of UHF digital radio station, with the advantages of mature technology, high rate, and small delay. However, 4G communication is restricted in remote areas where base stations have weak air-to-air signals or where it is difficult to deploy base stations. In such case, the BeiDou satellite communication link can give play to its own advantages. The short message function, which is the unique communication function of the BeiDou satellite, can achieve reliable communication in remote areas such as the sea, the fields, and the mountainous areas thanks to its advantages, such as wide coverage and strong anti-interference ability [24].

Considering the applicable scenarios and transmission range of different links, a combination of three data links, UHF data transfer communication, 4G cellular mobile communication, and BeiDou satellite communication, was adopted in the system, so as to make full use of the advantages of the three communication links to ensure the communication of the entire communication system in the whole airspace. Figure 1 shows the air-ground segment data link of the health management system.

2.1.2. Research on Ground-Ground Segment Communication Technology. During the health data management of the aeroengine, the ground control center needs to process and transmit the obtained health data to multiple departments. To realize the transmission of data to multiple departments, ground-ground segment communication needs to meet the requirements for one-to-many or many-to-many interactive communication, so as to realize data sharing.

To allow the health management data to be accurately and timely distributed to other departments and to realize the sharing of engine's health management data, after research and tests, the ground-ground segment adopted the user datagram protocol (UDP) network communication protocol for communication.

UDP network communication protocol is a connectionless transport layer protocol, with the advantages such as small resource consumption, fast processing, and high real-time performance. In addition, it supports one-to-many and many-to-many interactive communication, which can achieve the function of transmitting health management data to different departments. Figure 2 exhibits the ground-ground segment data link of the health management system.

2.2. Research on the Technology of Compression and Encryption of Key Health Management Information. As the "heart" of the aircraft, the aeroengine represents the highest level of industrial technology of a country and is hailed as the "Pearl" on the "Crown" of industry. Thus, its health management data are evidently vital. In the communication system for air-to-ground data of health management, the data need to be processed to prevent it from being stolen, attacked, and destroyed. Before transmitted, the data can be encrypted to reduce threats and ensure data security to the greatest extent.

The security encryption algorithms for information transmission have been extensively studied, and in this system, DES security encryption algorithm was used. DES is a world-recognized standard encryption format, which performs encryption by 64-bit data packets before replacement and iteration [25, 26]. The length of the key is 56 bits, and processing such as replacement and cyclic shift is also required. In the initial IP replacement, the 64-bit plaintext is disorganized according to an $8 * 8$ replacement table to increase the strength of encryption; subkey is used in 16 iterative computations; the initial ciphertext formed by each packet undergoes the inverse initial replacement again, which is the inverse operation of the initial replacement, to obtain the final ciphertext of the packets. The specific process is displayed in Figure 3.

2.3. Research on the Technology of Resuming Transfer from the Break Point of Key Health Management Information. Resuming transfer from the break point means that, during the data upload or download process, when the connection is reestablished after an interruption, the part that has never been uploaded or downloaded will continue to be transferred without repeating the transmission of the existing content. This technology can effectively save time and improve the efficiency of the system [27, 28].

There are two key issues that need to be paid attention to in resuming transfer from the break point:

(1) When the transmission is interrupted, the local download information is saved: a temporary file needs to be created to save the local download information, including the number of downloaded bytes, the file pointer, and the total file size, so that the download information will be stored in a temporary file when the transmission is interrupted, and the file pointer will point to the end of the downloaded file.

(2) The local download information needs to be checked when resuming the transfer from the break point: first, it is necessary to determine whether the downloaded file exists locally; if it exists, the download information of the downloaded file is read, and the pointer is moved to the next position of the downloaded bytes at the same time so that the transfer of the file can be resumed from the break point.

Figure 4 shows the flowchart of resuming transfer from the break point.

2.4. Research on the Technology of Air-to-Ground Data Communication of Health Management with Adaptive Rate. Different communication links are applicable to different working scenarios, working ranges, and transmission distances. If only one communication link is used for communication, the communication effect will vary among different scenarios where the aircraft performs tasks. The application of multiple communication links can not only take advantage of each communication link to ensure communication quality but also improve the survivability of the entire communication system. Therefore, it is necessary to use multiple data links for communication.

When multiple communication links exist in the system, how to coordinate these communication links to allow them to work efficiently is a key issue that needs to be solved. Therefore, an adaptive switching mechanism based on three communication links was investigated. Considering the differences in the transmission range and update frequency among UHF digital radio station, 4G cellular mobile communication, and BeiDou satellite communication, the default priority of the communication links (from high to low) was set as follows: UHF digital radio station, 4G cellular mobile communication, and BeiDou satellite communication. Even if the information of several communication links is received simultaneously, the ground control center will only process and distribute the data of the communication link with the highest priority currently. This mechanism conducts analysis according to the packet loss and transmission delay to judge the communication quality of the

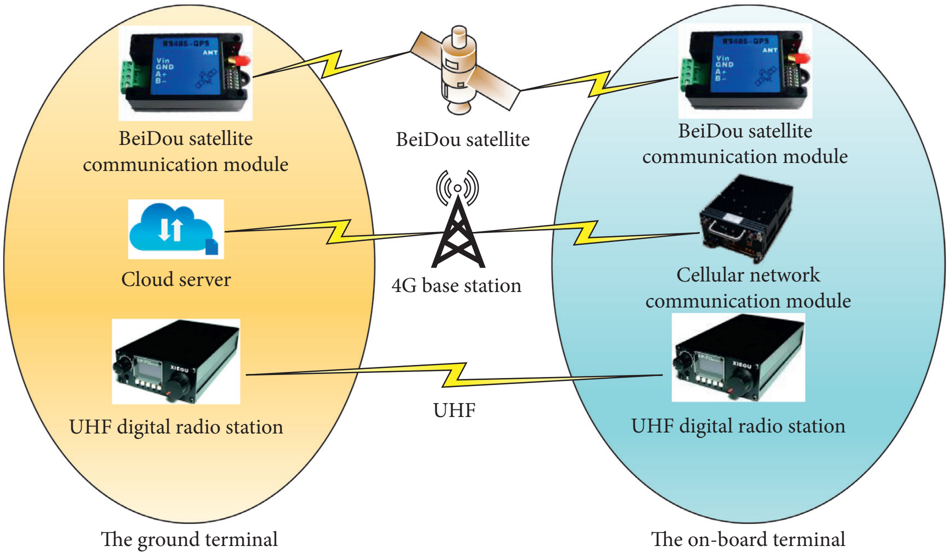

BeiDou satellite communication module
BeiDou satellite
BeiDou satellite communication module
Cloud server
4G base station
Cellular network communication module
UHF digital radio station
UHF
UHF digital radio station
The ground terminal
The on-board terminal

Figure 1: Diagram of air-ground segment data link.

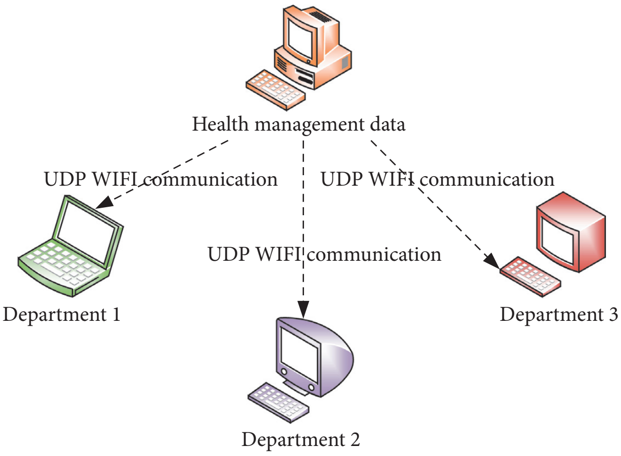


Figure 2: Diagram of the ground-ground segment data link.

existing communication link. If the communication quality deteriorates and the optimal data cannot be provided, it will switch to other communication links for communication according to the priority. For example, when the transmission delay is greater than 500 ms or the packet loss rate is greater than 1%, the communication condition is considered bad, and the communication link needs to be switched, as shown in Figure 5.

3. Construction of a Prototype System for Real-Time Communication of Air-to-Ground Health Management Data

3.1. Construction of a Two-Way Data Transfer Architecture for Air-Ground Segment and Ground-Ground Segment Data. The main cross-linking relationship in the prototype system for real-time communication of air-to-ground data of health management is displayed in Figure 6. The on-board flight control system transmits data with the engine simulation equipment through serial communication, and the engine simulation equipment and the on-board terminal of data link realize data interaction through serial communication. The ground terminal of the data link realizes information interaction with the ground control center through the serial port, and the ground control center conducts information interaction with the outfield portable maintenance aids (PMA), the simulation equipment of the maintenance center, and the support center through the network. The on-board terminal and the ground terminal of data link communicate through three links: UHF digital radio station, 4G cellular network communication module, and BeiDou satellite communication module, and switching is performed among the three communication methods through an adaptive switching mechanism to select an optimal communication method, so as to ensure reliable and stable communication.

The flight control system sends data such as flight status information to the engine simulation equipment, which calculates the telemetry data of the simulated engine according to the received parameters such as the amount of the throttle lever of the flight control system, and transmits it to the on-board terminal of data link. The data link on-board terminal then frames the received telemetry data from the flight control system and the engine, as well as the status data from the on-board terminal of data link, and then transmits the data to the data link ground terminal through the cellular network communication module, BeiDou satellite communication, and UHF digital radio station after the data are encrypted, encoded, and modulated. The ground RF front end of the data link ground terminal converts the received radio frequency signal before transmitting it to the ground terminal of data link, which transfers the data to the ground control center after processing such as demodulation, decoding, and decryption. The ground control center analyzes and reframes the received data and then distributes it to the simulation equipment of the outfield PMA terminal, maintenance center, and support center through network communication.

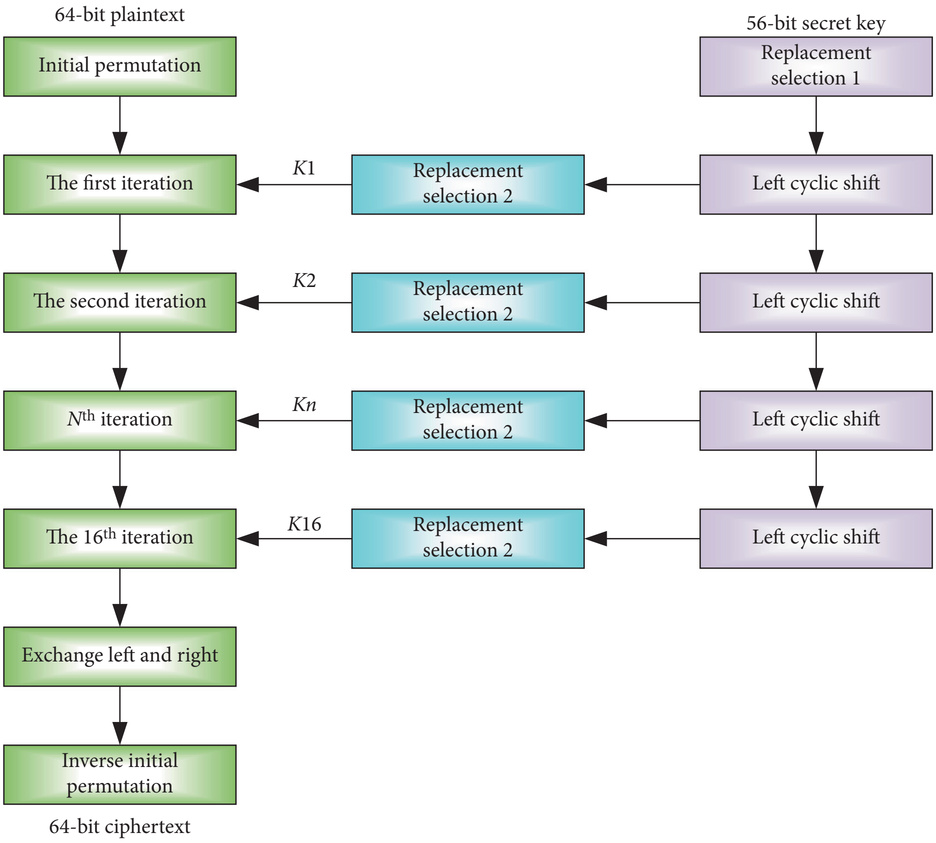


Figure 3: Encryption process of DES algorithm.

3.2. Implementation of Each Component of the Prototype System for Real-Time Air-to-Ground Data Communication of Health Management. The real-time communication system of air-to-ground data of health management is mainly composed of the on-board subsystem and the ground subsystem. The two subsystems are connected by the on-board terminal and ground terminal of data link.

3.2.1. On-Board Subsystem. The on-board subsystem is mainly composed of on-board data link terminal, engine simulation equipment, and flight control system.

(1) Engine simulation equipment: the STM32F405 hardware platform with a basic frequency of 168 MHz was used for simulated operation, as shown in Figure 7, so that information such as flight status can be obtained from the on-board flight control system. Then, the engine nonlinear on-board real-time model was adopted to calculate the operating data of the simulated engine; finally, these data were sent to the data link on-board terminal, with the remote control command received from the on-board terminal of data link simultaneously.

(2) The on-board terminal of data link: the UHF communication module is shown in Figure 8, 4G communication module is shown in Figure 9, and BeiDou module is shown in Figure 10; the three data link on-board terminals are, respectively, connected to the data link comprehensive integrated unit via the serial port, the data link integrated unit is a single-chip microcomputer based on STM32L476RGT6, which mainly completes the functions of channel encoding and decoding, data packets deframing, data dispreading, modulation and demodulation, subsystem built-in test, and fault diagnosis and processing of the data link on-board terminal and then realize data communication with the engine simulation equipment and the ground terminal of data link, as shown in Figure 11, in which USART1 is connected to the 4G core board; USART2 is connected to the RS232 female port; UART4 is connected to the RDSS module; UART5 is connected to the UHF data transfer module; LPUART1 is connected to the engine simulation module.

(3) Flight control system: to ensure the success and safety of the flight test, this system uses the open-source flight control board Pixhawk 2.1, as shown in Figure 12. On this basis, the secondary development was carried out, and a self-defined protocol was used for communication.

3.2.2. Ground Subsystem. The ground subsystem consists of data link ground terminal, ground control center, and the simulation equipment of outfield PMA, support center, and maintenance center, as shown in Figure 13.

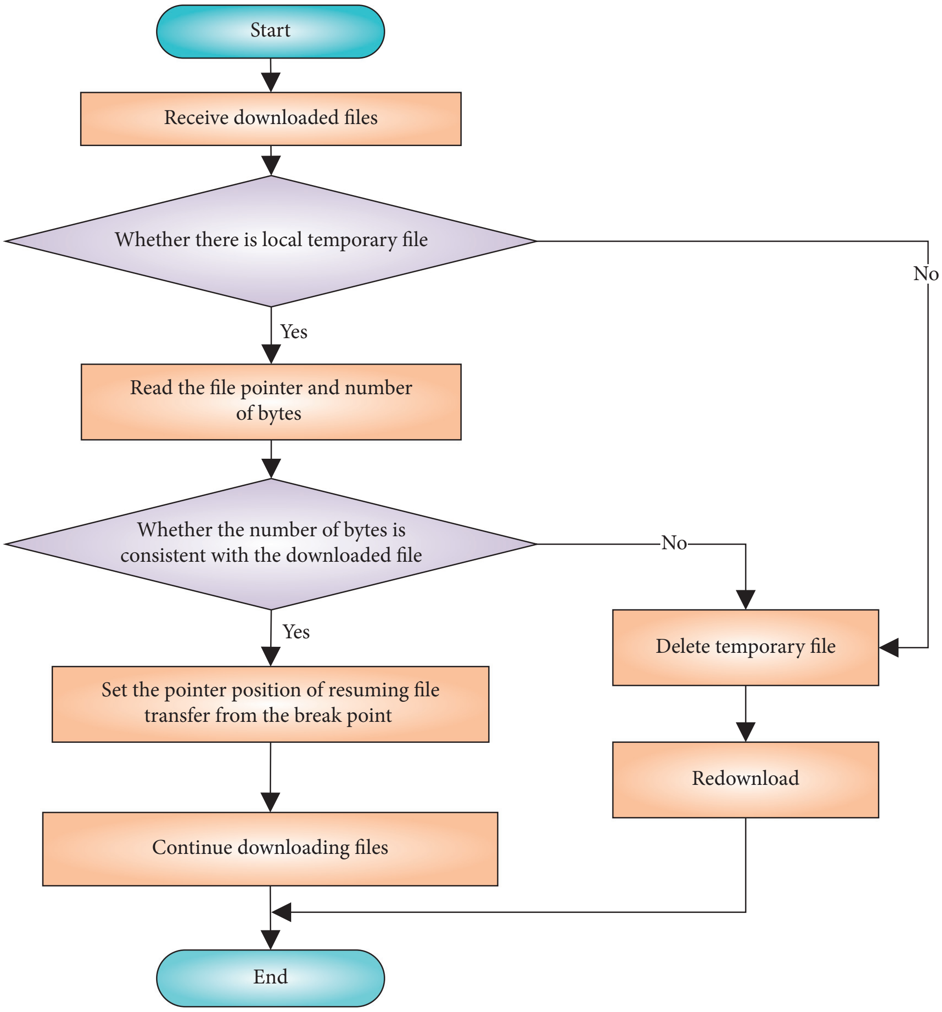

Start
Receive downloaded files
Whether there is local temporary file
No
Yes
Read the file pointer and number of bytes
Whether the number of bytes is consistent with the downloaded file
No
Yes
Set the pointer position of resuming file transfer from the break point
Delete temporary file
Redownload
Continue downloading files
End

Figure 4: Flowchart of resuming transmission from the break point.

(1) Data link ground terminal: it mainly completes the functions of channel encoding and decoding, data packets deframing, data dispreading, modulation and demodulation, subsystem built-in test, and fault diagnosis and processing of the data link ground terminal and realizes communication with the ground control center and data link on-board terminal.

(2) Ground control center: it mainly completes the analysis of data from the ground terminal of the data link, displays the data in the form of visual charts, and sends the user's operation command to the ground terminal of data link.

(3) The simulation equipment of outfield PMA, support center, and maintenance center: it completes the communication with the ground control center and receives the data from ground control center for analysis and processing.

4. Simulated Flight Test on Real-Time Communication and Transmission of Health Management Air-Ground Data

4.1. Method of the Engine's Health Management Data Simulation. Due to the high cost and difficulty of collecting the health management data of real engine, interpolation was used to build a high-confidence real-time on-board engine model to simulate and calculate the working data of the engine. The parameters of each section of the current engine were obtained through the interpolation of the throttle lever command given by the flight control, and then the data of the engine were sent to the on-board terminal of data link through the serial port.

4.2. Establishment of a Flight Test Platform for Real-Time Communication and Transmission of Air-to-Ground Data. Owing to the high cost and great risk of using aircraft and

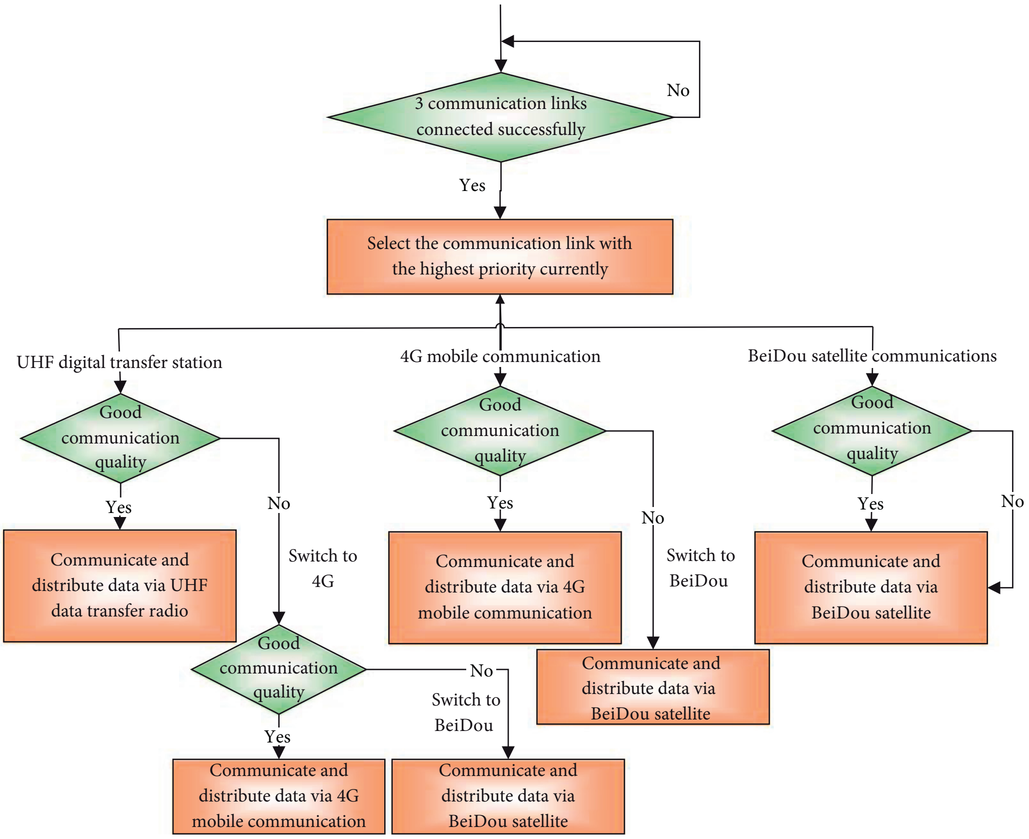

3 communication links connected successfully
No
Yes
Select the communication link with the highest priority currently
UHF digital transfer station
4G mobile communication
BeiDou satellite communications
Good communication quality
Yes
No
Communicate and distribute data via UHF data transfer radio
Switch to 4G
Communicate and distribute data via 4G mobile communication
Switch to BeiDou
Communicate and distribute data via BeiDou satellite
Communicate and distribute data via BeiDou satellite
Good communication quality
Switch to BeiDou
Communicate and distribute data via 4G mobile communication
Communicate and distribute data via BeiDou satellite

Figure 5: Flowchart of multilink adaptation.

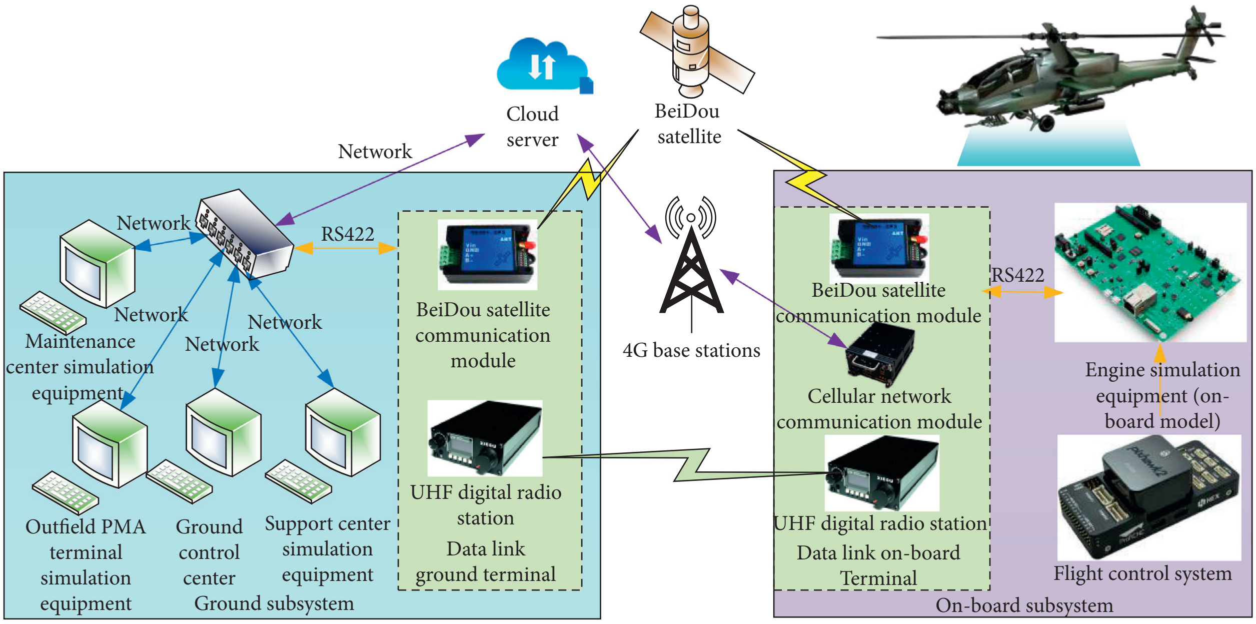

Cloud server
BeiDou satellite
Network
Network
RS422
Network
Network
Network
Maintenance center simulation equipment
BeiDou satellite communication module
UHF digital radio station
Data link ground terminal
Outfield PMA terminal simulation equipment
Ground control center
Support center simulation equipment
Ground subsystem
4G base stations
BeiDou satellite communication module
Cellular network communication module
UHF digital radio station
Data link on-board Terminal
RS422
Engine simulation equipment (on-board model)
Flight control system
On-board subsystem

Figure 6: Transmission architecture of air-ground segment and ground-ground segment.

engine for data transmission simulation test, a vertical take-off and landing UAV with fixed wings was used for test in this paper, as shown in Figure 14. Combining the advantages of rotary-wing UAV and the fixed-wing UAV, this UAV can not only achieve vertical take-off and landing but also fly horizontally with fixed wings. The functions of fixed-point take-off and landing, hovering, and high-speed cruise ensures that this UAV can operate smoothly in areas with complex terrain such as mountains, hills, and jungles and in places with dense buildings. It is characterized by fast

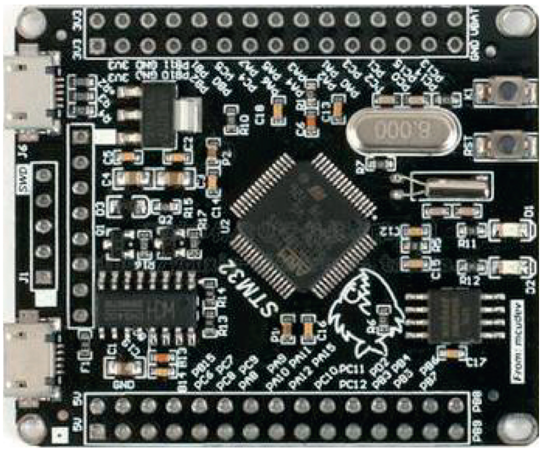
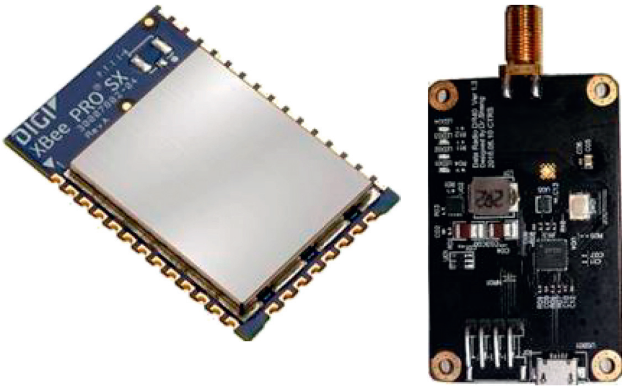
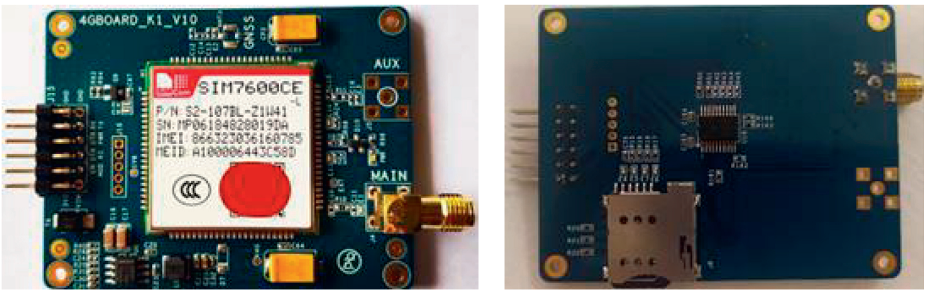


Figure 7: STM32F405 engine simulation module.



(a) (b)

Figure 8: UHF communication module.

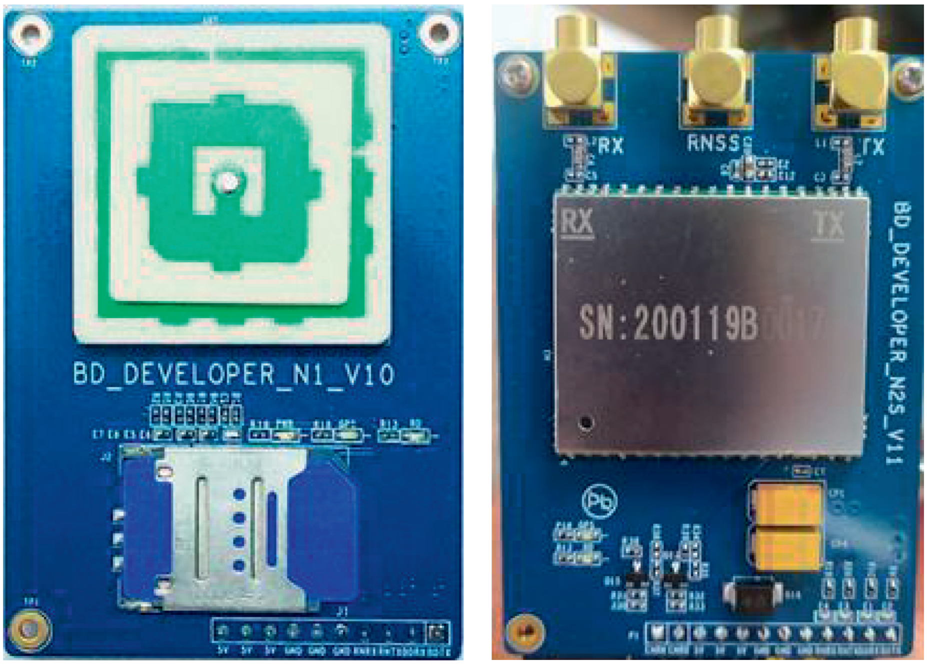


(a) (b)

Figure 9: 4G communication module.

cruising, long flying distance, long endurance, and no terrain requirements for take-off and landing.

The schematic diagram of the test platform is displayed in Figure 15. The outfield PMA, maintenance center, support center, and ground control center are, respectively, integrated on different computers. The outfield PMA, maintenance center, and support center are connected to the ground control center through the network interface, and the ground control center is connected to the ground terminal of data link through serial port to receive the telemetry data from the ground terminal of data link for analysis and processing and forwards the received telemetry data to the outfield PMA, maintenance center, and support center for analysis and processing. In addition to the on-board flight control system, the UAV test platform is also equipped with engine simulation equipment to simulate the real-time working status of the engine and obtain data. After receiving the aircraft status



(a)

(b)

Figure 10: BeiDou communication module.

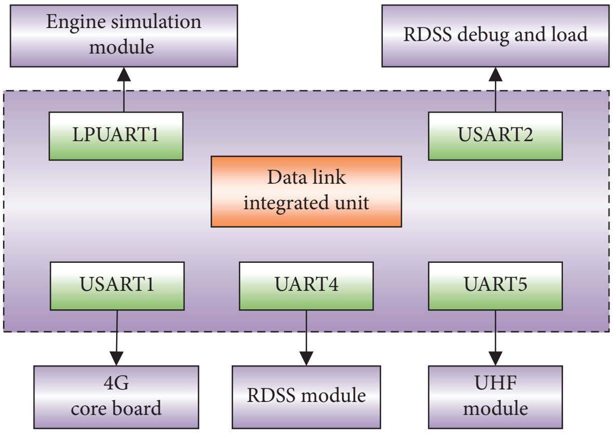


Figure 11: STM32 baseboard serial port connection.

Figure 12: Flight control module.

input from the on-board flight control system, the engine simulation equipment uses the on-board engine model to simulate changing the engine status and sends it to the on-board terminal of data link.

4.3. Simulation Test of Air-to-Ground Data Transmission of Aircraft Engine, Outfield PMA, Maintenance Center, and Support Center.

To verify the technology of real-time air-to-ground data communication with adaptive rate of the aeroengine health management system in the whole air-space, the on-board engine model based on STM32F405 was mounted on the UAV as the engine simulation equipment for the data link communication with the ground terminal, and then the air-to-ground data transmission simulation was carried out.

According to the engine data, UAV data, and the physical characteristics of the communication links, three formats of communication protocols were defined, and three types of data messages were generated. Through the flight test data of real UAV and the engine simulation data, the communication of the air-ground links for engine's health management data was tested, involving the communication tests of the UHF data radio communication, the 4G cellular mobile communication link, and the BeiDou satellite communication link and the test on the adaptive switching among the three links; when the packet loss rate of the current communication link is too high or the transmission delay is too large, it can be manually or automatically switched to other communication links. In addition, with the test on the function of ground-ground segment UDP data

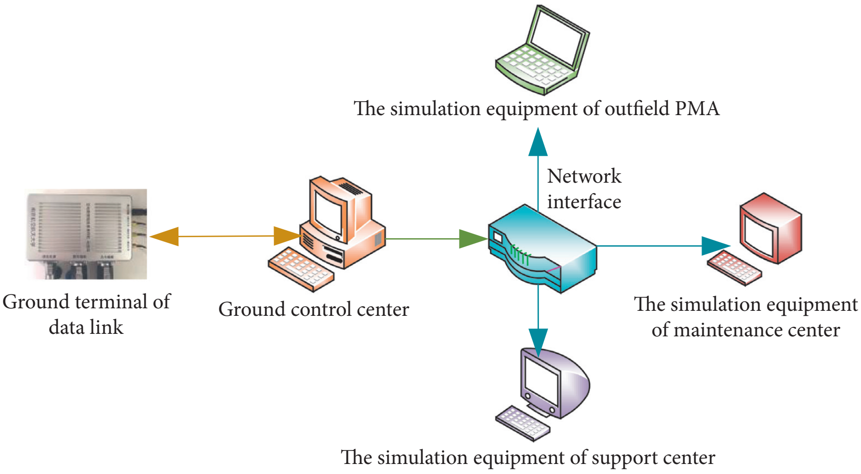

The simulation equipment of outfield PMA
Network interface
Ground terminal of data link
Ground control center
The simulation equipment of maintenance center
The simulation equipment of support center

Figure 13: Composition of the ground subsystem.

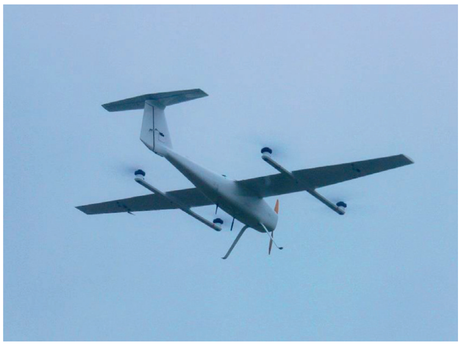


Figure 14: Vertical take-off and landing UAV with fixed wings.

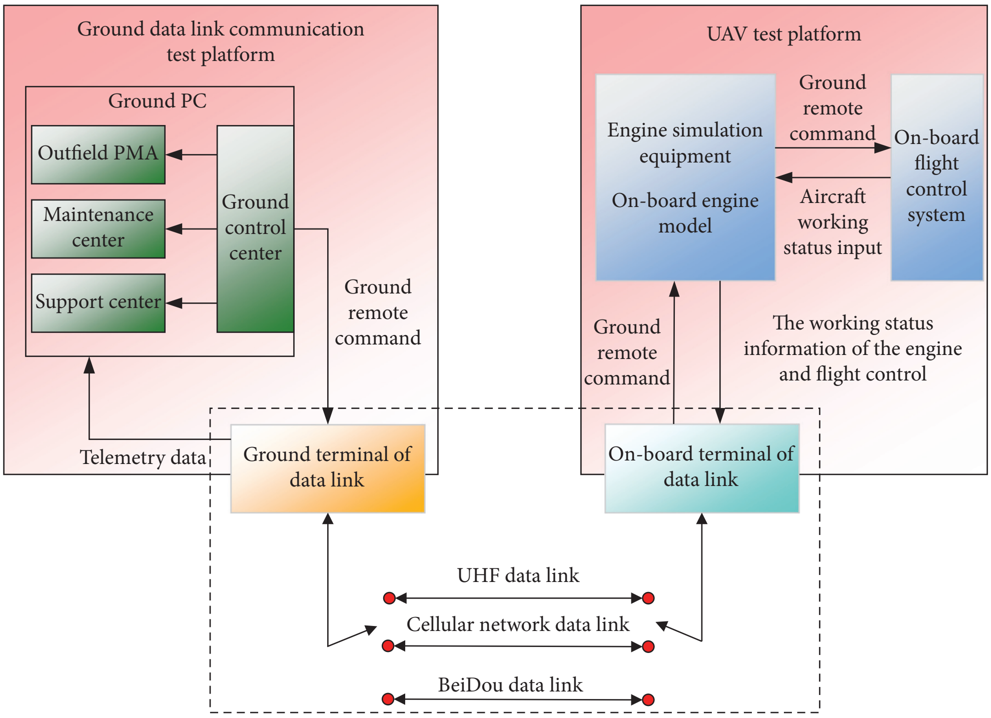
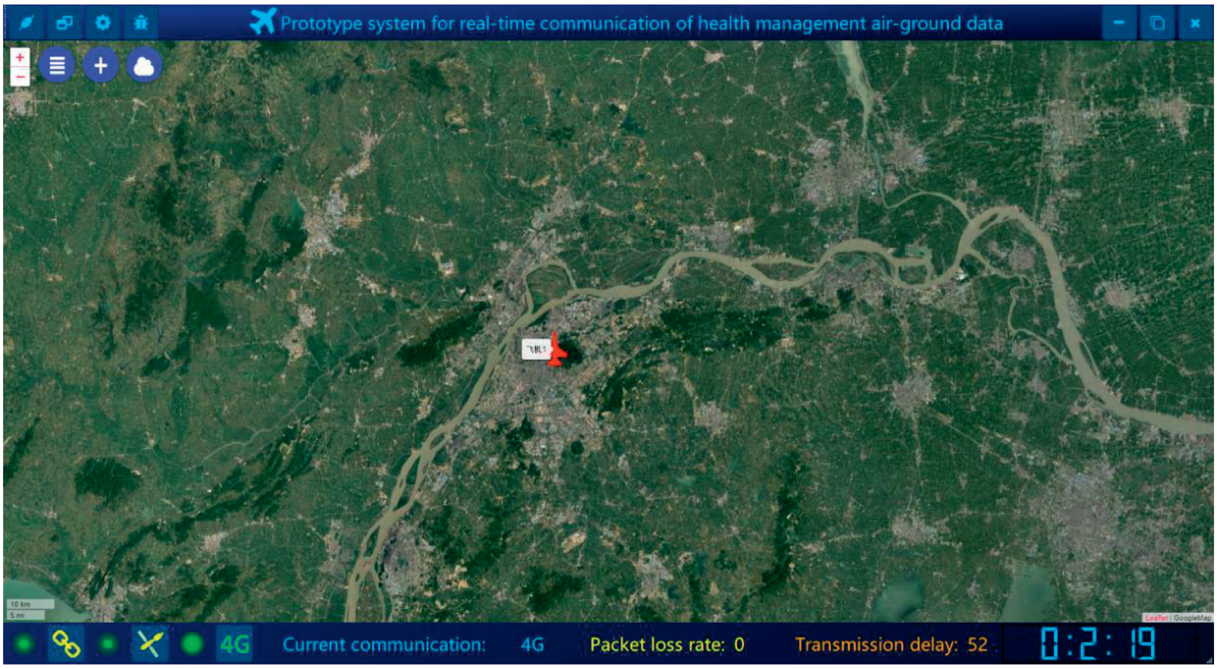

Ground data link communication test platform
Ground PC
Outfield PMA
Maintenance center
Support center
Ground control center
Ground remote command
Telemetry data
Ground terminal of data link
UAV test platform
Engine simulation equipment
On-board engine model
Ground remote command
Aircraft working status input
On-board flight control system
Ground remote command
The working status information of the engine and flight control
On-board terminal of data link
UHF data link
Cellular network data link
BeiDou data link

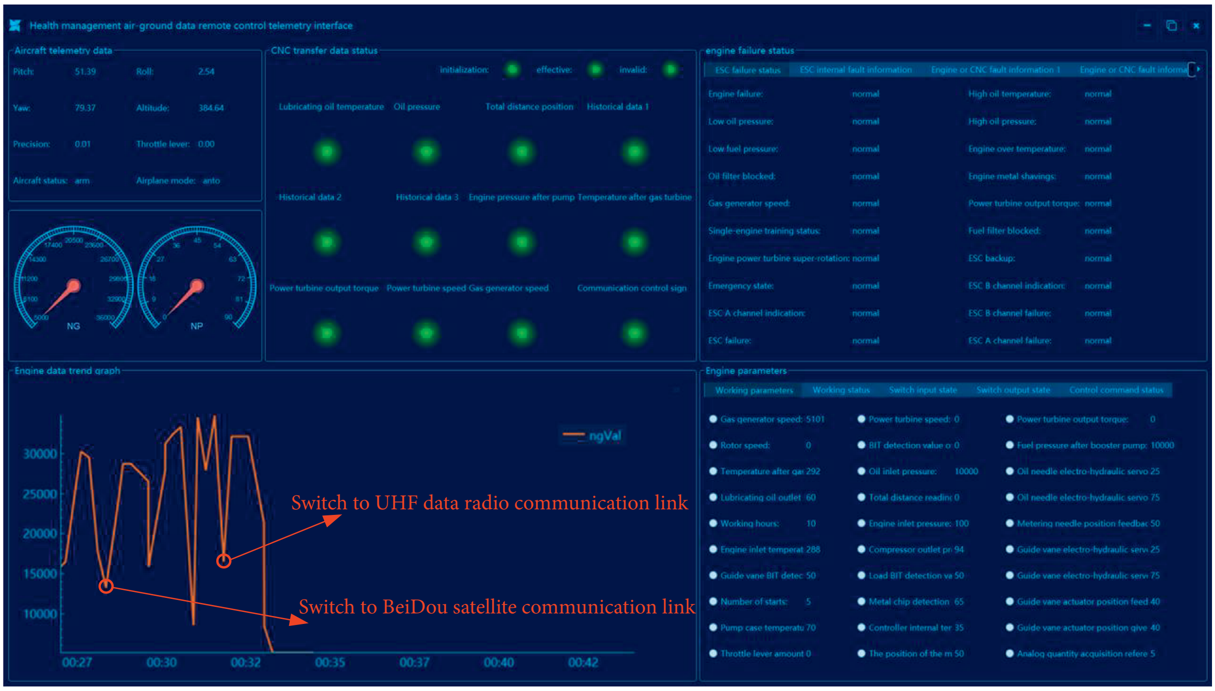
Figure 15: Schematic diagram of transmission simulation test platform.



Prototype system for real-time communication of health management air-ground data

Current communication: 4G   Packet loss rate: 0   Transmission delay: 52   0:2:19

Figure 16: 4G cellular mobile communication link air-to-ground data communication test.



Health management air-ground data remote control telemetry interface

Aircraft telemetry data

Pitch:	51.39	Roll:	2.54
Yaw:	79.37	Altitude:	384.64
Precision:	0.01	Throttle lever:	0.00
Aircraft status:	arm	Airplane mode:	anto

CNC transfer data status

initialization: ● effective: ● invalid: ●

Lubricating oil temperature ● Oil pressure ● Total distance position ● Historical data 1 ●

Historical data 2 ● Historical data 3 ● Engine pressure after pump ● Temperature after gas turbine ●

Power turbine output torque ● Power turbine speed ● Gas generator speed ● Communication control sign ●

engine failure status

ESC failure status | ESC internal fault information | Engine or CNC fault information 1 | Engine or CNC fault informa

Engine failure:	normal	High oil temperature:	normal
Low oil pressure:	normal	High oil pressure:	normal
Low fuel pressure:	normal	Engine over temperature:	normal
Oil filter blocked:	normal	Engine metal shavings:	normal
Gas generator speed:	normal	Power turbine output torque:	normal
Single-engine training status:	normal	Fuel filter blocked:	normal
Engine power turbine super-rotation:	normal	ESC backup:	normal
Emergency state:	normal	ESC B channel indication:	normal
ESC A channel indication:	normal	ESC B channel failure:	normal
ESC failure:	normal	ESC A channel failure:	normal

Engine data trend graph

Switch to UHF data radio communication link

Switch to BeiDou satellite communication link

Engine parameters

Working parameters | Working status | Switch input state | Switch output state | Control command status

● Gas generator speed: 5101	● Power turbine speed: 0	● Power turbine output torque: 0
● Rotor speed: 0	● BIT detection value o 0	● Fuel pressure after booster pump: 10000
● Temperature after ga 292	● Oil inlet pressure: 10000	● Oil needle electro-hydraulic servo 25
● Lubricating oil outlet 60	● Total distance readin 0	● Oil needle electro-hydraulic servo 75
● Working hours: 10	● Engine inlet pressure: 100	● Metering needle position feedbac 50
● Engine inlet temperat 288	● Compressor outlet pr 94	● Guide vane electro-hydraulic serv 25
● Guide vane BIT detec 50	● Load BIT detection va 50	● Guide vane electro-hydraulic serv 75
● Number of starts: 5	● Metal chip detection 65	● Guide vane actuator position feed 40
● Pump case temperatu 70	● Controller internal ter 35	● Guide vane actuator position give 40
● Throttle lever amount 0	● The position of the m 50	● Analog quantity acquisition refere 5

Figure 17: Display of the UAV/engine data switching between three communication links.

distribution, all tests passed, with the transmission delay within 100 ms and the packet loss rate less than 1%. The entire system can accurately and effectively communicate real-time data in different flight scenarios, which greatly improves the reliability and survivability of the system. Figure 16 shows the communication test of 4G cellular mobile air-to-ground data. Figure 17 displays the UAV/engine data switching from 4G communication link to UHF data radio communication link and then to BeiDou satellite communication link, and Figure 18 shows the communication between the ground control center and the outfield PMA, maintenance center, and support center.

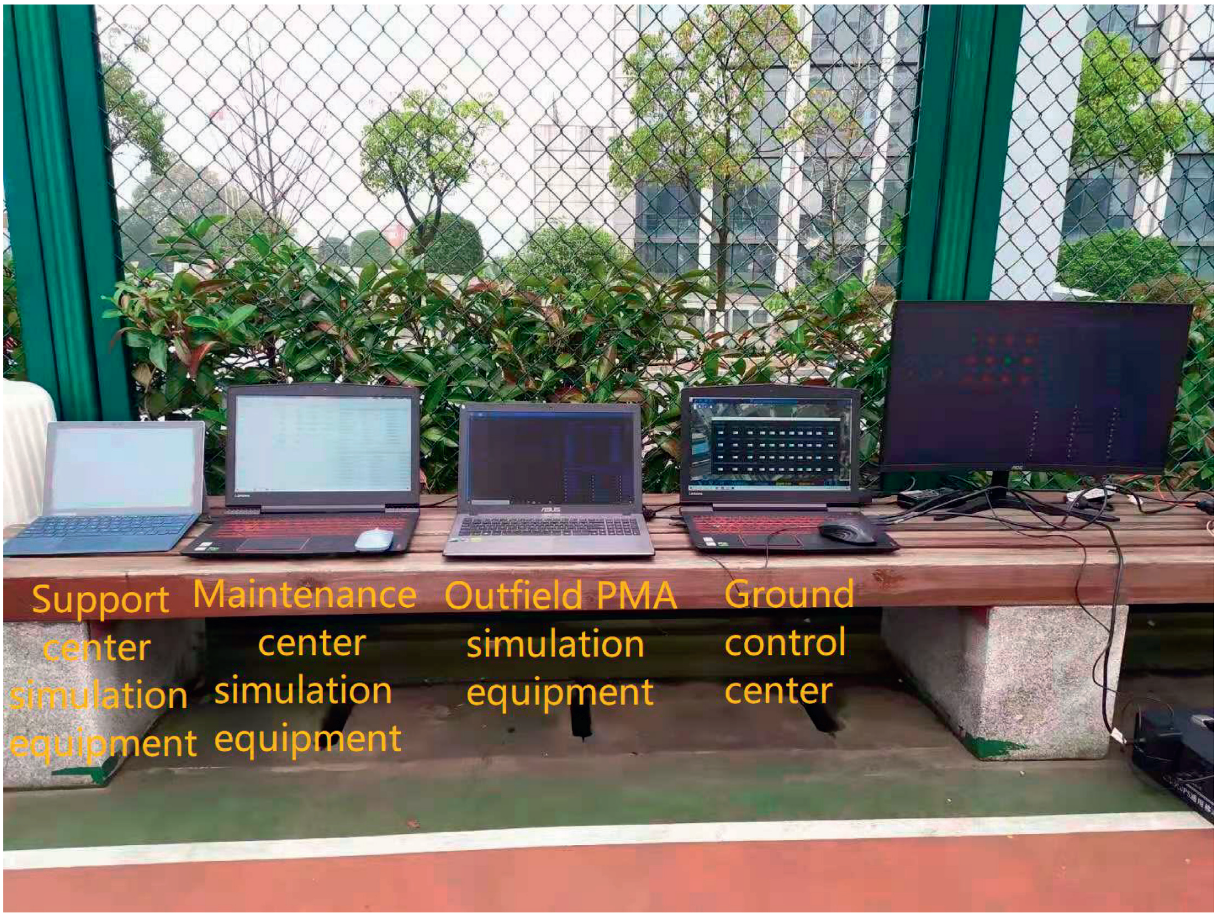

Support center simulation equipment
Maintenance center simulation equipment
Outfield PMA simulation equipment
Ground control center

Figure 18: Communication between ground control center and outfield PMA, maintenance center, and support center.

5. Problem Statement and Its Proposed Solution of Real-Time Air-to-Ground Data Communication Technology for Aeroengine Health Management System with Adaptive Rate in the Whole Airspace

(1) At present, the communication rate of BeiDou satellite communication link is not high. It is only used in special areas without 4G signal coverage for emergency. In the future, BeiDou satellite communication link can be improved by using multiple BeiDou communication modules for simultaneous communication or military grade module with a communication rate of 1000 bytes per second.

(2) At present, the communication between 4G cellular module and server is based on TCP/IP. In the future, UDP can be used to achieve a higher communication rate.

(3) The current system only supports the resuming transfer from the break point of single file and can be improved later to realize the multiple files' resuming transfer from the break point.

6. Conclusion

In this paper, an aeroengine health management system was described, which realized the real-time communication of air-to-ground data in the whole airspace with adaptive rate. The system focuses on the multilink adaptive switching among UHF data transfer link, 4G cellular mobile network link, and BeiDou satellite link and also uses DES symmetric encryption algorithm and the technology of resuming transfer from the break point to ensure the security and integrity of data transmission. The simulation of air-to-ground data transmission showed that this system can realize the monitoring of aircraft and engine data and the adaptive switching among multiple links, which verified the feasibility of the system in practical application.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research work was supported by the National Natural Science Foundation of China (no. 51906103), Aviation Science Foundation of China (2018ZB52), and Aeronautics Power Foundation (6141B09050384).

References

[1] I. Y. Tumer and A. Bajwa, "A survey of aircraft engine health monitoring systems," *in Proceedings of the 35th Joint Propulsion Conference and Exhibit*, Los Angeles, CA, USA, June, 1999.

[2] Z. Lin and N. Zhang, "Analysis of VHF ground-to-air data link system of China's civil aviation," *China Radio*, vol. 2015, no. 7, pp. 45–47, 2015.

[3] C. R. Mercer, D. L. Simon, G. W. Hunter et al., *Fundamental Technology Development for Gas-Turbine Engine Health Management*, NASA Glenn Research Center, Cleveland, OH, USA, 2007.

[4] M. J. Roemer, *Engine Health Monitoring System for Advanced Diagnostic Monitoring for Gas Turbine Engines*, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA, 1998.

[5] Y. Ying, Y. Cao, S. Li, and J. Li, "Nonlinear steady-state model based gas turbine health status estimation approach with improved particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 940757, 2015.

[6] Z. Wang, J. Li, K. Fan, and S. Li, "The off-design performance simulation of marine gas turbine based on optimum scheduling of variable stator vanes," *Mathematical Problems in Engineering*, vol. 2017, Article ID 2671251, 2017.

[7] M. T. Yildirim and B. Kurt, "Aircraft gas turbine engine health monitoring system by real flight data," *International Journal of Aerospace Engineering*, vol. 2018, Article ID 9570873, 2018.

[8] L. Gou, H. Li, H. Zheng, H. Li, and X. Pei, "Aeroengine control system sensor fault diagnosis based on CWT and CNN," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5357146, 2020.

[9] H. Zhang, G. Bai, and L. Song, "Multiobjective design optimization framework for multicomponent system with complex nonuniform loading," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7695419, 2020.

[10] G. Mazzuto, S. Antomarioni, F. E. Ciarapica, and M. Bevilacqua, "Health indicator for predictive maintenance based on fuzzy cognitive maps, grey wolf, and K-nearest neighbors algorithms," *Mathematical Problems in Engineering*, vol. 2021, Article ID 8832011, 2021.

[11] Y. Yang and Z. Gao, "Power management problem for civil aircraft under more electric environment," *International Journal of Aerospace Engineering*, vol. 2020, Article ID 8474375, 2020.

[12] L. Rui, Y. Guo, and W. Wu, "Integrated simulation platform for the design and verification of aero-engine sensor fault diagnosis," *Computer Measurement & Control*, vol. 18, no. 3, pp. 527–529, 2010.

[13] H. Yang, *Aero-Engine Gas Path Fault Diagnosis Based on Intelligent Technology*, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2008.

[14] X. Lv, "Discussion on the development of air-to-ground data link," *Air Traffic Management*, vol. 2005, no. 5, pp. 29–31, 2005.

[15] B. Sun, "Talking about the development and application of air-ground data link," *Military-Civilian Dual-Use Technology and Products*, vol. 2016, no. 12, p. 69, 2016.

[16] J. Huang, "Analysis of the architecture and application of civil aviation communication network," *Information&Communications*, vol. 2020, no. 10, pp. 80–82, 2020.

[17] H. Wang and M. Wang, "A simulation study on V/UHF air-ground data link," *Aircraft Design*, vol. 2008, no. 1, pp. 58–63, 2008.

[18] H. Liu, "Discussion on common interference problems and countermeasures of VHF communication system (R/S200)," *Digital Communication World*, vol. 2020, no. 2, p. 98, 2020.

[19] Z. Huang, "Present situation and development prospect of 4G network technology development," *Digital Communication World*, vol. 2017, no. 4, pp. 46–48, 2017.

[20] J. Shen, "Overview of the development of 4G mobile communication technology," *Wireless Internet Technology*, vol. 2017, no. 7, pp. 11–13, 2017.

[21] J. Guo, "The construction, development and application of air-to-ground data link of China's civil aviation," *China Civil Aviation*, vol. 2006, no. 3, pp. 64–66, 2006.

[22] L. Wen, "Beidou satellite navigation system and its application in civil aviation navigation," *Information Recording Materials*, vol. 20, no. 7, pp. 148–149, 2019.

[23] S. Wang, R. Wang, Z. Chen et al., "Overview of aeroengine health management," *Gas Turbine Test and Research*, vol. 22, no. 1, pp. 51–58, 2009.

[24] The Civil Aviation Administration of China, "Roadmap for the implementation of the Beidou satellite navigation system in civil aviation of China," *Civil Aviation Management*, vol. 2020, no. 1, p. 76, 2020.

[25] F. Liu, D. Wang, B. Yu, and F. Yu, "Instant messaging system based on hybrid 3DES and RC4 algorithm," *Computer Systems & Applications*, vol. 29, no. 8, pp. 80–89, 2020.

[26] X. Geng, "Research on file encryption based on DES algorithm," *China Computer & Communication (Theoretical Edition)*, vol. 32, no. 3, pp. 44–46, 2020.

[27] Y. Li, M. Cheng, X. Xu, and Y. Yang, "A Research on the continue transferring from breakpoint and multi-threading mechanism for plane media remote transmission," *Journal of Beijing Institute of Graphic Communication*, vol. 20, no. 6, pp. 53–56, 2012.

[28] Y. Zheng, *The Broken-Point Continually Transferring and Multi-Thread Download Component Based on HTTP/FTP Protocol*, University of Jinan, Jinan, China, 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9958257, 15 pages
https://doi.org/10.1155/2021/9958257

Research Article

Variability of SST and ILD in the Arabian Sea and Sea of Oman in Association with the Monsoon Cycle

Sartaj Khan,[1,2] Shengchun Piao,[1,2] Imran U. Khan,[1] Bingchen Xu,[1,2] Shazia Khan,[3] Muhammad Asim Ismail,[1] and Yang Song[1,2]

[1]*College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China*
[2]*Acoustic Science Technology Laboratory, Harbin Engineering University, Harbin 150001, China*
[3]*Department of Biological Sciences, International Islamic University, Islamabad 44000, Pakistan*

Correspondence should be addressed to Yang Song; song_yang@hrbeu.edu.cn

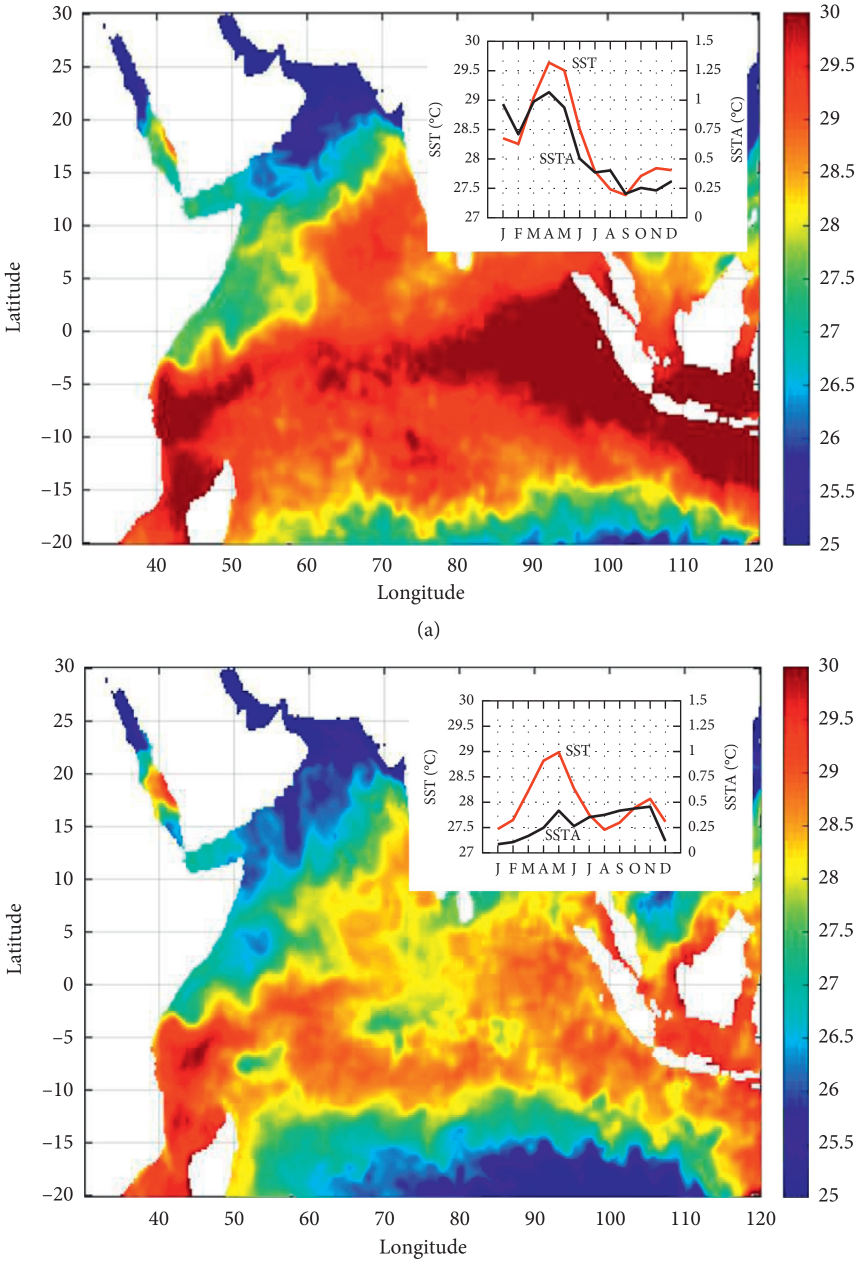
Received 5 March 2021; Revised 2 May 2021; Accepted 12 June 2021; Published 23 June 2021

Academic Editor: Bhawani Shankar Chowdhry

Sea surface temperature (SST) and isothermal layer depth (ILD) are important oceanic parameters and could play a significant role in understanding the upper thermal structure as well as improve the predictive capability of monsoons in the tropical oceans. In a disparate departure from the past research, the present study investigates the seasonal variability of SST and ILD in association with the monsoon cycle in the Arabian Sea and Sea of Oman regions by examination of Argo datasets for 2016-17. In this study, the ILD climatology is determined from temperature profiles provided by the Argo floats based on a threshold technique $(T(z) \geq \mathrm{SST} - 1^{\circ}\mathrm{C})$ to investigate the region of stronger and weaker monsoon wind forcing. For SST, values of temperature are used nearest to the sea surface (depth $(z) \leq 5\,\mathrm{m}$). The region is split into four distinct zones for an accurate description of the monsoon cycle: the south Arabian Sea, the central Arabian Sea, the north Arabian Sea, and the Sea of Oman. It is observed that summer monsoon is more pronounced in the south-central basin of the Arabian Sea, where ILD is deepening (>100 m in September 2016) mainly due to stronger wind forcing in this region. On the contrary, the Sea of Oman region is displayed with smaller ILD amplitude (<10 m in June 2016) with larger SST, meaning that this region is weakly influenced by the summer monsoon. The seasonal relationship established between ILD variability and monsoon cycle for 2016-17 shows that ILD could be a useful indicator for predicting summer monsoon in the Arabian Sea regions. Our analysis results indicate that the dynamics for SST variability are different in these regions and are influenced either by large-scale atmospheric forcing, such as the El Niño-Southern Oscillation (ENSO) and the Indian Ocean Dipole (IOD), or by the effects of mesoscale variations occurring along the Oman-Arabian coast.

1. Introduction

The dynamics of sea surface temperature (SST) variability in the Indian Ocean vary in different seasons and regions. Previous research shows that the strong seasonality of the SST is the result of the combined effect of oceanic and atmospheric processes at the air-sea interface (mainly controlled by seasonal changes in the incoming solar radiation), and the oceanic and atmospheric circulation such as El Niño/La Niña-Southern Oscillation (ENSO) in the Pacific Ocean [1–3] or the internal independent mechanism such as Indian Ocean Dipole (IOD) can affect the interannual variability of SST [4–8]. The annual cycle of monsoons mainly exhibits bimodal distribution and significantly affects the upper thermal structure, which is mainly responsible for regional circulation and heat/salt transport in the Arabian Sea [9]. The annual cycle of SST in the tropical Indian Ocean mainly consists of four stages: (1) a warming stage from about February to May; (2) cooling from May to August; (3) warming from August to October; and (4) cooling from October to January (Figure 1). This pattern is in contrast to the annual cycle of the SST in most of other regions of the world ocean, which display only two phases: warming during spring and summer; and cooling during autumn and



(a)

(b)

Figure 1: Annual mean SST in the tropical Indian Ocean for 2016 (a) and 2017 (b). Monthly mean SST and SST anomaly cycles are shown at the top right of the map data. Datasets were derived from the NOAA OISST (Optimum Interpolation Sea Surface Temperature) blended product, version 2.1. This product includes satellite observations Advanced Very High-Resolution Radiometer (AVHRR) and Advanced Microwave Scanning and is available from 1981, with a 0.25° × 0.25° spatial resolution and daily temporal interval.

winter. All available evidence suggests that this unusual behavior of the tropical Indian Ocean is due to the influence of the southwest monsoon (summer season) that dominates the Arabian Sea during the northern hemisphere summer. The energetic circulation of wind during this period is known to have an effect on the SST. In the coastal regions, upwelling typically occurs, which brings up colder water and then spreads offshore [10], whereas, in the open sea, the loss of energy and heat on the surface lower the SST. These changes in SSTs and winds over the oceanic may have influence on weather and climate of the adjacent land area and Indian rainfall [3, 11]. Recently, the SST in the east Pacific Ocean was strongly influenced by the powerful El Niño in 2016 [4, 12] and the weak La Niña in 2017 [13]. In early 2016, the tropical Indian Ocean warmed up in the presence of El Niño and the maximum SST (monthly mean) increased beyond 29.5°C in April-May (Figure 1(a)). The powerful El Niño and weak La Niña caused extreme negative and weak positive IOD events in the Indian Ocean during 2016 and 2017, respectively. The extreme negative and weak positive IOD events have had a significant impact on the monsoon cycle in the Indian Ocean and have caused more flood and rainfall in many parts of Australia, Indonesia, and the Indian subcontinent, as well as drought conditions in East Africa [12, 14]. As a result, almost half of the world's population living in this region has an impact on food production, water insecurity, and even human health.

Apart from SST, isothermal layer depth (ILD) is an important oceanic parameter that could help us understand the effects of monsoonal winds variability in the tropical Indian Ocean. It is, therefore, critical to document ILD variability and its relation to the monsoon cycle, because it may have a significant influence on summer monsoon qualitative predictability. Using various Argo and satellite datasets, an attempt is made in the present study to investigate the influence of monsoons on SST cycle and ILD amplitude in the Arabian Sea and Sea of Oman regions. It is important to highlight that the SST cycle in the south-central basin of the Arabian Sea is mainly affected by the ENSO and IOD events. A notable feature of the Arabian Sea circulation is the presence of strong (weak) upwelling along the Oman-Arabian coast that occurs in late spring and peaks in summer. The low upwelling warms SST along the Oman-Arabian coast, which is considered important for Indian monsoon rainfall and remains a key area in this study. We believe that these results, based on satellite and Argo datasets, will be useful for scientists engaged in research on atmospheric and oceanic studies, air-sea interaction studies, and monsoon studies. The rest of the paper is organized as follows: Section 2 describes the area under investigation, as well as various satellite and Argo datasets, followed by a brief derivation of ILD and SST from temperature profiles provided by Argo floats. Section 3 describes the results and discussion, including SST and ILD variability in association with monsoon cycles in the Arabian Sea and Sea of Oman regions. Finally, the main findings are summarized in Section 4.

2. Materials and Methods

2.1. Study Area. The Arabian Sea is the northwestern part of the tropical Indian Ocean with land boundaries in the west, north, and east. The geographical extent of the Arabian Sea is shown in Figure 2. As shown in the figure, the sea is bounded by India (to the east), Pakistan and Iran (to the north), and the Arabian Peninsula in the west. In the southwest, the sea is connected with the Red Sea through the Gulf of Aden, while, in the northwest, it is connected with the Persian Gulf through the Sea of Oman and Strait of Hormuz. The Sea of Oman (formerly known as the Gulf of Oman) extends between 22 and 26°N and 56–60°E. It is broad and deep (240 × 450 km and from 200 m to typically 4000 m deep at the Arabian Sea) and opens on the Arabian Sea and the Indian Ocean at its southeast part. The areas under investigation are identified by four distinct zones including three in the Arabian Sea, south Arabian Sea (Zone 1; Eq–5°N/54–61°E), central Arabian Sea (Zone 2; 11–15°N/57–69°E), and north Arabian Sea (Zone 3; 17–23°N/60–67°E), and one in the Sea of Oman (Zone 4; 23–25°N/58–65°E).

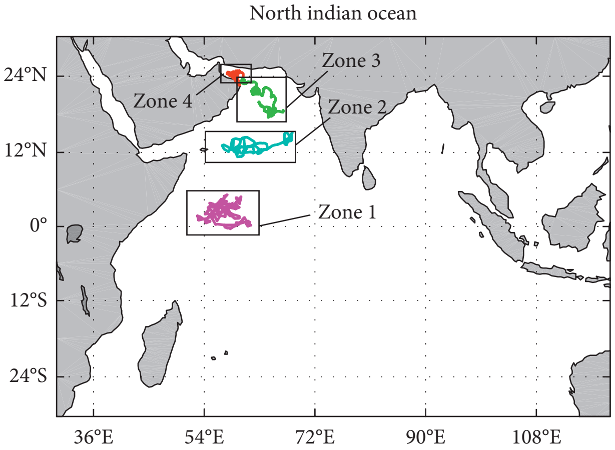


Figure 2: Geographical extent of the north Indian Ocean; the areas under investigation are marked with black boxes representing the south Arabian Sea (Zone 1; Eq–5°N, 54–61°E), central Arabian Sea (Zone 2; 11–15°N, 57–69°E), north Arabian Sea (Zone 3; 17–23°N, 60–67°E), and Sea of Oman (Zone 4; 23–25°N, 58–60°E). Trajectories of Argo floats are shown within the boxes.

2.2. Data Sources

2.2.1. Argo Data. The most important data for this study are temperature and salinity profiles provided by Argo floats. The recent expansion of the Argo array in the Indian Ocean has enhanced the coverage and resolution of temperature and salinity profiles [15–17]. Argo provides greater spatial and temporal resolution relative to other in situ observations (i.e., ship-based measurements, gliders, and moorings). Usually, Argo floats sample during their ascent phase and stop sampling 3 to 4 m below the sea surface. The cycle period (the time interval between two surfacings of the float) of the floats ranged from 3 to 10 days. In this study, T/S (temperature/salinity) profiles of four Argo floats were used for SST and SSS (sea surface salinity) measurements. All these profiles were obtained from January 01, 2016, to December 31, 2017, and were periodically downloaded from the Ifremer website [18]. As the dynamics of SST variability are different for different seasons and regions, therefore, monthly mean of SST data was analyzed separately for each region in this paper. Reanalysis of time-space variations of SST and SSS was performed in Matlab version "Matlab r2017a". Before analysis, the datasets were linearly interpolated with a one-meter depth interval. For SST and SSS, values of temperature and salinity were used closest to the sea surface (depth (z) ≤5 m), as it would be the nearest surface observation that would be comparable to both satellite SST and SSS measurements [19]. Details of Argo floats used in this study are described in Table 1.

2.2.2. Satellite Data. In addition to Argo floats, various satellite datasets were used in this study. The monthly mean sea-air temperature (SAT) data at 1 atmospheric pressure were provided by NCEP Global Data Assimilation System (GDAS) and are available at 2.5° (latitude) × 2.5° (longitude) spatial resolution. In addition, the precipitation and wind

Table 1: Details of Argo floats used in the study.

Argo float (WMO ID)	Deployment region	Description
2902659	South Arabian Sea (Zone 1)	This float was deployed on 28 October 2014 in the south Arabian Sea region near the equator around 1°N, 65.2°E by First Institute of Oceanography (Qingdao China) as Argo China Project. The profiler cycle period of this float was 240 h with maximum 2000 m depth. The float provided a total of 201 profiles out of which 72 profiles between EQ–5°N, 56–61°E were analyzed. Trajectory of the float is shown in Figure 2.
2902123	Central Arabian Sea (Zone 2)	This float was released on 10 March 2014 in the central Arabian Sea region around 14°N, 67°E by Indian National Centre for Ocean Information Services (INCOIS) as Indian Argo Project. The profiler cycle period of this float was 10-days with a maximum of 2000 m depth. The float provided a total of 201 profiles out of which 72 profiles between 11 and 15°N, 57–69°E were analyzed. Trajectory of the float is shown in Figure 2.
2902062	North Arabian Sea (Zone 3)	This float was launched on 21 December 2015 in the north Arabian Sea region around 23°N, 60.25°E by Naval Oceanographic Office (NAVO) as U.S. Argo Project. The cycle period of this float was 3 to 5 days, with a maximum of 1500 m depth. The float provided a total of 293 profiles out of which 187 profiles between 17 and 23°N, 60–67°E were analyzed. Trajectory of the float 2902062 is shown in Figure 2.
2902065	Sea of Oman (Zone 4)	Trajectory of the float 2902065 is shown in Figure 2. This float was deployed on 09 November 2015 in the Sea of Oman region around 24°N, 58.9°E by Naval Oceanographic Office (NAVO) as U.S. Argo Project. The Argo drift had a 3 to 5-day cycle with a maximum of 1500 m depth. The float provided a total of 297 profiles out of which 198 profiles between 23 and 25°N, 58–60°E were analyzed.

speed at 10 m above sea surface (U10) were used. The monthly mean precipitation data (mm/day) were derived from the Global Precipitation Climatology Project Version 2.3 (GPCPv2.3). The GPCPv2.3 data were available with air-sea 2.5° (latitude) × 2.5° (longitude) spatial resolution [20, 21]. The Woods Hole Oceanographic Institution (WHOI) Objectively Analyzed Fluxes (OAFlux) project provided the mean wind speed at 10 m above sea surface. The OAFlux is an ongoing research and development project for global air-sea fluxes (http://oaflux.whoi.edu). The monthly mean SSS data were obtained from the SMAPv3 (Soil Moisture Active Passive version 3; [22]) and are accessible online at http://www.remss.com/missions/smap. Although SMAP was designed to measure space soil moisture, its L-band radiometer can also be used to measure SSS.

2.3. Criteria for ILD Determination. In the Arabian Sea, the propagation of the fluctuations in the thermocline is a key factor in maintaining the SST cycle. The changes in the thermocline depths are represented by the changes in the ILD. In this study, the threshold technique was used to derive the ILD from the temperature profiles provided by the Argo floats. For ILD, the threshold technique determines the depth at which the temperature (ΔT) alters by a fixed value relative to the value at a near-surface reference depth [23]. The ILD climatologies have been created in the past by many authors for individual ocean basins [24–26]. The criterion for determining the isotherm temperature varies from 1°C [26, 27] to 0.8°C [28, 29] to 0.5°C [30–32] to 0.2°C [33–37]. In this study, the ILD was estimated by using Rao et al. [26] formulation. The reference level, however, varies from near-surface [30] to 10 m depth [36]. The criteria used in this research were the depth range, where the temperature of given depth (z) is within 1°C of the near-surface temperature, i.e., ILD = depth, where $\mathrm{T(z)} \geq \mathrm{SST} - \Delta\mathrm{T}$, and $\Delta T = 1$°C [26, 27]. Seawater density δ (T, S, z) was determined from temperature and salinity profiles, where T, S, and z represent temperature (°C), salinity, and depth (m), respectively.

The above criterion is applied to a single temperature and salinity profile in the Arabian Sea and the Sea of Oman regions (Figure 3). As can be seen from the Figure 3(b), the appearance of the halocline layer just below the surface layer indicates a density variation within the isothermal layer. It means that the difference between isothermal and isopycnal layers depth is positive, which is often in the tropics [38]. As discussed later in this study, the deeper isothermal layer in the Arabian Sea is most likely due to strong wind mixing activity at the air-sea interface, which represents stronger monsoonal force in the Arabian Sea region during the summer compared to the Oman Sea region, where the depth of the isothermal layer is always shallow.

3. Results and Discussion

3.1. South Arabian Sea. Monthly variations of SST and SAT in the south Arabian Sea region in 2016 and 2017 are shown in Figures 4(a)and 4(b), respectively. As shown in Figure 4(a), this region appears with a semiannual cycle in SST distribution, where low SST occurs in both summer and winter monsoons. The annual minimum SST reaches 26.8°C in July 2016 during summer monsoon and in January 2017 during winter monsoon. On seasonal average, the winter monsoon warming is not similar and shows different variability in SST in both 2016 and 2017. Significant variations in the SST cycle are observed during the winter monsoon when the surface temperature is warmer by almost 2°C during winter months (January to February) in 2016 than in 2017. It is observed that variability of SAT patterns is consistent with SST and remains high at the beginning of 2016 compared to 2017. This warming in SST during 2016 is concurrent with the extreme El Niño event that occurred in

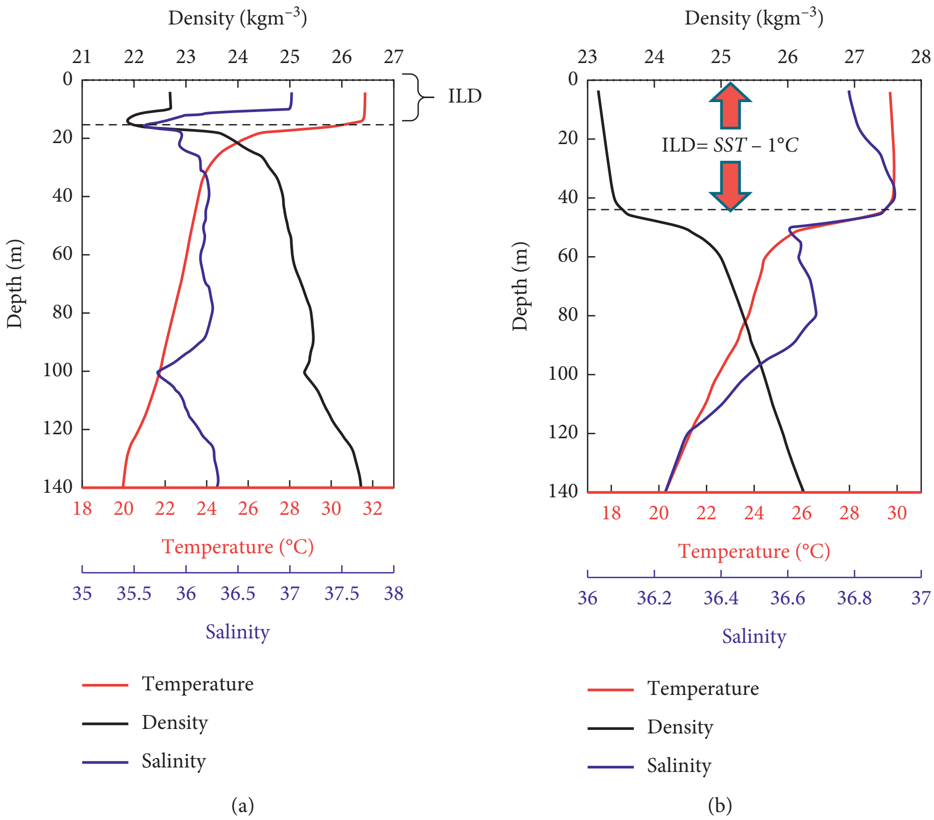


Figure 3: Typical profiles of temperature and salinity from the Sea of Oman and Arabian Sea regions. ILD is calculated from the depth, where the temperature decreases by 1°C from the temperature at the sea surface (or reference temperature). Float position was 24.8°N, 59.8°E in July 2017 in the Sea of Oman (a) and 22.5°N, 62.5°E in June 2016 (b).

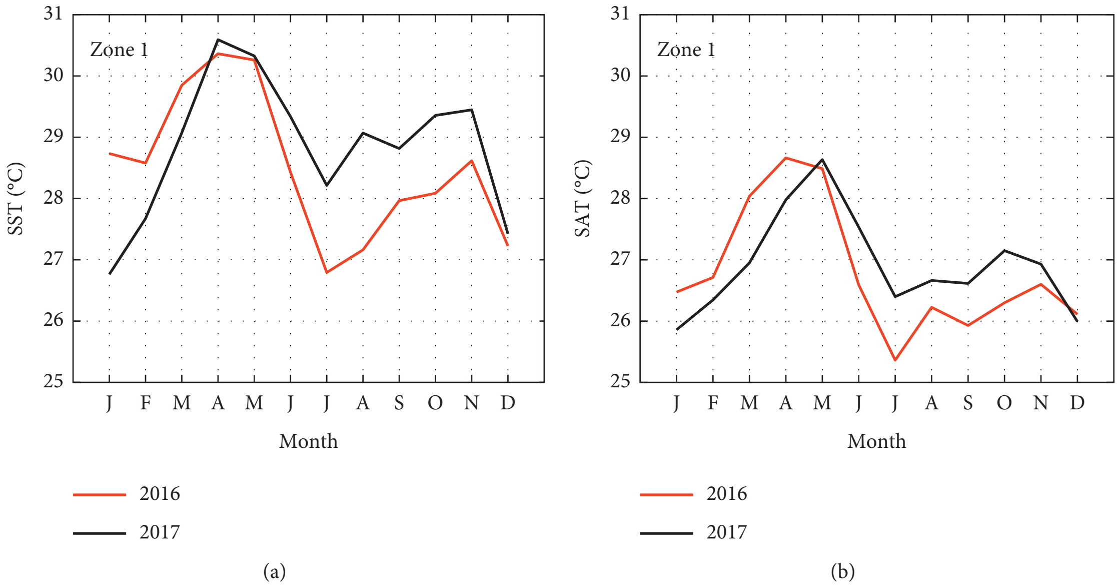


Figure 4: (a) Monthly mean cycle of SST in the south Arabian Sea (Zone 1) during 2016-17 based on Argo datasets. (b) Monthly mean cycle of SAT at 1 atmospheric pressure in the South Arabian Sea (Zone 1) during 2016-17. SAT datasets were provided by NCEP Global Data Assimilation System (GDAS), and they are available at 2.5° × 2.5° spatial resolution.

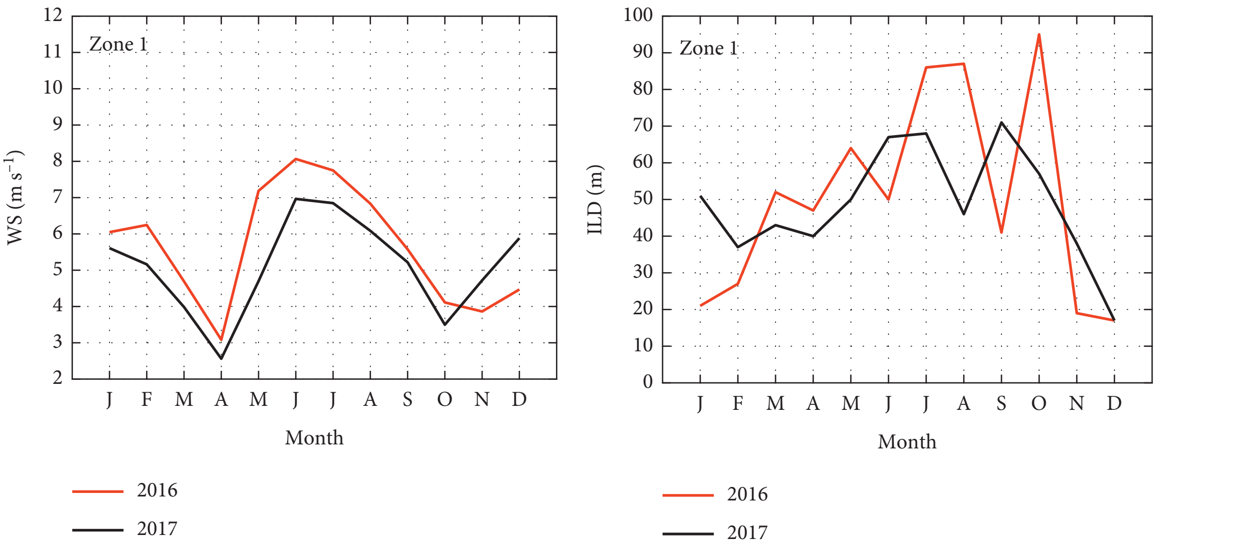
the Pacific Ocean in early 2016 [12]. The strong El Niño significantly affected the equatorial Pacific Ocean and the tropical Indian Ocean, where positive SST anomalies persisted throughout 2016 (Figure 1(a)).

During the presummer monsoon, SST is getting warm due to clear skies, rise in solar insolation, and reduced winds (WS <3 m/s in April; Figure 5(a)). These moderate winds and strong surface heating are instrumental in warming up the ocean surface, resulting in high surface temperature (SST >30°C in April-May; Figure 4(a)). However, as shown in Figure 5(a), the wind speed shows an increasing trend as the monsoon sets. SST gets cool and reaches above 27°C (28°C) on average in summer 2016 (2017) from July to September due to strong winds and cooling in the air temperature. During this period, the effect of wind force is high as compared to the premonsoon, and the turbulent mixing causes the cool surface waters to sink into a deeper layer, resulting in a deeper thermocline (up to 90 m) in July-August of 2016. In comparison, the SST in this region is cooler by almost 1°C in 2016 than in 2017, which is concurrent with the negative (positive) IOD events in 2016 (2017) with a low (high) SST in the western tropical Indian Ocean compared to the eastern part. The strong correlation, which is observed between SST and SAT patterns, represents a strong coupling between the ocean and the atmosphere in this region. These variability patterns highlight the fundamental role of SAT on SST. Quite different behavior is, however, seen in this region, where ILD amplitude is always smaller in the winter monsoon and larger in the summer and post-summer monsoons. As can be seen from Figure 5(b), the two-year distribution of ILD reveals significant fall in ILD amplitude during the summer monsoon in 2017 than in 2016. The smaller ILD amplitude from December to March indicates the weaker nature of the convective mixing during the winter monsoon, where larger amplitude from June to September represents the dominant nature of the winds stirring and Ekman pumping during the summer monsoon. In short, the mechanisms describing the air-sea feedback, as well as the relationship between wind and thermodynamic parameters, indicate that SST is mainly driven by wind force in this region. The smaller ILD amplitude from December to March indicates the weaker nature of the convective mixing during the winter monsoon, where larger amplitude from June to September represents the dominant nature of the winds stirring and Ekman pumping during the summer monsoon.

3.2. Central Arabian Sea. Monthly cycle of SST and SAT in the central Arabian Sea region in 2016 and 2017 is shown in Figures 6(a)and 6(b), respectively. Like the south Arabian Sea, this region also exhibits a semiannual cycle of low SST in both summer and winter monsoons. The annual minimum SST reaches 26°C in August 2016 during the summer monsoon and 25.9°C in February 2017 during the winter monsoon. As in the southern basin, the two-year distribution of SST in this region has shown a remarkable warming in the 2016 winter monsoon compared to 2017. Winter SST is warmer at around 2°C during the winter months (January to February) in 2016 than in 2017. This trend is, however, in contrast with the late summer and autumn seasons, in which SST remained warm in 2017 compared to 2016. This warming in the SST in winter 2016 is concurrent with the extreme El Niño event that occurred in the Pacific Ocean in early 2016. The premonsoon period (April to May) in the central Arabian Sea is characterized by clear sky, light winds, deep solar insolation, and strong heat gain. In this period, SST rapidly increases and reaches 31°C in May 2016 (the annual high).

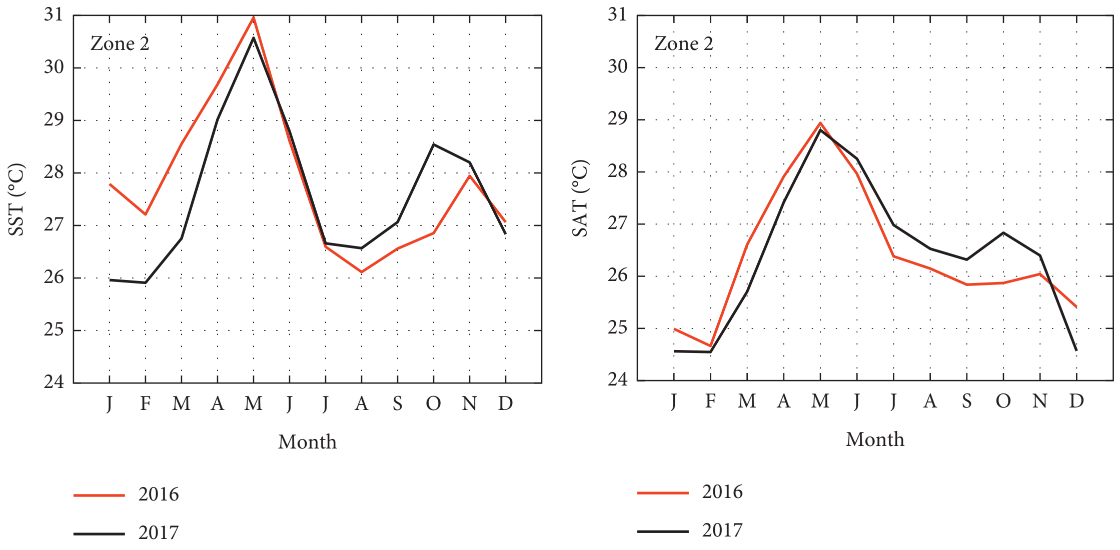
In the summer monsoon period, the central Arabian Sea region expresses some of the strongest and balanced winds forcing, which is characterized by strong winds, moist air, and a decrease in solar insolation due to cloudy cover. The WS is showing an increasing trend as the monsoon sets in both 2016 and 2017 (Figure 7(a)). SST gets cool and reaches ~ 26°C in summer monsoon due to strong wind speed (WS >11 m/s) and cooling in SAT (<26°C). During this period, the effect of wind force is much higher than that of the premonsoon, and the turbulent mixing causes the cool surface waters to sink into a deeper layer, resulting in a deeper thermocline (ILD ~ 120 m in September 2016). The two-year distribution shows almost similar patterns in SST during the first half of summer monsoon (June to July); however, the second half (August to September) shows significant cooling (about 1°C) in 2016 mainly due to negative (positive) IOD events in 2016 (2017) discussed earlier in previous section. The strong correlation, which is observed between SST and SAT pattern variability, represents a strong link between the ocean and the atmosphere in this region. As can be seen from Figure 7(b), the two-year distribution of ILD shows an increase in ILD amplitude during the late summer of 2016 compared to 2017. The larger ILD amplitude (ILD >100 m) in July-September represents the dominant nature of the winds stirring and Ekman pumping during the summer monsoon. In short, the mechanisms describing the air-sea feedback, as well as the relationship between wind and thermodynamic parameters, indicate that SST is mainly driven by wind force in the central and south basins of the Arabian Sea.

3.3. North Arabian Sea. The monthly cycle of SST and SAT in the north Arabian Sea region in 2016 and 2017 is shown in Figures 8(a)and 8(b), respectively. As shown in Figure 8(a), the seasonal cycle of temperature variations in this region in the upper surface is characterized by cooling from January to February and significantly warming from May to July and postmonsoon (October to November). The winter minimum temperature reaches 24°C in February 2016; however, the SST is getting warm during the presummer monsoon due to a rise in solar insolation and reaches 30°C in June 2016 with high SSS (>36.3) because of freshwater loss due to high evaporation. A patch of high saline water (above 36 on average; see Figure 9(c)) is observed in this region during the entire period. The presence of this high saline water near the Ras al Hadd region indicates the intrusion of highly saline water from the Persian Gulf into the northern Arabian Sea via the Sea of Oman [39]. However, the wind speed shows an increasing trend during the summer monsoon (Figure 10(a)). SST is getting cool and reaches 26°C in



(a) (b)

Figure 5: (a) Monthly mean cycle of wind speed at 10 m above sea surface in the south Arabian Sea (Zone 1) during 2016-17 based on WHOI Objectively Analyzed Air-Sea Fluxes project datasets. (b) Monthly mean cycle of ILD amplitude (ΔT = 1°C)derived from Argo datasets.



(a) (b)

Figure 6: As in Figure 4, but for the central Arabian Sea (Zone 2). (a) Monthly mean cycle of SST. (b) Monthly mean cycle of SAT.

August 2016 due to high wind speed (WS >9 m/s) and cooling in the air temperature (~26.5°C)(see Figure 8(b)).

On the seasonal average, the summer monsoon warming is not similar and shows different variability in both 2016 and 2017. The two-year distribution shows that the winter warming of SST is almost equal for both 2016 and 2017; however, significant variations in the SST cycle are observed during the summer monsoon months (July to September), where about 1.5°C decrease (increase) in SST is recorded in 2016 (2017) (Figure 8(a)). The wind speed was slightly high in 2016 than in 2017 summer months. The cooling (warming) in SST in 2016 (2017) is most probably due to the summer increased (reduced) upwelling that occurred along the western coast of the Arabian Sea near Ras al Hadd region around 22°N, 64°E band. The warmer ocean temperatures add more moisture to the atmosphere due to increased evaporation and latent heat, contributing to more rainfall and flooding in Bangladesh during the summer of 2017 [13, 40]. On the other hand, the strong upwelling along the Oman-Arabian coast pulled the cold water up and shallow the thermocline depth in summer of 2016 (ILD ~ 30min August-September 2016; Figure 10(b)).

3.4. *Sea of Oman.*

The monthly cycle of SST and SAT in the Sea of Oman region in 2016 and 2017 is shown in Figures 11(a)and 11(b), respectively. In this region, the

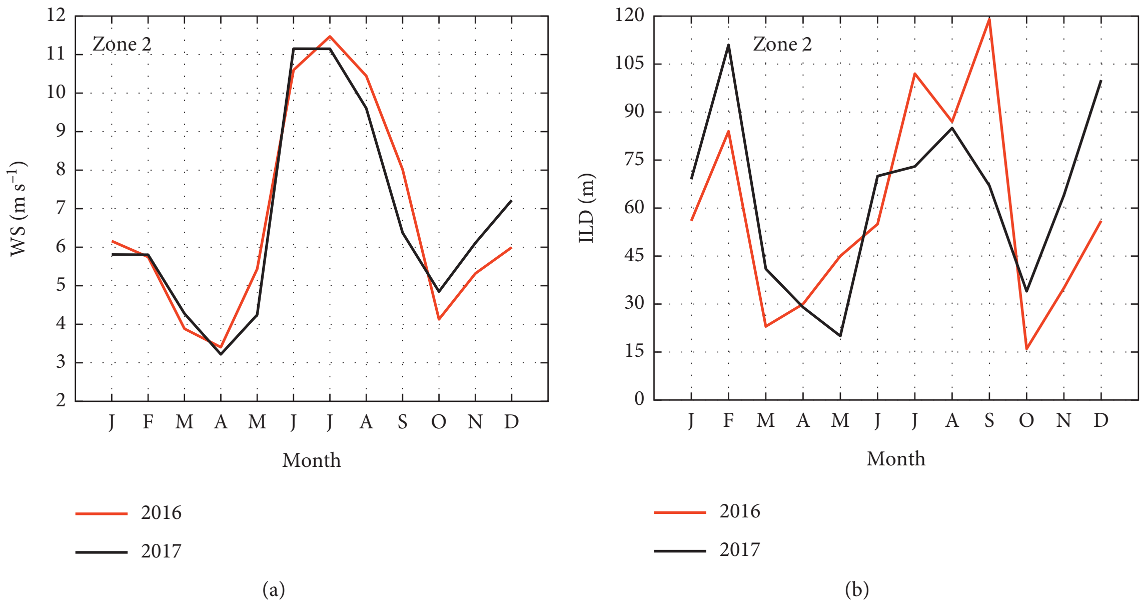


Figure 7: As in Figure 5, but for the central Arabian Sea (Zone 2). (a) Monthly mean of WS at 10 m above sea surface. (b) Monthly mean cycle of IDL (m).

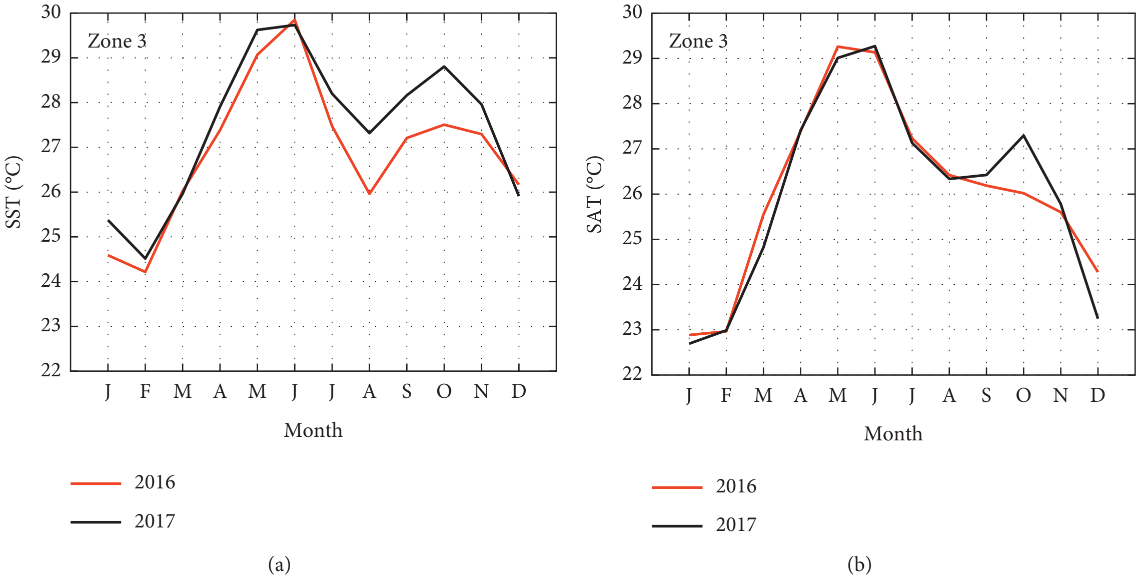
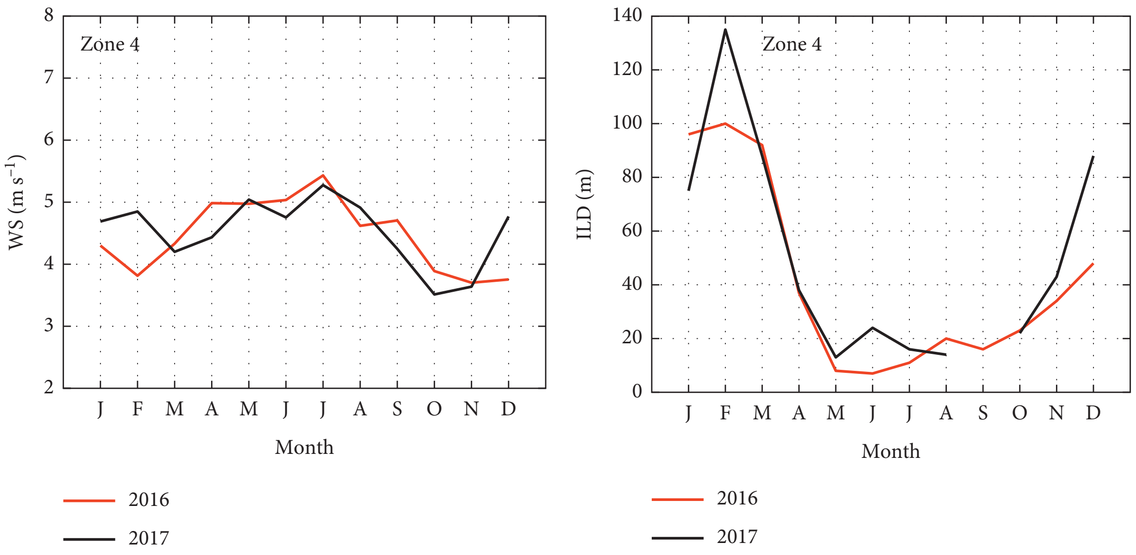


Figure 8: As in Figure 4, but for the north Arabian Sea (Zone 3). (a) Monthly mean cycle of SST. (b) Monthly mean cycle of SAT.

seasonal cycle of temperature variations in the upper sea surface is characterized by cooling during the winter monsoon (December to March) and predominantly warming in the summer monsoon (June to September). SST remains cool during the winter monsoon, with minimum sea and air temperatures at around 23.8°C and 22°C, respectively (in February 2017). However, there is a steady increase in SST during the premonsoon period (April to May) due to clear skies, reduced winds, and a rise in solar insolation. These moderate winds and strong surface heating are instrumental in warming up the ocean surface starting from 25°C in April 2017 and reaching 31°C in May 2017 (Figure 11(a)).

Upon the arrival of the summer monsoon, SST warms up to 31°C in August 2016 and June 2017 with SSS above 36.2 each (see Figure 12(d)). The presence of high salinity water (>36.5) in the Sea of Oman is most probably due to excess evaporation over precipitation and highly saline water intrusion from the Persian Gulf during the summer monsoon ([39]; see Figure 13(d)). The postmonsoon period in this region is characterized by the departure of the summer monsoon, having almost similar characteristics of surface forcing as that of premonsoon. SST starts to gradually decrease again and reaches the second low of 26.1°C in October 2016.

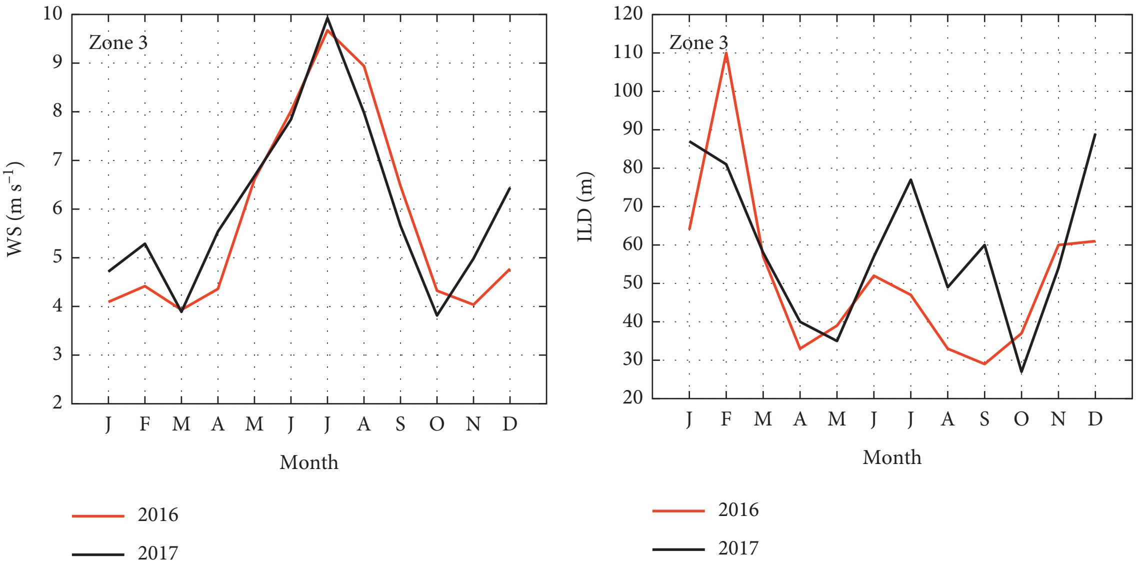
The two-year distribution of SST in the Sea of Oman region reveals that the winter warming is almost similar for



(a)

(b)

FIGURE 9: As in Figure 5, but for the Sea of Oman (Zone 4). The discontinue line in (b) is the missing data in the Argo datasets in this region. (a) Monthly mean of WS at 10 m above sea surface. (b) Monthly mean cycle of IDL (m).

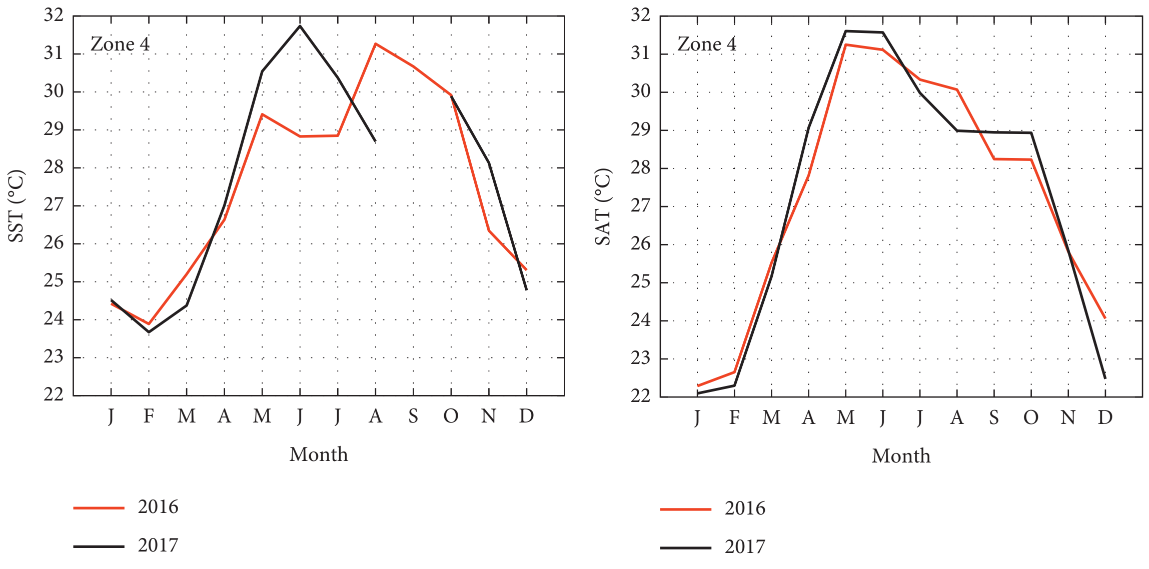


(a)

(b)

FIGURE 10: As in Figure 5, but for the north Arabian Sea (Zone 3). (a) Monthly mean of WS at 10 m above sea surface. (b) Monthly mean cycle of IDL (m).

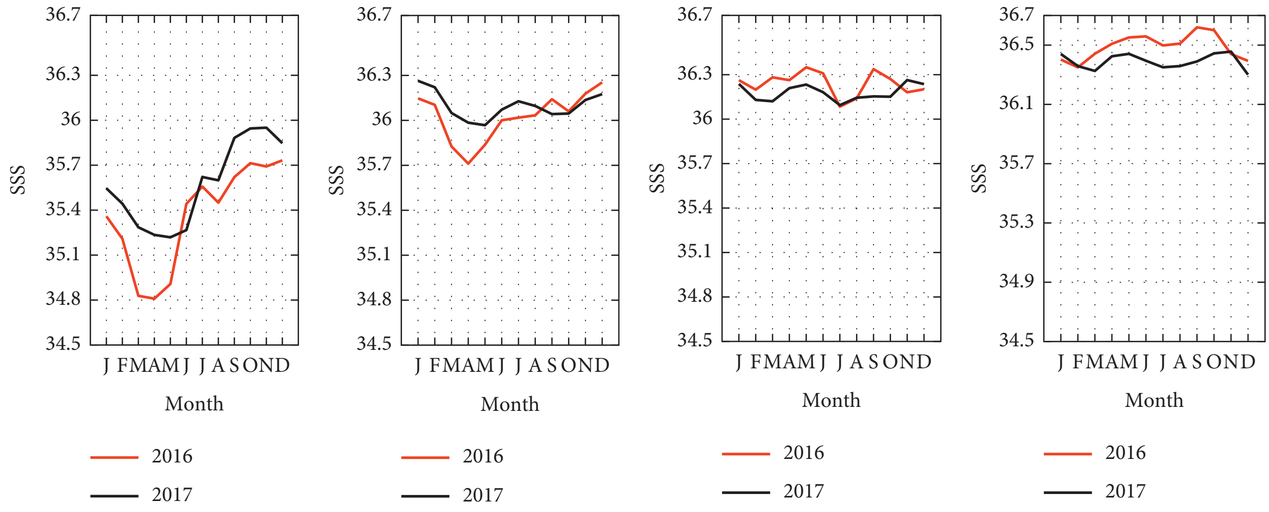
both years; however, a significant change in the SST cycle is observed during the premonsoon month (May) and the first half of summer monsoon (June to July). As can be seen from Figure 11(a), an abrupt cooling in SST up to 3°C is recorded in 2016 from May to July. This fall in temperature is due to the annual upwelling events, which occur along the southern coast of Oman near the Ras al Hadd region between 22.5°N and 59.5°E band (figure is not shown). This is because the upwelling water of about 25°C may originate below the pycnocline and, therefore, is colder than the surface water, which is 29°C or more and forms a shallow thermocline around 5 to 10 m deep (see Figure 9(b)). The depth of the thermocline, however, varies from the sea surface depending on tides and the direction, as well as the speed of coastal winds and currents. This falling trend in SST in the Sea of Oman during summer has been confirmed by other researchers [41–45]. Recently, Watanabe et al. [46] observed an abrupt SST fall in summer during various years in the Sea of Oman based on in situ SST data. The vertical profile of seawater temperature gathered by temperature sensors in 2010 also suggests that the thermocline was closer to the surface during summer upwelling events. Additionally, in



(a)

(b)

FIGURE 11: As in Figure 4, but for the Sea of Oman (Zone 4). The discontinue line in (a) is the missing data in the Argo datasets in this region. (a) Monthly mean cycle of SST. (b) Monthly mean cycle of SAT.



(a)

(b)

(c)

(d)

FIGURE 12: Monthly cycle of sea surface salinity (SSS) variability for 2016 and 2017 in Zone 1 (a), Zone 2 (b), Zone 3 (c), and Zone 4 (d). The data are obtained from the SMAPv3 (Soil Moisture Active Passive version 3) project.

situ chlorophyll-a and satellite-based SST confirmed that upwelling also occurred in July 2008 [47]. They observed a variation in the cold water near 20°C that underlies the solar-heated surface water of 30°C or more and found a short thermocline at a depth of 5–15 m. The upwelling phenomenon that occurs during the summer usually affects the SST and is the key environmental factor restricting coral growth and reef development along the southern coast of Oman from Dhofar to Ras al Hadd [48].

3.5. Precipitation Variability and SSS Circulation. The monthly mean precipitations (mm/day) in Zone 1 to 4 for 2016 and 2017 are shown in Figures 13(a)–13(d), respectively. The corresponding variability in SSS is shown in Figures 12(a)–12(d), respectively, for Zones 1 to 4.

The two-year distribution in the south Arabian Sea region shows a high precipitation rate (mm/day) in early 2016 than in 2017 (Figure 13(a)). The corresponding effect on SSS can be clearly seen in Figure 12(a), in which low SSS is observed at the beginning of 2016. The presence of high SSS during the winter and presummer monsoons expresses a negative P-E (precipitation minus evaporation) or strong evaporation and is characterized by low rainfall during 2017 compared to 2016. The summer monsoon, however, reveals reduced precipitation in this region in both 2016 and 2017.

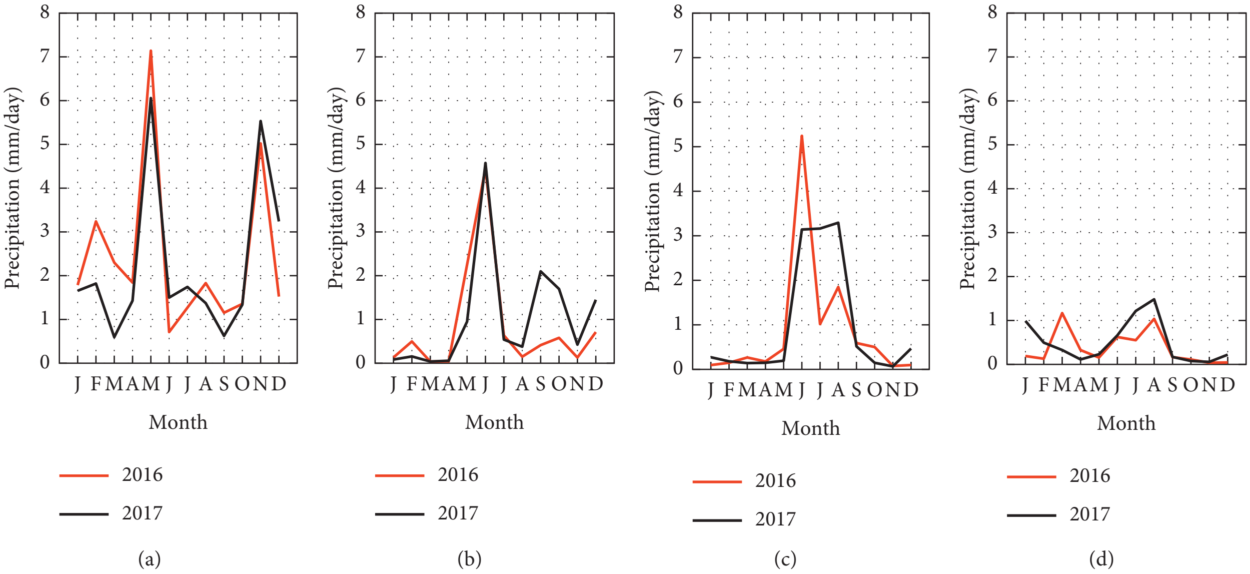


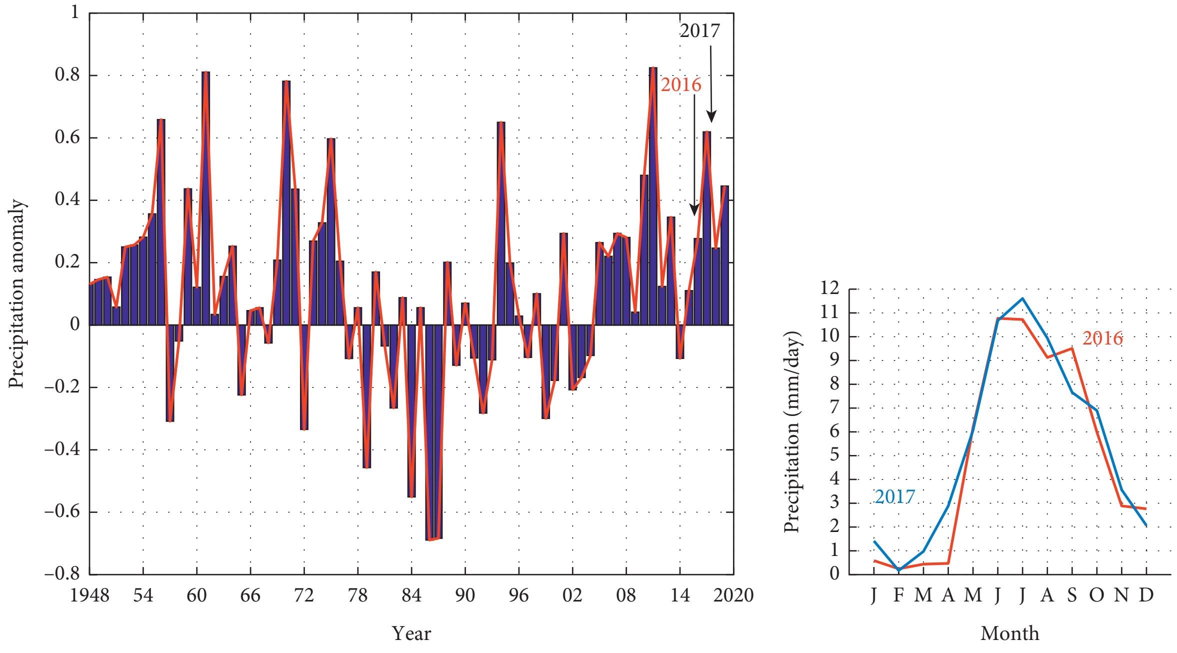
Figure 13: Monthly mean precipitation rate (mm/day) for 2016 and 2017 in Zone 1 (a), Zone 2 (b), Zone 3 (c), and Zone 4 (d). The data are provided by GPCPv2.3 for reanalysis and available at 2.5° × 2.5° resolution.

The moderate precipitation rate in this region in June-October in 2016 is mainly due to the strong negative IOD, which brought drought conditions in East Africa and reduced East African Short Rains (EASR) from October to December. These observations are consistent with Lu et al. (2018) [14], who recently investigated about 1 mm/day reduction in EASR in 2016, with a 50% reduction in normal rainfall in some regions.

On a seasonal scale, the monthly rainfall pattern for both winter and summer monsoons in the central Arabian Sea is low compared to the south Arabian Sea region; however, higher SSS (>36 on average) is observed over the entire period (see Figures 13(b)and 9(b)). The two-year distribution shows a slight increase in precipitation during the 2016 winter monsoon compared to 2017. The low SSS is most likely due to excess precipitation in the early months of 2016 compared to 2017. The presummer and first half of the summer monsoon show almost similar precipitation patterns but significantly reduced from the south basin in both 2016 and 2017. The subsurface high saline water (>36.9) at a depth of about 80 m observed from May to June 2017 at a latitude of 13°N (figure is not shown) is mainly due to the excess evaporation over precipitation during summer and the intrusion of Red Sea Outflow Water (>40) into the Arabian Sea via the Gulf of Aden [48–50].

On an annual average, the Bay of Bengal region expresses above average rainfall in both years with an increased trend in 2017, consistent with SST warming in the north Arabian Sea region due to low upwelling along the Oman-Arabian coast in summer 2017 (Figure 14). However, evidence of precipitation increase with SST, in the Arabian Sea, is not observed (Figure 15). It is interesting to mention here that SST was warmed by the weak upwelling that occurred along the Oman-Arabian coast in summer of 2017, which led to an increased rainfall in the west coast of Indian and western Bay of Bengal [51]. The low (high) precipitation, shown in Figure 13(b) in 2016 (2017), is concurrent with negative (positive) IOD and consistent with the results of Chanda et al. 2018 [52].

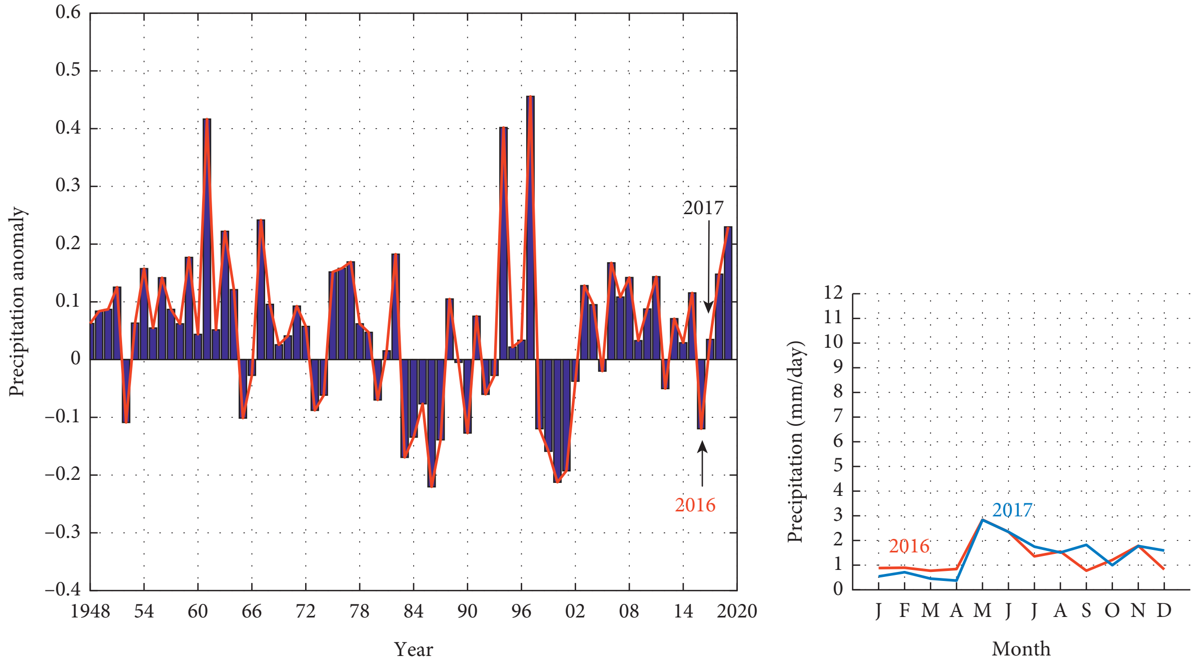
The monthly mean precipitation in the Arabian Sea and Sea of Oman regions indicates that the south Arabian Sea region is strongly influenced by the monsoon cycle with increased precipitation rate (Figure 13). However, a decreasing trend is observed towards north of the equator with minimum precipitation rate in the Sea of Oman region, meaning that this region is less influenced by the monsoon. Interestingly, a similar but opposite trend is observed in SSS circulation in these regions (Figure 12). For SSS circulation, it is perceived that the presence of highly saline water in the north and central basins of the Arabian Sea was due to excess evaporation over precipitation and intrusion of high saline water from two marginal seas: the Persian Gulf and the Red Sea. On the other side, seawater at the south Arabian Sea was significantly less in saline (<35) than that of central and north Arabian Sea regions. This decrease in salinity was mostly because of positive P-E (precipitation minus evaporation) or strong precipitation near the equator and intrusion of less saline water from the Bay of Bengal. It clearly indicates the coexistence of two water masses in the Arabian Sea region: (1) the Arabian Sea High Salinity Water



(a)

(b)

Figure 14: (a) Time series dataset of precipitation anomaly (1948–2019) in the Bay of Bengal (10–20°N, 80–100°E). The anomaly datasets are obtained from the NOAA precipitation reconstruction (PREC) project. (b) The monthly mean precipitation in 2016 and 2017 in the Bay of Bengal region.



(a)

(b)

Figure 15: As in Figure 14, but for the Arabian Sea region. (a) Arabian Sea (Eq–25°N, 50–70°E). (b) Arabian Sea.

(ASHSW) that is in excess of salinity (>36.9) and occupying the North and Central basins of the Arabian Sea and (2) the Bay of Bengal Water (BBW) that was lower in salinity (<35) and occupying the south basin of the Arabian Sea [53].

4. Summary and Conclusions

The present research has demonstrated the capability of Satellite and Argo datasets and their effectiveness in observing the upper-ocean thermal structure variability. The dynamics and behavior of the Arabian Sea are different from other seas of the world oceans, as it is situated at the intersection of all the three oceans and is highly exposed to the tropical climate. The present study provides a detailed description of the SST and ILD variability in association with the monsoon cycle, as this may have a significant impact on summer monsoon qualitative predictability (meaning the prediction of a weaker or stronger summer monsoon relative to the climatological mean) in the tropical oceans. In addition, adequate information on the climatological space-time variability in SST and ILD and the forcing mechanisms that determine this variability would provide critical information for oceanographers and modelers of space-time upper-ocean thermal structure variations in a region of important air-sea interaction.

On the seasonal average, we investigated that the winter monsoon warming in the south-central basin of the Arabian Sea was not similar and showed different variability in SST in both 2016 and 2017. The significant variability observed in the SST cycle in the south-central basin was the result of a large-scale dynamic such as El Niño in the Pacific Ocean that occurred in early 2016 [12]. In addition, we have noticed significant cooling (warming) in the SST cycle during the summer of 2016 (2017) in the south-central regions. This cooling (warming) in SST during the summer is associated with the evolution of extreme negative (weak positive) IOD events that occurred in the summer of 2016 (2017) and affected the western Indian Ocean with negative (positive) SST anomalies [4, 12, 13]. A notable feature of the SST cycle in the Arabian Sea and Sea of Oman regions is the existence of a strong upwelling along the Oman-Arabian coast that occurred during summer season. Most of the SST variability in the Sea of Oman and the north basin of the Arabian Sea was basically due to annual signatures of small-scale variations, especially along the Oman-Arabian coast near Ras al Hadd region that occurred in late spring and peak in summer. In short, it can be summarized that the dynamic mechanism of seasonal variations in the SST cycle in the Arabian Sea and Sea of Oman regions can be either the result of large-scale interannual atmospheric force such ENSO in the Pacific Ocean and internal independent ocean mechanism such as IOD or the annual signatures of the oceanic mesoscale variations that have occurred along the Oman-Arabian coastlines.

Surprisingly, quite different behavior was seen in the south-central Arabian Sea regions, where the ILD amplitude was always smaller in the winter monsoon and larger in the summer and postmonsoons. The smaller ILD amplitude in December to March indicated the weaker nature of the convective mixing during the winter monsoon, while larger amplitude in June to September represents the dominant nature of the winds stirring and Ekman pumping during the summer monsoon. On the other side, the emergence of thinner ILD amplitude in the summer monsoon is most probably due to the absence of wind action in the Sea of Oman region. The larger SST peaks and smaller ILD amplitudes during the summer monsoon period indicated that the Sea of Oman was shielded to some extent from the monsoons. In short, it can be summarized that, during the summer monsoon, winds forcing plays a significant role in mixing and deepening of ILD, where the convective fluxing or Ekman pumping were the essential dynamics that were responsible for deeper ILD during the winter monsoon. Consequently, out of four zones, the central Arabian Sea was strongly influenced by summer monsoon winds, where ILD was larger in amplitude with smaller SST peaks, while, on the other side, the Sea of Oman region showed smaller ILD amplitude with larger SST peaks and was, therefore, weakly affected by the monsoon. In this study, we provided sufficient information to predict the strength of a stronger or weaker monsoon from ILD amplitude, with space and time changing in the Arabian Sea and Sea of Oman regions. However, the prediction of large-scale dynamics and the impact of other ocean basins affecting the monsoon cycle, which are responsible for extreme weather conditions in the Arabian Sea, is still less explored and needs to be addressed in future research.

Data Availability

The Argo datasets are freely available online and can be downloaded from the website www.argo.ucsd.edu. The monthly mean SST and SST anomaly datasets are obtained from NOAA OISST (Optimum Interpolation Sea Surface Temperature) blended product, version 2.1. The monthly mean wind speed datasets are provided by the OAFlux project and available online at http://oaflux.whoi.edu. The long time precipitation datasets (1948–2019) are obtained from the NOAA precipitation reconstruction (PREC).

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors are thankful to the International Argo Project, NCEP Global Data Assimilation System (GDAS), Global Precipitation Climatology Project (GPCP), WHOI Objectively Analyzed Air-Sea Fluxes (OAFlux) project, and SMAPv3 (Soil Moisture Active Passive version 3) for providing the data. They wish to acknowledge the Acoustic Science and Technology Laboratory of Harbin Engineering University for analysis and graphics. This work was supported by an International Postgraduate Scholarship awarded by the Chinese Scholarship Council and National Key Research and Development Program of China (no. 2016YFC1400100).

References

[1] G. T. Walker, “Correlation in seasonal variations of weather. IX. A further study of world weather,” *Memoirs of the Indian Meteorological Department*, vol. 24, no. Part 9, pp. 275–332, 1924.

[2] V. Krishnamurthy and J. Shukla, “Intraseasonal and interannual variability of rainfall over India,” *Journal of Climate*, vol. 13, no. 24, pp. 4366–4377, 2000.

[3] K. Arpe, L. Dümenil, and M. A. Giorgetta, “Variability of the Indian monsoon in the ECHAM3 model: sensitivity to sea surface temperature, soil moisture, and the stratospheric quasi-biennial oscillation,” *Journal of Climate*, vol. 11, no. 8, pp. 1837–1858, 1998.

[4] S. Khan, S. Piao, G. Zheng et al., “Sea surface temperature variability over the tropical Indian Ocean during the ENSO and IOD events in 2016 and 2017,” *Atmosphere*, vol. 12, no. 5, p. 587, 2021.

[5] P. J. Webster, A. M. Moore, J. P. Loschnigg, and R. R. Leben, “Coupled ocean-atmosphere dynamics in the Indian Ocean during 1997-98,” *Nature*, vol. 401, no. 6751, pp. 356–360, 1999.

[6] N. H. Saji, B. N. Goswami, P. N. Vinayachandran, and T. Yamagata, “A dipole mode in the tropical Indian Ocean,” *Nature*, vol. 401, no. 6751, pp. 360–363, 1999.

[7] R. Murtugudde, J. P. McCreary Jr, and A. J. Busalacchi, “Oceanic processes associated with anomalous events in the Indian Ocean with relevance to 1997-1998,” *Journal of Geophysical Research: Oceans*, vol. 105, no. C2, pp. 3295–3306, 2000.

[8] G. Reverdin, D. L. Cadet, and D. Gutzler, “Interannual displacements of convection and surface circulation over the equatorial Indian Ocean,” *Quarterly Journal of the Royal Meteorological Society*, vol. 112, no. 471, pp. 43–67, 1986.

[9] R. Murtugudde, R. Seager, and P. Thoppil, “Arabian Sea response to monsoon variations,” *Paleoceanography*, vol. 22, no. 4, 2007.

[10] D. W. A. Leetmaa, “Arabian Sea cooling a preliminary heat budget,” *Oceanosraphy*, vol. 10, pp. 307–312, 1980.

[11] J. Shukla, “Effect of Arabian sea-surface temperature anomaly on Indian summer monsoon: a numerical experiment with the GFDL model,” *Journal of the Atmospheric Sciences*, vol. 32, no. 3, pp. 503–511, 1975.

[12] World Meteorological Organization (WMO), *Statement on the State of the Global Climate in 2016*, WMO, Geneva, Switzerland, 2016.

[13] World Meteorological Organization (WMO), *Statement on the State of the Global Climate in 2017*, WMO, Geneva, Switzerland, 2017.

[14] B. Lu, H. L. Ren, A. A. Scaife et al., “An extreme negative Indian Ocean Dipole event in 2016: dynamics and predictability,” *Climate Dynamics*, vol. 51, no. 1-2, pp. 89–100, 2018.

[15] F. Ninove, P. Y. Le Traon, E. Remy, and S. Guinehut, “Spatial scales of temperature and salinity variability estimated from Argo observations,” *Ocean Science*, vol. 12, p. 7, 2016.

[16] D. Roemmich, G. Johnson, S. Riser et al., “The Argo Program: observing the global oceans with profiling floats,” *Oceanography*, vol. 22, no. 2, pp. 34–43, 2009.

[17] B. Zhang, Y. Liu, D. Shang, and I. U. Khan, “A method for predicting radiated acoustic field in shallow sea based on wave superposition and ray,” *MDPI Applied Sciences*, vol. 10, no. 3, pp. 1–13, 2020.

[18] http://www.ifremer.fr/coriolis/cdc/default.htm.

[19] J. Boutin, N. Martin, X. Yin, J. Font, N. Reul, and P. Spurgeon, “First assessment of SMOS data over open ocean: part II-sea surface salinity,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 5, pp. 1662–1675, 2012.

[20] R. F. Adler, G. J. Huffman, A. Chang et al., “The version-2 global precipitation climatology project (GPCP) monthly precipitation analysis (1979-present),” *Journal of Hydrometeorology*, vol. 4, no. 6, pp. 1147–1167, 2003.

[21] G. J. Huffman, R. F. Adler, D. T. Bolvin, and G. Gu, “Improving the global precipitation record: GPCP version 2.1,” *Geophysical Research Letters*, vol. 36, no. 17, p. L17808, 2009.

[22] T. Meissner, F. J. Wentz, and D. M. L. Vine, “The salinity retrieval algorithms for the NASA aquarius version 5 and SMAP version 3 releases,” *Remote Sensing*, vol. 10, no. 7, p. 1121, 2018.

[23] A. B. Kara, P. A. Rochford, and H. E. Hurlburt, “Mixed layer depth variability over the global ocean,” *Journal of Geophysical Research: Oceans*, vol. 108, no. C3, p. 571, 2003.

[24] K. H. Bathen, “On the seasonal changes in the depth of the mixed layer in the North Pacific Ocean,” *Journal of Geophysical Research*, vol. 77, no. 36, pp. 7138–7150, 1972.

[25] P. Lamb, “On the mixed-layer climatology of the north and tropical Atlantic,” *Tellus*, vol. 36, pp. 292–305, 1984.

[26] R. R. Rao, R. L. Molinari, and J. F. Festa, “Evolution of the climatological near-surface thermal structure of the tropical Indian Ocean: 1. Description of mean monthly mixed layer depth, and sea surface temperature, surface current, and surface meteorological fields,” *Journal of Geophysical Research*, vol. 94, no. C8, pp. 10801–10815, 1989.

[27] R. R. Rao and R. Sivakumar, “Seasonal variability of sea surface salinity and salt budget of the mixed layer of the north Indian Ocean,” *Journal of Geophysical Research: Oceans*, vol. 108, no. C1, pp. 9–14, 2003.

[28] A. B. Kara, P. A. Rochford, and H. E. Hurlburt, “Mixed layer depth variability and barrier layer formation over the North Pacific Ocean,” *Journal of Geophysical Research: Oceans*, vol. 105, no. C7, pp. 16783–16801, 2000.

[29] N. K. Vissa, A. N. V. Satyanarayana, and B. Prasad Kumar, “Comparison of mixed layer depth and barrier layer thickness for the Indian Ocean using two different climatologies,” *International Journal of Climatology*, vol. 33, no. 13, pp. 2855–2870, 2013.

[30] K. Wyrtki, “The thermal structure of the eastern Pacific Ocean,” *Deutschen Hydrographischen Zeitschrift, Erga¨nzungsheft A*, vol. 8, pp. 6–84, 1964.

[31] J. F. Price, R. A. Weller, and R. Pinkel, “Diurnal cycling: observations and models of the upper ocean response to diurnal heating, cooling, and wind mixing,” *Journal of Geophysical Research*, vol. 91, no. C7, pp. 8411–8427, 1986.

[32] J. Sprintall and M. Tomczak, “Evidence of the barrier layer in the surface layer of the tropics,” *Journal of Geophysical Research*, vol. 97, no. C5, pp. 7305–7316, 1992.

[33] S. Hosoda, T. Ohira, K. Sato, and T. Suga, “Improved description of global mixed-layer depth using Argo profiling floats,” *Journal of Oceanography*, vol. 66, no. 6, pp. 773–787, 2010.

[34] R. O. R. Y. Thompson, “Climatological numerical models of the surface mixed layer of the ocean,” *Journal of Physical Oceanography*, vol. 6, no. 4, pp. 496–503, 1976.

[35] C. de Boyer Montégut, G. Madec, A. S. Fischer, A. Lazar, and D. Iudicone, “Mixed layer depth over the global ocean: an examination of profile data and a profile-based climatology,” *Journal of Geophysical Research: Oceans*, vol. 109, no. C12, 2004.

[36] C. de Boyer Montégut, J. Mignot, A. Lazar, and S. Cravatte, "Control of salinity on the mixed layer depth in the world ocean: 1. General description," *Journal of Geophysical Research: Oceans*, vol. 112, no. C6, 2007.

[37] B. An, C. Zhang, D. Shang, Y. Xiao, and I. U. Khan, "A combined finite element method with normal mode for the elastic structural acoustic radiation in shallow water," *Journal of Theoretical and Computational Acoustics*, vol. 28, no. 4, Article ID 2050004, 2020.

[38] M. Tomczak and J. S. Godfrey, *Regional Oceanography: An Introduction*, Elsevier, Amsterdam, Netherlands, 2013.

[39] S. Khan, S. Yang, X. Bingchen, D. Bradley, N. Ahmed, and P. Shengchun, "Subsurface highly saline water variability in the Gulf of Oman," *OCEANS 2019-Marseille*, pp. 1–4, IEEE, Marseille, France, 2019.

[40] P. S. Suthinkumar, C. A. Babu, and H. Varikoden, "Spatial distribution of extreme rainfall events during 2017 southwest monsoon over Indian subcontinent," *Pure and Applied Geophysics*, vol. 176, no. 12, pp. 5431–5443, 2019.

[41] N. Quinn and D. Johnson, "Cold water upwellings cover Gulf of Oman coral reefs," *Coral Reefs-Journal of the International Society for Reef Studies*, vol. 15, no. 4, p. 214, 1996.

[42] S. L. Coles, "Reef corals occurring in a highly fluctuating temperature environment at Fahal Island, Gulf of Oman (Indian Ocean)," *Coral Reefs*, vol. 16, no. 4, pp. 269–272, 1997.

[43] J. Wiggert, R. Hood, K. Banse, and J. Kindle, "Monsoon-driven biogeochemical processes in the Arabian Sea," *Progress in Oceanography*, vol. 65, no. 2-4, pp. 176–213, 2005.

[44] M. R. Claereboudt, *Reef Corals and Coral Reefs of the Gulf of Oman*, Historical Association of Oman, Oman, 2006.

[45] T. Watanabe, A. Suzuki, S. Minobe et al., "Permanent El Niño during the Pliocene warm period not supported by coral evidence," *Nature*, vol. 471, no. 7337, pp. 209–211, 2011.

[46] T. K. Watanabe, T. Watanabe, A. Yamazaki, M. Pfeiffer, D. Garbe-Schönberg, and M. R. Claereboudt, "Past summer upwelling events in the Gulf of Oman derived from a coral geochemical record," *Scientific Reports*, vol. 7, no. 1, p. 4568, 2017.

[47] A. R. Al-Azri, S. A. Piontkovski, K. A. Al-Hashmi, J. I. Goes, and H. R. Do Gomes, "Chlorophyll a as a measure of seasonal coupling between phytoplankton and the monsoon periods in the Gulf of Oman," *Aquatic Ecology*, vol. 44, no. 2, pp. 449–461, 2010.

[48] A. S. Bower, H. D. Hunt, and J. F. Price, "Character and dynamics of the Red Sea and Persian Gulf outflows," *Journal of Geophysical Research: Oceans*, vol. 105, no. C3, pp. 6387–6414, 2000.

[49] A. S. Bower, W. E. Johns, D. M. Fratantoni, and H. Peters, "Equilibration and circulation of Red Sea outflow water in the Western Gulf of Aden," *Journal of Physical Oceanography*, vol. 35, no. 11, pp. 1963–1985, 2005.

[50] S. Pous, X. Carton, and P. Lazure, "Hydrology and circulation in the Strait of Hormuz and the Gulf of Oman results from the GOGP99 experiment: 2. Gulf of Oman," *Journal of Geophysical Research: Oceans*, vol. 109, no. C12, 2004.

[51] T. Izumo, C. B. Montégut, J.-J. Luo, S. K. Behera, S. Masson, and T. Yamagata, "The role of the Western Arabian Sea upwelling in Indian monsoon rainfall variability," *Journal of Climate*, vol. 21, no. 21, pp. 5603–5623, 2008.

[52] A. Chanda, S. Das, A. Mukhopadhyay et al., "Sea surface temperature and rainfall anomaly over the Bay of Bengal during the El niño-southern oscillation and the extreme Indian Ocean dipole events between 2002 and 2016," *Remote Sensing Applications: Society and Environment*, vol. 12, pp. 10–22, 2018.

[53] C. de Boyer Montégut, F. Durand, R. Bourdallé-Badie, and B. Blanke, "Role of fronts in the formation of Arabian Sea barrier layers during summer monsoon," *Ocean Dynamics*, vol. 64, no. 6, pp. 809–822, 2014.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9955212, 10 pages
https://doi.org/10.1155/2021/9955212

Research Article

Estimating Children Engagement Interacting with Robots in Special Education Using Machine Learning

George A. Papakostas,[1] George K. Sidiropoulos,[1] Chris Lytridis,[1] Christos Bazinas,[1] Vassilis G. Kaburlasos,[1] Efi Kourampa,[2] Elpida Karageorgiou,[2] Petros Kechayas,[3] and Maria T. Papadopoulou[4]

[1] *HUman-MAchines INteraction Laboratory (HUMAIN-Lab), Department of Computer Science, International Hellenic University, 65404 Kavala, Greece*
[2] *Family Center KPG, 54352 Thessaloniki, Greece*
[3] *Department of Clinical Psychology, Papageorgiou General Hospital, Aristotle University of Thessaloniki, 56403 Thessaloniki, Greece*
[4] *Division of Child Neurology and Metabolic Disorders, 4th Department of Pediatrics, Papageorgiou General Hospital, Aristotle University of Thessaloniki, 56403 Thessaloniki, Greece*

Correspondence should be addressed to George A. Papakostas; gpapak@cs.ihu.gr

Received 10 March 2021; Accepted 30 May 2021; Published 19 June 2021

Academic Editor: Bhawani Shankar Chowdhry

The task of child engagement estimation when interacting with a social robot during a special educational procedure is studied. A multimodal machine learning-based methodology for estimating the engagement of the children with learning difficulties, participating in appropriate designed educational scenarios, is proposed. For this purpose, visual and audio data are gathered during the child-robot interaction and processed towards deciding an engaged state of the child or not. Six single and three ensemble machine learning models are examined for their accuracy in providing confident decisions on in-house developed data. The conducted experiments revealed that, using multimodal data and the AdaBoost Decision Tree ensemble model, the children's engagement can be estimated with 93.33% accuracy. Moreover, an important outcome of this study is the need for explicitly defining the different engagement meanings for each scenario. The results are very promising and put ahead of the research for closed-loop human centric special education activities using social robots.

1. Introduction

Nowadays, we are witnessing the fourth industrial revolution commonly known in Europe as Industry 4.0 [1]. One of the most important parts of this revolution is the extension of the robots' usage beyond the industrial environments to social activities interacting directly with humans. This new kind of robot named social robots shows increased interaction capabilities, characterized by a certain degree of intelligence, and is very much safe to interact with children in any type of education.

Our interest here is the case of special education, which draws increased attention from modern societies aiming at providing equal opportunities to children with special needs to develop their skills. Recent studies have demonstrated the positive role of social robots in delivering special education in person [2, 3] as well as in distance [4].

The ultimate goal of an advanced child-robot interaction is the establishment of a high level of an intelligence communication channel, in a closed-loop configuration with the child being at the center of the educational scenario. This goal can be achieved by developing efficient sensing mechanisms to the robot side, such as automatic engagement measuring, which will permit the robot to adapt its behavior or even the execution of the educational scenario [5], towards increasing the success—increased knowledge transfer and achievement of the learning objectives—of the education delivery. Therefore, the development of a robust methodology for measuring the engagement state of the children in special education constitutes a challenging problem to tackle.

Children with learning disabilities (LD) are identified as having typical intelligence but manifest specific difficulties that interfere with their task performance and academic achievement [6]. This repeated failure and frustration experienced by the children with LD reduce their self-efficacy leading to a sense of helplessness, which is associated with lack of motivation and academic disengagement [7–9]. As academic engagement refers to active participation and attention and focuses on the task during the learning process, disengagement refers to apathy and lack of interest. The degree to which students are engaged is a critical precursor to learning, as without academic engagement, students are unlikely to benefit from instructions [10]. In other words, the more students are engaged, the more they learn [11]. Therefore, the development of a robust methodology for measuring the engagement state of children with LD constitutes a challenge.

Although several methods [12] for measuring the engagement level during child-robot interaction have been presented in the literature, all these attempts were focused on children with Autism Spectrum Disorder (ASD), and their experimental study was limited with a small number of children.

Taking into consideration the fact that children with learning disabilities are associated with maladaptive engagement compared with their typically developed peers [13, 14], it would be of great importance to have knowledge of each child's engagement level through social robots in order to use them in intervention programs that aim to promote child's learning by increasing their involvement in all kinds of learning tasks. Thus, as confirmed through researches, interventions using social robots as a tool to support the learning process have been demonstrated to enhance students' motivational skills, maintenance of engagement, and compliance during instructional interactions [15, 16]. The research of Pistoia et al. [17], which is one of the first attempts to investigate the use of a social robot in students with dyslexia, confirms that the presence of the robot to support the learning process showed high levels of response and engagement during child-robot interaction.

In this context, this work contributes along with the following directions:

(1) A definition of the "Intelligent Interaction" based on psychology is provided

(2) A machine learning-based methodology that allows a social robot to interact with intelligence with the child is proposed

(3) The proposed methodology is evaluated with a large amount of in-house developed real data

(4) For the first time the case of children with learning difficulties is considered for measuring their engagement state during interaction with the social robot

(5) The need for customized engagement measuring methods based on the characteristics of the deployed scenario is touched for the first time

The rest of the paper is organized as follows: Section 2 provides a snapshot of the related work and Section 3 presents the definition of the "Intelligent Interaction," the information of the designed educational scenarios, and the details of the proposed methodology. Section 4 provides the experimental study with the corresponding results. Section 5 discusses the results, concludes this study, and lays out the future work.

2. Related Work

Sidner et al. [18] proposed and Ahmad et al. [19] rephrased a general definition of the concept of engagement during human-robot interaction: "Engagement is the process by which interactors start, maintain, and end their perceived connection to each other during the interaction." Measuring engagement of humans, executing a specific activity, constitutes a highly informative indication for analyzing the effectiveness of the activity design. This measurement can help the improvement of the design towards achieving the desired outcomes relative to the executed activity.

For this purpose, several methodologies have been proposed to measure the engagement of a user playing a video game [20], of a person when working [21], of students in a classroom [22], of TV viewers [22], of a consumer when purchasing products [23], and so on. Measuring engagement of a child with special needs during an educational process and/or intervention is very challenging due to the specially designed scenarios and interaction schemes, which must attract their attention and maintain engagement.

Early outstanding work for measuring the engagement of children with a game companion was proposed by Castellano et al. [24, 25] by using a multimodal processing scheme based on visual and contextual information. Moreover, Hernandez et al. [26] proposed a method to measure the engagement of children, which were difficult to engage during social interactions. In [26], wearable sensors were used to measure the electrodermal activity of the children and a Support Vector Machine (SVM) classifier was applied to classify the children being engaged or not. In [27], acoustic and linguistic data were utilized to detect the social engagement in conversational interactions of children with ASD and their parents, using an SVM classifier. The first in-depth study of measuring the engagement of children when interacting with social robots was proposed by Anzalone et al. [28]. In this work, the researchers analyzed visual information in a static and dynamic perspective, in several case studies of ASD child-robot interaction. Rudovic et al. [29] presented a very interesting study regarding the engagement measuring across cultures, which revealed that the engagement level of 30 ASD children can be increased by taking into account the cultural differences.

Recently, with the advent of deep learning technology, several attempts have been pointed out for measuring engagement during a child-robot interaction using advanced intelligent models. Rudovic et al. [30] proposed the CultureNet model based on the typical ResNet-50 architecture for estimating the engaged or not engaged children of different cultures interacting with NAO robot in robot-assisted therapy for children with Autism Spectrum Condition (ASC). In [31], Hadfield et al. proposed a deep learning model consisting of three fully connected layers and a single LSTM layer, while the used features are computed using visual data relative to the position of the child's body parts. The reported results were of almost 80% accuracy, but the limited number (3) of Typical Developed (TD) children can justify the quite low accuracy. In a very recent work, Del Duchetto et al. [32] tried to measure the engagement level in human-robot interaction utilizing Convolutional Neural Networks (CNNs) and LSTM model. The novelty of the work in [33] is the tackling of the engagement estimation as a regression problem, aiming at providing a scalar value for the engagement level during human-robot interaction. The reported results were very promising with Mean Squared Error (MSE) 0.126.

Although the previous approaches have contributed significantly to the engagement estimation in human-robot interaction, they possess some limitations: (1) they were applied mostly on adults or children with TD or ASD, without examining other categories of children with special needs, such as children with learning difficulties; (2) they were experimented with a limited number of children; and (3) they did not study the engagement estimation in the framework of appropriately designed intervention scenarios or the designed scenarios were few and very simple.

It is important to realize the need to analyze and measure the engagement of children with learning difficulties. Considering dyslexia as the most frequent learning difficulty, Uta Frith [33] proposed a three-level theoretical framework for the interpretation of dyslexia, namely, behavioral, cognitive, and biological. In this context, Frith also distinguished the role of the environmental level that interacts with the abovementioned three levels. Therefore, dyslexia students interacting with a social robot can learn easier due to its interactive and fun performance, which also allows students to take their time during a learning task. In addition, a social robot engages pupils in mental information processing and captures their attention [34].

After reviewing the applications of social robots in special education from the international literature, we found that the usage of social robots in supporting the educational procedure of children with learning difficulties is limited. This observation contradicts the educational needs of a large percentage of the world's population, which accounts for 10–15% [35]. We believe that the high percentage of the population showing learning difficulties imposes the targeting of this part of the population as a potential application field for using social robots.

The current study aims to complement the previous works by investigating the engagement measuring when children with learning difficulties are interacting with the social robot NAO. The number of children that participated in the experiments was 10, while child psychologists carefully designed 10 scenarios, executed by each child.

3. Materials and Methods

3.1. Intelligent Interaction: A Definition. In order to understand the real needs for an engagement measuring methodology, it is crucial to provide a definition of what is the meaning of an "Intelligent Interaction."

Considering the work of the psychologist Howard Gardner [36] regarding the type of intelligence, nine different types of intelligence can be considered. From these nine types of intelligence, the following five deal with the interaction of a human with the surrounding environment:

(1) Linguistic intelligence: ability to find the right words to express what do you mean

(2) Visual-spatial intelligence: having awareness of the surrounding environment

(3) Bodily-kinesthetic intelligence: coordinating the mind with the body

(4) Interpersonal intelligence: sensing children's feelings and motives

(5) Logical-mathematical intelligence: quantifying things, making hypotheses, and proving them

From the engineering point of view though, the previous interaction-oriented intelligence can be summarized to the following two levels of intelligence:

(1) 1st level of intelligence: ability to analyze the sensory data in order to understand the surrounding environment

(2) 2nd level of intelligence: establishing a human-like closed-loop communication with the child

The above two levels of intelligence enclose the aforementioned five types of intelligence defined in terms of psychology and can be the ultimate goals of any research dealing with human-robot interaction.

An important part of the above two levels of intelligence is the measuring of the child's engagement state by processing the sensory data (1st level) for adapting the robot's behavior and/or the educational scenario towards establishing a closed-loop communication channel (2nd level).

3.2. Educational Scenarios. For the sake of this study, five child psychologists (three from the "Family Center KPG, Thessaloniki, Greece" and two from the "Department of Clinical Psychology, Papageorgiou General Hospital, Thessaloniki, Greece") of our research team designed ten different educational scenarios for children with learning difficulties, as part of the national project titled "Social Robots as Tools in Special Education (SRTSE)" [37]. It is worth noting that each child executed each scenario on different days. More precisely, each child executed two scenarios per week and the average duration of each scenario was 35 minutes.

Table 1 shows what types of activities are included in each scenario.

The scenarios include the following types of activities:

(i) Meet/greet
(ii) Text decoding, comprehension, and reading
(iii) Phonology composition, decomposition, discrimination, and addition
(iv) Memory
(v) Robot-child relaxation game
(vi) Story listening and telling
(vii) Sentence structuring
(viii) Strategic visual representation

3.3. Proposed Methodology. Two are the main features of the proposed methodology: (1) the usage of multimodal data consisting of visual and audio modalities and (2) the usage of a machine learning model that provides the decision about the engagement state of the child. In the following subsections, the modules of the designed methodology depicted in Figure 1 are described in detail.

3.3.1. Multimodal Sensing. The sensing capabilities of the used social robot mainly control the type of sensory data to process in order to decide the engagement state of the child during the interaction. Our study considers the well-known NAO robot as the robot that is able to interact with the child, but other social robots [38] could also be used. This robot is equipped with two identical RGB video cameras located in the forehead and a microphone; thus, it can provide visual and audio sensing capabilities.

(1) Visual Sensing. The visual sensing capabilities of the NAO robot permit the acquisition of video frames that include the child's body and face. From each video frame, the body pose is extracted using the library [39], consisting of 25 key points (2 on the torso, 6 on the hands, 12 on the legs, and 5 on the head), as depicted in Figure 2(a). In addition, 68 key points called facial landmarks are extracted (see Figure 2(b)), from the child's face using the OpenFace library [40]. It is worth noting that the computed facial landmarks are used to define the child's emotional state in compliance with the Facial Action Coding System (FACS) [41]. Finally, the eye contact between the child and the robot is detected using the OpenGaze library [42] and following the methodology proposed by Xucong Zhang et al. [43].

(2) Audio Sensing. During the interaction with the child, the robot needs to keep facing the child at all times, in order for the robot to record and analyze the child's speech, by providing additional information related to the engagement state of the child.

3.3.2. Feature Extraction. The abovementioned multimodal sensing mechanism aims at collecting sensory raw data. This data, which has the form of 2D Cartesian points belonging to the child, is further processed to construct more informative descriptions named features. The feature extraction procedure is applied on the video frames (640 × 480 pixels resolution) captured every 0.7 secs (1.4 fps) by using non-overlapping sliding windows of 60 secs. Although the camera of the NAO robot has 2.5 fps for 640 × 480 video resolution, in a WiFi connection mode, in our case, the real-time performance of our system is 1.4 fps due to the execution of the algorithms. Moreover, it is decided to set the processing time window to 60 secs, in order to include enough event transitions and to help the manual annotation of the data. The features that are finally computed are the following:

(1) Feature 1: number of blinks: the blinks count of the child on average
(2) Feature 2: mean movement of the body in pixels
(3) Feature 3: if the child's body was turned away from the robot (0 or 1)
(4) Feature 4: percentage of the time window within which there was eye contact by the child
(5) Feature 5: emotion (happy, sad, surprised, fear, anger, disgust, or contempt)
(6) Feature 6: emotion intensity (0–5)
(7) Feature 7: if the child's head was turned away from the robot (0 or 1)
(8) Feature 8: mean response time (set to −1 if the scenario did not require a response from the child)
(9) Feature 9: mean voice level (in RMS)
(10) Feature 10: percentage of the time window within which the child was silent
(11) Feature 11: percentage of the time window within which the child was speaking

It should be noted that almost all the above visual features are computed by tracking and processing the extracted key points. For example, for a specific frame, the emotion is determined by combining the FACS corresponding to each feeling (Table 2), averaging their intensities, and choosing the emotion that has the highest intensity. In addition, features 3 and 7 are determined by counting the number of states (0 and 1) in the time window and choosing the one with the highest number of occurrences. Lastly, to determine if the child is speaking or not, we check if the voice volume is higher than 350 RMS and the mean voice level considers levels where the child is speaking.

To summarize, for each 60 secs video frame, a feature vector $FV \in R^{11}$ is assigned, which is also manually annotated by three experienced child psychologists to an engaged time slot or not. The extracted features from the educational scenarios are used to train the machine learning model, so it will able to detect the engagement state of the child.

3.3.3. Machine Learning Models. Herein, the detection of the child's engagement state (engaged or not engaged) is accomplished by solving a typical two-class classification problem by using a machine learning classifier. Machine learning has been proved to be an efficient technology in

Table 1: Educational scenarios' characteristics.

Scenario	Types of activities included
S1	Meet/greet, text decoding, phonology (de)composition, memory, and robot-child relaxation game
S2	Meet/greet, phonetic discrimination, text reading, decoding, and comprehension
S3	Meet/greet, story listening and telling, and sentence structuring
S4	Text comprehension and visual representation
S5	Phonemic addition, sentence playback from memory, and robot-child relaxation game
S6	Meet/greet, sentence playback from memory, and reading enhancement
S7	Meet/greet, phonetic awareness, and robot-child relaxation game
S8	Meet/greet, acoustic vocal discrimination, and acoustic syllable discrimination
S9	Memory enhancement and text decoding
S10	Text reading and robot-child relaxation game

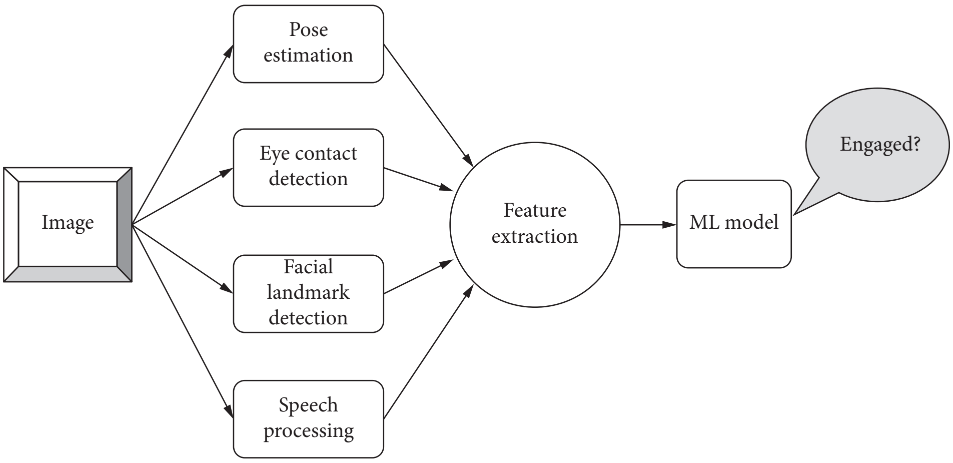
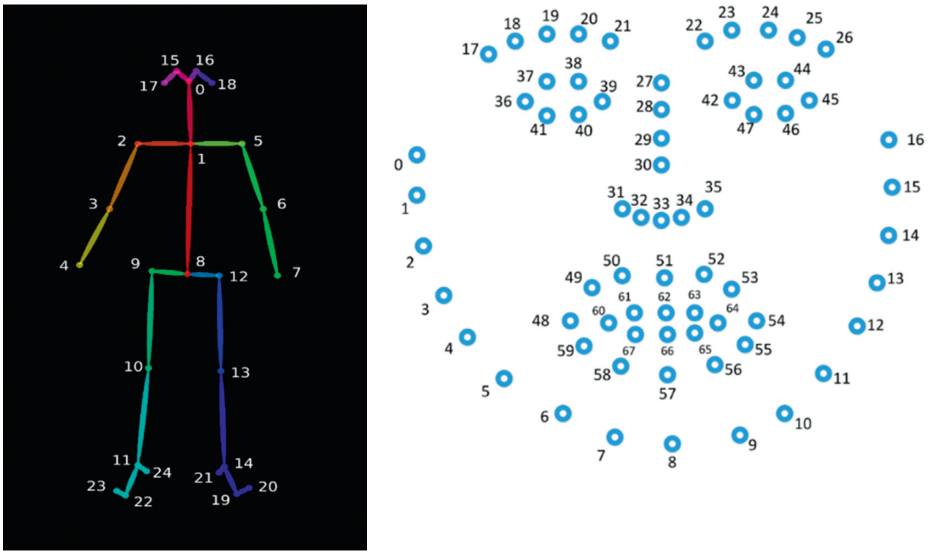


Figure 1: Block diagram of the proposed methodology.



(a)

(b)

Figure 2: Visual sensing extracted data. (a) Body pose detection [39]. (b) Facial landmark (action units) [44] detection.

many disciplines such as signal processing [45] and computer vision [46]. More precisely, six traditional machine learning models, the Support Vector Machine (SVM) model with two different kernels (RBF and poly), the Decision Tree, the K-NN, the Naïve Bayes (NB), the Multilayer Perceptron (MLP), and the Extreme Learning Machine (ELM) classifiers, are examined.

Additionally, three ensemble models are also considered, the Random Forest (RF), the AdaBoost Decision Tree, and AdaBoost Naïve Bayes ones. The advantage of the ensemble

Table 2: Facial action coding system (FACS).

Emotion	Action units (AU)	Description
Happiness/joy	6 + 12	Cheek raiser and lip corner puller
Sadness	1 + 4 + 15	Inner brow raiser, brow lowerer, and lip corner depressor
Surprise	1 + 2 + 5 + 26	Inner brow raiser, outer brow raiser, upper lid raiser, and jaw drop
Fear	1 + 2 + 4 + 5 + 7 + 20 + 26	Inner brow raiser, outer brow raiser, brow lowerer, upper lid raiser, lid tightener, lip stretcher, and jaw drop
Anger	4 + 5 + 7 + 23	Brow lowerer, upper lid raiser, lid tightener, and lip tightener
Disgust	9 + 15	Nose wrinkler, lip corner depressor, and lower lip depressor
Contempt	12 + 14	Lip corner puller and dimpler

classifiers is that they combine “weak learners” with strong ones, by reducing the bias and variance of the learner. The first ensemble uses the Bagging [47] and the last two use the AdaBoost [48] training techniques.

Most of the machine learning models owing to a set of configuration parameters that enables them to adjust their performance are subject to the considered problem and must be carefully selected.

4. Experimental Study

In order to study the performance of the proposed methodology, a set of experiments was arranged. The experiments were carried out using the scikit-learn [49] Machine Learning Library for Python, on Python version 2.7. Moreover, the experiments were conducted on a laptop computer equipped with Intel i7-6700HQ CPU, 8 GB DDR4 RAM, and GTX 960M GPU.

4.1. Dataset Design. For the sake of the experiments, 10 children participated in the ten scenarios (see Table 1), 2 girls and 8 boys, aged from 9 to 10 years. Each scenario is executed in a classroom with the participation of a child, the NAO robot, and a child psychologist sitting behind the NAO robot. The robot also needs to keep facing the child at all times, in order for the speech recognition module of the robot to work more accurately, since in this position the microphones are oriented to the source of the sound [50]. From the recorded video files, a dataset of 819 samples, with 11 features for each sample, was designed. From these samples, 99 samples corresponded to children being engaged, while 720 samples corresponded to children being not engaged. Three experienced child psychologists derived the ground truth data after manual annotation. Since this dataset is imbalanced, an oversampling technique was employed, called Synthetic Minority Oversampling Technique (SMOTE) [51], in order to balance the dataset, by containing the same number of samples for each class. The final balanced dataset includes 1440 samples (720 per class).

4.2. Settings of the Experiments. A 10-fold cross-validation grid search technique [52] was applied in order to select the best parameters set for each model. The resulting parameters that optimize the accuracy of each model are presented in Table 3.

The performance of each model was evaluated using the Precision, Recall, Accuracy, and F-measure indices [53]. These measures are widely used in machine learning to evaluate the performance of a model. They are taking into account the True Positive (TP) and True Negative (TN) cases, which correspond to those cases correctly identified as positive or negative, respectively, and False Positive (FP) and False Negative (FN) cases, which are falsely identified as positive or negative, respectively.

Accuracy is the proportion of the total number of correct predictions and is calculated from the equation

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (1)$$

Precision is the proportion of the correct predicted positive results and is calculated from the equation

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (2)$$

Recall is the proportion of correct positive results and is calculated from the equation

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (3)$$

F-measure combines both Precision and Recall and is the harmonic mean of those indices, calculated as follows:

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

4.3. Results. A k-fold (with $k = 10$) cross-validation technique is followed for the evaluation of each machine learning model in estimating the children’s engagement state. According to this training and testing protocol, the initial dataset of 1440 samples is divided into 10 equal and nonoverlapped subsets of 144 samples. Each one of these subsets is used to test the model trained with the remaining nine subsets. The process is repeated k times by using different subsets for testing only once. The results of the k experiments are averaged in order to conclude the generalization ability of each model. Table 4 summarizes the prediction performance of each model.

The results of Table 4 reveal two important conclusions. The first one is that the initial hypothesis that the children’s engagement can be measured by using multimodal data consisting of combined behavioral, pose, and emotional information is justified experimentally since the accuracy of the models is very high (up to 93.33%).

The second conclusion is that the AdaBoost Decision Tree model outperforms the other models, by a significant

Table 3: Settings of the ML models.

ML model	Best parameters
SVM (RBF)	$C = 10$, tol = 0.1, and gamma = 0.001
SVM (poly)	$C = 10$, tol = 0.0001, and gamma = "scale"
Decision Tree	Splitter = "best," min samples leaf = 10, criterion = "gini," max features = none, and max depth = 4
k-NN	n neighbors = 2, weights = "uniform," leaf size = 20, and algorithm = "ball tree"
Naïve Bayes	(Nothing to configure)
MLP	Solver = "Adam," learning rate = "constant," hidden layer sizes = (80, 40), tol = 10.0, and alpha = 0.01
ELM	Alpha = 100, n_hidden = 80, and rbf_width = 0.256
Random Forest	Max features = "sqrt," *n* estimators = 4, criterion = "gini," max depth = 15, and min samples leaf = 15
AdaBoost Decision Tree	Criterion = "entropy", max depth = 15, max features = "auto," splitter = "best," and min samples leaf = 5
AdaBoost Naïve Bayes	(Nothing to configure)

Table 4: Predicted results.

ML model	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
SVM (RBF)	91.53	91.86	91.53	91.51
SVM (poly)	57.92	61.50	57.92	54.74
Decision Tree	87.01	88.59	87.01	86.83
k-NN	88.33	88.81	88.33	88.30
Naïve Bayes	78.12	81.41	78.12	77.49
MLP	78.75	79.73	78.75	78.50
ELM	89.72	90.16	89.16	89.66
Random Forest	88.54	89.61	88.54	88.42
AdaBoost Decision Tree	**93.33**	**94.09**	**93.33**	**93.28**
AdaBoost Naïve Bayes	82.29	83.48	82.29	82.09

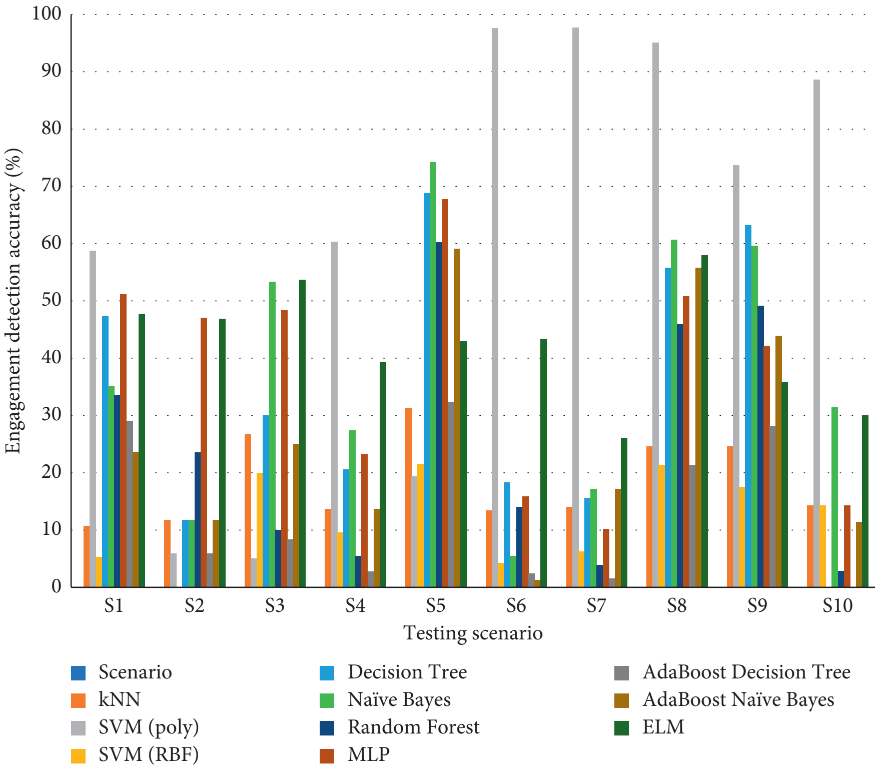


Figure 3: Engagement detection accuracy for the case of the modified leave-one-out training strategy.

factor in some cases, followed by the SVM (RBF). Despite the high accuracy of the AdaBoost Decision Tree model, the additional high Precision, Recall, and F-measure constitute the evidence that the model is able to estimate the children's engagement of unseen data with minimum False Positive (FP) and False Negative (FN) decisions. The outperformance of the ensemble methods was expected since these models are more complex and they provide the final decision considering the outcomes of multiple single classifiers working in a complementary way. Among the ensemble models, the AdaBoost shows the best performance, a result that reveals the ability of the sequential topology of the bootstrapping to improve the classification performance. On the other hand, the Bagging topology of the Random Forest model implies that the weak classifiers of the model operate with similar data, meaning that the dataset is quite homogenous, without including significant variations.

Moreover, in order to examine the bias of the machine learning models in the data of a specific scenario, a modified leave-one-out training strategy was applied with the samples that were left out in each case where all the samples corresponding to a specific scenario and the training were done with all other samples of the other scenarios. For each test case, the training data were again augmented to tackle the existing issue of the imbalanced number of samples per class. For example, in the first fold, the models were trained with the samples corresponding to all the scenarios except the first one and then tested with the samples corresponding to the first scenario and so on for each fold. Figure 3 depicts the performance of the machine learning models for each scenario when its data samples are used to test the models.

The results presented in Figure 3 reveal that the SVM (poly) model shows the lowest bias in the training data since it has the highest detection accuracy in 7 out of 10 training folds. Moreover, an interesting observation of this experiment is the different “definitions” of children’s engagement state in each scenario, since the performance of each model varies with the scenario type.

5. Discussion and Conclusion

The task of engagement detection of a child with learning difficulties interacting with a social robot for establishing a two-way intelligent interaction was studied in this work. The detection procedure was tackled as a two-class classification problem solved with high success by applying a machine learning model. The proposed methodology uses multimodal data (visual and audio) that describe the behavior of the child during the interaction. The initial hypothesis that an engaged child with learning difficulties can be identified by processing the body and head poses, the facial expressions, the eye contact, and the speech was accepted following the proposed method. However, this study brought to light the possible different “definitions” of engagement that apply in each educational scenario. This outcome is very important since it paves the way for more customized engagement measuring techniques oriented to the specific scenarios under deployment, towards providing an optimal interaction strategy.

In addition to the investigation of developing scenario-based engagement measuring methods, future work will consider the time parameter for each extracted feature and the handling of them as time series by deploying regression ML models such as Long Short Term Memory (LSTM) for predicting the engagement level at discrete time steps.

Data Availability

The data used in this research will be provided upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research had been cofinanced by the European Union and Greek National Funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH–CREATE–INNOVATE (Project code T1EDK-00929).

References

[1] M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” in *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 3928–3937, IEEE, Hawai, HI, USA, January 2016.

[2] L. I. Ismail, T. Verhoeven, J. Dambre, and F. Wyffels, “Leveraging robotics research for children with autism: a review,” *International Journal of Social Robotics*, vol. 11, no. 3, pp. 389–410, 2019.

[3] V. Holeva, V.-A. Nikopoulou, M. Papadopoulou, E. Vrochidou, G. A. Papakostas, and V. G. Kaburlasos, “Toward robot-assisted psychosocial intervention for children with autism spectrum disorder (ASD),” in *Proceedings of the Social Robotics*, M. A. Salichs, S. S. Ge, E. I. Barakova et al., Eds., pp. 484–493, Springer International Publishing, Madrid, Spain, November 2019.

[4] C. Lytridis, C. Bazinas, G. Sidiropoulos et al., “Distance special education delivery by social robots,” *Electronics*, vol. 9, no. 6, p. 1034, 2020.

[5] M. Ahmad, O. Mubin, and J. Orlando, “A systematic review of adaptivity in human-robot interaction,” *Multimodal Technologies and Interaction*, vol. 1, no. 3, p. 14, 2017.

[6] H. L. Swanson, “Learning disabilities: assessment, identification, and treatment,” in *The Oxford Handbook of School Psychology*, M. A. Bray and T. J. Kehle, Eds., pp. 334–350, Oxford University Press, Oxford, UK, 2011.

[7] M. Boekaerts, E. De Koning, and P. Vedder, “Goal-directed behavior and contextual factors in the classroom: an innovative approach to the study of multiple goals,” *Educational Psychologist*, vol. 41, no. 1, pp. 33–51, 2006.

[8] P. R. Pintrich, E. M. Anderman, and C. Klobucar, “Intraindividual differences in motivation and cognition in students with and without learning disabilities,” *Journal of Learning Disabilities*, vol. 27, no. 6, pp. 360–370, 1994.

[9] G. D. Sideridis, “On the origins of helpless behavior of students with learning disabilities: avoidance motivation?” *International Journal of Educational Research*, vol. 39, no. 4-5, pp. 497–517, 2003.

[10] K. Singh, M. Granville, and S. Dika, “Mathematics and science achievement: effects of motivation, interest, and academic engagement,” *The Journal of Educational Research*, vol. 95, no. 6, pp. 323–332, 2002.

[11] R. Ben-ari and P. Kedem-Friedrich, “Restructuring heterogeneous classes for cognitive development: social interactive

perspective," *Instructional Science*, vol. 28, no. 2, pp. 153–167, 2000.

[12] C. Lytridis, C. Bazinas, G. A. Papakostas, and V. Kaburlasos, "On measuring engagement level during child-robot interaction in education," in *Robotics in Education*, M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh, and D. Obdržálek, Eds., Springer International Publishing, Cham, Switzerland, pp. 3–13, 2020.

[13] J. A. Baxter, J. Woodward, and D. Olson, "Effects of reform-based mathematics instruction on low achievers in five third-grade classrooms," *The Elementary School Journal*, vol. 101, no. 5, pp. 529–547, 2001.

[14] G. D. Sideridis, "Social, motivational, and emotional aspects of learning disabilities," *International Journal of Educational Research*, vol. 43, no. 4-5, pp. 209–214, 2005.

[15] A. Ramachandran, C.-M. Huang, E. Gartland, and B. Scassellati, "Thinking aloud with a tutoring robot to enhance learning," in *Proceedings of the 2018 ACM/IEEE International Conference*, Chicago IL, USA, March 2018.

[16] P. Baxter, E. Ashurst, R. Read, J. Kennedy, and T. Belpaeme, "Robot education peers in a situated primary school study: personalisation promotes child learning," *PLoS One*, vol. 12, no. 5, Article ID e0178126, 2017.

[17] M. Pistoia, S. Pinnelli, and G. Borrelli, "Use of a robotic platform in dyslexia-affected pupils: the ROBIN project experience," *International Journal of Education and Information Technologies*, vol. 9, pp. 45–49, 2015.

[18] C. L. Sidner, C. Lee, C. Kidd, N. Lesh, and C. Rich, "Explorations in engagement for humans and robots," 2005, https://arxiv.org/abs/cs/0507056.

[19] M. I. Ahmad, O. Mubin, and J. Orlando, "Adaptive social robot for sustaining social engagement during long-term children-robot interaction," *International Journal of Human-Computer Interaction*, vol. 33, no. 12, pp. 943–962, 2017.

[20] E. N. Wiebe, A. Lamb, M. Hardy, and D. Sharek, "Measuring engagement in video game-based environments: investigation of the user engagement scale," *Computers in Human Behavior*, vol. 32, pp. 123–132, 2014.

[21] T. C.-t. Fong and S.-m. Ng, "Measuring engagement at work: validation of the Chinese version of the utrecht work engagement scale," *International Journal of Behavioral Medicine*, vol. 19, no. 3, pp. 391–397, 2012.

[22] Z. Wang, C. Bergin, and D. A. Bergin, "Measuring engagement in fourth to twelfth grade classrooms: the Classroom Engagement Inventory," *School Psychology Quarterly*, vol. 29, no. 4, pp. 517–535, 2014.

[23] B. J. Calder, M. S. Isaac, and E. C. Malthouse, "How to capture consumer experiences: a context-specific approach to measuring engagement," *Journal of Advertising Research*, vol. 56, no. 1, pp. 39–52, 2016.

[24] G. Castellano, A. Pereira, I. Leite, A. Paiva, and P. W. McOwan, "Detecting user engagement with a robot companion using task and social interaction-based features," in *Proceedings of the 2009 International Conference on Multimodal Interfaces*, pp. 119–126, Cambridge, MA, USA, November 2009.

[25] G. Castellano, I. Leite, A. Pereira, C. Martinho, A. Paiva, and P. W. McOwan, "Detecting engagement in HRI: an exploration of social and task-based context," in *Proceedings of the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 421–428, IEEE, Washington, DC, USA, September 2012.

[26] J. Hernandez, I. Riobo, A. Rozga, G. D. Abowd, and R. W. Picard, "Using electrodermal activity to recognize ease of engagement in children during social interactions," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 307–317, Seattle, WA, USA, September 2014.

[27] A. Chorianopoulou, E. Tzinis, E. Iosif, A. Papoulidi, C. Papailiou, and A. Potamianos, "Engagement detection for children with autism spectrum disorder," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5055–5059, IEEE, New Orlean, LA, USA, March 2017.

[28] S. M. Anzalone, S. Boucenna, S. Ivaldi, and M. Chetouani, "Evaluating the engagement with social robots," *International Journal of Social Robotics*, vol. 7, no. 4, pp. 465–478, 2015.

[29] O. Rudovic, J. Lee, L. Mascarell-Maricic, B. W. Schuller, and R. W. Picard, "Measuring engagement in robot-assisted autism therapy: a cross-cultural study," *Frontiers in Robotics and AI*, vol. 4, p. 36, 2017.

[30] O. Rudovic, Y. Utsumi, J. Lee et al., "A deep learning approach for engagement intensity estimation from face images of children with autism"" in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 339–346, IEEE, Madrid, Spain, October 2018.

[31] J. Hadfield, G. Chalvatzaki, P. Koutras, M. Khamassi, C. S. Tzafestas, and P. Maragos, "A deep learning approach for multi-view engagement estimation of children in a child-robot joint attention task," 2018, https://arxiv.org/abs/1812.00253.

[32] F. Del Duchetto, P. Baxter, and M. Hanheide, "Are you still with me? continuous engagement assessment from a robot"s point of view," 2020, https://arxiv.org/abs/2001.03515.

[33] U. Frith, "Paradoxes in the definition of dyslexia," *Dyslexia*, vol. 5, no. 4, pp. 192–214, 1999.

[34] K. Hamdan, A. Amorri, and F. Hamdan, ""Robot technology impact on dyslexic students" English learning"" *International Journal of Educational and Pedagogical Sciences*, vol. 11, pp. 1949–1954, 2017.

[35] S. Cramer, "Dyslexia and literacy, International: DI-Duke report," 2016, Available online: https://www.dyslexia-and-literacy.international/wp-content/uploads/2016/04/DI-Duke-Report-final-4-29-14.pdf Accessed on March 10, 2021.

[36] H. Gardner, *Multiple Intelligences: The Theory In Practice; Multiple Intelligences: The Theory In Practice*, vol. 304, Basic Books, New York, NY, USA, 1993, 978-0-465-01821-5.

[37] "Social robots as tools in special education (SRTSE)," 2018, Available online: http://www.koiro3e.eu/en/homen/ Accessed on March 10, 2021.

[38] G. A. Papakostas, A. K. Strolis, F. Panagiotopoulos, and C. N. Aitsidis, "Social robot selection: a case study in education," in *Proceedings of the 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–4, Split, Croatia, September 2018.

[39] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2d pose estimation using part affinity fields," 2018, https://arxiv.org/abs/1812.08008.

[40] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: facial behavior analysis toolkit," in *Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 59–66, IEEE, Xian, China, May 2018.

[41] J. L. Tracy, R. W. Robins, and R. A. Schriber, "Development of a FACS-verified set of basic and self-conscious emotion expressions," *Emotion*, vol. 9, no. 4, pp. 554–559, 2009.

[42] X. Zhang, Y. Sugano, and A. Bulling, "Evaluation of appearance-based methods and implications for gaze-based

applications," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, Scotland, UK, May 2019.

[43] X. Zhang, Y. Sugano, and A. Bulling, "Everyday eye contact detection using unsupervised gaze target discovery," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 193–203, Québec City, Canada, October 2017.

[44] T. Baltrušaitis, M. Mahmoud, and P. Robinson, "Cross-dataset learning and person-specific normalisation for automatic action unit detection," *IEEE* in *Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 6, pp. 1–6, Ljubljana, Slovenia, May 2015.

[45] G. A. Papakostas, K. I. Diamantaras, and F. A. Palmieri, "Emerging trends in machine learning for signal processing," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 6521367, 2 pages, 2017.

[46] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018.

[47] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[48] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Computational Learning Theory*, P. Vitányi, Ed., pp. 23–37, Barcelona, Spain, March 1995.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[50] J. Kennedy, S. Lemaignan, C. Montassier et al., "Child speech recognition in human-robot interaction: evaluations and recommendations," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 82–90, Vienna, AT, USA, March 2017.

[51] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[52] P. I. Kiratsa, G. K. Sidiropoulos, E. V. Badeka, C. I. Papadopoulou, A. P. Nikolaou, and G. A. Papakostas, "Gender identification through facebook data analysis using machine learning techniques," in *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, pp. 117–120, Athens, Greece, November 2018.

[53] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 7703165, 10 pages
https://doi.org/10.1155/2021/7703165

Research Article

Takagi–Sugeno Fuzzy Controller and Sliding Mode Controller for a Nonholonomic Mobile Robot

Hafedh Abid

Laboratory of Sciences and Techniques of Automatic Control and Computer Engineering (Lab-STA) Sfax, National School of Engineering of Sfax, University of Sfax, Sfax, Tunisia

Correspondence should be addressed to Hafedh Abid; abidhafedh@gmail.com

Received 16 April 2021; Revised 18 May 2021; Accepted 8 June 2021; Published 18 June 2021

Academic Editor: Zain Anwar Ali

This paper focuses on the nonholonomic wheeled mobile robot. We have presented a scheme to develop controllers. Two controllers have been developed. The first concerns the kinematic behavior, while the second relates to the dynamic behavior of the mobile robot. For the kinematic controller, we have used a Takagi–Sugeno fuzzy system to overcome the nonlinearities present in model, whereas for the second controller, we have used the sliding mode approach. The sliding surface has the identical structure as the proportional integral controller. The stability of the system has been proved based on the Lyapunov approach. The simulation results show the efficiency of the proposed control laws.

1. Introduction

In the last decades, the path of travel is considered as one of the critical problems in the field of mobile robotics. The trajectory tracking consists of guiding the robot through intermediate points to reach the final destination. This tracking is carried out under a constraint time, which means that the robot must reach the goal within a predefined time. In the literature, the problem is treated as the tracking of a reference robot that moves to the desired trajectory with a certain rhythm. The real robot must follow precisely the reference and reduce the distance error, by varying its linear and angular velocities [1, 2]. There are many works that have focused on tracking the trajectory of the mobile robots, and they consider the mobile robot as a particle; in this case, the inputs are velocities. Their aims are kinematic models. In [3], the kinematic control law approach supposes that the control signal generates the exact motion commanded. On the contrary, some works consider the kinematic aspect and the dynamic aspect for the mobile robot. In this case, the actuator inputs signals are torques instead of velocities [4]. In [5], Lee et al. suggest a technique for designing the tracking control of wheeled mobile robots based on a new sliding surface with an approach angle. In [6], authors proposed a robust backstepping controller for the uncertain kinematic model of the wheeled mobile robot based on a nonlinear disturbance observer in order to cope with model uncertainties and the external disturbances. Topalov [7] proposed an adaptive fuzzy approach for the kinematic controller. This method was able to decrease the effect of unmodeled disturbances. In [8], a dynamic Petri recurrent fuzzy neural network was proposed. In [9], the proposed controller combines nonlinear time varying feedback with an integral sliding mode controller. The latter is obtained by introducing an integral term in the switching manifold.

In [10], a robust adaptive mobile robot controller is presented using backstepping for kinematics and dynamics motions, and the adaptive process was based on the neural network. In [11], a classical parallel distributed compensation (PDC) control law, based on Takagi–Sugeno fuzzy modeling, is proposed. The controller comprises sixteen rules in which the control gains have been calculated using LMI techniques. In [12], the authors present an adaptive controller with consideration of unknown model parameters.

In [13], the authors suggest a controller of a mobile robot in Cartesian coordinates with an approach angle based on the sliding mode. In [14], the authors combine hybrid backstepping kinematic control with the adaptive integral sliding mode kinetic control of the three-wheeled mobile robot.

Most of the works deal with nonholonomic wheeled mobile robot, which is used for kinematic motion of a classical controller arising from the backstepping method [2, 10, 12, 14, 15].

This paper includes two main contributions. First, a new controller based on Takagi–Sugeno fuzzy systems for kinematic motion. This lastly uses three fuzzy rules. The second contribution consists of developing for the dynamic part a controller based on the sliding mode. The sliding surface, which is based on linear and angular velocities of the robot, has the similar structure as the proportional integral controller. The switching control term of the latter controller combines the two sliding surfaces.

The remainder of this paper is organized as follows. Section 2 is devoted to the description of the kinematic and dynamic models of the two-wheeled mobile robot. Section 3 that is reserved to the controllers design includes two subsections, the first is reserved to the development of the new T-S type fuzzy controller of the kinematic behavior, whereas the second is consecrated to the design of the dynamic motion controller using the sliding mode approach. The stability analysis is checked in the both precedent subsections by the Lyapunov approach. Then, Section 4 is sacred to the presentation of the simulation results.

2. Mobile Robot Modeling

In this section, we are interested in the modeling of the robot, which is composed of two driving wheels and a drive shaft in the center, as shown in Figure 1. Indeed, the Section 2.1 is reserved for kinematic modeling, while Section 2.2 concerns dynamic modeling.

We define the current position (x_c, y_c) and the heading angle θ, which constitute the coordinates of the middle point of the mobile robot and the angle between the heading direction and the x-axis to describe the current posture position of the mobile robot. Figure 1 depicts the current posture position of a two wheels mobile robot in Cartesian frame coordinates.

The nonholonomic constraint of a wheeled mobile robot is given by the following equation:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0. \tag{1}$$

2.1. Fuzzy Kinematic Model of Robot. Based on the Newton–Euler equations [16] and the previous hypotheses, the state equations of the mobile robot are represented by the following equations' system [17]:

$$\begin{cases} \dot{x} = v\cos\theta, \\ \dot{y} = v\sin\theta, \\ \dot{\theta} = \omega, \end{cases} \tag{2}$$
$$v = \sqrt{\dot{x}^2 + \dot{y}^2},$$

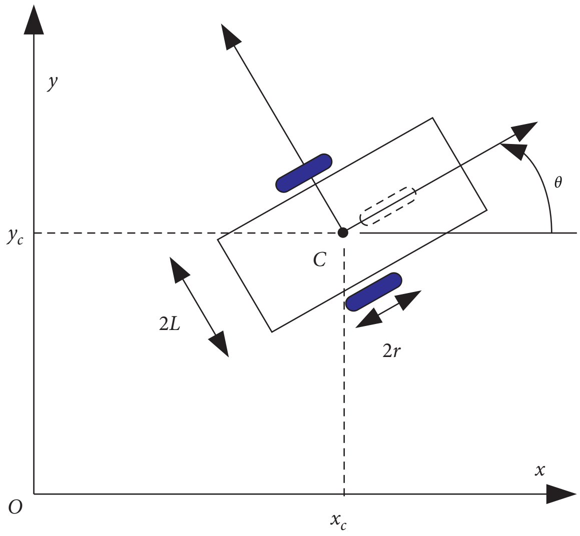


Figure 1: Representation of the navigation environment.

where (x, y), v, and ω represent, respectively, the instantaneous position coordinates of point C of the mobile robot in the global Cartesian frame and the measurements at point C of the linear and angular speeds of the robot. The state variables of mobile robot are $q = [\,x \;\; y \;\; \theta\,]^T$:

$$\begin{aligned} v_d &= \sqrt{\dot{x}_d^2 + \dot{y}_d^2}, \\ w_d &= \frac{\dot{x}_d\ddot{y}_d - \dot{y}_d\ddot{x}_d}{\dot{x}_d^2 + \dot{y}_d^2}, \end{aligned} \tag{3}$$

where v_d and w_d represent, respectively, the desired linear and angular velocity.

The state kinematic model of the mobile robot in Cartesian frame coordinates is given by the following expression:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = J(\theta)V_m, \tag{4}$$

with

$$\begin{aligned} V_m &= [\,v \;\; \omega\,]^T, \\ J(\theta) &= \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}^T. \end{aligned} \tag{5}$$

In order to develop a T-S fuzzy controller, which stabilizes the system and allows the robot to follow the desired path, we need a fuzzy model. In this context, we proceed to determine a fuzzy model of the robot.

The posture vector error is not specified in the global frame coordinate system, but quite as a vector error in the local frame coordinate system of the robot: $q_e(t) = [\,e_1 \;\; e_2 \;\; e_3\,]^T$.

The posture vector error q_e is computed based on the actual posture vector $q(t) = [\,x \;\; y \;\; \theta\,]^T$ and the reference posture vector $q_d(t) = [\,x_d \;\; y_d \;\; \theta_d\,]^T$:

$$\dot{q}_d(t) = J(\theta_d)V_{m\,d}, \tag{6}$$

where $V_{m\,d} = [\,v_d \;\; \omega_d\,]^T$.

So,

$$\tilde{q} = q_d - q = \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}. \tag{7}$$

The relation between the local frame and the global frame, as shown in Figure 2, is given by the following equation:

$$q_e = \mathfrak{R}_e\tilde{q}, \tag{8}$$

where

$$\mathfrak{R}_e = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

Equation (8) allows transforming the magnitudes described in the global coordinate system to the local coordinate system:

$$\dot{q}_e = \dot{\mathfrak{R}}_e\tilde{q} + \mathfrak{R}_e\dot{\tilde{q}}. \tag{10}$$

However, by differentiating equation (10), which contains the linear speed and the angular speed terms, we obtain the derivative of the error vector, which is expressed by the following equation:

$$\dot{q}_e = \begin{cases} \dot{e}_1 = \omega e_2 - v + v_d\cos e_3, \\ \dot{e}_2 = -\omega e_1 + v_d \sin e_3, \\ \dot{e}_3 = \omega_d - \omega. \end{cases} \tag{11}$$

The posture error model can be rewritten as follows:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_d \\ w_d \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{12}$$

We note that equation (12) contains trigonometric nonlinearities which are cos (e_3) and sin (e_3). However, the nonlinearities depend on the error e_3, whose range of variation is from −pi/2 to pi/2.

The advantage of the T-S type fuzzy approach is that it allows describing the nonlinear model by linear submodels. Indeed, each submodel represents a local linear relation between the inputs and the outputs and all the nonlinearities are reported in the premises of the fuzzy rules [18].

Based on the theory of T-S fuzzy systems, the nonlinear model (12) can be transformed into three local models, which are inferred by fuzzy rules. The three local models are described by the following systems of equations:

From the weights assigned to each rule, the state vector of the fuzzy models is inferred as follows (which corresponds to a barycentric aggregation).

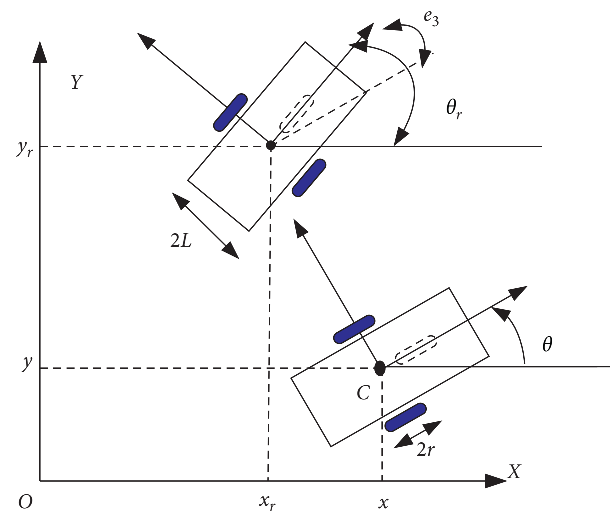


Figure 2: Trajectory tracking.

The member ship function for the error e_3 is given in Figure 3.

The rules of the local models are given by the following expression:

$$\begin{aligned} &\text{if } e_3 \text{ is } \mu_1, \quad \text{then } \dot{q}_e = A_1V_{m\,d} + B_1V_m, \\ &\text{if } e_3 \text{ is } \mu_2, \quad \text{then } \dot{q}_e = A_2V_{m\,d} + B_2V_m, \\ &\text{if } e_3 \text{ is } \mu_3, \quad \text{then } \dot{q}_e = A_3V_{m\,d} + B_3V_m. \end{aligned} \tag{13}$$

The T-S fuzzy model of equation (12) is given by the following equation:

$$\dot{q}_e = \sum_{i=1}^{3} \mu_i\left(A_iV_{m\,d} + B_iV_m\right), \tag{14}$$

where μ_i and A_i and B_i represent, respectively, the weight assigned to each rule and the matrices associated to the local model.

With,

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 0 \\ e_3 & 0 \\ 0 & 1 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ A_3 &= \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, \\ B_1 = B_2 = B_3 &= \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{15}$$

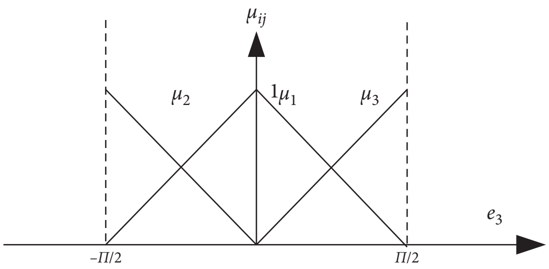


FIGURE 3: Membership function.

2.2. Dynamic Model of Robot. The dynamic equation of the wheeled mobile robot is given by the following equation:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) = B(q)\tau - A^T(q)\lambda, \tag{16}$$

where $C(q,\dot{q})$ is the centripetal and Coriolis matrix, $F(\dot{q})$ is the friction force, τ represents the torque vector, and $A^T(q) = 0$:

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_g \end{bmatrix},$$
$$B(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}, \tag{17}$$
$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where m and J_g represent, respectively, the mass and the moment inertia of the wheeled mobile robot. L and r represent, respectively, the distance separating the two driving wheels and the wheel radius. Without considering disturbances and uncertainties, the latest equation becomes as

$$\overline{M}(q)\dot{V}_m = \overline{B}(q)\tau, \tag{18}$$

where

$$\overline{M}(q) = \begin{bmatrix} m & 0 \\ 0 & J_g \end{bmatrix},$$
$$\overline{B}(q) = \frac{1}{r}\begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix},$$
$$V_m = \begin{bmatrix} v \\ w \end{bmatrix}, \tag{19}$$
$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}.$$

The expressions of linear and angular velocities of the mobile robot, (v, w), depend on the left and right linear velocities of the motors. They are expressed by the following equations:

$$\begin{aligned} v &= \frac{v_r + v_l}{2}, \\ w &= \frac{v_r - v_l}{2L}. \end{aligned} \tag{20}$$

3. Design of Robot Controllers

In this work, we consider the kinematic and dynamic behavior of the robot. The purpose of the control design is to allow the robot to follow the virtual robot. The latter represents the reference robot and provides the desired path defined by the following vector: $q_d(t) = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$.

The architecture of the control scheme of the robot, includes six blocks, as shown in Figure 4. The first block generates the desired states, whereas the second block transforms the error from the local frame into the general frame. The third and fourth blocks are reserved, respectively, for kinetic and dynamic controllers. The fifth and sixth blocks, respectively, describe the behavior of the kinematic and dynamic models of the robot.

3.1. Fuzzy Kinematic Controller. In this section, we are interested in the search for a T-S type fuzzy controller, which guarantees the convergence of the kinematic errors towards zero in the local coordinate system and allows the robot to follow the desired path.

Based on the T-S fuzzy model (14), the rules for the local controllers are given by the following expressions:

$$\begin{aligned} &\text{if } e_3 \text{ is } \mu_1, \quad \text{then } V_m = \begin{bmatrix} v = K_1 e_1 + v_d & \omega = K_3 e_3 + \dfrac{v_d e_2}{e_3} + \omega_d \end{bmatrix}^T, \\ &\text{if } e_3 \text{ is } \mu_2, \quad \text{then } V_m = \begin{bmatrix} v = K_1 e_1 + v_d & \omega = K_3 e_3 + \dfrac{v_d e_2}{e_3} + \omega_d \end{bmatrix}^T, \\ &\text{if } e_3 \text{ is } \mu_3, \quad \text{then } V_m = \begin{bmatrix} v = K_1 e_1 + v_d & \omega = K_3 e_3 + \dfrac{v_d e_2}{e_3} + \omega_d \end{bmatrix}^T. \end{aligned} \tag{21}$$

The global T-S fuzzy controller is given by the following equation:

$$\begin{aligned} V_m = \Bigg[v_c = \sum_{i=1}^{3} \mu_i v_i = K_1 e_1 + \mu_1 v_d \omega_c = \sum_{i=1}^{3} \mu_i \omega_i \\ = \omega_d + K_3 e_3 + \mu_1 v_d e_2 + (\mu_2 - \mu_3)\frac{v_d e_2}{e_3} \Bigg]^T. \end{aligned} \tag{22}$$

If $e_3 = 0$, then $\mu_1 = 1$ and $\mu_2 = \mu_3 = 0$, so

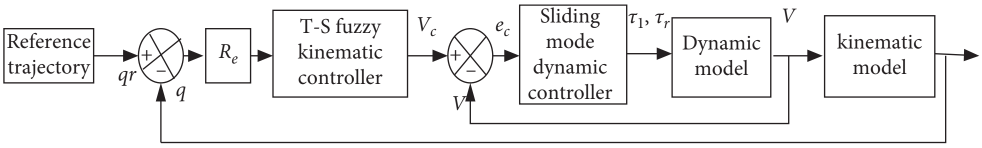


FIGURE 4: Architecture of the robot controller.

$$V_m = \left[\nu_c = \sum_{i=1}^{3} \mu_i \nu_i = K_1 e_1 + \mu_1 \nu_d \omega_c \right. \\ \left. = \sum_{i=1}^{3} \mu_i \omega_i = \omega_d + K_3 e_3 + \mu_1 \nu_d e_2 \right]^T. \tag{23}$$

3.1.1. Stability Analysis. To check the stability of the robot, we use Lyapunov's theory. However, we choose the following Lyapunov candidate function:

$$V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}e_3^2. \tag{24}$$

The derivative of Lyapunov function is as follows.

So,

$$\dot{V} = \dot{e}_1 e_1 + \dot{e}_2 e_2 + \dot{e}_3 e_3, \\ \dot{V} = (\mu_1(\omega e_2 - \nu + \nu_d) + \mu_2(\omega e_2 - \nu) + \mu_3(\omega e_2 - \nu))e_1 \\ + (\mu_1(-\omega e_1 + \nu_d e_3) + \mu_2(-\omega e_1 + \nu_d) + \mu_3(-\omega e_1 - \nu_d))e_2 \\ + (\mu_1(\omega_d - \omega) + \mu_2(\omega_d - \omega) + \mu_3(\omega_d - \omega))e_3. \tag{25}$$

So,

$$\dot{V} = ((-\nu + \mu_1 \nu_d))e_1 + \left(\mu_1 \nu_d e_2 + \omega_d + (\mu_2 - \mu_3)\frac{\nu_d e_2}{e_3} - \omega \right) e_3. \tag{26}$$

If we choose the following linear and angular velocities,

$$\nu_c = K_1 e_1 + \mu_1 \nu_d, \\ \omega_c = \sum_{i=1}^{3} \mu_i \omega_i = \omega_d + K_3 e_3 + \mu_1 \nu_d e_2 + (\mu_2 - \mu_3)\frac{\nu_d e_2}{e_3}. \tag{27}$$

Equation (26) becomes

$$\dot{V} = -K_1 e_1^2 - K_3 e_3^2 \le 0, \quad \text{if } e_3 \ne 0. \tag{28}$$

If $e_3 = 0$, then $\mu_1 = 1$ and $\mu_2 = \mu_3 = 0$. So, $\nu_c = K_1 e_1 + \nu_d$ and $\omega_c = \sum_{i=1}^{3} \mu_i \omega_i = \omega_d + K_3 e_3 + \nu_d e_2$.

Also,

$$\dot{V} = -K_1 e_1^2 - K_3 e_3^2 \le 0. \tag{29}$$

The derivative of the Lyapunov function is negative and the stability of the system is guarantee.

3.2. Dynamic Controller Based on Sliding Mode. In this section, we are interested in the development of a controller, which guarantees the convergence of the posture error q_e towards zero for any arbitrary reference trajectory. However, we have developed a controller based on the sliding mode approach because the latter is considered a robust approach [19, 20]. In this case, we define two sliding surfaces. The first surface depends on linear velocity, while the second uses angular velocity, $S = [\, s_v \;\; s_w \,]^T$:

$$S(t) = \begin{bmatrix} S_v \\ S_w \end{bmatrix} = e_e(t) + K \int_0^t e_e(\delta) \mathrm{d}\delta, \tag{30}$$

where s_v and s_w are given, respectively, by equations (31) and (32), $e_e = (V_c - V_m) = [\, e_v \;\; e_w \,]^T$. With $e_v = v_c - v$ and $e_w = w_c - w$,

$$S_v(t) = e_v(t) + k_v \int_0^t e_v(\delta) \mathrm{d}\delta, \tag{31}$$

$$S_w(t) = e_{wv}(t) + k_w \int_0^t e_w(\delta) \mathrm{d}\delta. \tag{32}$$

However, the derivatives of the sliding surfaces $s_v(t)$ and $s_w(t)$ are given by the following expressions:

$$\dot{S}_v(t) = \dot{e}_v(t) + k_v e_v(t), \\ \dot{S}_w(t) = \dot{e}_w(t) + k_w e_w(t). \tag{33}$$

The dynamic motion of the robot is described by equation (7) which can be transformed as

$$\dot{V}_m = (\overline{M}(q))^{-1} \overline{B}(q) \tau. \tag{34}$$

Equation (34) can be written as

$$\dot{V}_m = \widetilde{B} \tau, \tag{35}$$

where $\widetilde{B} = (\overline{M})^{-1} \overline{B}$

Based on the sliding mode theory, the controller includes two terms which are known as equivalent control law and switching control. The global control law is expressed as

$$u = \tau = u_{\text{eq}} + u_s = \tau_{\text{eq}} + \tau_s. \tag{36}$$

The equivalent control law u_{eq} is computed by recognizing that $\dot{S} = 0$ which is a necessary condition for the state trajectory to stay in the sliding surface [19, 20]. The derivative of the sliding surface is

$$\dot{S}(t) = \dot{e}_e(t) + K e_e(t), \tag{37}$$

with $\dot{e}_e = (\dot{V}_c - \dot{V}_m)$ and $e_e = (V_c - V_m)$.

Thus, substituting (35) for (37), we obtain

$$\dot{S}(t) = \dot{V}_c - \widetilde{B}\tau + K e_e(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{38}$$

with

$$\begin{aligned}\overline{M}^{-1} &= \frac{1}{mJ_g}\begin{bmatrix} J_g & 0 \\ 0 & m\end{bmatrix},\\ \widetilde{B} = \overline{M}^{-1}\overline{B} &= \frac{1}{rmJ_g}\begin{bmatrix} J_g & J_g \\ mL & mL\end{bmatrix},\\ \widetilde{B}^{-1} &= \frac{r}{2L}\begin{bmatrix} mL & J_g \\ mL & -J_g\end{bmatrix},\\ \overline{B}^{-1} &= \frac{1}{2Lr}\begin{bmatrix} L & 1 \\ L & -1\end{bmatrix}.\end{aligned} \tag{39}$$

However, the equivalent control law $u_{\text{eq}} = \tau_{\text{eq}}$ can be computed as

$$u_{\text{eq}} = \tau_{\text{eq}} = \widetilde{B}^{-1}\dot{V}_c + \widetilde{B}^{-1}Ke_e(t), \tag{40}$$

with

$$\dot{V}_c = \begin{bmatrix} \dot{\nu}_c = K_1\dot{e}_1 + \mu_1\dot{\nu}_d \\ \dot{\omega}_c = \dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\dfrac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3^2}\end{bmatrix}. \tag{41}$$

So,

$$\begin{aligned}\begin{bmatrix} u_{\text{eqr}} \\ u_{\text{eql}}\end{bmatrix} &= \begin{bmatrix} \dfrac{rm}{2} & \dfrac{rJ_g}{2L} \\ \dfrac{rm}{2} & -\dfrac{rJ_g}{2L}\end{bmatrix}\left(\begin{bmatrix} K_1\dot{e}_1 + \mu_1\dot{\nu}_d \\ \dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\dfrac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3}\end{bmatrix} + \begin{bmatrix} \lambda e_v \\ \lambda e_w\end{bmatrix}\right),\\ \begin{bmatrix} u_{\text{eqr}} \\ u_{\text{eql}}\end{bmatrix} &= \begin{bmatrix} \dfrac{rm}{2} & \dfrac{rJ_g}{2L} \\ \dfrac{rm}{2} & -\dfrac{rJ_g}{2L}\end{bmatrix}\begin{bmatrix} K_1\dot{e}_1 + \mu_1\dot{\nu}_d + \lambda e_v \\ \dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\left(\dfrac{\dot{\nu}_d e_2}{e_3} + \dfrac{\nu_d e_2}{e_3} - \dfrac{\nu_d e_2\dot{e}_3}{e_3^2}\right) + \lambda e_w\end{bmatrix},\\ \begin{bmatrix} u_{\text{eqr}} \\ u_{\text{eql}}\end{bmatrix} &= \begin{bmatrix} \dfrac{rm}{2}\left(K_1\dot{e}_1 + \mu_1\dot{\nu}_d + \lambda e_v\right) + \dfrac{rJ_g}{2L}\left(\dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\left(\dfrac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3^2}\right) + \lambda e_w\right) \\ \dfrac{rm}{2}\left(\left(K_1\dot{e}_1 + \mu_1\dot{\nu}_d + \lambda e_v\right)\right) - \dfrac{rJ_g}{2L}\left(\dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\left(\dfrac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3^2}\right) + \lambda e_w\right)\end{bmatrix}.\end{aligned} \tag{42}$$

Finally, we obtain

$$\begin{aligned}u_{\text{eqr}} &= \frac{rm}{2}\left(K_1\dot{e}_1 + \mu_1\dot{\nu}_d + \lambda e_v\right) + \frac{rJ_g}{2L}\left(\dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\left(\frac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3^2}\right) + \lambda e_w\right),\\ u_{\text{eql}} &= \frac{rm}{2}\left(\left(K_1\dot{e}_1 + \mu_1\dot{\nu}_d + \lambda e_v\right)\right) - \frac{rJ_g}{2L}\left(\dot{\omega}_d + K_3\dot{e}_3 + \mu_1\left(\dot{\nu}_d e_2 + \nu_d\dot{e}_2\right) + \left(\mu_2 - \mu_3\right)\left(\frac{(\dot{\nu}_d e_2 + \nu_d\dot{e}_2)e_3 - \nu_d e_2\dot{e}_3}{e_3^2}\right) + \lambda e_w\right).\end{aligned} \tag{43}$$

The switching control term is generally chooses as $u_s = -\eta \operatorname{sign}(S)$, with $\eta > 0$. This term can be canceled when the system reaches the sliding surface [20]. In this paper, the switching control law is chosen as follows:

$$\tau_s = \begin{bmatrix} -k_{s1}\left(\operatorname{sgn}\left(s_v + s_w\right)\right) \\ -k_{s1}\left(\operatorname{sgn}\left(s_v - s_w\right)\right) \end{bmatrix}. \tag{44}$$

3.2.1. Reaching Condition and Stability Analysis. To verify the reaching condition, we need to just check the following condition. The Lyapunov candidate function is chosen as

$$V = \frac{1}{2} S^T S. \tag{45}$$

The derivative can be expressed as

$$\begin{aligned} \dot{V} &= S^T \dot{S}, \\ \dot{V} &= S^T \left[\dot{V}_c - \widetilde{B}\tau + K e_e(t) \right], \\ \dot{V} &= S^T \left[\dot{V}_c - \widetilde{B}\left(\tau_{\text{eq}} + \tau_s\right) + K e_e(t) \right], \\ \dot{V} &= S^T \dot{S} = S^T \left[\dot{V}_c - \widetilde{B}\left(\tau_{\text{eq}} + \tau_s\right) + K e_e(t) \right]. \end{aligned} \tag{46}$$

Based on equation (40), $\dot{V}_c - \widetilde{B}\tau_{\text{eq}} + K e_e(t) = 0$

So,

$$\begin{aligned} \dot{V} &= S^T \dot{S} = S^T \begin{bmatrix} \dfrac{rm}{2} & \dfrac{rJ_g}{2L} \\ \dfrac{rm}{2} & -\dfrac{rJ_g}{2L} \end{bmatrix} \begin{bmatrix} -k_s\left(\operatorname{sgn}\left(s_v + s_w\right)\right) \\ -k_s\left(\operatorname{sgn}\left(s_v - s_w\right)\right) \end{bmatrix}, \\ \dot{V} &= S^T \begin{bmatrix} -k_s \dfrac{rm}{2} \operatorname{sgn}\left(s_v + s_w\right) - k_s \dfrac{rJ_g}{2L} \operatorname{sgn}\left(s_v - s_w\right) \\ -k_s \dfrac{rm}{2} \operatorname{sgn}\left(s_v + s_w\right) + k_s \dfrac{rJ_g}{2L} \operatorname{sgn}\left(s_v - s_w\right) \end{bmatrix}, \\ \dot{V} &= \begin{bmatrix} s_v & s_w \end{bmatrix} \begin{bmatrix} -k_s \dfrac{rm}{2} \operatorname{sgn}\left(s_v + s_w\right) - k_s \dfrac{rJ_g}{2L} \operatorname{sgn}\left(s_v - s_w\right) \\ -k_s \dfrac{rm}{2} \operatorname{sgn}\left(s_v + s_w\right) + k_s \dfrac{rJ_g}{2L} \operatorname{sgn}\left(s_v - s_w\right) \end{bmatrix}, \\ \dot{V} &= \frac{k_s rm \left(s_v + s_w\right)}{2} \operatorname{sgn}\left(s_v + s_w\right) - \frac{k_s r J_g \left(s_v - s_w\right)}{2L} \operatorname{sgn}\left(s_v - s_w\right), \\ \dot{V} &= -\frac{k_s rm}{2} \left|\left(s_v + s_w\right)\right| - \frac{k_s r J_g}{2L} \left|\left(s_v - s_w\right)\right|. \end{aligned} \tag{47}$$

We note $k_{1s} = (k_s rm)/2$ and $k_{2s} = (k_s r J_g)/2L$. So,

$$\dot{V} = -k_s \left| s_v + s_w \right| - k_{2s} \left| s_v - s_w \right| < 0. \tag{48}$$

4. Simulation Results

In this part, we present the simulation results of the nonholonomic wheeled mobile robot. The parameters of the robot (see Figure 1) are $m = 17\,\text{kg}$, $J = 0.537\,\text{kg}\cdot\text{m}^2$, $L = 0.24\,\text{m}$, and $r = 0.1\,\text{m}$ and initial position $\begin{bmatrix} x_o & y_o & \theta \end{bmatrix} = \begin{bmatrix} 5 & 5 & \pi/12 \end{bmatrix}$. The reference speed has been chosen as the variable as shown by the following equation:

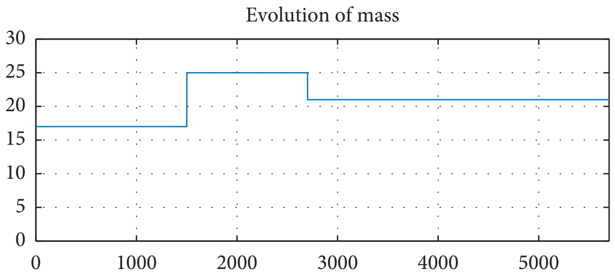


Figure 5: Evolution of mass of the robot.

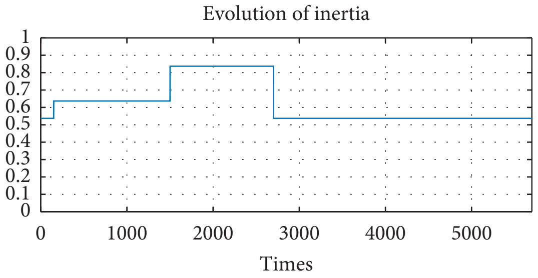


Figure 6: Evolution of inertia of the robot.

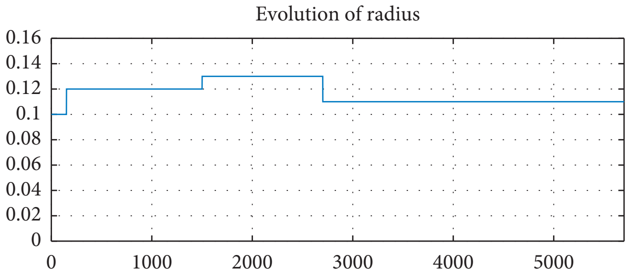


Figure 7: Evolution of radius of the robot.

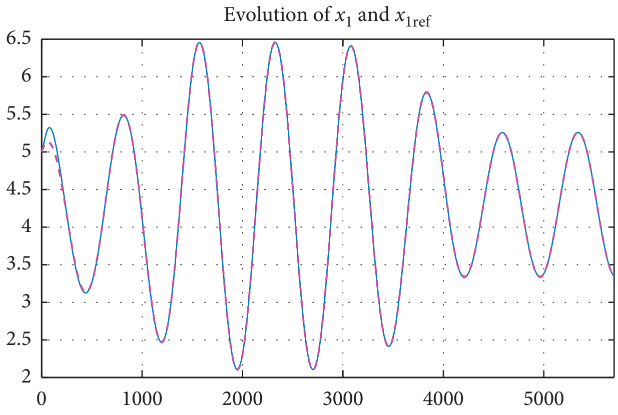


Figure 8: Evolution of the robot trajectory along the x-axis (blue denotes the trajectory and magenta denotes the reference trajectory).

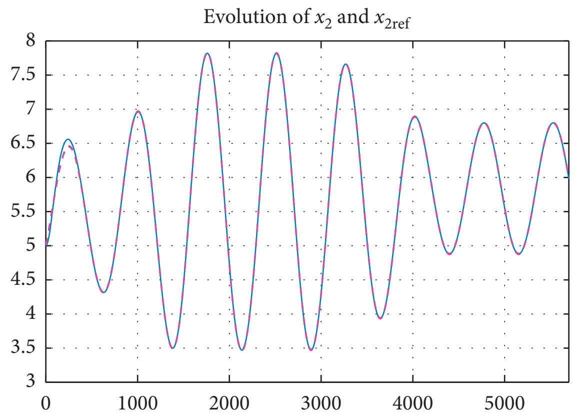


Figure 9: Evolution of the robot trajectory along the y-axis (blue denotes the trajectory and magenta denotes the reference trajectory).

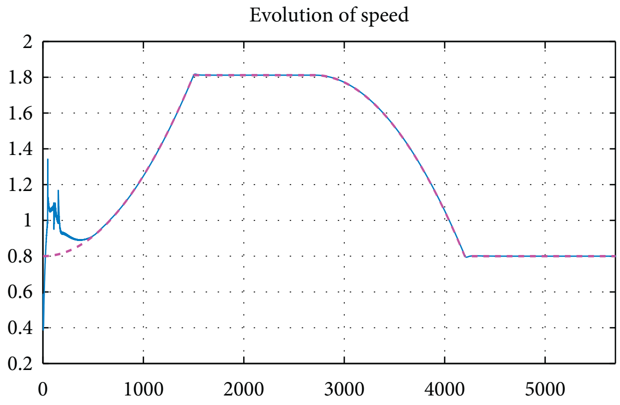


Figure 10: Evolution of linear speed v.

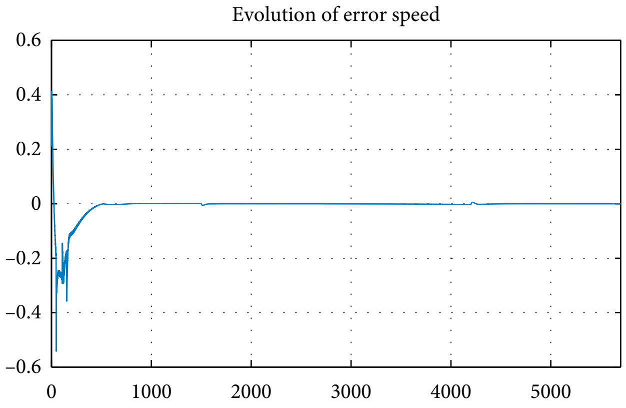


Figure 11: Evolution of the error of the linear speed $ev = v_r - v$.

$$\begin{cases} v_r = v_{ro} + 910^{-7}t, & \text{for } t \leq 15\text{ s}, \\ v_r = v_r, & \text{for } 15 \leq t \leq 27\text{ s}, \\ v_r = v_r - 910^{-7}t, & \text{for } t{>}27\text{ s and } v_r \geq 0.8. \end{cases} \tag{49}$$

The initial speeds have been chosen as $v_{ro} = 0.8$, $v_d = 0.78$, and $v_g = 0.38$.

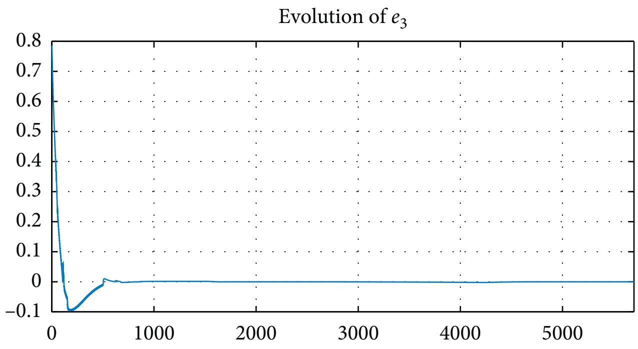


Figure 12: Evolution the angular error e_3.

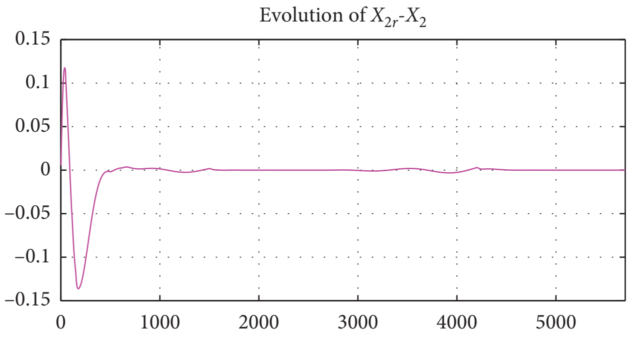


Figure 13: Evolution of the error $x_{r2} - x_2$.

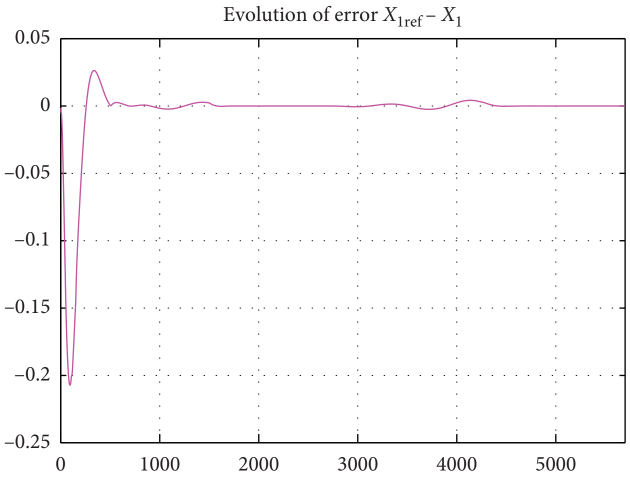


Figure 14: Evolution the error $x_{r1} - x_1$.

Figures 5–7 show, respectively, the evolution in time of mass, inertia, and radius of the robot.

Figures 8 and 9 show, respectively, the evolution in time of the robot along the x- and y-axis (the trajectories of robot and its reference).

Figures 10 and 11 show, respectively, the behavior evolution of linear speed v and the error speed $ev = v_r - v$.

Figures 8–10 show that all the signals follow quickly the reference. The curves of the reference signals and the responses overlap in a very short time after start-up.

Figure 12 shows the evolution the angular error e_3.

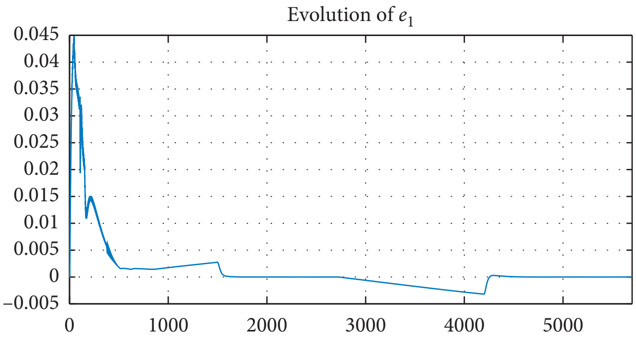


Figure 15: Evolution of e_1.

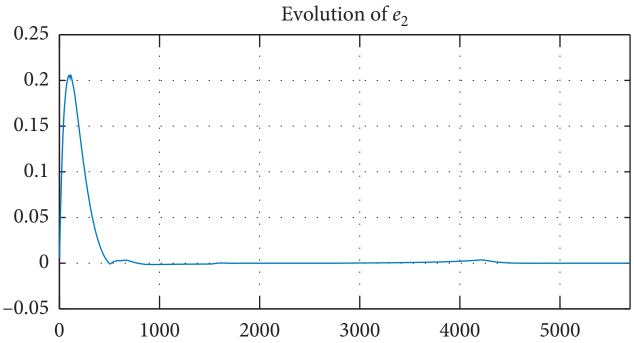


Figure 16: Evolution of e_2.

Figures 13 and 14 show, respectively, the behavior evolution of errors $x_{2\text{ref}} - x_2$ and $x_{1\text{ref}} - x_1$.

Figures 15 and 16 show, respectively, the behavior evolution of errors e_1 and e_2.

Figures 12–16 show that all the errors rapidly tend towards zero. This clearly shows that the system converges towards the desired trajectories in a very short time. This proves the effectiveness of the proposed command.

5. Conclusion

In this paper, we have presented a scheme to control a nonholonomic robot. We have proposed two controllers. The first concerns the kinematic behavior, while the second relates to the dynamic behavior of the mobile robot. To overcome the nonlinearities present in the model, we have used a Takagi–Sugeno fuzzy system for the kinematic controller. For the second controller, we have used the sliding mode technique, which is known as a robust technique. The proposed sliding surface has the same structure as the proportional integral controller. The Lyapunov approach has been used to prove stability of the system. The results presented in Section 4 show the efficiency of the proposed control laws.

Data Availability

There is no specific data used to support the conclusions of the study. All the necessary information exists in the paper.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609–615, 2000.

[2] G. Zidani, D. Said, and L. Chrifi-Alaoui, A. Benmakhlouf and S. Chaouch, Back- stepping controller for a wheeled mobile robot," in *Proceedings of the 4th International Conference on Systems and Control (ICSC'15)*, Sousse,Tunisia, April 2015.

[3] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control*, vol. 46, no. 11, pp. 1777–1782, 2001.

[4] J. Yang and J. kim, "Sliding mode motion control of nonholonomic mobile robots," *IEEE Control Systems*, vol. 19, no. 2, pp. 15–23, 1999.

[5] J.-K. Lee, J.-B. Park, and Y.-H. Choi, "Tracking control of nonholonomic wheeled mobile robot based on new sliding surface with approach angle," in *Proceedings of the 3rd IFAC Symposium on Telematics Applications The International Federation of Automatic Control*, Seoul, Republic of Korea, November 2013.

[6] M. A. Moqbel Obaid, A. R. Husain, and A. A. Mohammed Al-kubati, "Robust backstepping tracking control of mobile robot based on nonlinear disturbance observer," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 2, pp. 901–908, 2016.

[7] A. V. Topalov, J.-H. Kim, and T. P. Proychev, "Fuzzy-net control of non-holonomic mobile robot using evolutionary feedback-error-learning," *Robotics and Autonomous Systems*, vol. 23, no. 3, pp. 187–200, 1998.

[8] R.-J. Chia-Ming Liu and C.-M. Liu, "Design of dynamic Petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2667–2683, 2009.

[9] A. Bessas, B. Atallah, and F. Boudjema, "Integral sliding mode control for trajectory tracking of wheeled mobile robot in presence of uncertainties," *Journal of Control Science and Engineering*, vol. 2016, Article ID 7915375, 10 pages, 2016.

[10] R. Fierro and F. L. Lewis, "Control of a nonholomic mobile robot: backstepping kinematics into dynamics," *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149–163, 1997.

[11] E.-H. Guechi, J. Lauber, M. Dambrine, G. Klancar, and S. Blažic, "PDC control design for non-holonomic wheeled mobile robots with delayed outputs," *Journal of Intelligent and Robotic Systems*, vol. 60, no. 3, pp. 395–414, 2010.

[12] Y. Hung-Hsiu and H.-K. Huang, "Experimental verification for adaptive tracking control of a nonholonomic mobile robot," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Taipei, Taiwan, November 2007.

[13] J. K. Lee, Y. H. Choi, Y. H. Choi, and J. B. Park, Sliding mode tracking control of mobile robots with approach angle in cartesian coordinates," *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 718–724, 2015.

[14] A. Nikranjbar, A. Masoud Haidari, and B. Ali Asghar Atai, "Adaptive sliding mode tracking control of mobile robot in dynamic environment using artificial potential fields," *Journal of Computer and Robotics*, vol. 11, no. 1, pp. 1–14, 2018.

[15] K. L. Besseghieur, R. Trębiński, W. Kaczmarek, and J. Panasiuk, “Trajectory tracking control for a nonholonomic mobile robot under ROS,” *Journal of Physics Conference Series*, vol. 1016, Article ID 012008, 2018.

[16] J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley, Harlow, England, 1em plus 0.5em minus 0.4em, 1989.

[17] M. K. Bugeja, S. G. Fabri, and L. Camilleri, “Dual adaptive dynamic control of mobile robots using neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 129–141, 2009.

[18] Y. Morère, “Mise en oeuvre de lois de commande pour les modèles flous de type TakagiSugeno,” Université de Valenciennes et du Hainaut-Cambrésis, Valenciennes, France, Thèse de doctorat, 2001.

[19] H. Abid, C. Mohamed, and T. Ahmed, “An indirect adaptive fuzzy sliding mode controller for a class of SISO non-linear systems,” *International Journal of Modelling, Identification and Control*, vol. 4, no. 2, 2008.

[20] Utkin, “Variable structure systems with sliding mode,” *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 212–222, 1997.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2599500, 14 pages
https://doi.org/10.1155/2021/2599500

Research Article

Design Space Exploration for High-Speed Implementation of the MISTY1 Block Cipher

Raza Hasan,[1] Yasir Khizar ,[2] Salman Mahmood,[3] and Muhammad Kashif Sheikh [3]

[1]*Department of Computing, Middle East College, Knowledge Oasis Muscat, Seeb, Oman*
[2]*College of Information Science, Nanjing University of Aeronautics & Astronautics, Nanjing, China*
[3]*Department of Information Technology, Malaysian University of Science & Technology, Petaling Jaya, Malaysia*

Correspondence should be addressed to Yasir Khizar; khizar_yasir@hotmail.com

Received 13 April 2021; Revised 22 May 2021; Accepted 1 June 2021; Published 16 June 2021

Academic Editor: Zain Anwar Ali

This paper proposes 2 × unrolled high-speed architectures of the MISTY1 block cipher for wireless applications including sensor networks and image encryption. Design space exploration is carried out for 8-round MISTY1 utilizing dual-edge trigger (DET) and single-edge trigger (SET) pipelines to analyze the tradeoff w.r.t. speed/area. The design is primarily based on the optimized implementation of lookup tables (LUTs) for MISTY1 and its core transformation functions. The LUTs are designed by logically formulating S9/S7 s-boxes and FI and {FO + 32-bit XOR} functions with the fine placement of pipelines. Highly efficient and high-speed MISTY1 architectures are thus obtained and implemented on the field-programmable gate array (FPGA), Virtex-7, XC7VX690T. The high-speed/very high-speed MISTY1 architectures acquire throughput values of 25.2/43 Gbps covering an area of 1331/1509 CLB slices, respectively. The proposed MISTY1 architecture outperforms all previous MISTY1 implementations indicating high speed with low area achieving high efficiency value. The proposed architecture had higher efficiency values than the existing AES and Camellia architectures. This signifies the optimizations made for proposed high-speed MISTY1 architectures.

1. Introduction

With the advances in high-speed wireless applications, the quest to provide secure transfer of data has been of major concern [1, 2]. The efforts are underway to provide a real-time encryption solution for high data transmissions with minimum overhead in terms of power [3–5]. This study primarily focuses on high-speed implementations of a 64 bit MISTY1 block cipher for a wide range of applications, i.e., wireless networks, Ethernet devices, image encryption, and radio network controllers (RNCs) [6].

A 64 bit block cipher MISTY1 is an ISO standardized algorithm designed by Mitsubishi Corporation Electric Limited. It is used to handle a 64 bit block of data or less, e.g., 8 byte personal identification numbers (PINs), and is based on a provable 2^{-56} probability against linear/differential cryptanalysis [7–10]. The differential/integral attacks on MISTY1 require large data as well as computational complexities making it practically infeasible for breaking the MISTY1 block cipher. The hardware architecture of MISTY1 and its major subfunctions FO and FI constitute a repetitive loop structure [11]. Therefore, the MISTY1 algorithm is suitable for the implementations of resource-constrained and high-speed applications.

To meet the requirement of the Internet of Things, cryptographic algorithms are frequently optimized for area reduction and high throughput implementation or to achieve a good tradeoff between throughput and speed [12–25]. For low-area design, reutilization/logic optimization methodologies have been widely adopted thereby implementing s-boxes using combinational logic [12–20]. A single-round MISTY1 architecture designed for compact implementation is proposed in [20] consisting of only odd-round functions, i.e., 2 × FL functions, 1 × FO function, and 1 × 32 bit XOR. Later, more compact MISTY1 architectures were proposed comprising only one S9/S7 s-box in the FI function [12]. The compact MISTY1 architectures constitute an area of 3041 and 2331 NAND gates, respectively [12].

Finally, 2 × area-efficient MISTY1 design schemes are proposed in [17] based on the combined substitution unit and threshold throughput requirements. The architectures consist of a very low area of 1853/1546 NAND gates and are the most compact implementations to date. In addition, we analyzed the throughput values of the aforementioned studies and found that the compact MISTY1 architectures attained low throughput values, i.e., ≤500 Mbps, and are therefore unsuitable for high-speed applications [12–14, 17, 20].

Contrary to low-area cryptographic hardware architectures, high-speed encryption algorithms utilize LUTs/RAMs or optimized combinational logic for s-boxes using pipelined schemes [20–25]. In the recent era, the focus of the studies has also shifted on the efficient implementations measured in the form of throughput-to-area ratio. Owing to high-speed and efficient implementation requirements, the architecture presented in [20] utilizes FPGA RAM blocks for the implementation of S7/S9 s-boxes. However, the straightforward implementation of LUTs for S9/S7 s-boxes (given in MISTY1 specifications) and longer path delay where 4 × XOR operations are executed in a single clock cycle followed by RAM resulted in a large circuit area and reduced throughput values. The architecture presented in [21] utilizes the double-edge trigger methodology for MISTY1 high-speed pipeline implementation but has a longer path delay. Moreover, no architectural modifications/structural optimizations are made for high-speed MISTY1 implementation. On the contrary, although the MISTY1 architecture proposed in [22] achieves high speed, it costs a large area implementing a large number of pipelines. In this study, an effort has been made for high-speed and efficient MISTY1 implementation. In the last couple of years, multiple studies have been published regarding different block ciphers. In [26], researchers proposed a block cipher based on the chaotic generator and implemented it on Xilinx FPGA to prove its effectiveness. Similarly, in [27], Muthalagu and Jain took an existing block cipher algorithm and enhanced its performance to reduce the encryption time.

The unique contributions of the proposed MISTY1 $n = 8$-round pipelined architectures are as follows:

Optimized implementation of MISTY1 S9/S7 s-boxes and transformation functions, i.e., FL, FI, FO, and 32-bit XOR, by logic formulation of 4, 5, and 6 bit input LUTs for area reduction

Designing of MISTY1 and its transformation functions to attribute for the distribution of parallel processing in order to obtain a highly efficient pipelined architecture

High-speed exploration of 8-round MISTY1 architectures by employing SET and DET techniques

This paper is organized into five sections with the introduction, i.e., Section 1, followed by optimizations/designing of LUTs for the implementation of MISTY1 transformation functions described in Section 2. Section 3 proposes 2 × high-speed MISTY1 architectures based on SET and DET pipeline schemes. FPGA implementation results/analysis are described in Section 4. Lastly, a brief conclusion is given in Section 5.

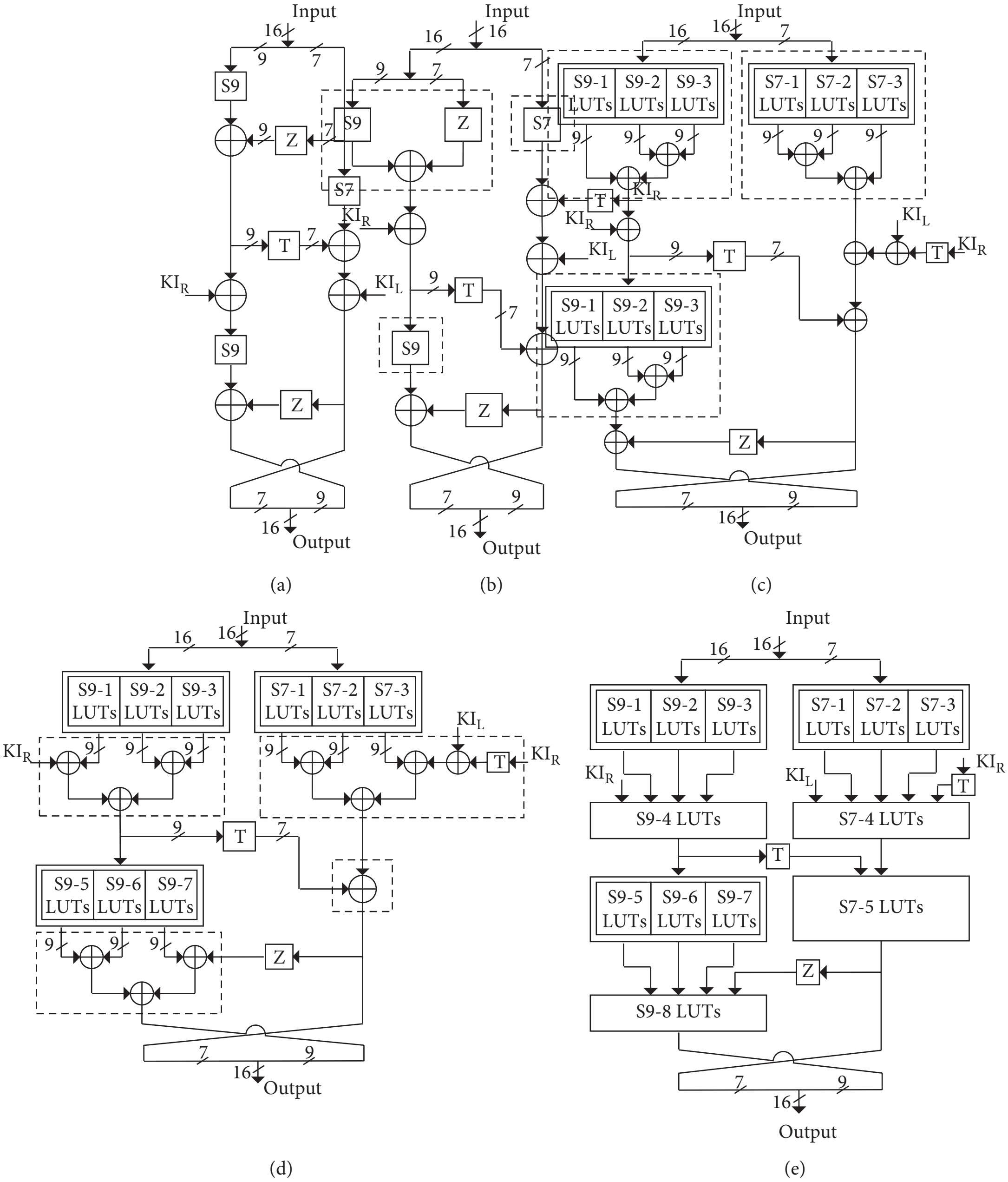
2. Optimized Implementation of MISTY1 Transformation Functions

2.1. FI Function. The optimizations made in the design/implementation of the proposed FI function and its constituent S9 and S7 substitution functions are elaborated in Figures 1(a)–1(e). Figures 1(a) and 1(b) depict the FI function and the equivalent FI with modified S9/S7 paths, respectively. The modifications in Figure 1(b) indicate simultaneous execution of leftmost 9 bits and rightmost 7 bits where the subscripts ‘L’ and ‘R’ represent the leftmost and rightmost bits, respectively. T stands for the TRUNCATE function, and the plus sign showing the summer function is actually the XOR gate. The XOR gate with $\mathrm{KI_R}$ is adding on the LSB side to reduce the path delay. The LSB bits are dependent on MSB bits, and the addition of $\mathrm{KI_R}$ eliminates the dependency of MSB on LSB bits. We have optimized the LUTs of LSB bits by combining S7 and XOR gate. The hardware cost is reduced by the optimization of LUTs for both MSB and LSB sides. In the next step shown as Figure 1(c), the dotted lines of Figure 1(b) are replaced by LUTs {(S9-1 ~ S9-3), (S9-5 ~ S9-7), and (S7-1 ~ S7-3)} concatenated by XOR gates. The upper-left LUTs (S9-1 ~ S9-3) are described in Table 1 as per the modified logic expressions (i.e., S9 is used in conjunction with the zero-extended XOR operation), whereas lower-left LUTs (S9-5 ~ S9-7) can be obtained by eliminating ($x_{10}, x_{11}, \ldots, x_{16}$) bits from the given expressions.

The LUTs for (S7-1 ~ S7-3) are employed as 4 bit and 5 bit input LUTs as described in [21]. In the steps shown in Figures 1(d) and 1(e), the XOR gates of Figure 1(c) are reordered to configure S9-4, S9-8, S7-4, and S7-5 LUTs. The proposed FI function has the primary advantage of reduced LUTs and can be executed in a maximum of 4 clock cycles. Table 2 summarizes the area reduction of 66.7% and 41.3% with the proposed FI function compared to [20, 22], respectively.

2.2. FO Function and 32-Bit XOR. MISTY1 FO transformation function is appended with the 32 bit XOR operation in odd and even rounds (except for the last round) as depicted in Figure 2(a). Therefore, the proposed LUT-based architecture of the FO function comprises {FO + 32 bit XOR}. Figure 2(b) depicts a modified FO function indicating parallel operations for left/right 16 bits. The dotted lines are also mentioned in Figure 2(b), dividing the FO function into 4 sections with each section having side-by-side logic operations. The proposed FO function is deliberated in Figure 2(c) comprising 4 LUT blocks for left and right 16 bits, respectively.

The LUTs of the first and third section include the XOR operations, whereas the second and fourth sections comprise FI functions and XOR operations. However, the left-hand side of the second section symbolized by $\mathrm{FI_1}$ is composed of (FI + XOR), whereas the right-hand side of the second section includes only the FI function. Similarly, the left-hand side of the fourth section shown as $\mathrm{FI_3}$ comprises (FI + (2 × XORs)) as compared to the right-hand side XOR operation. Thus, the FI function described in Section 2.1 is modified as



(a) (b) (c)

(d) (e)

Figure 1: LUT-based FI function formation. (a) FI function. (b) Modifications for the LUT-based FI function. (c) LUT-XOR structure. (d) Reordering of XORs for the formation of LUTs. (e) LUT-based FI function.

per the design requirements of FI_1 and FI_3 as shown in Figures 3 and 4, respectively.

It is evident from Figures 3(a)–3(c) and 4(a)–4(c) that changes required to incorporate XORs into the FI function will mainly require the alterations in the last part of the aforementioned FI function. Therefore, new LUTs are added in the lower right part shown as S7-6 and S7-7 for FI_1 and FI_3, respectively. In addition, S9-8 of Figure 1(e) is replaced by newly formed LUTs S9-9 and S9-10 in the lower left section of FI_1 and FI_3 functions, respectively.

A uniformly distributed LUT-based FO function and inclusion of 32 bit XOR reduce the (initial) latency as well as the pipeline requirements of proposed MISTY1 architectures. The reduction in pipelines and latency thought is not evident from the figures, yet the proposed implementation significantly reduces the area. Table 3 summarizes the area of (FO + 32 bit XOR) showing 53.3% and 44.4% reduction compared to [20, 22]. The proposed FO function is based on the clock cycle operation required to execute FI_1/FI_2/FI_3 functions and will be explained in detail in Section 3.

2.3. Proposed FL Function and Area Estimation of MISTY1 Architectures.

A reference FL function is shown in Figure 5(a) followed by Figure 5(b) showing FL-1 and FL-2 representing 4/3 bit input LUTs for left and right 16 bits,

Table 1: Logic formulation for S9 s-boxes as S9-1, S9-2, and S9-3.

yi	S9-I	Logic expressions
y9	S9-1	x9x5⊕x9x4⊕x8x4
y8		x9x7⊕x6⊕x8x6⊕x7x6
y7		x9x8⊕x8x6⊕x5⊕x9x5⊕x7x5⊕x6x5
y6		x9⊕x8x7⊕x7x5⊕x7x1⊕x5x1
y5		x9x6⊕x7x6⊕x9x4⊕x3⊕x6x4⊕x4x3⊕x7x3
y4		x9x6⊕x8x5⊕x8x3⊕x6x5⊕x5x3
y3		x9x8⊕x8x5⊕x9x1⊕x6⊕x5x1⊕x1
y2		x9x8⊕x8⊕x9x5⊕x9x2⊕x5x2
y1		x9⊕x9x8⊕x9x4⊕x9x2⊕x9x1
y9	S9-2	x8x3⊕x7x3⊕x7x2⊕1
y8		x9x3⊕x7x3⊕x9x1⊕1
y7		x9x3⊕x5x4⊕x4x3⊕x1
y6		x8x4⊕x8x2⊕x6x4⊕x5x4⊕x4
y5		x5x3⊕x3x2⊕x2x1
y4		x6x2⊕x4x2⊕x3x2⊕x2
y3		x7x4⊕x7x2⊕x5x4⊕x4x2
y2		x8x7⊕x7x6⊕x8x3⊕x8x1⊕x6x3
y1		x8x7⊕x7x4⊕x4x3
y9	S9-3	x6x2⊕x6x1⊕x5x1
y8		x6x5⊕x5x4⊕x2⊕x6x1⊕x4x1
y7		x8x2⊕x6x2⊕x16
y6		x4x3⊕x3x2⊕x15
y5		x8⊕x7x1⊕x14
y4		x9x1⊕x7⊕x2x1⊕x13
y3		x3x1⊕x2x1⊕x12⊕1
y2		x4⊕x3x2⊕x11⊕1
y1		x6x3⊕x6x1⊕x5⊕x3x1⊕x10⊕1

Table 2: Area reduction of the proposed FI function implemented on Xilinx Virtex-7.

Method	Ftns	LUTs				Area (slices)	% reduction	
		2-1	3-1	4-1	5-1		With respect to [10]	With respect to [12]
Prop.	S9-1	—	—	3	6	27	66.7	41.3
	S9-2	—	—	6	3			
	S9-3	—	—	6	3			
	S9-4	—	—	9	—			
	S9-5	—	—	4	5			
	S9-6	—	—	9	—			
	S9-7	—	1	6	2			
	S9-8	—	—	9	—			
	S7-1	—	—	1	6			
	S7-2	—	—	6	1			
	S7-3	—	—	6	1			
	S7-4	—	—	—	7			
	S7-5	7	—	—	—			
	FI	7	1	65	34			
[10]	FI	—				81		
[12]	FI	—		—		46		

respectively. Thus, area for $n = 8$-round MISTY1 architecture can be computed by summation of LUTs required for $10 \times \text{FL}$ functions, $8 \times (\text{FO} + 32\,\text{bit XOR})$ functions, and extended key generation function, i.e., $8 \times \text{FI}_2$ functions. Table 4 summarizes the area for proposed MISTY1 architectures.

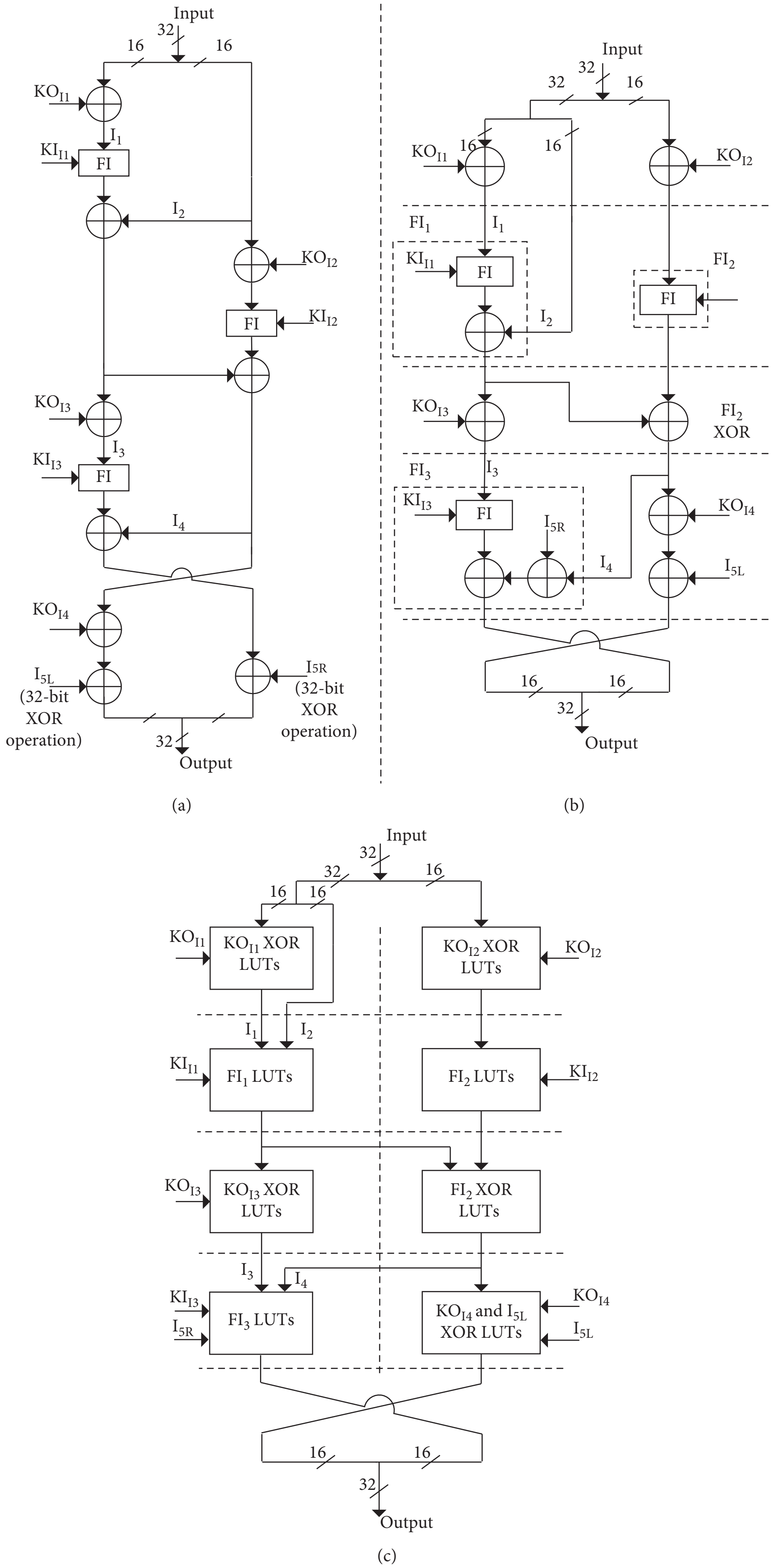


Figure 2: LUT-based architecture for the FO function. (a) FO function + 32 bit XOR. (b) Modified FO function. (c) LUT-based FO function.

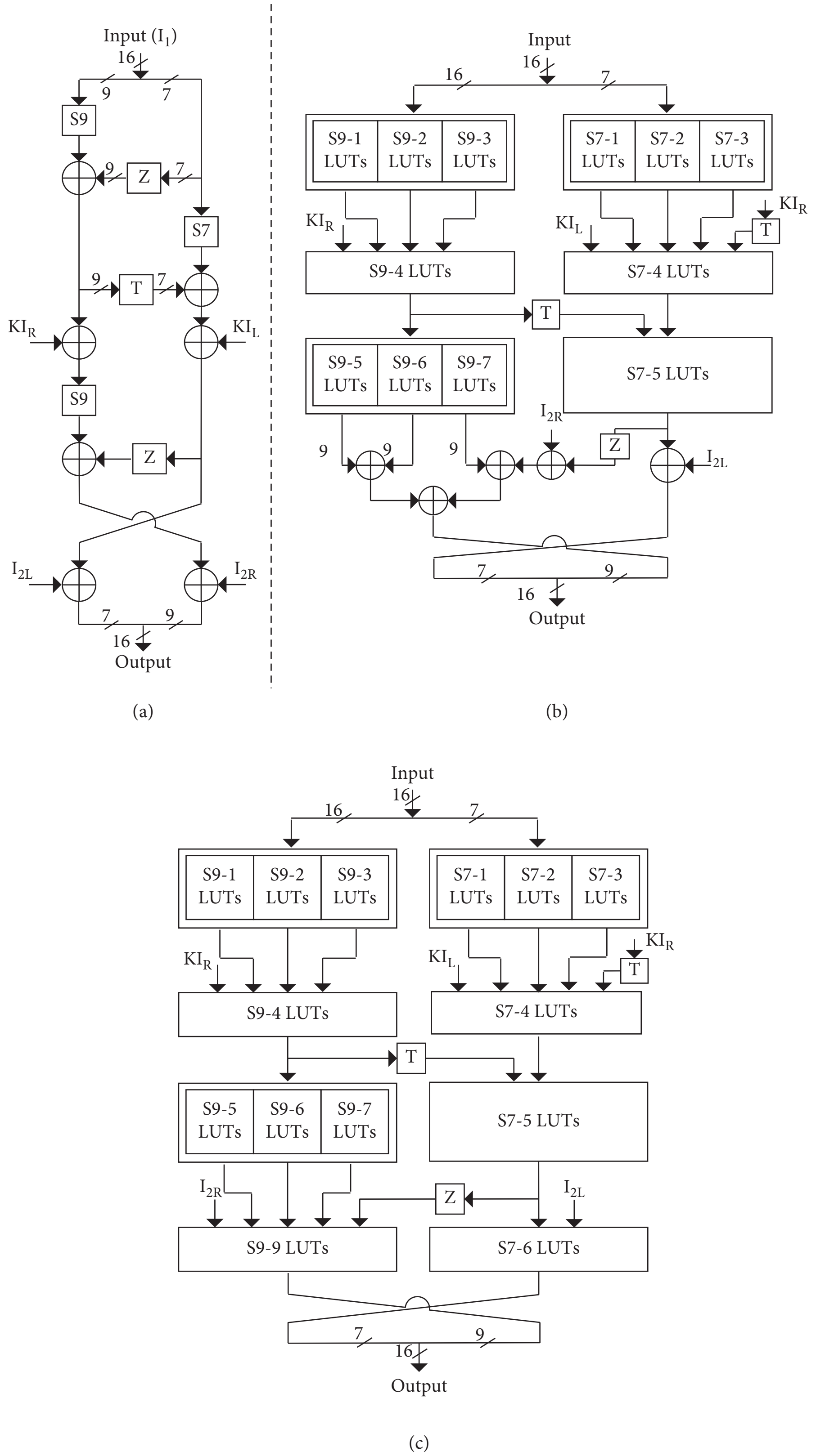


FIGURE 3: FI_1 function. (a) FI_1 function. (b) LUT-XOR structure of the FI_1 function. (c) LUT-based FI_1 function.

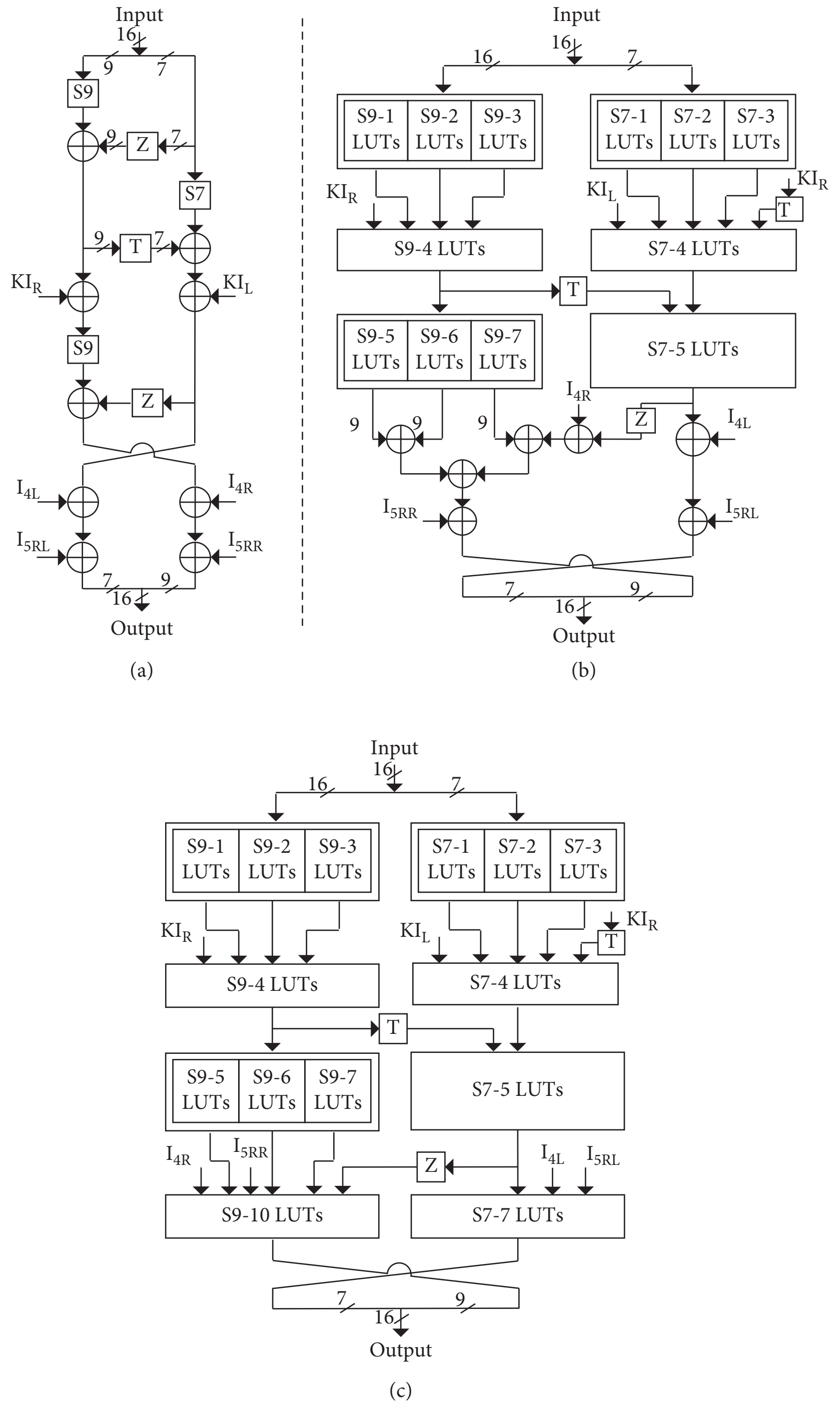


Figure 4: FI_3 function. (a) FI_3 function. (b) LUT-XOR structure of the FI_3 function. (c) LUT-based FI_3 function.

3. Design Space Exploration for High-Speed MISTY1 Architectures

3.1. Architecture 1: DET Pipeline Architecture for High-Speed MISTY1. A high-speed MISTY1 pipelined architecture is shown in Figure 6, whereas the respective FO and FI functions (only the FI_2 function is shown for reference) are depicted in Figures 7(a) and 7(b). High-speed MISTY1 comprises 8-round architecture with 5-stage and 10-stage pipelines in odd and even rounds, respectively. The number of pipelines in odd and even rounds of MISTY1 is based on the number of clock cycles required to execute FO/FI functions. A double-edge-triggered pipeline is employed with each LUT triggering on alternate clock cycles. This reduces the pipeline requirements of the MISYT1 architecture; however, it has a path delay of $2 \times$LUTs as mentioned in [11]. The proposed MISTY1 architecture can process $41 \times$plaintexts and outputs the

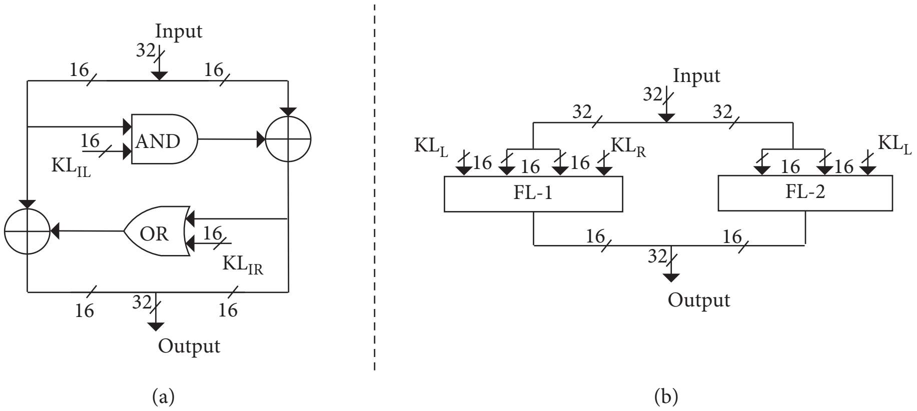


Figure 5: FL function. (a) FL function. (b) LUT-based FL function.

Table 3: Area reduction of the proposed FO function.

Method	FTN	LUTs					Area	% reduction w.r.t.	
		2-1	3-1	4-1	5-1	6-1		[10]	[12]
Prop.	S9-1 ~ S9-7	—	1	43	19	—	114	53.3	41.2
	S7-1 ~ S7-5	7	—	13	15	—			
	S9-8	—	—	9	—	—			
	S9-9	—	—	—	9	—			
	S9-10	—	—	—	2	7			
	S7-6	7	—	—	—	—			
	S7-7	—	7	—	—	—			
	KO_{i1}	16	—	—	—	—			
	KO_{i2}	16	—	—	—	—			
	KO_{i3}	16	—	—	—	—			
	KO_{i4} and I_{5L}	—	16	—	—	—			
	FI_2 XOR	16	—	—	—	—			
	FI_1	14	1	56	43	—			
	FI_2	7	1	65	34	—			
	FI_3	7	8	56	36	7			
	{FO + XOR}	92	26	177	113	7			
[19]	FO			—			244		
[21]	FO			—			194	—	

Table 4: LUT area for MISTY1 architectures.

Function	LUTs					Remarks
	2-1	3-1	4-1	5-1	6-1	
FO	736	208	1416	904	56	8 × FO
FL	—	160	160	—	—	10 × FL
Key gen	112	8	440	352	—	8 × FI
Total	848	376	2016	1256	56	$n = 8$-round MISTY1

required ciphertext of 64 bits per clock cycle. Thus, high-speed MISTY1 is obtained with DET pipelines and highly optimized FO/FI function implementations.

$$T_{\text{Path Delay}}\frac{T}{2} = T_{\text{C2Q (LUT)}} + T_{\text{P (LUT)}} + T_{\text{Setup (LUT)}}. \tag{1}$$

3.2. Architecture 2: MISTY1 SET Pipeline Architecture for Very High-Speed MISTY1. Very high-speed MISTY1 and its respective FO and FI functions (FI_1 and FI_3 functions are presented here for reference) employing single-edge-triggered pipelines are depicted in Figures 8 and 9.

It is evident that the FI_1 function requires 4 clock cycles, whereas the corresponding FO function is executed in 9 clock

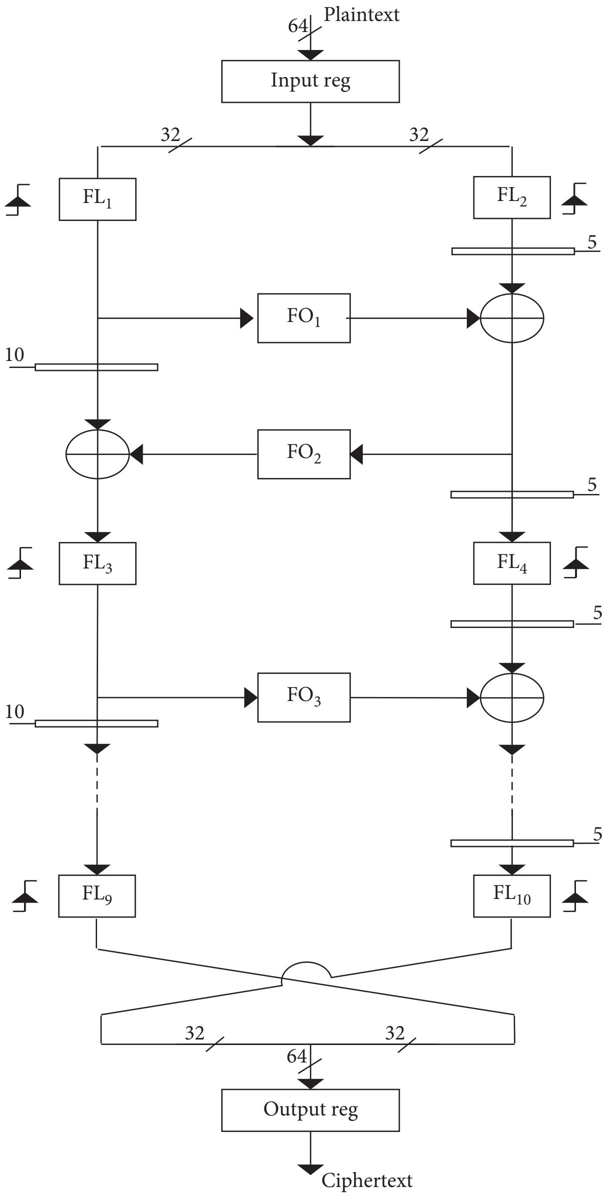


Figure 6: DET pipeline architecture for high-speed MISTY1.

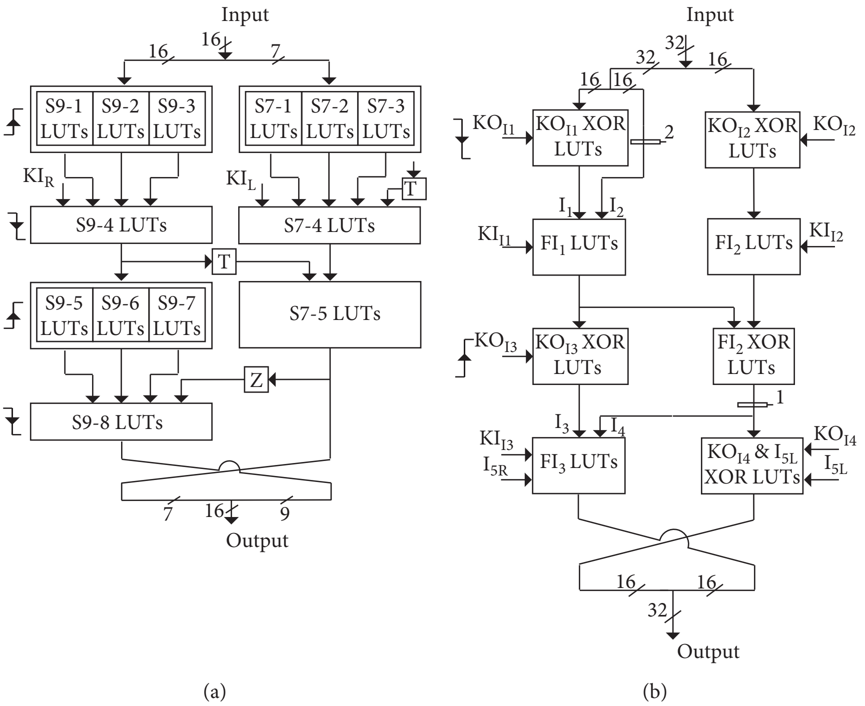


Figure 7: (a) FI_2 function with double-edge triggering accessing 2 × successive LUTs in 1 × clock cycle. (b) FO function with double-edge triggering.

cycles. The pipeline registers are inserted in the FO function as well as MISTY1 architecture to synchronize LSB and MSB bits. The path delay of the SET-based pipelined architecture is 1 × LUT, and therefore, the architecture achieves very high speed. By increasing the pipeline stages, the latency, i.e., the initial ciphertext generation, increases and is found as 77 clock cycles. The proposed architecture is highly suitable for high-speed applications of the order of 40 Gbps.

4. Hardware Implementation Results and Comparison

The proposed MISTY1 high-speed architectures are implemented on FPGA Xilinx Virtex-7, XC7VX690T. The performance comparison/analysis is carried out with existing high-speed Camellia, AES, and MISTY1 architectures. Table 5 depicts the performance parameters, i.e., throughput, area, and efficiency, of the proposed and existing design schemes.

The proposed MISTY1 architectures outperform all previous MISTY1 implementations indicating high speed with low area achieving high efficiency value. The throughput values obtained are 43/25.2 Gbps with a high efficiency of 28.5/18.9 Mbps/slices for very high-speed/high-speed MISTY1 architectures, respectively. For a fair comparison, the referred MISTY1 architectures [20, 22] are implemented using the same FPGA device, i.e., Xilinx Virtex-7. The architectures thus represent highly efficient and high-speed MISTY1 implementations to date. Besides, the proposed architectures have higher efficiency values compared to the existing AES and Camellia architectures (as per our study). This signifies the optimizations made for proposed high-speed MISTY1 architectures.

5. Conclusion

In this paper, we proposed MISTY1 8-round pipelined architectures characterizing high-speed and efficient implementations. The structural optimizations and logic modifications in MISTY1 transformation functions readily reduced the LUTs and pipeline requirements. The proposed high-speed MISTY1 architectures using the SET and DET pipeline explore the speed/area tradeoffs for FPGA implementations. The design/optimization schemes can be extended for the high-speed implementation of the KASUMI algorithm. The high-speed designs have applications in wireless sensor networks, image encryption, and network controllers.

5.1. Future Work. This paper deals only with a high-speed MISTY1 block cipher. In the future, we shall make an energy-efficient MISTY1 block cipher using capacitance scaling, clock gating, clock enable, thermal scaling, voltage scaling, and other energy-efficient techniques. In the future, we shall check the thermal stability of MISTY1. The implementation of the MISTY1 block cipher is on 28 nm technology-based Virtex-7 FPGA in this paper. There is an open scope to reimplement this MISTY1 block cipher design on both 20 nm technology-based Ultrascale Virtex FPGA and 16 nm technology-based Ultrascale Plus Virtex FPGA.

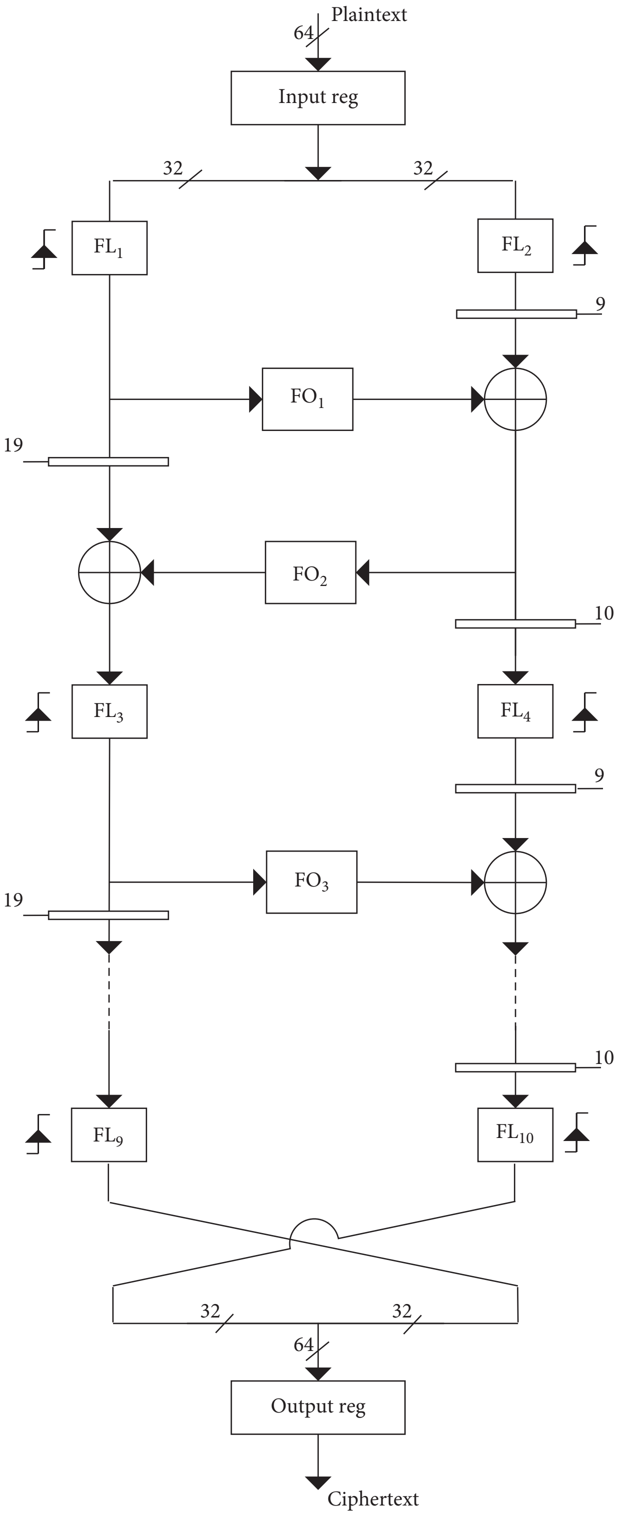


Figure 8: SET pipeline architecture for very high-speed MISTY1.

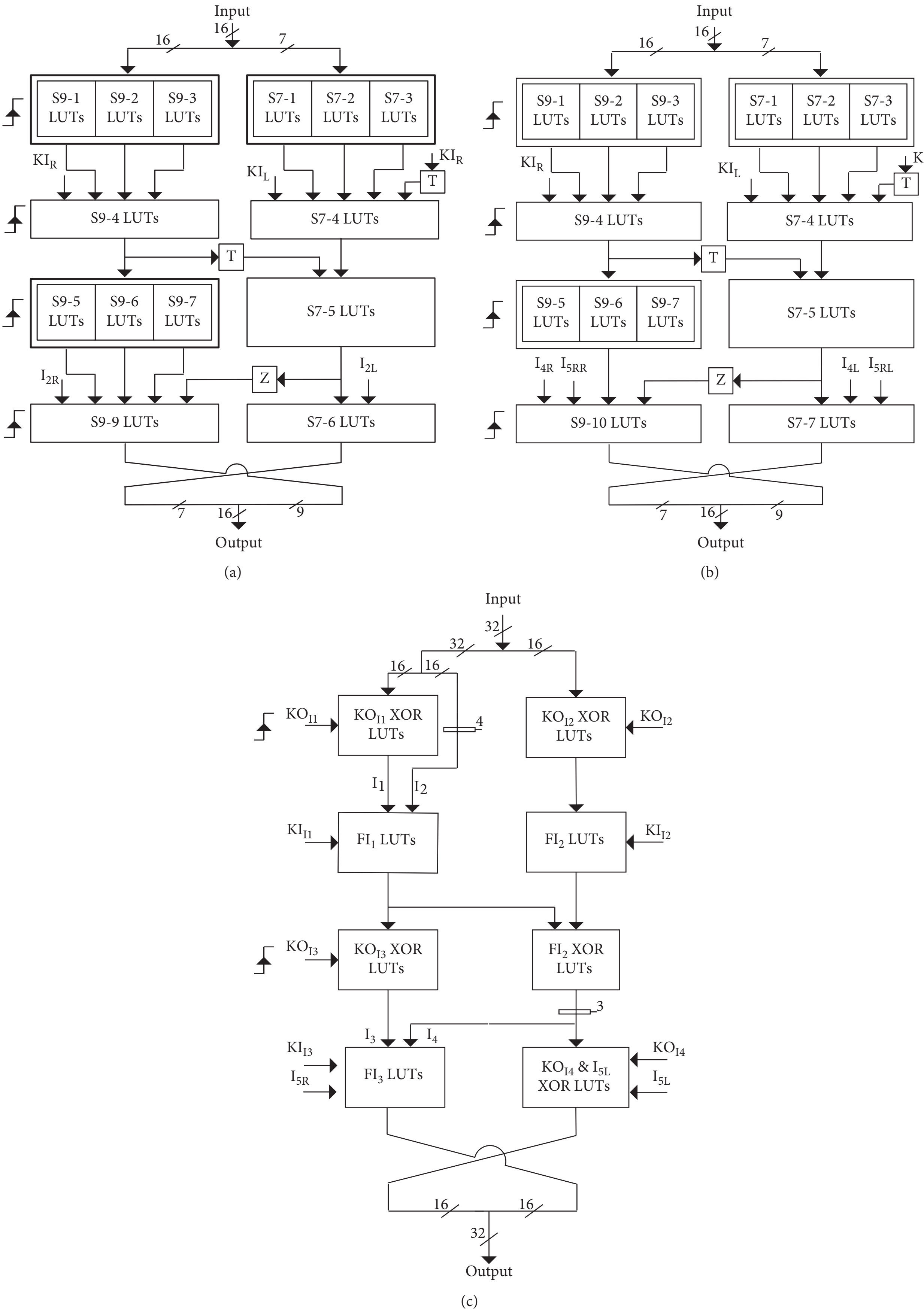


Figure 9: (a) FI_1 function, (b) FI_3 function, and (c) FO function with single-edge-triggered pipeline implementation for very high-speed MISTY1.

Table 5: FPGA implementation and comparison.

Ref.	Algorithm	Area (slices)	Speed (Gbps)	Freq (MHz)	Eff. (Mbps/slices)
[24]	AES	35,328	260	508	7.36
[25]	AES	4339	75.92	593	17.50
[23]	Camellia	2805	28.4	221.6	10.12
[20]*	MISTY1	1865	0.56	79	0.3
[20]*	MISTY1	4732	7.2	96	1.52
[20]**	MISTY1	2920	21.9	342	7.5
[20]*	MISTY1	4039	12.6	168	3.12
[20]**	MISTY1	2920	21.9	342	7.5
[21]*	MISTY1	1265	16.3	254.5	12.9
[22]*	MISTY1	6322	10.18	159	1.61
[22]**	MISTY1	2506	38.9	607.5	15.5
[22]*	MISTY1	6322	19.4	303	3.07
[22]**	MISTY1	2506	38.9	607.5	15.5
Ours	MISTY1	1509	43	673.8	28.5
	MISTY1	1331	25.2	393.7	18.9

*Results published in papers cited. **Results obtained by authors with the implementation on the same FPGA.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Kumar, B. Pandey, T. Das, and B. S. Chowdhry, "Mobile DDR IO standard based high performance energy efficient portable ALU design on FPGA," *Wireless Personal Communications*, vol. 76, no. 3, pp. 569–578, 2014.

[2] B. Pandey, "Energy efficient design and implementation of ALU on 40nm FPGA," in *Proceedings of the International Conference on Energy Efficient Technologies for Sustainability*, IEEE, Nagercoil, India, April 2013.

[3] B. Pandey, "Clock gating based energy efficient ALU design and implementation on FPGA," in *Proceedings of theInternational Conference on Energy Efficient Technologies for Sustainability*, IEEE, Nagercoil, India, April 2013.

[4] B. Pandey, "FSM based green memory design and its implementation on ultrascale plus FPGA," *Journal of Critical Reviews*, vol. 7, pp. 454–458, 2020.

[5] R. Sharma, B. Pandey, V. Jha, S. Saurabh, and S. Dabas, *Input-output standard-based energy efficient UART design on 90 nm FPGA System and Architecture*, Springer, Singapore, pp. 139–150, 2018.

[6] I. Kaur, L. Rohilla, A. Nagpal, B. Pandey, and S. Sharma, *Different configuration of low-power memory design using capacitance scaling on 28-nm field-programmable gate array System and Architecture*, Springer, Singapore, pp. 151–161, 2018.

[7] V. Thind, S. Pandey, D. M. Akbar Hussain, B. Das, M. F. L. Abdullah, and B. Pandey, *Timing constraints-based high-performance DES design and implementation on 28-nm FPGA System and Architecture*, Springer, Singapore, pp. 123–137, 2018.

[8] S. H. A. Musavi, B. S. Chowdhry, T. Kumar, B. Pandey, and W. Kumar, "IoTs enable active contour modeling based energy efficient and thermal aware object tracking on FPGA," *Wireless Personal Communications*, vol. 85, no. 2, pp. 529–543, 2015.

[9] E. Aerabi, M. Bohlouli, M. H. A. Livany, M. Fazeli, A. Papadimitriou, and D. Hely, "Design space exploration for ultra-low-energy and secure IoT MCUs," *ACM Transactions on Embedded Computing Systems*, vol. 19, no. 3, pp. 1–34, 2020.

[10] J. Yang and T. Johansson, "An overview of cryptographic primitives for possible use in 5G and beyond," *Science China Information Sciences*, vol. 63, pp. 1–22, 2020.

[11] M. Matsui, "New block encryption algorithm MISTY," *Fast Software Encryption*, vol. 1267, pp. 54–68, 1997.

[12] A. Yasir, N. Wu, and X. Zhang, "Compact hardware implementations of MISTY1 block cipher," *Journal of Circuits, Systems and Computers*, vol. 27, no. 3, Article ID 1850037, 2017.

[13] D. Yamamoto, J. Yajima, and K. Itoh, "Compact architecture for ASIC implementation of the MISTY1 block cipher," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93-A, no. 1, pp. 3–12, 2010.

[14] Yasir, N. Wu, X. Q. Zhang, and M. R. Yahya, "Highly optimised reconfigurable hardware architecture of 64 bit block ciphers MISTY1 and KASUMI," *Electronics Letters*, vol. 53, no. 1, pp. 10–12, 2017.

[15] AbdoulRjoub, "Low power/high speed optimization approaches of MISTY algorithm," in *Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, UAE, Silchar, India, July 2016.

[16] S. Mathew, S. Satpathy, V. Suresh et al., "340 mV-1.1 V, 289 Gbps/W, 2090-gate NanoAES hardware accelerator with area-optimized encrypt/decrypt GF(2 4) 2 polynomials in 22 nm tri-gate CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 4, pp. 1048–1058, 2015.

[17] A. Yasir, N. Wu, X. Chen, and M. Rehan Yahya, "Area-efficient hardware architectures of MISTY1 block cipher," *Radioengineering*, vol. 27, no. 2, pp. 541–548, 2018.

[18] N. W. Yasir, A. A. Zain, M. Mujtaba Shaikh, M. RehanYahya, and M. Aamir, "Compact and high speed architectures of KASUMI block cipher," *Wireless Personal Communication*, vol. 106, no. 4, pp. 1787–1800, 2018.

[19] Yasir, fnm Ning Wu, and A. A. Siddiqui, "Performance Comparison of KASUMI and hardware architecture optimization of f8 and f9 algorithms for 3g UMTS Networks," in

Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 420–424, Islamabad, Pakistan, January 2017.

[20] P. Kitsos, M. D. Galanis, and O. Koufopavlou, "Architectures and fpga implementations of the 64-bit Misty1 block cipher," *Journal of Circuits, Systems and Computers*, vol. 15, no. 6, pp. 817–831, 2006.

[21] Yasir, N. Wu, X. Chen, M. R. Yahya, and X. Zhang, "FPGA based highly efficient MISTY1 architecture," *IEICE Electronics Express*, vol. 14, no. 18, Article ID 20170841, 2017.

[22] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, "Efficient FPGA implementation of block cipher MISTY1," in *Proceedings of the International Parallel and Distributed Processing Symposium*, Nice, France, April 2003.

[23] A. F. Martínez-Herrera, C. Mancillas-López, and C. Mex-Perera, "GCM implementations of Camellia-128 and SMS4 by optimizing the polynomial multiplier," *Microprocessors and Microsystems*, vol. 45, pp. 129–140, 2016.

[24] AbolfazlSoltani, "An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA," *Microprocessors and Microsystems*, vol. 39, p. 7, 2015.

[25] Q. Liu, Z. Xu, and Y. Yuan, "High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion," *IET Computers & Digital Techniques*, vol. 9, no. 3, pp. 175–184, 2015.

[26] M. Madani and C. Tanougast, "FPGA implementation of an enhanced chaotic-KASUMI block cipher," *Microprocessors and Microsystems*, vol. 80, Article ID 103644, 2021.

[27] R. Muthalagu and S. Jain, "Improved KASUMI block cipher for GSM-based mobile networks," *Journal of Cyber Security Technology*, vol. 4, no. 4, pp. 197–210, 2020.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9953283, 9 pages
https://doi.org/10.1155/2021/9953283

Research Article

Correlation Determination between COVID-19 and Weather Parameters Using Time Series Forecasting: A Case Study in Pakistan

Humera Batool [1] **and Lixin Tian** [1,2,3]

[1] *School of Mathematical Sciences, Nanjing Normal University, Nanjing, Jiangsu 210023, China*
[2] *Center for Energy Development and Environmental Protection, Jiangsu University, Zhenjiang, Jiangsu 212013, China*
[3] *Research Centre of Energy-Interdependent Behavior and Strategy, Nanjing Normal University, Nanjing, Jiangsu 210023, China*

Correspondence should be addressed to Humera Batool; drhumerabatool@hotmail.com

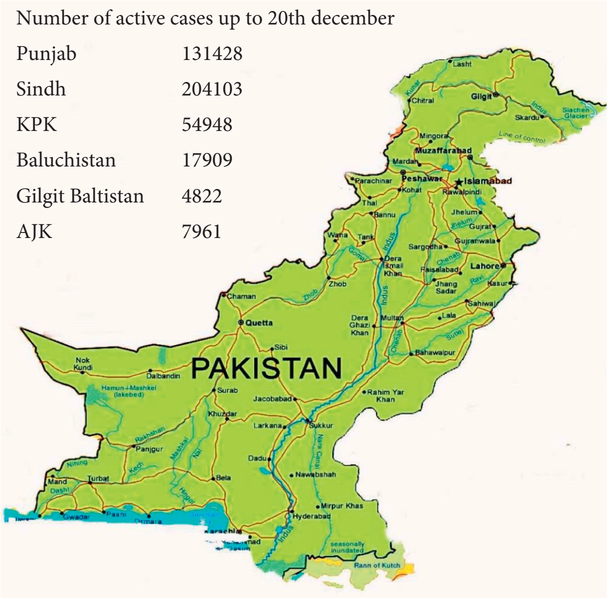
Received 21 March 2021; Revised 24 April 2021; Accepted 29 May 2021; Published 15 June 2021

Academic Editor: Bhawani Shankar Chowdhry

Infectious diseases like COVID-19 spread rapidly and have led to substantial economic loss worldwide, including in Pakistan. The effect of weather on COVID-19 spreading needs more detailed examination, as some studies have claimed to mitigate its spread. COVID-19 was declared a pandemic by WHO and has been reported in about 210 countries worldwide, including Asia, Europe, the USA, and North America. Person-to-person contact and international air travel between the nations were the leading causes behind the spreading of SARS-CoV-2 from its point of origin, besides the natural forces. However, further spread and infection within the community or country can be aided by natural elements, such as the weather. Therefore, the correlation between COVID-19 and temperature can be better elucidated in countries like Pakistan, where SARS-CoV-2 has affected at least 0.37 million people. This study collected Pakistan's COVID-19 infection and mortality data for ten months (March–December 2020). Related weather parameters, temperature, and humidity were also obtained for the same course of time. The collected data were processed and used to compare the performance of various time series prediction models in terms of mean squared error (MSE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE). This paper, using the time series model, estimates the effect of humidity, temperature, and other weather parameters on COVID-19 transmission by obtaining the correlation among the total infected cases and the number of deaths and weather variables in a particular region. Results depict that weather parameters hold more influence in evaluating the sum number of cases and deaths than other factors like community, age, and the total population. Therefore, temperature and humidity are salient parameters for predicting COVID-19 affected instances. Moreover, it is concluded that the higher the temperature, the lesser the mortality due to COVID-19 infection.

1. Introduction

A viral infection named COVID-19 was initially discovered in mid-December 2019 in Wuhan city of China [1], which spread across the whole world, and eventually WHO declared it as a pandemic [2]. Figure 1 shows the map along with the total number of confirmed cases in the province. Up to November 22, 2020, there was 58,475,749 COVID-19 cases, 1,385,775 deaths, and 40,459,596 recoveries across the world, out of which 371,508 total cases, 7,603 deaths, and 328, 931 recoveries were in Pakistan [3]. Although SARS-CoV-2 originated from China, the world's biggest population, it was effectively controlled in China's epicenter and other regions since February 2020 [4]. Daily COVID-19 cases in Pakistan peaked at 6,825 on June 14, 2020; then, it declined to 331 on August 3, 2020; and from the first week of November 2020, again it showed ascending pattern. Albeit there is a cure, the main focus is to curb the spread through national blockades and quarantine measures [5]. Such a high daily number of cases warrants an immediate plan of action to control it effectively and its need to prepare for future outbreaks in Pakistan and other nations.



Number of active cases up to 20th december

Punjab	131428
Sindh	204103
KPK	54948
Baluchistan	17909
Gilgit Baltistan	4822
AJK	7961

FIGURE 1: Map of Pakistan showing active cases in all regions of Pakistan, that is, Gilgit Baltistan (GB), Azad Jammu and Kashmir (AJK), Khyber Pakhtunkhwa (KPK), Baluchistan, Islamabad, Punjab, Sindh, as of December 20, 2020

Recently, scientists elucidated close affiliation between weather parameters and main COVID-19 epidemic areas. Moreover, these areas are located in a relatively temperate region in the northern partition [6]. Although pandemic is a global issue, the outbreak epicenters of the world have a mean temperature of 5°C–11°C with 47%–79% humidity in the first two months of the year 2020. Based on these facts, our primary hypothesis is that virus spread is curtailed in high-temperature and humidity areas rather than areas having average temperature and humidity.

Initially, two cases in Pakistan appeared on February 26, 2020. In parallel, three more cases were recorded in subsequent hours from different cities and there was no affinity/contact among these COVID-19 victims. Unexpectedly, an increase in the number of affected persons on April 14, 2020, was witnessed with the highest number of cases in Punjab, that is, 2826. Sindh was the second with 1452 patients, KPK was the third, having 800 patients, Gilgit Baltistan was in fourth place, having 233 patients, and Baluchistan and Islamabad had 321 and 131 cases, respectively. In contrast, AJK had the least number of cases, i.e., 43 [7]. As there is no proper cure for this pandemic yet discovered and multiple forms of SARS-CoV-2 are also dependent on seasonality [8], these all factors make SARS-CoV-2 spread more alarming and lethal. Short-term forecasting is inevitable to maintain the balance between social, economic, and health aspects in subsequent months [9]. To illustrate the nature of SARS-CoV-2 and to forecast its transmission, there is a dire need to explore its effect on weather. In this regard, the systematic approach of our study includes the following:

(a) Using existing data to predict the number of actual COVID-19 affected cases and the total number of deaths in upcoming months with or without weather data in Pakistan.

(b) To determine the fragile range of climatic factors and verifying these factors at various periods through statistical analysis.

(c) Aiding Pakistan government institutions and policymakers to adopt new strategies to strengthen existing preventive measures to combat the COVID-19 pandemic.

Demongeot et al. [10] identified that the virulence of SARS-CoV-2 and their lethal strains get downregulated in hot and humid climate conditions. The presumed temperature-dependent virulence of COVID-19 also got an eminent interest in the medical field. Instead of the above, our study aims to determine critical factors relying on temperature and transmission kinetics of COVID-19, which increases with cold and dry weather.

Sajadi et al. [11] explained a simplified model describing a zone at high virulence of the COVID-19 outbreak. Bloom-Feshbach et al. [12] elaborated that COVID-19 prevails high in cold and temperate climates than warm and tropical climates, which acknowledges respiratory influenza viruses. For natural distancing calculation and estimation, Prem et al. [13] utilized an age-structured susceptible-exposed-infected-removed (SEIR) model. This study illustrated that if arrival to work initiated in April, physical distancing measures would be most efficient. Eikenberry et al. [14] stated that the SEIR model aimed to evaluate the potential colony impact of the adoption of masks by the public on the mobility and control of the COVID-19. The study recommended using masks nationwide and implementing their use strictly.

Research work related to applying machine learning tools to elucidate the impact of weather parameters on transmission and circulation of COVID-19 seemed lacking and needs more attention. In addition, ascending temperature may or may not lower SARS-CoV-2 spread, and likewise role of other weather factors is also still under debate.

Therefore, past studies are concise to various models, and findings are also not authentic. Hence, it is time to understand the relationship between weather variables and the epidemic spreading of COVID-19 in Pakistan.

2. Materials and Methods

2.1. Data Collection. The daily cumulative total number of confirmed cases and the total number of deaths were obtained from the official website of the National Institute of Health (NIH) in Islamabad, Pakistan. The National Institute of Health is an independent health research department under the Ministry of National Health Services of Pakistan. It is located in Islamabad and is engaged in various research activities and vaccine making. Daily COVID-19 diagnosed cases, recoveries, deaths, and COVID-19 diagnostic tests conducted across Pakistan were updated on the official website of NIH [15].

2.2. Examination. The data were collected from March 10 to December 20, 2020, both for COVID-19 and weather, and was further divided into training and testing datasets. The training dataset comprises the data from March 10 to November 15, 2020, and the testing dataset has data from November 15 to December 20, 2020. Test data was further analyzed for a cumulative number of cases and deaths with and without weather data. Figure 2 shows the division of complete data into training and testing datasets.

2.3. Methods. We have applied simple machine learning models, deep learning techniques, and statistical models to predict the total number of cases and total deaths with and without weather data for COVID-19. Time series prediction models such as ARIMA, linear regression, SVM, MLP, RNN, LSTM, and GRU were used. Statistical performance of time series prediction models was measured in terms of mean squared error (MSE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE). For all these experiments, we used Python version 3.8, Scikit-learn version 0.21.0, and deep learning library Keras v.2.2.5 using tensor flow at the backend.

2.4. Autoregressive Integrated Moving Average (ARIMA). There are three types of ARIMA, namely, autoregression, data-dependent integration, and parameter estimation. All these three types are implemented according to the issue that needs to be focused on [16, 17]. The time series form of the process is

$$\begin{aligned} x^t &= \Theta_0 + \Theta_1 x^{t-1} + \Theta_2 x^{t-2} + \cdots + \Theta_p x^{t-p} \\ &\quad + \varepsilon^t - \Theta_1 \varepsilon^{t-1} - \cdots \Theta_q \varepsilon^{t-q}. \end{aligned} \tag{1}$$

In the previous equation, x^t and ε^t depict the original value and random error at time t. Model parameters are $Øa$ ($a = 1, 2..., p$) and $Øb$ ($a = 0, 1, 2,..., q$). An unexpected error is defined by ε^t and considered with zero mean and $\sigma 2$ of standard variance. Equation (1) represents the ARIMA model and is applied to various applications for problem-solving.

Taking value $q = 0$, in equation (1) works as an AR model with order p, and for $p = 0$, it becomes MA model with order q. Thus, (p, q) are both inevitable factors for ARIMA model determination.

2.5. Linear Regression. Linear regression can be defined as

$$Y = \alpha + bX + \varepsilon, \tag{2}$$

where Y = dependent variable, X = independent variable, α = intercept, b = regression parameter as slope, and ε = random error.

The disadvantage of linear regression is that it usually correlates among an average of input and input variables. Unfortunately, a simple average is not a full illustration of a single variable.

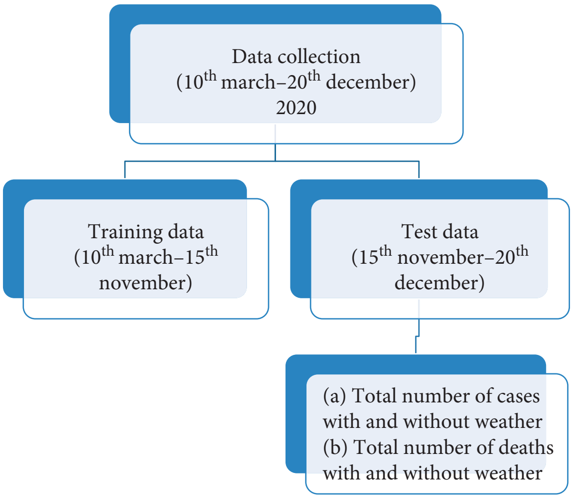

Data collection (10th march–20th december) 2020
Training data (10th march–15th november)
Test data (15th november–20th december)
(a) Total number of cases with and without weather
(b) Total number of deaths with and without weather

Figure 2: Diagram showing division of complete data into training and testing datasets.

2.6. Support Vector Regression (SVM). SVR involves evaluating the support vectors (points) near the hyperplane to upgrade boundary among two-point groups acquired by variation between objective value and threshold. SVR is employing kernel functions to elucidate nonlinear issues, which calculates the affinity between two values. We used the linear kernel function in our study. The main leverage of SVR is that it can capture the nonlinearity of the prediction and then use it to raise the prediction case. In the same scenario, it is beneficial to adopt this view in the case studies used because the sample is inadequate [18]. SVR for the complicated data is

$$y = f(x) = \sum_{i=1}^{M} w_i x_i + b, \tag{3}$$

where wi = input weights, y = actual values, b = bias, and M = total number of data samples. This comparison illustrates the purpose of use of SVR and $||W||$ = magnitude of vector:

$$\min_{w} \frac{1}{2} w^2. \tag{4}$$

Enabling SVM consists of two inadequate variables, that is, ε and ε^*. They are used to guard against anomalies, and 1/2$||W||$2 is used for the precision of function. Both specifications rely on the C parameter. Then, equation (4) will transform into the following equation:

$$\min \frac{1}{2} w^2 + C \sum_{i=1}^{M} \left(\varepsilon_i + \varepsilon_i^*\right). \tag{5}$$

With the suppression,

$$\begin{aligned} & y_i - w^T x_i \le \varepsilon + \varepsilon_{i'}^*, \quad i = 1, 2, 3 \ldots M, \\ & W^T X_i y_i - y_i \le \varepsilon + \varepsilon_{i'}^*, \quad i = 1, 2, 3 \ldots M, \\ & \varepsilon_i, \varepsilon_i^* \ge 0. \end{aligned} \tag{6}$$

Finally, SVR task is accessed as

$$f(x) = \sum_{i=1}^{M} \left(\propto_i^* + \propto_i \right) k \left(X_i - X \right) + b. \tag{7}$$

2.7. Multilayered Perceptron (MLP). Multilayered perceptron (MLP) is the frequently used artificial neural network (ANN) for modeling and forecasting. For evaluating tasks in simple and semicomplicated datasets, this method provides considerable accuracy. It is a wholly joined feed-forward artificial neural network in which neurons are overlapped [19]. MLP has layers: an input, output, and hidden layer. The output layer in the presented research is the total number of cases and deaths. The MLP used in this study has three neurons in the input layer, and each neuron corresponds to an input data point (total cases, total deaths, and days since infection). MLP as the method has ease of implementation. In comparison to complex forms, MLP results in high-quality models while keeping robustness and accuracy in prediction.

Because MLP regressor can only revert an individual value, an adaptive model must be used if the issue hinders multiple output values. Although there may be resemblances among the models, training the whole model means that the dataset will be tested, so a better predictive model can be gained to address each issue. In the present study, three independent MLPs were employed.

2.8. Recurrent Neural Networks (RNNs). In deep learning, it is assumed that classified models are more prospering than flat models in regression tasks [20]. As RNN holds hidden states allocated across time, it favors them to accumulate previous information. Moreover, due to their capability of analyzing the variable length of data, they are abundantly used in forecasting [21]. Our research aims to analyze and evaluate the proposed prediction model, RNN, with different hyperparameters. The essential aspect of RNNs is to consider the impact of previous data on the generated output. Most importantly, RNN is effective for learning time information [22]. LSTM and GRU are two robust RNN models. These illustrations have depicted sublime outcomes in precision and accuracy compared to the classic time series models, and commonly used networks have identified that they can attain multiple outputs in various purposeful domains with time series [23, 24]. Figure 3 shows the conceptual framework of the applied proposed model depicting splitting of data into training and testing data. Further, testing data was evaluated using MSE, RMSE, and MAPE, while training data was validated through time series prediction models ARIMA, linear regression, SVR, MLP, RNN, GRU, and MAPE.

2.9. Gated Recurrent Units (GRU). GRU was presented by [25], which solves vanishing gradient with a standard RNN. GRU is reciprocal to LSTM, but it joins LSTM into one update gate. The GRU further combines cell and concealed form. It consists of a cell containing multiple operations which are duplicated and could be a neural network. When the neural network is applied through BPTT, it can prevent gradient vanishing [26]. The GRU layer, comprising reset gates and update gates, can learn long-term and short-term interdependence from the flow [25]. The mathematical interrelationship among different GRU factors is given by

$$\begin{aligned}
\text{updata gate}\, \mathcal{Z}_t &= o\left(\mathcal{X}_t \mathcal{W}_{xz} + \mathcal{H}_{t-1} \mathcal{W}_{hz} + b_z\right), \\
\text{reset gate}\, \mathcal{R}_t &= o\left(\mathcal{X}_t \mathcal{W}_{xr} + \mathcal{H}_{t-1} \mathcal{W}_{hr} + b_r\right), \\
\text{cell gate}\, \mathcal{H}_t^{\sim} &= tanh\left(\mathcal{X}_t \mathcal{W}_{xh} + \left(R_t o H_{t-1}\right) \mathcal{W}_{hh} + b_h\right), \\
\text{new state}\, \mathcal{H}_t &= \mathcal{Z}_t o \mathcal{H}_{t-1} (1 - \mathcal{Z}_t) o \mathcal{H}_t^{\sim}, \\
\mathcal{W}_{xr}, \mathcal{W}_{xz}, \mathcal{W}_{hr} &= \text{weight parameters}, \\
b_r,\, b_z &= \text{bias parameters}.
\end{aligned} \tag{8}$$

2.10. Long Short-Term Memory (LSTM). The common application of LSTM is in speech recognition and data prediction. Its robust performance in evaluating future predictions by modeling the issue as a series regression problem caught various scientists' attention due to its applications such as activity recognition, prediction, risk resolve, and fall detection [27, 28]. As a deep learning methodology, it leads to other forecasting methods [29]. LSTM is a type of RNN, and its original purpose is to eliminate errors in previous algorithms when backpropagating the information contained in the most recent input event [30]. There are two reasons for using LSTM. First, it returns the error to the machine to calibrate the model in the first training phase. At the same time, errors have been deliberately applied in mechanical gates. Second, the LSTM network is impartial to lag among events in the time series. Therefore, when we are trying to derive an unknown prediction model, the LSTM algorithm is more effective as compared to ANN's (such as hidden Markov and SVR) or other prediction methodologies (such as ARIMA) [29].

For the flow of information, LSTM has input, output, and duplicate gates. These gates are composed of weighted sum logistic functions, and the weights can be gained through backpropagation, all along the process of training. The input gate manages the unit state and the forget gate. The output is accomplished from the output gate or hidden state, and it illustrates the memory used by the direction. This structure permits the network to remember for a long duration, while the traditional single RNN does not have such memory. The ideal feature of LSTM is its extended quality to capture long-term dependencies and powerful ability to process time-series data. For example, given the input time-series X_t and the number of hidden units as h, the gates have the following equation:

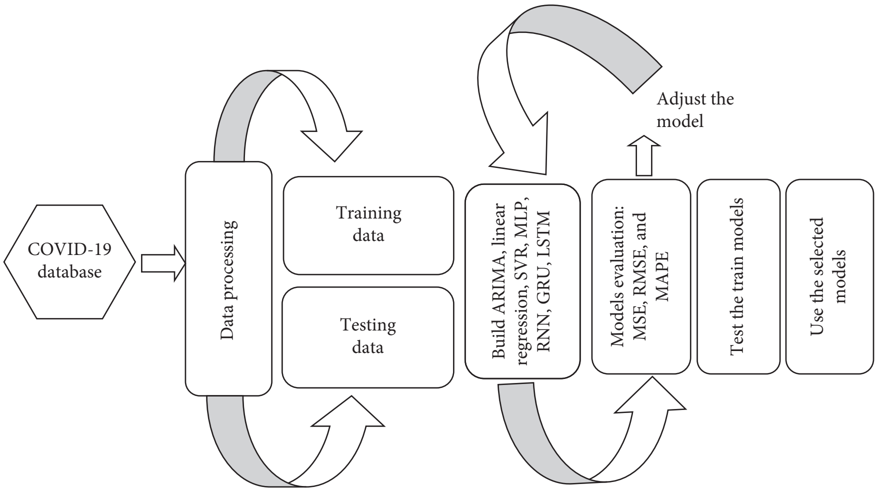


FIGURE 3: Conceptual framework of the proposed forecasting methods.

$$
\begin{aligned}
\text{Input gate: } \mathrm{I}_{\mathrm{t}} &= o\left(\mathscr{X}_t \mathscr{W}_{\mathrm{xi}} + \mathscr{H}_{\mathrm{t-1}} \mathscr{W}_{\mathrm{hi}} + b_{\mathrm{i}}\right),\\
\text{Forward gate: } F_{\mathrm{t=0}} &\left(\mathscr{X}_t \mathscr{W}_{\mathrm{xf}} + \mathscr{H}_{\mathrm{t-1}} \mathscr{W}_{\mathrm{hf}} + b_{\mathrm{f}}\right),\\
\text{Output gate: } O_{\mathrm{t=0}} &\left(\mathscr{X}_t \mathscr{W}_{\mathrm{xo}} + \mathscr{H}_{\mathrm{t-1}} \mathscr{W}_{\mathrm{ho}} + b_{\mathrm{o}}\right),\\
\text{Intermediate cell state: } \mathscr{C}_{\mathrm{t}}^{\sim} &= \tanh\left(\mathscr{X}_t \mathscr{W}_{\mathrm{xc}} + \mathscr{H}_{\mathrm{t-1}} \mathscr{W}_{\mathrm{hc}} + b_{\mathrm{c}}\right),\\
\text{Cell state (next memory input) } \mathscr{C}_{\mathrm{t}} &= \mathscr{F}_{\mathrm{t}}\, o \mathscr{C}_{t-1}\, \mathrm{o} \mathscr{C}_t^{\sim},\\
\text{New state: } \mathscr{H}_{\mathrm{t}} &= \mathscr{O}_{\mathrm{t}} \mathrm{o} \tanh\left(\mathscr{C}_t\right).
\end{aligned}
\tag{9}
$$

$\mathscr{W}_{\mathrm{xi}}, \mathscr{W}_{\mathrm{xf}}, \mathscr{W}_{\mathrm{xo}}$, and $\mathscr{W}_{\mathrm{hc}}, \mathscr{W}_{\mathrm{hf}}, \mathscr{W}_{\mathrm{ho}}$ are weight parameters and $\mathrm{b_i}, \mathrm{b_f}, \mathrm{b_o}$ denote bias parameters. $\mathscr{W}_{\mathrm{xc}}, \mathscr{W}_{\mathrm{hc}}$ = weight parameter, b_c is bias parameter, and o = elementwise multiplication. The estimation of $\mathscr{C}_{\mathrm{t}}$ depends on the output information's from memory cells ($\mathscr{C}_{\mathrm{t-1}}$) and the current time step $\mathscr{C}_{\mathrm{t}}^{\sim}$.

2.11. Performance Metrics. Measure the average of the squares of the errors. It is the average squared difference between the estimated values and the actual value. MSE is a risk function, corresponding to the expected value of the squared error loss:

$$
\mathrm{MSE} = \sqrt{\frac{1}{x}} \sum_{i-1}^{n} \left(Y_{i+} - Y_t^{\sim}\right)^2. \tag{10}
$$

3. Root-Mean-Squared Error

Root-mean-square error is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed:

$$
\mathrm{RMSE} = \sqrt{\frac{1}{x}} \sum\nolimits_{t-1}^{n} \left(Y_t - Y_t^{\wedge}\right)^2. \tag{11}
$$

3.1. Mean Absolute Percentage Error. The mean absolute percentage error (MAPE) is a measure of prediction accuracy of a forecasting method in statistics, for example, in trend estimation, also used as a loss function for regression problems in machine learning. It usually expresses the accuracy as a ratio defined by the formula

$$
\mathrm{MAPE} = \frac{100}{x} \sum\nolimits_{t-1}^{n} \left| Y_t - Y_t^{\wedge} \div Y_t \right|_{\%}. \tag{12}
$$

4. Results

Evaluation of COVID-19 transmission using mathematical models requires training on a large number of datasets. The size of the dataset affects the performance of the proposed algorithms and holds a considerable role in training. The dataset is classified into two parts, the training and the testing datasets. A training dataset is employed during model development, whereas testing datasets are used to validate datasets that are not previously used [31, 32].

The interrelationship between COVID-19 and weather factors in the case of Pakistan is examined in this study. The number of confirmed COVID-19 cases (dependent variable) was log-transformed to make it work as normal distribution as the original data is highly skewed in the selected area. For a specified period up to November 15, 2020, training data evaluates the statistics of cases by considering Pakistan's humidity and temperature data. The hypothesis is that high humidity and temperature (weather variables) shall coincide with a lowered count of SARS-CoV-2 cases. Figures 4(a) and 4(b) illustrate a scatter plot among the number of proved infections compared to thermal readings and humidity in Pakistan.

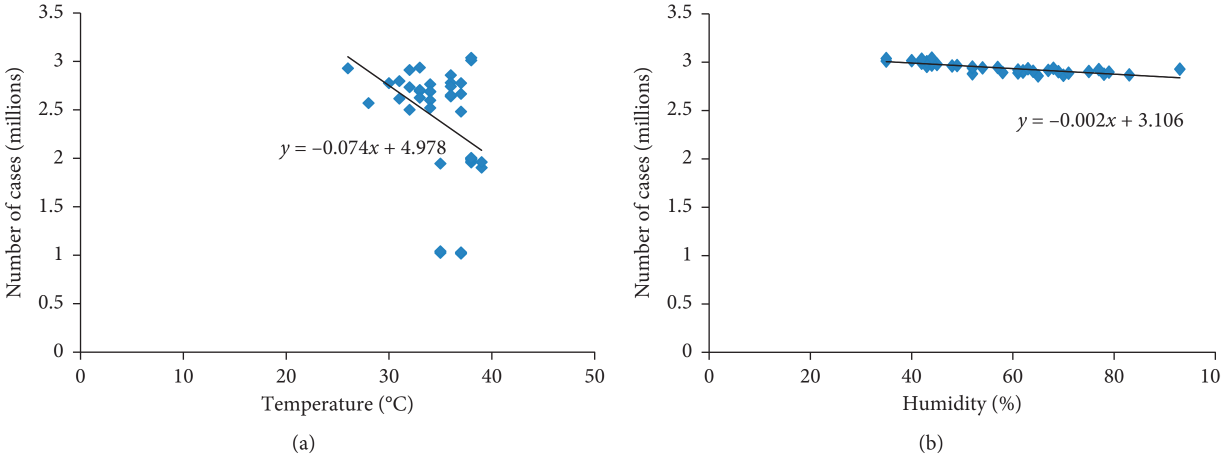


Figure 4: Scatter plot showing the number of cases in Pakistan based on temperature (a) and humidity (b).

Figures 5(a) and 5(b) depict a scatter plot among the number of deaths compared to temperature and humidity in Pakistan. From these findings, it is understood that, as atmospheric moisture and thermal reading decline (increase in temperature), the numbers of infected cases and death rates also decline.

When temperature and humidity showed ascending pattern, the infection rate descends. However, this fact is unavoidable that when sunlight hours increase, interaction among people increases. As a result, the infection rate may elevate. The people residing in urban areas are also strongly influenced because it means a higher population density, making COVID-19 inferior. Several parameters which can affect COVID-19 spread could be considered as a potential carrier. Population density also matters in epidemic spreading. Older people are more susceptible to the epidemic. Figures 6(a) and 6(b) depict the total number of cases without weather, while Figures 6(c) and 6(d) denote deaths. In both cases, we can observe in Figures 6(a) and 6(c) that the difference between actual and predicted graph lines is more significant than Figures 6(b) and 6(d). Predictions evaluated with weather data showed that the addition of weather parameters improved predicted results.

In order to understand whether the weather parameters, that is, temperature and humidity inclusion, affect the results or not, we created more comprehensive time series prediction models using ARIMA, linear regression, SVR, MLP, GRU, and LSTM. The current time series prediction model gives better facilitation to elucidate the impact of weather parameters on epidemic spreading. In addition, these time series models aid in illustrating the authentic interrelationship among the number of proved cases, deaths, and weather factors. Tables 1 and 2 predict the total number of cases (actual vs. predicted) and Table 3 and 4 elaborate a total number of fatalities (actual vs. predicted) with and without including weather variables, where the performance of these models are indicated in terms of MSE, RMSE, and MAPE.

5. Discussion

This study aims to figure out an output of seven-time series prediction models with and without weather data on the total number of COVID-19 cases and their mortality. In Table 1, it is clear that LSTM achieved better results with lower MSE, RMSE, and MAPE values. For illustration, the LSTM model achieved MAPE values 0.022, 0.0217, 0.0208, 0.0198, 0.0176, 0.0164, and 0.0155 for the total number of cases with weather data in Pakistan. Thus, the results depict that prediction of new COVID-19 confirmed cases by LSTM has sublime performance. The efficiency of the actual versus predicted total number of cases with weather data for COVID-19 is promising and evident. LSTM's outperforming ability to handle fewer datasets than the other models (linear regression, SVR, MLP, RNN, and GRU) which possibly require lengthier data to evaluate correlated fluctuation in time series data has made it a better choice. Conversely, RNN and its updated version GRU accommodate comparatively balanced forecasting performance due to the evaluation metrics (RMSE and MAPE), and explained variance is executed ambiguously.

The performance of time series models ARIMA, linear regression, SVR, MLP, RNN, GRU, and LSTM in MSE, RMSE, and MAPE predicting the total number of cases without considering weather data parameters (temperature and humidity) is shown in Table 2. It is clear that values of MSE, RMSE, and MAPE for all-time series prediction models were enhanced without the addition of weather data; for example, GRU showed values of 180.8178718, 13.4468536, and 0.018989281 for MSE, RMSE, and MAPE without weather data. In contrast, it was 140.0163399, 11.83285003, and 0.01641411, respectively, for the number of cases with weather parameters.

Similarly, Table 3 shows the application of time series prediction models on the number of deaths in Pakistan, considering the weather parameters, temperature, and humidity. LSTM shows the best MSE, RMSE, and MAPE values, that is, 1711, 41.36423576, and 0.492211157, respectively. But in Table 4, it is predicted that if we see performance of time series models ARIMA, linear regression, SVR, MLP, RNN, GRU, and LSTM in terms of MSE, RMSE, and MAPE without temperature and humidity, the accuracy of models descends. In both Tables 3 and 4, the ARIMA model shows the least accuracy and high error values, and LSTM predicts the least values of MSE, RMSE, and MAPE.

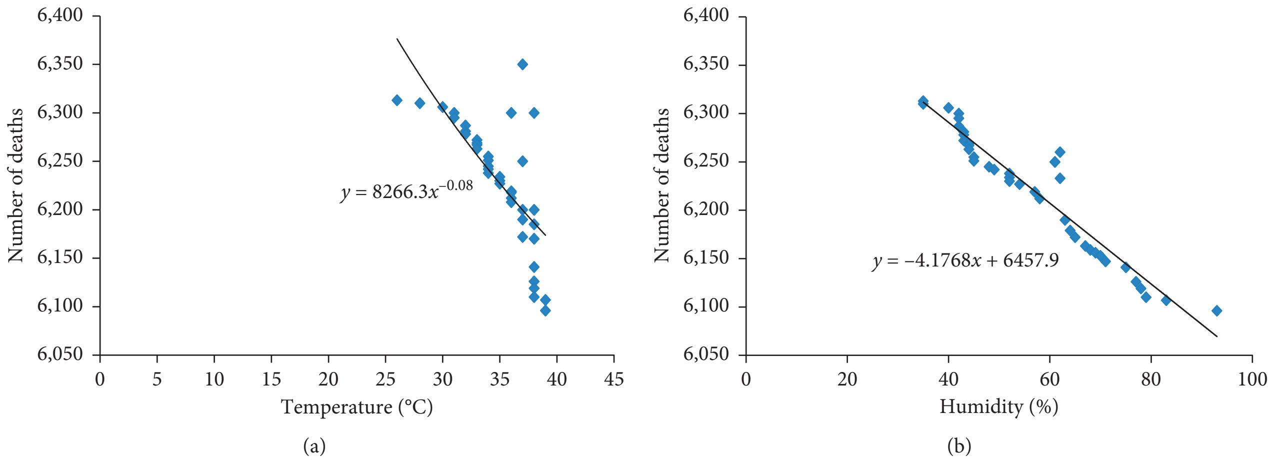


Figure 5: Scatter plot showing the number of deaths in Pakistan based on temperature (a) and humidity (b).

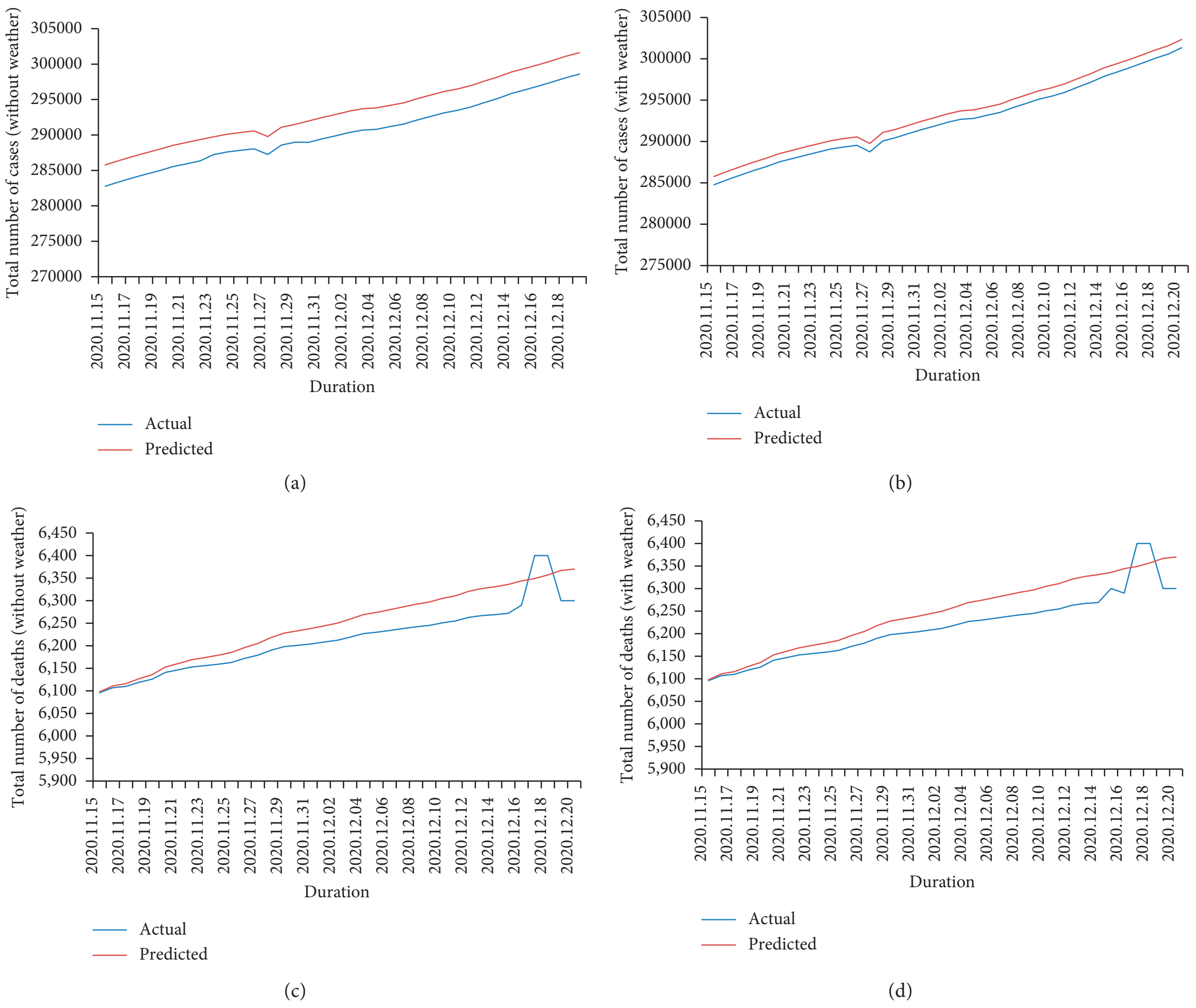


Figure 6: Total number of COVID-19 cases (actual vs. predicted) from November 15 to December 20, 2020, in Pakistan: (a) without weather and (b) with the weather. Total number of deaths due to COVID-19 (actual vs. predicted) from November 15 to December 20, 2020: (c) without weather and (d) with the weather.

Table 1: Validation metric for total number of cases (actual vs. predicted) with weather.

Methods	MSE	RMSE	MAPE
ARIMA	**374.0833333**	**19.34123402**	**0.022592539**
Linear regression	348.6187913	18.67133609	0.021763819
SVR	310.8357811	17.63053547	0.020891829
MLP	274.3420311	274.3420311	0.019862762
RNN	190.9222324	13.81746114	0.017662732
GRU	140.0163399	11.83285003	0.01641411
LSTM	103.5873714	10.17778814	0.015592019

Table 2: Validation metric for total number of cases (actual vs. predicted) without weather.

Methods	MSE	RMSE	MAPE
ARIMA	407.1029011	20.17679115	0.025818721
Linear regression	380.0182181	19.49405597	0.023817828
SVR	330.1928918	18.17121052	0.022661666
MLP	301.18281	17.35461927	0.020109929
RNN	210.4517527	14.50695532	0.019827818
GRU	180.8178718	13.4468536	0.018989281
LSTM	**135.9179218**	**11.65838418**	**0.017928821**

Table 3: Validation metric for total number of deaths (actual vs. predicted) with weather.

Methods	MSE	RMSE	MAPE
ARIMA	2214.672	47.05316142	0.65601851
Linear regression	2167.525	46.55668588	0.625752876
SVR	2112.525	45.96221274	0.614681416
MLP	1953.075	44.20124433	0.600669098
RNN	1891.225	43.48821679	0.590657314
GRU	1797.575	42.39781834	0.544018402
LSTM	**1711.188**	**41.36423576**	**0.492211157**

Table 4: Validation metric for total number of deaths (actual vs. predicted).

Methods	MSE	RMSE	MAPE
ARIMA	2270.907	47.644517	0.666156352
Linear regression	2190.502	46.8027777	0.647128252
SVR	2135.556	46.20606021	0.639812675
MLP	1970.124	44.38468204	0.628978829
RNN	1910.327	43.70354677	0.61928292
GRU	1821.201	42.67551991	0.585261726
LSTM	**1745.017**	**41.77319715**	**0.541672819**

Based on our study results shown in Tables 1–4, it can be illustrated that weather parameters like moisture and temperature can pervade SARS-CoV-2. From the results, we can conclude that there can be elevated epidemic spreading when atmospheric temperature and humidity descend. While on the other hand, when both temperature and humidity are high, the infection rate of SARS-CoV-2 declines. In forecasting the total number of cases and total deaths with and without weather data in Pakistan, the current research illustrated comparability among deep learning models using time series models ARIMA, linear regression, SVR, MLP, RNN, GRU, and LSTM to training datasets. The present study findings elucidate the sublime performance of LSTM over other models by showing high accuracy and precision compared to other time series prediction models.

Table 5: Parameter settings of the studied approaches.

Methods	Parameters	Values
ARIMA	(p, d, q)	(1, 1, 14)
Linear regression		
SVR	C	3
	Epsilon	0.0000001
	Degree	3
	Tolerance	0.000001
MLP	Learning rate	0.004
	Time step	4
	Invisible units	64
	Training epochs	1000
RNN	Learning rate	0.005
	Time step	5
	Invisible units	16
	Training epochs	1000
GRU	Learning rate	0.0005
	Time step	5
	Invisible units	16
	Training epochs	1000
LSTM	Learning rate	0.0005
	Time step	5
	Invisible units	16
	Training epochs	1000

In this study, we focused on the number of cases and death cases from Pakistan. First, each model is trained. Then, we forecast each variable. Parameters of the constructed ARIMA, linear regression, SVR, MLP, RNN, GRU, and LSTM models based on training datasets are presented in Table 5.

6. Conclusion

The present investigation analyzed the effect of prime weather factors (temperature and humidity) on the number of reported cases and deaths due to COVID-19 in Pakistan. Different time series prediction models such as ARIMA, linear regression, SVR, MLP, RNN, GRU, and LSTM were used, and the execution of each model was analyzed in terms of MSE, RMSE, and MAPE. Results illustrated that the LSTM could better predict the COVID-19 spread as compared to other models. From the present results, we can deliberately conclude that weather holds significance in COVID-19 prediction. Thus, it is advised to wear masks and personal protective wears, keep social distancing, and continue isolation (on infection/suspect) until the temperature rises or the vaccine is fully deployed. Further, predicting the COVID-19 spread/incidence by considering other weather parameters like rainfall, wind speed, and so forth shall provide additional clues to mitigate the epidemic.

Data Availability

The data used to elaborate the results and findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Humera Batool conceptualized the study, developed the methodology, performed formal analysis, reviewed and edited the article, performed validation, and performed visualization. Lixin Tian reviewed the article, performed supervision, and performed project administration.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 71690242, 91546118, and 11731014).

References

[1] C. Paules, H. Marston, and A. Fauci, "Infeksi coronavirus—lebih dari sekedar pilekCoronavirus Infections-More Than jold," *JAMA*, vol. 323, pp. 707-708, 2020.

[2] Q. Bukhari, J. M. Massaro, R. B. D'agostino, and S. Khan, "Effects of weather on coronavirus pandemic," *International journal of environmental research and public health*, vol. 17, p. 5399, 2020.

[3] Worldometer, *Covid-19 coronavirus pandemic*, Worldometer, Yorkville, IL, USA, 2020.

[4] K. Kupferschmidt and J. Cohen, "Can china's covid-19 strategy work elsewhere?" *Science*, vol. 367, no. 6482, pp. 1061-1062, 2020.

[5] J. Hamzelou, "World in lockdown," *New Science*, vol. 245, pp. 30611–30614, 2020.

[6] M. M. Sajadi, P. Habibzadeh, A. Vintzileos, S. Shokouhi, F. Miralles-Wilhelm, and A. Amoroso, "Temperature, humidity and latitude analysis to predict potential spread and seasonality for covid-19," *SSRN*, vol. 9, p. 3550308, 2020.

[7] M. Yousaf, S. Zahir, M. Riaz, S. M. Hussain, and K. Shah, "Statistical analysis of forecasting covid-19 for the upcoming month in Pakistan," *Chaos Solitons Fractals*, vol. 138, p. 25, 2020.

[8] L. M. Casanova, S. Jeon, W. A. Rutala, D. J. Weber, and M. D. Sobsey, "Effects of air temperature and relative humidity on coronavirus survival on surfaces," *Applied Environment Microbiology*, vol. 76, pp. 2712–2717, 2010.

[9] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus covid-19," *PloS One*, vol. 15, 2020.

[10] J. Demongeot, Y. Fleet-Berliac, H. Seligmann et al., "Temperature decreases spread parameters of the new covid-19 case dynamics," *Biology*, vol. 9, 2020.

[11] M. Sajadi, P. Habibzadeh, A. Vintzileos, S. Shokouhi, F. Miralles-Wilhelm, and A. Amoroso, "Temperature, humidity, and latitude analysis to estimate potential spread and seasonality of coronavirus disease 2019 (covid-19)," *JAMA Netw OpenJAMA network open*, vol. 3, p. 11834, 2020.

[12] K. Bloom-Feshbach, W.J. Alonso, and V. Charu, "Latitudinal variations in seasonal activity of influenza and respiratory syncytial virus (RSV): A global comparative review," *PloS One*, vol. 8, p. 14, 2013.

[13] K. Prem, Y. Liu, T. W. Russell et al., "The effect of control strategies to reduce social mixing on outcomes of the covid-19 epidemic in Wuhan, China: A modelling study," *Lancet Public Health*, vol. 5, pp. e261–e270, 2020.

[14] S.E. Eikenberry, M. Mancuso, and E. Iboi, "To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the covid-19 pandemic," *Infection Disease Model*, vol. 5, pp. 293–308, 2020.

[15] National Institute of Health, Islamabad, Pakistan, https://www.nih.org.pk/, 2020.

[16] J. Contreras, R. Espinola, F.J. Nogales, and A.J. Conejo, "Arima models to predict next-day electricity prices," *IEEE transactions on power systems*, vol. 18, pp. 1014–1020, 2003.

[17] R. Adhikari and R.K. Agrawal, "An introductory study on time series modelling and forecasting," 2013, http://arxiv.org/abs/1302.6613.

[18] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1996.

[19] D.S. Hui, I.A. E, T.A. Madani et al., "The continuing 2019-ncov epidemic threat of novel coronaviruses to global health - the latest 2019 novel coronavirus outbreak in Wuhan, China," *International Journal of Infectious Diseases*, vol. 91, pp. 264–266, 2020.

[20] Y. Bengio, *Learning Deep Architectures for AI*, Now Publishers Inc, The Netherlands, 2009.

[21] A. Graves, "Generating sequences with recurrent neural networks," 2013, http://arxiv.org/abs/1308.0850.

[22] A. Zeroual, F. Harrow, A. Dairi, and Y. Sun, "Deep learning methods for forecasting covid-19 time-series data: A comparative study," *Chaos Solitons FractalsChaos, solitons, and fractals*, vol. 140, p. 15, 2020.

[23] F. Harrow, F. Kadri, and Y. Sun, "Forecasting of photovoltaic solar power production using lstm approach," *Advanced Statistical Modeling, Forecasting, and Fault Detection in Renewable Energy Systems*, Springer, Berlin, Germany, 2020.

[24] A.S. Ashour, A. Attar, N. Dey, H. Abd Elkader, and M. Elnaby, "Long short term memory based patient- dependent model for fog detection in Parkinson's disease," *Pattern recognition letters*, vol. 131, 2019.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, http://arxiv.org/abs/1406.1078.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modelling," 2014, http://arxiv.org/abs/1412.3555.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, pp. 1735–1780, 1997.

[28] G. Forbes, S. Massie, and S. Craw, "Fall prediction using behavioural modelling from sensor data in smart homes," *Artificial Intelligence Review*, vol. 53, pp. 1071–1091, 2020.

[29] R. Law, G. Li, D. Fong, and X. Han, "Tourism demand forecasting: A deep learning approach," *Annals of Tourism Research*, vol. 75, pp. 410–423, 2019.

[30] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transition Neural Network*, vol. 5, pp. 157–166, 1994.

[31] T. Trappenberg, *Machine learning with sklearn*, pp. 38–65, Oxford University Press, Oxford, UK, 2019.

[32] W. M. Lee, *Getting Started with Scikit-learn for Machine Learning*, pp. 93–117, John Wiley & Sons, Inc. Hoboken, NJ, USA, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2334236, 16 pages
https://doi.org/10.1155/2021/2334236

Research Article

Constraint Satisfaction for Motion Feasibility Checking

Seokjun Lee and Incheol Kim

Department of Computer Science, Kyonggi University, San 94-6, Yiui-Dong, Youngtong-Gu, Suwon-Si 443-760, Republic of Korea

Correspondence should be addressed to Incheol Kim; kic@kyonggi.ac.kr

Received 9 April 2021; Revised 9 May 2021; Accepted 18 May 2021; Published 27 May 2021

Academic Editor: Zain Anwar Ali

Task and motion planning (TAMP) is a key research field for robotic manipulation tasks. The goal of TAMP is to generate motion-feasible task plan automatically. Existing methods for checking motion feasibility of task plan skeletons have some limitations of semantic-free pose candidate sampling, weak search heuristics, and early value commitment. In order to overcome these limitations, we propose a novel constraint satisfaction framework for checking motion feasibility of task plan skeletons. Our framework provides (1) a semantic pose candidate sampling method, (2) novel variable and constraint ordering heuristics based on intra- and inter-action dependencies in a task plan skeleton, and (3) an efficient search strategy using constraint propagation. Based upon these techniques, our framework can improve the efficiency of motion feasibility checking for TAMP. From experiments using the humanoid robot PR2, we show that the motion feasibility checking in our framework is 1.4x to 6.0x faster than previous ones.

1. Introduction

In recent years, Artificial Intelligence (AI) has become an increasingly common presence in robotic solutions, introducing flexibility and learning capabilities in previously rigid applications. In this study, we address the issues in applying AI constraint satisfaction problem (CSP) solving techniques to task and motion planning (TAMP) [1–37], which is a key research field of robotic manipulation tasks. A typical TAMP involves the combination of task planning [38–40] that generates a sequence of actions for satisfying a goal condition in the current state based on a symbolic abstract action model and motion planning [41–43] that checks the motion feasibility for determining if the motions can be executed in a physical space. The ultimate goal of TAMP is to combine these to generate a motion-feasible task plan.

Task planning determines the logical order of actions. It knows symbolic knowledge such as type of action, parameters, preconditions, and effects. However, in task planning, the goal pose (or configuration), path (or trajectory), etc., to execute each action are unknown. This disadvantage can be supplemented through motion planning. While abstract symbolic knowledge is unknown in motion planning, it can generate collision-free paths for executing an action in a physical space. To combine the two planning methods, which have different advantages and disadvantages, an intermediate (interface) layer is required. Figure 1 shows an example of combining task and motion planning to generate a motion-feasible task plan. The left side of Figure 1 shows the task layer including the task planning process.

The right side of Figure 1 shows the motion layer including the motion planning process. With the help of the task layer, the interface layer generates a task plan, that is, a task plan skeleton that includes unbound pose parameters. Once the task plan skeleton is generated, the interface layer generates candidates of pose parameters of each action. As motion planning cannot generate the goal pose on its own, the interface layer must play a role in generating candidates of pose parameters. The interface layer then checks if there is a collision-free path between candidates of neighboring pose parameters with the help of the motion layer. If all of collision-free paths exist, then the candidates are assigned to the pose parameters of the task plan skeleton to generate a motion-feasible task plan. Otherwise, it creates a new task planning problem from motion-related errors (e.g., obstacles), and repeats the same interfacing process.

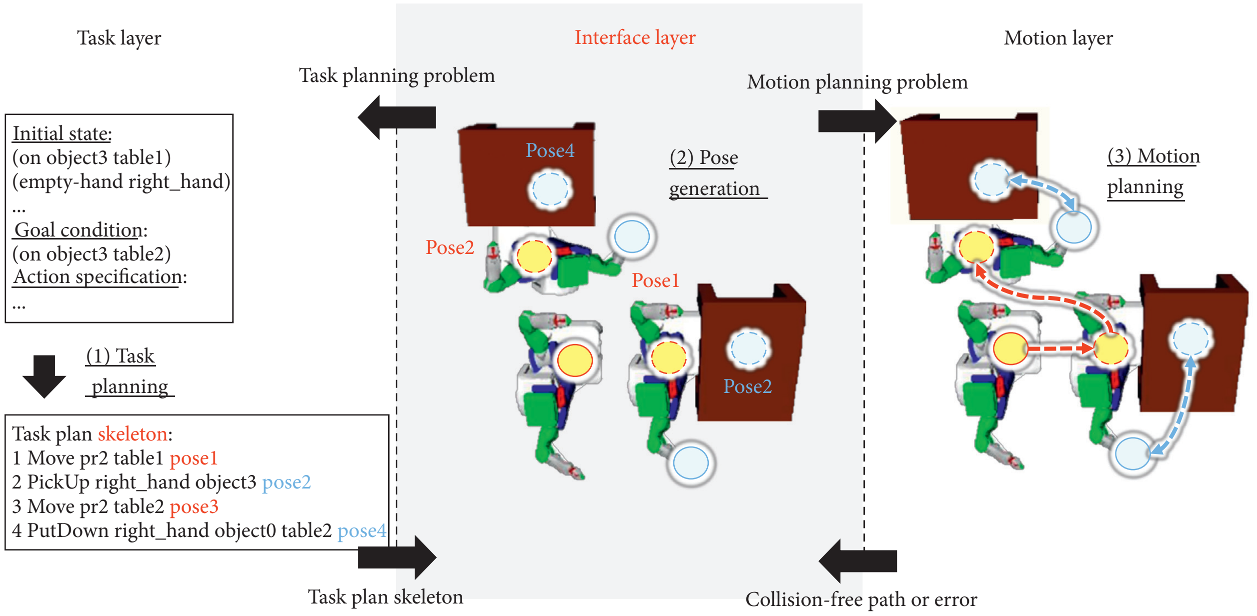


Figure 1: An example of combining task and motion planning using the interface layer.

This paper focuses on checking the motion feasibility of the task plan skeleton in the interface layer. Motion feasibility checking involves finding motion-feasible values of unbound pose parameters included in the task plan skeleton. This is a challenging problem with a very complex search space. It expands the motion planning problem that finds the collision-free path of a single behavior to the problem of validating the pose candidates and collision-free paths of multiple actions. To solve this problem effectively, the followings are required: (1) theoretical modelling of the search problem to find motion-feasible values of pose parameters, (2) a method for generating candidates of the value of pose parameters in continuous space, and (3) search strategies and heuristics (e.g., using constraints that the pose parameters must satisfy).

The previous study of Lozano-Pérez and Kaelbling [18] attempted to model and to solve the motion feasibility checking problem as a traditional CSP (Constraint Satisfaction Problem) [44, 45]. Garrett et al. [37] attempted to model and to solve the motion feasibility checking problem as on-the-fly CSP. Lagriffoul et al. [17] attempted to model and to solve the motion feasibility checking problem as a general search tree. These studies have limitations, including generating pose candidates with a high probability of failure due to simple sampling methods, or developing search strategies based on only general-purpose heuristics that is not suitable for motion feasibility checking. Furthermore, they attempted to determine values in advance that must be redetermined at the time of execution, such as IK (Inverse Kinematics) and collision-free path with high variability. It complicates the search problem and makes planning and execution inefficient in the long run.

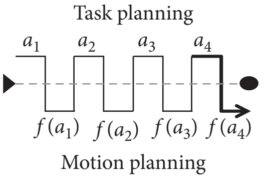
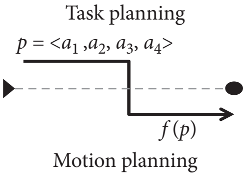
This paper proposes an efficient constraint satisfaction framework to address the motion feasibility checking problem. Motion feasibility checking is the problem that finds a valid (or motion-feasible) value of pose parameters. The valid value means that it satisfies constraints such as kinematics and collision-free path. In our framework, the motion feasibility checking is modelled as a constraint satisfaction problem (CSP). The CSP formulation has the advantage that the various types of constraints can be represented equally in a concise form. In addition, effective general-purpose or special-purpose search strategies and heuristics can be used to solve both discrete and continuous constraint satisfaction problems. (1) Our framework proposes a method of generating pose candidates guided by the semantics of unbound pose parameters and already bound task parameters. This method reduces the search space substantially. Meanwhile, actions belonging to the task plan skeleton have inter-action dependencies because of logical order. In addition, parameters and preconditions within an action have intra-action dependencies. (2) Based on these inter- and intra-action dependencies, novel, variable, and constraint ordering heuristics are proposed in our framework. (3) Moreover, our framework employs an efficient search strategy with constraint propagation [44] to detect failure early or speed up the search substantially. To verify the practical applicability and performance of the proposed framework, this study performs various experiments using humanoid robots that can perform mobile manipulation.

The rest of the paper is organized as follows. Section 2 introduces the related works in the field of TAMP. Section 3 provides the problem statement. In Section 4, the proposed framework with CSP modelling and heuristic search methods is presented in detail. In Section 5, experimental results are reported. Finally, Section 6 summarizes our work and discusses some limitations and future works.

2. State of Art

We categorize previous works according to their motion feasibility checking methods, as shown in Table 1. There are two different methods for checking motion feasibility. The

Table 1: Categorization of previous works.

Type	Illustration
Eager [1–14, 37]	 Task planning a_1 a_2 a_3 a_4 $f(a_1)$ $f(a_2)$ $f(a_3)$ $f(a_4)$ Motion planning
Lazy [15–37]	 Task planning $p = <a_1, a_2, a_3, a_4>$ $f(p)$ Motion planning

eager method [1–14] alternately performs task planning and motion-feasibility checking step-by-step. On the contrary, the lazy method [15–37] postpones its motion-feasibility checking until a complete skeleton of task plan is built. The eager method has the advantage of being able to identify in advance future motion-unfeasible actions that the task planner cannot identify during the task planning process. However, the cost of generating the motion plan is high because motion planning is continuously interleaved during task planning. Although the lazy method cannot identify in advance the motion feasibility of the actions during task planning, it limits the search space of the motion planning to a small range since the task plan to achieve the goal condition is already defined. In addition, it examines different knowledge contained in the task plan skeleton and applies a wide range of search strategies and heuristics. This paper utilizes the lazy method.

Especially, Lozano-Pérez and Kaelbling [18], Garrett et al. [37], and Lagriffoul et al. [17] are closely related to our study, because they focus on motion feasibility checking of task plan skeletons using constraint satisfaction methods. First, Lozano-Pérez and Kaelbling [18] attempted to model the motion feasibility checking problem as a traditional CSP. In [18], the depth-first backtracking search algorithm and constraint propagation considering the dependency between constraints are used. However, [18] used only general-purpose variable ordering heuristics such as MRV (Minimum Remaining Value), which did not consider some dependencies between actions and in an action. Furthermore, [18] generated pose candidates with high probability of failure and dealt with only the parameters and constraints related to the manipulation actions.

Next, Garrett et al. [37] attempted to model the motion feasibility checking problem as on-the-fly CSP. In particular, [37] proposed a conditional sampling method based on the constraint network. However, the CSP modelled by [37] had some variables that cause high cost of motion planning, such as IK and trajectories. It makes the search problem more complex. The method of [37] is also less scalable because it only considers general-purpose CSP heuristics.

Finally, Lagriffoul et al. [17] attempted to model the motion feasibility checking problem as a general search tree. The nodes of the tree represent pose parameters, and the edges represent collision-free trajectories. Based on the depth-first backtracking search, it finds the values of pose parameters with verified validity for the trajectory. In particular, [17] attempted to reduce the number of backtracks by propagating linear constraints before visiting the next nodes. It is the same as forward checking in CSP. However, search tree does not represent all constraints in a unified form, and its simple sampling method generates many samples with a high probability of failure. In addition, [17] dealt only with the parameters and constraints associated with manipulation actions.

In order to overcome the limitations of these existing studies, our framework provides (1) a semantic pose candidate sampling method, (2) novel variable and constraint ordering heuristics based on intra- and inter-action dependencies in a task plan skeleton, and (3) an efficient search strategy using constraint propagation. Based upon these techniques, our framework can improve the efficiency of motion feasibility checking for TAMP.

In addition, there are several recent notable works on TAMP (Task and Motion Planning). The authors of [32] proposed a TAMP framework using a top-k skeleton planner to produce diverse skeletons, guaranteeing that no better solution exists under a current domain description. Moreover, the framework uses a Monte-Carlo Tree Search (MCTS) to solve this stochastic decision-making problem over skeletons and concrete bindings of the action parameters. The authors of [33] proposed a novel online planning and execution system to solve a TAMP problem as a hybrid partially observable Markov decision process (POMDP) and use past plans to constrain the structure of solutions for the current planning problem. On the other hand, both [34, 35] addressed multi-agent or multi-robot TAMP problems. The authors of [34] dealt with a new problem of motion planning feasibility checking for task-agent assignment to perform complex tasks using multi-arm mobile manipulators. The authors of [35] presented a multi-robot integrated task and motion planning method capable of handling sequential subtasks dependencies within multiple decomposable tasks. Interestingly, [36] proposed a TAMP method for efficient and safe autonomous urban driving, different from robotic manipulation tasks.

3. Problem Statement

We propose an efficient constraint satisfaction framework to check the motion feasibility of the task plan skeleton. We assume that a task plan skeleton has already been generated. Then, we focus on checking its motion feasibility.

Figure 2 illustrates an example environment including a humanoid robot with two arms (e.g., PR2) to manipulate objects on tables. In Figure 2, PR2 must move the object from one table to the other.

We remark the task plan skeleton as follows:

Remark 1 (task plan skeleton). A task plan skeleton is a sequence of actions <*a*0, . . ., *ak*> following the states transition <*s*0, . . ., *sk*+1>, where a∈A is an action, s0 the

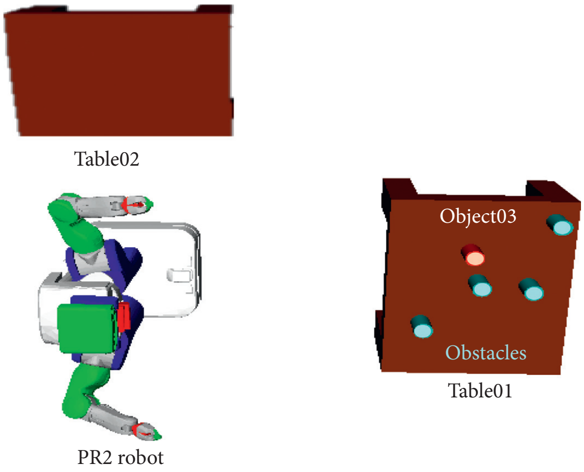


Figure 2: Task environment for mobile manipulation.

initial state, and *sk*+1 the goal state, which satisfies the goal condition. Meanwhile, each action *a* has pose references.

The pose references are discrete identifiers for representing the values of the poses still undetermined in the task planning phase. The pose references are regarded to have values that satisfy the preconditions of the actions. To generate the task plan skeleton, the pose references are temporarily bound to the pose parameters.

Table 2 shows an example of a task plan skeleton generated from the task environment in Figure 2. This task plan skeleton consists of a sequence of actions in which the robot moves *object03* to *table 02*.

The actions that comprise the task plan skeleton are specified according to the schema below.

Remark 2 (action schema). Action schema is a tuple (*α* (*Pt*, *Pp*), pre (*Pt*, *Pp*, *V*), eff (*Pt*, *Pp*, *V*)), where *α* is an action symbol; *Pt* is a set of task-level parameters such as object, robot, etc.; *Pp* is a set of pose parameters; pre (*Pt*, *Pp*, *V*) is a conjunction of predicates representing preconditions of action (where *V* are additional variables, $V \neq Pt$, *Pp*); and eff (*Pt*, *Pp*, *V*) is a conjunction of predicates representing effects of action.

Table 3 shows the specifications of the actions for navigation. In the parameters for each action, there are numerical parameters that were not in the previous action specification. For example, *baseLoc0* and baseLoc1 are pose parameters, where *baseLoc0* is the robot's current location and baseLoc1 the goal location. During the generation of the task plan skeleton, these parameters will be bound to pose references. Based on the preconditions of the action, the pose parameters must satisfy the state conditions at the motion level. For example, the baseLoc1 parameter of *MoveBasePickUp* must satisfy state conditions such as *reachableBasePose* and *validBasePath*. Section 4 details the specific meanings of these state conditions.

Table 4 shows the specifications of the actions for manipulation. First, *PickUpGeneral* is the action of moving a hand to grasp an object and then returning the hand back to its initial pose. This action contains many motions, thus requiring a relatively large number of pose parameters. This action includes the initial hand pose (*handPose0*), the pregrasping hand pose (*handPose*1), the grasping hand pose (*handPose*2), and the post-grasping hand pose (*handPose*3). *PutDownGeneral* is the action of putting the grasped object on a support plane such as a table and returning the hand to the initial pose. This action has pose parameters similar to the *PickUpGeneral* action. However, the *PickUpGeneral* action includes a pose parameter for the placement position of the object. Section 4 details the specific meanings of these state conditions.

Under the assumptions described above, we summarize the main problem for checking the motion feasibility of the task plan skeleton as follows.

Remark 3 (motion feasibility checking). Given a task plan skeleton *S*, the problem is to find a sequence of valid values <*v*1, . . ., *vk*> of pose references in *S*. The valid values <*v*1, . . ., *vk*> satisfy preconditions of corresponding action.

We attempt to model and solve this motion feasibility checking problem by converting it to a CSP [32, 33]. CSP refers to the problem of finding a value that satisfies a plurality of constraints within a domain. The valid values of the undetermined pose parameters must satisfy the state conditions presented in the task-planning problem so that they can be fully formulated as a CSP. As poses exist in a continuous space, discrete domains are created to find a solution in a reasonable amount of time. Thus, the CSP considered in this paper is a discrete CSP. Now, we summarize subproblems as follows:

Remark 4 (constructing discrete CSP). Given a task plan skeleton *S*, the problem is to construct a discrete CSP csp = (*V*, *D*, *C*) from the task plan skeleton, where *V* is a set of variables, *D* is a set of the respective domains of values, and *C* is a set of constraints.

Remark 5 (solving CSP). Given a discrete CSP csp, the problem is to search valid values of each variable in *V*.

4. Proposed Solution

The task plan skeleton contains unbound pose references. Values for the pose references must satisfy the motion-related constraints, such as IK and the collision-free path. If any value satisfying all constraints exists, then we can consider that the task plan skeleton is motion feasible. This is a constraint-based search problem. So, we model the problem for checking the motion feasibility as a CSP.

Figure 3 describes the process of motion feasibility checking. First, it generates CSP statements from a task plan skeleton. Second, it reduces the size of domain through preprocessing before solving CSP. Last, it finds a solution and decides if the task plan skeleton is motion feasible or not.

4.1. Formulating Constraint Satisfaction Problem from Plan Skeleton

4.1.1. Variables. First, we have to formulate the motion feasibility checking problem into a constraint satisfaction

Table 2: An example of a task plan skeleton to move an object to *table02* from *table01*.

1. (MOVEBASEFORPICKCUP pr2 object03 B_{init} B_{reach})
2. (PICKUPGENERAL rgirpper object03 H_{init} H_{pre} H_{grasp} H_{post})
3. (MOVEBASEFORPUTDOWN pr2 rgripper object03 table02 B_{reach} B_{place})
4. (PUTDOWNGENERAL rgirpper object03 table02 O_{place} H_{init} H_{low} H_{place} H_{ret})

Table 3: Action specifications for navigation.

action MoveBaseForPickUp
param robot, obj, baseLoc0, baseLoc1
precond robotAt (baseLoc0), objectAt (obj, objLoc), reachableBasePose (robot, obj, objLoc, baseLoc1), validBasePath (robot, baseLoc0, baseLoc1)
effect—robotAt (baseLoc0), robotAt (baseLoc1)
action MoveBaseForPutDown
param robot, hand, obj, plane, baseLoc0, baseLoc1
precond robotAt (baseLoc0), graspedBy (obj, hand), placeableBasePose (robot, obj, plane, baseLoc1), validBasePath (robot, baseLoc0, baseLoc1)
effect—robotAt (baseLoc0), robotAt (baseLoc1)

Table 4: Action specifications for manipulation.

action PickUpGeneral
param obj, hand, handPose0, handPose1, handPose2, handPose3
precond empty (hand), robotAt (baseLoc), objectAt (obj, objLoc), handAt (hand, handPose0), graspableHandPose (hand, obj, objLoc, baseLoc, handPose2), preGraspableHandPose (hand, obj, objLoc, baseLoc, handPose2, handPose1), poseGraspableHandPose (hand, obj, objLoc, baseLoc, handPose2, handPose3), validHandPath (hand, baseLoc, handPose0, handPose1), validHandPath (hand, baseLoc, handPose1, handPose2), validHandPath (hand, baseLoc, handPose2, handPose3), validHandPath (hand, baseLoc, handPose3, handPose0)
effect graspedBy (obj, gripper), —empty (gripper), —objectAt (obj, objLoc)
action PutDownGeneral
param obj, hand, plane, objLoc, handPose0, handPose1, handPose2, handPose3
precond graspedBy (obj, hand), robotAt (baseLoc), handAt (hand, handPose0), placeableObjectLocation (obj, plane, objLoc), lowerableHandPose (hand, obj, objLoc, baseLoc, handPose2), placeableHandPose (hand, obj, objLoc, baseLoc, handPose2, handPose1), retractableHandPose (hand, obj, objLoc, baseLoc, handPose2, handPose3), validHandPath (hand, baseLoc, handPose0, handPose1), validHandPath (hand, baseLoc, handPose1, handPose2), validHandPath (hand, baseLoc, handPose2, handPose3), validHandPath (hand, baseLoc, handPose3, handPose0)
effect—graspedBy (obj, gripper), objectAt (obj, objLoc)

problem (CSP). Usually, goal pose, grasp, IK, and path were modelled as CSP variables in a way a similar to that in [17, 18, 37]. But, we only model the goal pose to the CSP variable, that is, modelling the path to the CSP variable. The reason for this is as follows: (1) If too many variables are modelled, it increases the complexity of search. In addition, the domain of variable (for path especially) is very difficult and expensive. (2) To bind the value of anything other than the pose is not required necessarily in the planning phase. Considering changes of the environment during planning time, even if the IK, paths, etc., are determined at the planning phase, rechecking is required at the execution phase. It is appropriate that these are assigned once at the execution phase.

We categorize the poses into three types. The types include position of mobile base B, pose of hand H, and position of object O. The variables are extracted from the pose references of the actions the task plan skeleton. Table 2 is a mapping table of the abovementioned variables. As shown in Table 5, the pose references bound to each action are mapped as variables of the CSP. However, since the initial poses of mobile base, hand, and object are constants that can be read directly from the environment, these pose references are not mapped to variables of the CSP. In addition, a pose reference that has already been mapped as B_{reach} is not mapped in duplicate.

4.1.2. Domains. Each variable is represented by a 4×4 homogeneous matrix in continuous space. For execution, a discretized deterministic value must be bound to each variable. For this reason, we generate discrete domains based on the sampling guided by the semantic of the task plan skeleton. To generate the discrete domains, the geometric constants are retrieved from the value of the bounded parameter such as mobile base, hand, object, and support plane. For example, referring the value of bounded parameters, we can know that the domain of B_{reach} should be near the object and the orientation of B_{reach} should be towards the object. This can be represented by the following:

$$B_{\text{reach}} = \left(r_2 \cdot t_0 \cdot r_1 \cdot r_0\right) O_{\text{init}}, \tag{1}$$

where r_0 is a matrix representing an axis angle $a = (0, 0, 0)$, r_1 is a matrix representing a quaternion $q = (0, 0, 0, 1)$, t_0 is a matrix representing a pose $p = (1, 0, 0, 0, d, 0, 0)$, and r_2 is a matrix representing an axis angle $a = (0, 0, \theta)$. The details of the equations that derive the matrix from the axis angle, quaternion, etc., are provided in the appendix.

Equation (1) is to calculate the relative position and orientation of the mobile base relative to the coordinate system of the object. This equation has two parameters: d and θ, where d indicates the distance between the center of the robot and the center of the object, whereas θ indicates the direction of the robot towards the object. d can be appropriately determined in consideration of the arm length. θ is calculated by equation (2):

$$\theta = \frac{2i\pi}{n}, \quad \text{where } i = 1, \ldots, n, i, \in N,\ n \in N. \tag{2}$$

If the number of d is m, then the number of domains is mn.

Figure 4 shows the area of the domain of B_{reach}. In Figure 4(a), the circular dotted line that maintains the

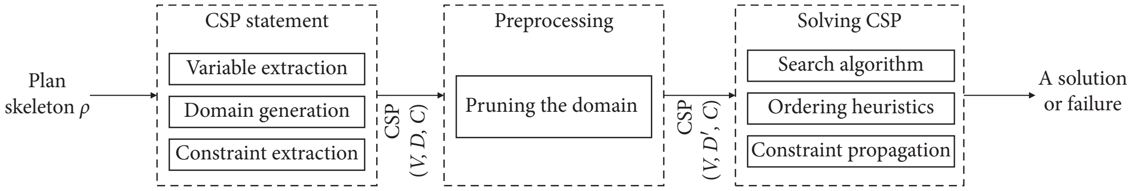


Figure 3: Motion feasibility checking process.

Table 5: Mapping pose references to CSP variables.

Action	Bounded parameters	Pose references	CSP variables
MoveBase ForPickUp	pr2, objeect03	O_{init}, B_{init}, B_{reach}	B_{reach}
PickUpGeneral	pr2, rgripper, object03	O_{init}, H_{init}, H_{pre}, H_{grasp},	H_{pre}, H_{grasp}, H_{post}
MoveBase ForPutDown	pr2, object03, table02	B_{reach}, B_{place}	B_{place}
PutDownGeneral	pr2, rgripper, object03 table02	O_{place}, H_{init}, H_{low}, H_{place}, H_{ret}	H_{low}, H_{place}, H_{ret}

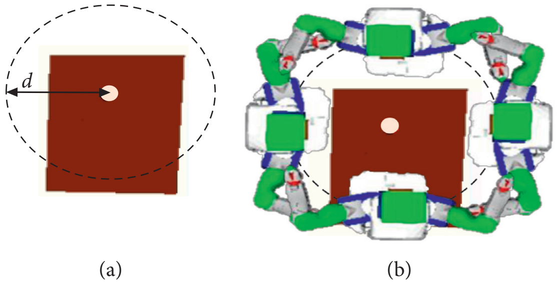


Figure 4: Domain generation of B_{each}. (a) The range of domain and (b) the discrete domain (when $m = 1$, $n = 4$).

distance d with the object is the range of domain. Figure 4(b) shows an example of a domain generated when m is 1 and n is 4. The domain of B_{place} is generated in the similar way as the domain of B_{reach}. However, it calculates the distance d on the basis of the center of the table.

Also, we can know that the domain of H_{grasp} should be located at a location very close to the object and the orientation of B_{reach} should be towards the object. This can be represented in an equation (3) as follows:

$$H_{\text{grasp}} = (t_1 \cdot r_2 \cdot t_0 \cdot r_1 \cdot r_0) O_{\text{init}}, \tag{3}$$

where r_0 is a matrix representing a quaternion $q = (0, 0, 0, 1)$, r_1 is a matrix representing an axis angle $a = (0, -(\pi/2), 0)$, t_0 is a matrix representing a pose $p = (1, 0, 0, 0, -d, 0, 0)$, r_2 is a matrix representing an axis angle $a = (0, 0, \theta)$, and t_1 is a matrix representing a pose $p = (1, 0, 0, 0, 0, 0, h)$.

This equation is to calculate the position and orientation of the hand relative to the coordinate system of the object. This equation has three parameters: d, θ, and h. d is the distance between the center of the hand and the center of the object, h is the height of the hand from the bottom of the object, and θ is the direction of the hand towards the object. d can be appropriately determined in consideration of the finger length. h may be determined in consideration of the volume of the hand, etc. If the number of d is m and the number of h is k, then the number of domains is mkn.

Figure 5 shows the area of the domain of H_{grasp}. In Figure 5(a), the circular dotted line that maintains the distance with the object d and the height h is the domain. Figure 5(b) shows an example of a domain generated when m is 1, k is 1, and n is 6. The domains of H_{pre} and H_{post} are sampled similar to the domain of H_{grasp}. However, d for the domain of H_{pre} is farther than d for the domain of H_{grasp}, and h for the domain of H_{post} is higher than h for the domain of H_{grasp}.

In the case of O_{place}, we can know that the domain of O_{place} should be located on the support plane. To prevent falling, the position should be slightly further inside from each corner of the support plane. When O_{place} is a matrix representing a pose $p = (1, 0, 0, 0, x, y, z)$, x, y, and z are constrained as below.

$$\begin{gathered} P_{\min X} + p < x < P_{\max X} - p, \\ P_{\min Y} + p < y < P_{\max Y}, \\ z = Pz. \end{gathered} \tag{4}$$

In equation (4), the support plane P is assumed to be rectangular. This equation has parameters x, y, z, and p. x and y show the position of the object on the support plane. z represents the height of the object. p represents the space inward from the edge of the support plane, that is, the padding. If the number of x is m, and if the number of y is n, then the number of domains is mn.

Figure 6 shows the area of the domain of O_{place}. The gray area on the table is the area of the domain. The gray area represents the remaining area of the support plane excluding padding. Figure 6(b) shows an example of a domain generated at linear intervals when m is 1 and n is 4.

4.1.3. Constraints. The constraint types are categorized into unary constraints (e.g., *reachableBasePose* (B, o)), binary constraints (e.g., *placeableBasePose* (B, O)), and nonbinary constraints (e.g., *preGraspablaHandPose* (H, H', B, h, o))

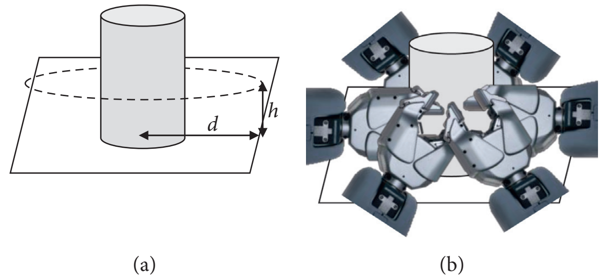


Figure 5: Domain generation of H_{grasp}. (a) The range of domain and (b) the discrete domain (when $m = 1$, $k = 1$, and $n = 6$).

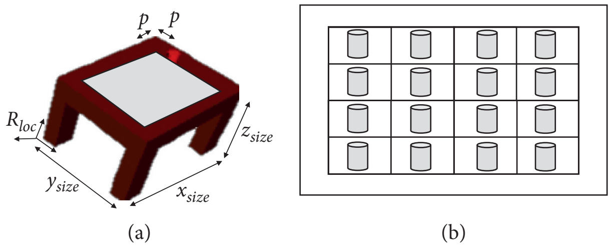


Figure 6: Domain generation of O_{place}. (a) The range of domain and (b) the discrete domain (when $m = 4$ and $n = 4$).

according to arity in the constraints. These constraints are all extracted from the preconditions of each action in the task plan skeleton. Table 6 is a mapping table for this. For convenience, among the parameters in the constraints, constants are excluded from the variables shown.

There is an issue of how to deal with the pose of movable object to be affected by the previous action. There are some ideas for this. In Lozano-Pérez and Garrett, some constraints have a variable of a movable object that was affected by previous action. For example, when they assume that there is a plan skeleton for two pick and place tasks, first *disjoint* (. . ., {$O1$}) constraint is modelled from the first place action and then *disjoint* (. . ., {$O2$, $O1$}) constraint is modelled from the second place action although the second place action does not have a parameter of variable $O1$. However, this is not scalable. In Lagriffoul et al., they avoided this issue by fixing the goal pose of each objects.

In this paper, we model the constraint that only has variables that correspond to self-action parameters. For instance, *placeableHandPose* constraint has only one variable for object pose regardless of the number of place actions. This constraint considers the poses of other movable objects as constraints that are loaded from the environment. This is very scalable to model the CSP from the plan skeleton. But it may assign the wrong pose (that collides with other movable objects) to a variable. To avoid this, we propose a variable ordering in section 4.3.

Meanwhile, we classify the constraints into the pose constraints and path constraints according to their semantics. The path constraints are *validBasePath* and *validHandPath*, and the remainder is the pose constraint. All constraint descriptions are summarized in a table in the appendix. The classification of constraints according to semantic is used in heuristic design.

4.2. Preprocessing. Preprocessing is performed to reduce the search space by reducing the size of domain sets. In the preprocessing step, node consistency is checked using the unary constraints. Through the node consistency, the values that satisfy the unary constraints are left and others are removed. In Table 6, the unary constraints are *reachableBasePose* (B) and *placeableObjectPose* (O).

Figure 7 shows an example of checking node consistency using *reachableBasePose* (*Breach*). As shown in Figure 7(a), when there are 4 domain values (①–④) for variable Breach, when the node consistency is checked for *reachableBasePose* (*Breach*), domain values ③ and ④ are removed as shown Figure 7(b). Domain value ③ is removed because the target object does not exist in the robot's configuration space accessible by the robot's arm, and domain value④ is removed because it collides with the table.

4.3. Ordering Heuristics. In the CSP, ordering heuristics (the search order of variables, the call order of constraints, etc.) are greatly influenced by the overall search time. We propose two ordering heuristics, variable ordering and constraint ordering. Variable ordering considers the inter-action dependency and the intra-action dependency.

4.3.1. Variable Ordering. First, our variable ordering considers the inter-action dependency. We mentioned the movable object issue of our modelling method in section 4.1.3. This variable ordering solves this issue. This takes into account the order between actions. All variables belonging to the earliest ordered actions are visited first. Then, we can get the priority of variable visits like Table 7. One thing to note is that after bounding values to all the variables of an action, the effect of this action is projected into the environment. As the effect of an action changes the environment, the variables of the next action must be visited. Then, when moving several objects from one table to another, the position of the next object can be determined considering the placement of the previously moved object.

Even if the ordering is conducted considering inter-action dependency, because the manipulation actions such as the *PickUpGeneral* and *PutDownGeneral* include multiple CSP variables, ordering between the variables in these actions is not completely performed. Ordering considering the intra-action dependency between variables orders multiple variables within one action. The intra-action dependency between the variables is as follows. For the *PickUpGeneral* action, H_{grasp} affects H_{pre}, and H_{post}, H_{pre}, and H_{post} do not affect each other. For *PutDownGeneral*, the placement position O_{place} affects H_{place}, H_{low}, and H_{ret}, H_{place} affects H_{low} and H_{ret}, H_{low} and H_{ret} do not affect each other. Figure 8 denotes this as a constraint network. Based on the constraints modelled by these dependencies, this paper determines the search order of the variables using MRV, a general-purpose CSP heuristic.

Table 8 shows the search order of the variables finally determined by this ordering. The variable ordering proposed in this paper is a mixture of generic variable ordering and special-purpose variable ordering.

Table 6: Mapping action preconditions into constraints.

Action	Precondition	CSP constraint
MoveBase *ForPickUp*	robotAt (B_{init}), objectAt (object03, O_{init}), reachableBasePose (pr2, object03, O_{init}, B_{reach}), validBasePath (pr2, B_{init}, B_{reach})	reachablaBasePose (B_{reach}) validBasePath (B_{init}, B_{reach})
PickUp General	empty (hand), robotAt (B_{reach}), objectAt (obj, O_{init}), handAt (rgripper, H_{init}), graspableHandPose (rgripper, object03, O_{init}, B_{reach}, H_{grasp}), preGraspableHandPose (rgripper, object03, O_{init}, H_{grasp}, H_{pre}), postGraspableHandPose (rgripper, object03, O_{init}, H_{grasp}, H_{post}), validHandPath (rgripper, B_{reach}, H_{init}, H_{pre}), validHandPath (rgripper, B_{reach}, H_{pre}, H_{grasp}), validHandPath (rgripper, B_{reach}, H_{grasp}, H_{post}), validHandPath (rgripper, B_{reach}, H_{post}, H_{init})	graspableHandPose (H_{grasp}, B_{reach}) preGraspableHandPose (H_{pre}, H_{grasp}, B_{reach}) postGraspableHandPose (H_{post}, H_{grasp}, B_{reach}) validHandPath (H_{init}, H_{pre}, B_{reach}) validHandPath (H_{pre}, H_{grasp}, B_{reach}) validHandPath (H_{grasp}, H_{post}, B_{reach}) validHandPath (H_{post}, H_{init}, B_{reach})
MoveBase *ForPutDown*	robotAt (B_{reach}), graspedBy (object03, rgripper), placeableBasePose (pr2, object03, table02, B_{place}), validBasePath (pr2, B_{reach}, B_{place})	placeableBasePose (B_{place}) validBasePath (B_{reach}, B_{place})
PutDown General	graspedBy (object03, rgripper), robotAt (B_{place}), handAt (rgripper, H_{init}), placeableObjectLocation (object03, table02, O_{place}), placeableHandPose (rgripper, object03, B_{place}, O_{place}, H_{place}), lowerableHandPose (rgripper, object03, B_{place}, O_{place}, H_{place}, H_{low}), retractableHandPose (rgripper, object03, B_{place}, O_{place}, H_{place}, H_{ret}), validHandPath (rgripper, B_{place}, H_{init}, H_{low}), validHandPath (rgripper, B_{place}, H_{low}, H_{place}), validHandPath (rgripper, B_{place}, H_{place}, H_{ret}), validHandPath (rgripper, B_{place}, H_{ret}, H_{init})	placeableObjectPose (O_{place}) placeableHandPose (B_{place}, O_{place}, H_{place}) lowerableHandPose (B_{place}, O_{place}, H_{place}, H_{low}) retractableHandPose (B_{place}, O_{place}, H_{place}, H_{ret}) validHandPath (B_{place}, H_{init}, H_{low}) validHandPath (B_{place}, H_{low}, H_{place}) validHandPath (B_{place}, H_{place}, H_{ret}) validHandPath (B_{place}, H_{ret}, H_{init})

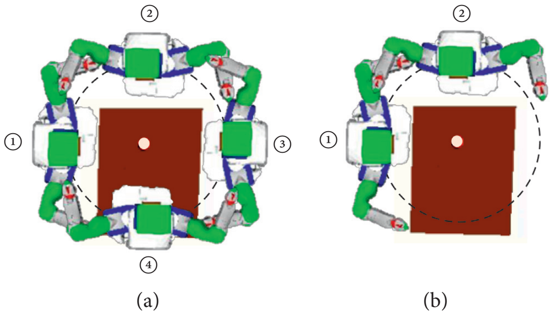


Figure 7: (a) All the domains and (b) domains pruned by *reachableBasePose* constraint.

Table 7: Variable ordering based on inter-action dependency.

Priority	Variable
1	B_{reach}
2	H_{grasp}, H_{pre}, H_{post}
3	B_{place}
4	O_{place}., H_{place}., H_{low}., H_{ret}

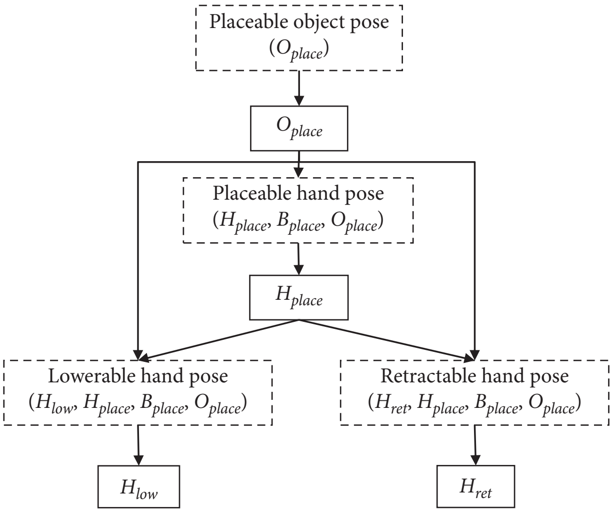

Placeable object pose
(O_{place})
O_{place}
Placeable hand pose
(H_{place}, B_{place}, O_{place})
H_{place}
Lowerable hand pose
(H_{low}, H_{place}, B_{place}, O_{place})
Retractable hand pose
(H_{ret}, H_{place}, B_{place}, O_{place})
H_{low}
H_{ret}

Figure 8: Constraint network for variable ordering in *PutDown-General* action.

4.3.2. Constraint Ordering. As mentioned in Section 4.1.3, in addition to the general classification method of classifying constraints according to the number of parameters, this paper classifies the constraints into pose constraints and path constraints according to the semantics of the constraints. The path constraints are validBasePath and validHandPath, and the remainders are the pose constraints.

Table 8: Variable ordering based on intra-action dependency.

Priority	Variable
1	B_{reach}
2	H_{grasp}
3	$H_{\text{pre,}}$ H_{post}
4	O_{place}
5	B_{place}
6	H_{place}
7	$H_{\text{low,}}$ H_{ret}

This constraint classification method can be used to determine the calling order of the constraints. The calling order of the constraints is determined as first calling the pose constraint and then the path constraint. Typically, to generate a single motion plan, a valid goal pose that satisfies the constraints is first determined, and then a trajectory to the goal pose is calculated. In practice, the cost of calculating the trajectory is large enough to comprise most of the cost for checking motion feasibility. Therefore, if the trajectory is calculated for an invalid goal pose, the search time is greatly wasted. The constraint calling order has the effect of pre-pruning invalid pose values, thus preventing unnecessary path constraint calls.

4.4. Constraint Propagation. In the search phase, a depth-first backtracking algorithm [44] involving constraint propagation is used to search the solution. Constraint propagation is a technique that tentatively removes the domains of other variables that violate the constraints based on the values bound to the currently visited variables during the backtracking process. Therefore, backtracking with constraint propagation can result in better performance than normal backtracking as it reduces the size of the domains to be visited. GAC (Generalized Arc Consistency) [45] is generally used to propagate nonbinary constraints fully. GAC is an extension of arc consistency [45] used to propagate constraints with more than two variables. However, full propagation of nonbinary constraints not only requires excessive costs for motion planning but also is difficult to see the reduction effect in the domains of the transitive variables. Therefore, we determined that it is more reasonable to use forward checking, which propagates the constraints only to variables neighboring the currently visited variables. In particular, when conducting forward checking, only the pose constraint check is performed to the exclusion of the path constraint check. This is because, first, it is reasonable to check the domains after first reducing them as much as possible because path constraint checking requires considerable computation time. In practice, the cost of path constraint checking accounts for most of the cost of motion feasibility checking. Second, the path constraints are dependent on the pose constraints. Thus, a valid goal pose must be determined first to enable the search for a collision-free path to the goal pose.

The proposed framework solves the constraint satisfaction problem by using the search method shown in Algorithm 1. Algorithm 1 describes a depth-first backtracking search that involves forward checking. In Algorithm 1, one of the domain's values is bound to the visited variable, and then forward checking is performed on the neighboring variables of the current variables among those still unbound. As mentioned above, only the pose constraint check is performed in forward checking. The path constraint check is applied when the values are bound to the visited variables, in which constraint checking is performed with the remaining variables.

Algorithm 2 describes the constraint propagation function. This algorithm propagates pose constraints to variables that are still unbound and neighboring the currently visited variable. When propagating a pose constraint, the size of the domain of the neighboring variables may be partially reduced by the pose constraint. If all domain values violate the constraint, then this is treated as failure. The part of the domain in which values are removed is in the revise function. The revise function conducts a pose constraint check and removes all domain values that violate the pose constraints.

5. Implementation and Evaluation

5.1. Implementation. Our framework was implemented in Ubuntu 16.04, Python 2.7. The FD (Fast Downward) library [46] was used for task planning, and the TrajOpt library [47] was used for motion planning. The robot task environment was implemented using the OpenRAVE [48] simulator. The modelling and solution methods of the CSP were implemented by extending the python-constraints open-source library [49]. The computing environment was an Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz, 16 GB memory.

5.2. Evaluation. We performed experiments to evaluate the generality, scalability, efficiency, and optimality of the proposed CSP framework for TAMP. The length of the task plan skeleton was set as shown in Table 9. P1 is a task plan skeleton that takes pick and place once, and P2 is a task plan skeleton that takes pick and place twice. The size of the domain set was set as shown in Table 10. Table 10 shows the size of the domain set generated for each variable. The domain set D_2 is twice as large as the domain set D_1. The sizes of the obstacles were set as shown in Table 11. The obstacles were placed at table T_{pick} where the target object is located, as well as at table T_{place} for placing the target object, at the same sizes. For the obstacle sizes, none were O_1, 3 were O_2, and 6 were O_3.

First, an evaluation was performed to confirm comprehensiveness of constraints which the proposed CSP framework dealt with. For this evaluation, the constraints of our framework were compared with those of previous studies. Table 12 shows the evaluation results. The constraints were largely divided into pose and path constraints, and more specifically, constraints on mobile base, hand, and object. The experimental results show that out of 16

```
function BWP (assignment = {}, csp= (V, D, C)) returns a solution or failure
if assignment is complete then return assignment end if
var = ordered_unassigned_variable (V)
for each val in D (var)
    if val is consistent with assignment then
        add {var = val} to assignment
        result = forward_checking (csp, var, assignment)
        if result is failure then return failure end if
        add neighbors to assignment
        result = BWP (assignment, csp)
        if result is not failure then return result end if
        remove {var = value} from assignment
    end if
end for
return failure
```

Algorithm 1: Backtracking with propagation.

```
function LFC (csp = (V, D, C), var, assignment) returns false if inconsistency is found and true otherwise, and update domain of
    neighbors
neighbors = all_neighbors (csp, var)
while neighbors is not empty do
    $N_i$ = neighbors.pop
    if revise ($N_i$, csp, assignment)
        if D ($N_i$) is empty then
            return false
        end if
    end if
end while
return true
function revise ($N_i$, csp, assignment) return true if the domain of $N_i$ is changed is Revised = false
    for each val in D ($N_i$)
        if val is inconsistent in pose constraints with assignment then remove val from D ($N_i$) is Revised = true
    end if
end for
```

Algorithm 2: Limited forward checking.

Table 9: Experiment setting: task plan skeleton.

Task plan skeleton	Action sequence
P_1	1 MoveBaseForPickUp (...) 2 PickUpGeneral (...) 3 MoveBaseForPutDown (...) 4 PutDownGeneral (...)
P_2	1 MoveBaseForPickUp (...) 2 PickUpGeneral (...) 3 MoveBaseForPutDown (...) 4 PutDownGeneral (...) 5 MoveBaseForPickUp (...) 6 PickUpGeneral (...) 7 MoveBaseForPutDown (...) 8 PutDownGeneral (...)

Table 10: Experiment setting: domain.

Variable	Domain	
	D_1	D_2
B_{reach}	16	32
H_{grasp}	8	16
H_{pre}	24	48
H_{post}	24	48
O_{place}	10	20
B_{place}	16	32
H_{place}	8	16
H_{low}	24	48
H_{ret}	24	48

Table 11: Experiment setting: obstacle.

Table	Obstacle		
	O_1	O_2	O_3
T_{pick}	0	3	6
T_{place}	0	3	6

constraints, 13 checks are possible in ours, 8 in Garrett, 7 in Lagriffoul, and 4 in Lozano-Pérez. Our framework showed more comprehensive coverage of constraints than existing works because it includes additional constraints on both

Table 12: Constraint list.

Constraint			Ours	Lagriffoul	Lozano-Pérez	Garrett
Pose	Base	Picking up	Yes	No	No	Yes
		Putting down	Yes	No	No	Yes
	Hand	Grasp	Yes	No	No	No
		Pre-grasp	Yes	No	No	No
		Post-grasp	Yes	No	No	No
		Re-grasp	No	Yes	No	No
		Placing	Yes	Yes	Yes	Yes
		Lowering	Yes	No	No	No
		Retracting	Yes	No	No	No
		Hand over	No	Yes	No	No
	Object	Placing	Yes	Yes	Yes	Yes
		Stacking	No	Yes	No	No
Path	Base	Collision-free path	Yes	No	No	Yes
		Collision-free path with object	Yes	No	No	Yes
	Hand	Collision-free path	Yes	Yes	Yes	Yes
		Collision-free path with object	Yes	Yes	Yes	Yes

base and hand poses for mobile manipulation tasks. On the contrary, our framework includes neither hand over constraint nor stacking constraint since it mainly focuses on pick and place tasks. However, because we focus on pick and place, it does not cover constraints such as stack and hand over.

Second, experiments were conducted to investigate the effect of the proposed preprocessing. Figure 9 shows the measurements of the search time with and without preprocessing. The horizontal axis is a problem instance with different values of the experimental parameters, and the vertical axis is the search time. The experimental results demonstrate that the search with the proposed preprocessing is faster than that without preprocessing. The set of constraints listed in Table 9 includes many unary constraints. Therefore, the proposed preprocessing was much effective to reduce the search space.

Third, experiments were conducted to analyze the performance of the proposed variable ordering method. In these experiments, our variable ordering method was compared with generic ordering methods such as MRV, DH (Degree Heuristic), and random ordering. Figure 10 shows the experimental results. The proposed variable ordering method shows the fastest search time, followed by MRV, DH, and random ordering. These results demonstrate that the proposed region-dependent ordering method considering the order between the actions has a positive effect on the search time.

Fourth, experiments were conducted to analyze effect of the proposed constraint ordering. Figure 11 shows the performance with and without the proposed constraint ordering. The experimental results demonstrate that the search time with constraint ordering is faster than the search time without constraint ordering. These results indicate that the proposed constraint ordering to pre-prune unnecessary path constraint checks positively impacts the search time.

Fifth, experiments were conducted to analyze effect of the proposed constraint propagation. Figure 12 shows the experimental results to compare search time of the proposed constraint propagation with those of generic constraint propagation methods such as forward checking (FC) and arc consistency (AC). The experimental results demonstrate that the proposed constraint propagation method has the shortest search time, followed by forward checking, arc consistency, and the method without constraint propagation. These results indicate that forward checking limited to the pose constraints as proposed in this paper is the most effective. Meanwhile, the results of Figure 12 demonstrate that forward checking (FC) is more effective than arc consistency (AC) for motion feasibility checking.

Sixth, experiments were conducted to compare the performance of our TAMP framework with existing constraint-based methods, namely, Lozano-Pérez [18] and Lagriffoul [17]. In Lozano-Pérez, both the MRV variable ordering heuristics and arc consistency checking with pose constraints are applied. In case of Lagriffoul, both an action-dependent variable ordering heuristics and forward checking with pose constraints are applied. Because their CSP modellings are much different from ours, we reimplement their heuristics and search strategies within our framework. The experimental results in Figure 13 demonstrate that the proposed framework used the shortest search time. Lozano-Pérez's method showed the lowest performance because its general-purpose heuristic, MRV, and arc consistency are not effective for multi-hop propagation. Lagriffoul's method is faster than that of Lozano-Pérez because of its unique variable ordering heuristics and forward checking. However, Lagriffoul's method only considers the inter-action dependency for variable ordering, while our framework considers both inter- and intra-action dependencies.

6. Discussion and Conclusion

This paper proposed an efficient constraint satisfaction framework for checking motion feasibility of task plan skeletons as a TAMP solution. The proposed framework not only presents mapping rules that can automatically generate a constraint satisfaction problem (CSP) from the task plan skeleton, but it also provides useful methods to solve the CSP such as pre-processing, unique variable ordering heuristics based on intra- and inter-action dependencies in a task plan skeleton, and constraint propagation to improve efficiency of motion feasibility checking. Through numerous experiments using the humanoid robot PR2, we confirmed high performance and efficiency of the proposed framework.

However, the current framework has some technical limitations. The proposed framework was applied to relatively simple task environments based on the assumption that the environments are fully observable, static, deterministic, and have a single robot. However, the real-world

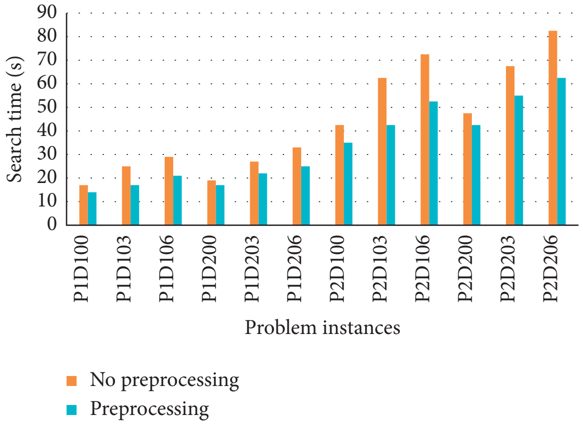


Figure 9: Experimental results of preprocessing.

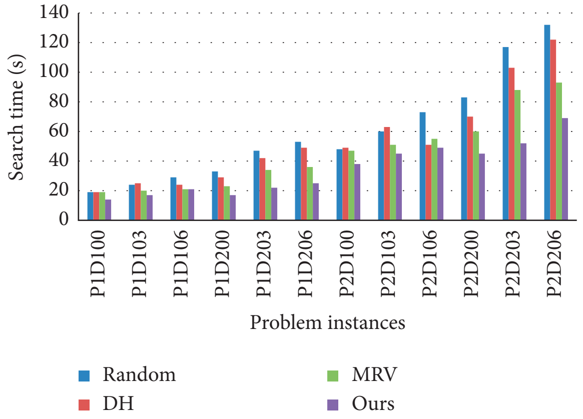


Figure 10: Experimental results of variable ordering.

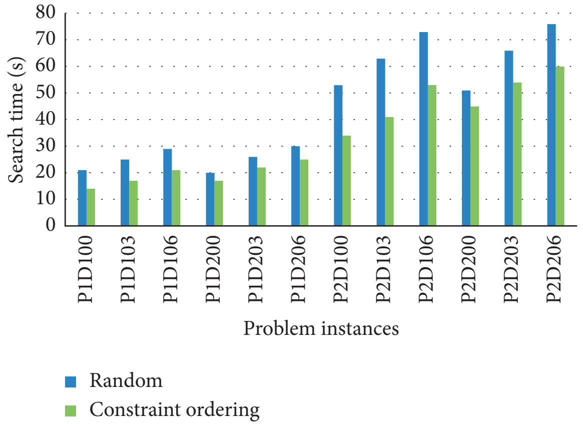


Figure 11: Experimental results of constraint ordering.

physical environments do not meet the assumption. They are partially observable, dynamic, stochastic, and have multiple robots or manipulators. Therefore, we plan to extend our framework to deal with dynamics of real-world environments by designing advanced action model and replanning capability. Furthermore, to address uncertainty in state

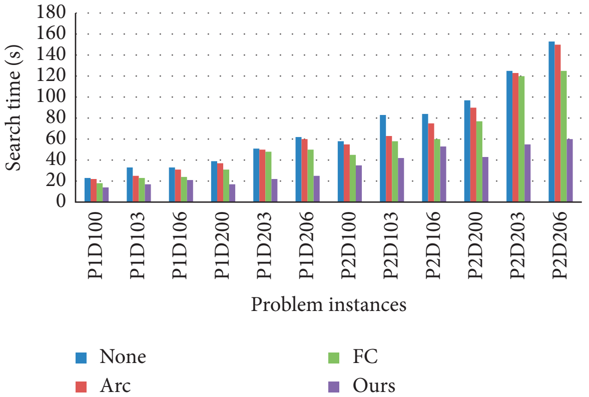


Figure 12: Experimental results of constraint propagation.

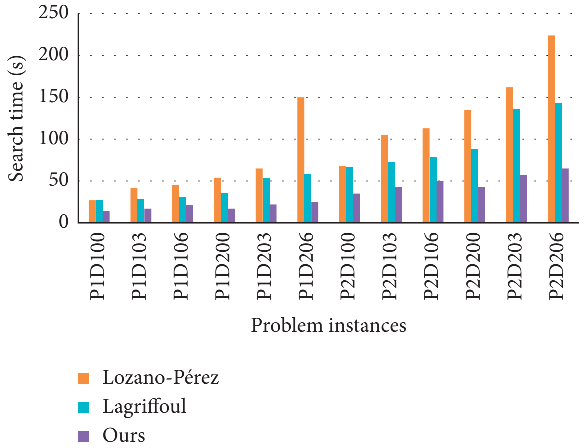


Figure 13: Performance comparison with previous works.

Table 13: Constraint descriptions.

Constraint	Description
reachablaBasePose (B)	When robot is located on base pose B, there must be no collision and the target object must be within radius of robot's arm.
placeableBasePose (B)	When robot is located on base pose B, there must be no collision and the center of table must be within radius of robot's arm.
graspableHandPose (H, B)	When robot's hand is located at hand pose H, robot should be able to grasp the target object by closing fingers. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H.
preGraspableHandPose (H, H', B)	When robot's hand is located at hand pose H, the hand must be a little farther than hand pose H' which is a graspableHandPose for target object. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H
postGraspableHandPose (H, H', B)	When robot's hand is located at hand pose H, the hand must be a little higher than hand pose H' which is a graspableHandPose for target object. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H.
placeableHandPose (B, O, H)	When robot's hand is located at hand pose H, robot should be able to place the target object on the target location O by opening fingers. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H.
lowerableHandPose (B, O, H', H)	When robot's hand is located at hand pose H, the hand must be a little higher than hand pose H' is a placeableHandPose for target location O. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H.
retractableHandPose (B, O, H', H)	When robot's hand is located at hand pose H, the hand must be a little farther than hand pose H' is a placeableHandPose for target location O. When robot is located on base pose B, there is at least one inverse kinematics for the hand pose H.
placeableObjectPose (O)	When target object is located on target location O, there must be no collision.
validBasePath (B, B')	There must be at least one collision-free path from base pose B to base pose B'.
validHandPath (H, H', B)	There must be at least one collision-free path from hand pose H to hand pose H' at base pose B.

recognition, the current framework will be extended to solve TAMP over belief state space. It will be also interesting to apply the proposed framework to multi-agent or multi-robot environments.

Appendix

Appendices A–D describe the functions that appear in Section 4.1.2. Table 13 describes the definition of constraints Section 4.1.3.

A. Matrix from Pose

Given a pose p, this function converts p to rotation matrix R. R is obtained as follows:

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 - q_0 q_3\right) & 2\left(q_1 q_2 + q_0 q_2\right) & x \\ 2\left(q_1 q_2 + q_0 q_3\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2 q_3 - q_0 q_1\right) & y \\ 2\left(q_1 q_3 - q_0 q_2\right) & 2\left(q_2 q_3 + q_0 q_1\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (A.1)$$

where

$$p = \left(q_0, q_1, q_2, q_3, x, y, z\right). \qquad (A.2)$$

B. Matrix from Quaternion

Given a quaternion q, this function converts q to rotation matrix R. R is obtained as follows:

$$R = \begin{bmatrix} 1 - Q2 - Q3 & \dfrac{2\left(q_1 q_2 - q_0 q_3\right)}{L} & \dfrac{2\left(q_1 q_3 - q_0 q_2\right)}{L} \\ \dfrac{2\left(q_1 q_2 - q_0 q_3\right)}{L} & 1 - Q1 - Q3 & \dfrac{2\left(q_2 q_3 - q_0 q_1\right)}{L} \\ \dfrac{2\left(q_1 q_3 - q_0 q_2\right)}{L} & \dfrac{2\left(q_2 q_3 - q_0 q_1\right)}{L} & 1 - Q1 - Q2 \end{bmatrix}, \qquad (A.3)$$

where

$$\begin{aligned} q &= \left(q_0, q_1, q_2, q_3\right), \\ L &= a_0^2 + a_1^2 + a_2^2 + a_3^2, \\ Q1 &= \frac{2a_1^2}{L}, \\ Q2 &= \frac{2a_2^2}{L}, \\ Q3 &= \frac{2a_3^2}{L}. \end{aligned} \qquad (A.4)$$

C. Quaternion from Axis Angle

Given an axis angle, this function converts the axis angle to quaternion q. q is obtained as follows:

$$q = \begin{cases} (1, 0, 0, 0), & \text{if } L = 0, \\ \left(\dfrac{L}{2}, sa_x, sa_y, sa_z\right), & \text{otherwise}, \end{cases} \qquad (A.5)$$

where

$$\begin{aligned} a &= \left(a_x, a_y, a_z\right), \\ L &= a_x^2 + a_y^2 + a_z^2, \\ s &= \frac{\sin\left(L/2\right)}{L}. \end{aligned} \qquad (A.6)$$

D. Matrix from Axis Angle

Given an axis angle, this function converts the axis angle to rotation matrix R. This function obtains a quaternion from the function of Appendix C and obtains R from the function of Appendix B.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Technology Innovation Program or Industrial Strategic Technology Development Program (Grant No. 10077538, Development of manipulation technologies in social contexts for human-care service robots) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

References

[1] K. Hauser and J. C. Latombe, "Integrating task and PRM motion planning: dealing with many infeasible motion planning queries," in *Proceedings of ICAPS Workshop on Bridging the Gap between Task and Motion Planning*, Thessaloniki, Greece, July 2009.

[2] E. Plaku and G. D. Hager, "Sampling-based motion and symbolic action planning with geometric and differential constraints," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'10)*, pp. 5002–5008, Anchorage, Alaska, May 2010.

[3] J. Guitton and J. L. Farges, "Taking into account geometric constraints for task-oriented motion planning," in *Proceedings of AAAI Workshop on Bridging the Gap between Task and Motion Planning*, pp. 26–33, Atlanta, GA, USA, July 2010.

[4] S. Alili, A. K. Pandey, E. A. Sisbot, and R. Alami, "Interleaving symbolic and geometric reasoning for a robotic assistant," in *Proceedings of ICAPS Workshop on Combining Action and Motion Planning*, Toronto, Canada, March 2010.

[5] L. De Silva, A. K. Pandey, M. Gharbi, and R. Alami, "Towards combining HTN planning and geometric task planning," in *Proceedings of RSS Workshop on Combined Robot Motion Planning and AI Planning for Practical Applications*, Berlin, Germany, June 2013.

[6] L. De Silva, A. K. Pandey, and R. Alami, "An interface for interleaved symbolic-geometric planning and backtracking," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13)*, pp. 232–239, Tokyo, Japan, November 2013.

[7] C. Dornhege, P. Eyerich, T. Keller, S. Trüg, M. Brenner, and B. Nebel, "Semantic attachments for domain-independent planning systems," in *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'09)*, pp. 114–121, Thessaloniki, Greece, September 2009.

[8] C. Dornhege, M. Gissler, M. Teschner, and B. Nebel, "Integrating symbolic and geometric planning for mobile manipulation," in *Proceedings of IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR)*, pp. 1–6, Denver, CO, USA, May 2009.

[9] C. Dornhege, P. Eyerich, T. Keller, M. Brenner, and B. Nebel, "Integrating task and motion planning using semantic attachments," in *Proceedings of AAAI Workshop on Bridging the Gap between Task and Motion Planning*, pp. 10–17, Atlanta, GA, USA, July 2010.

[10] A. Hertle, C. Dornhege, T. Keller, and B. Nebel, "Planning with semantic attachments: an object-oriented view," in *Proceedings of 20th European Conference on Artificial Intelligence (ECAI'12)*, pp. 402–407, Montpellier, France, August 2012.

[11] L. P. Kaelbling LP and T. Lozano-Pérez, "Hierarchical planning in the now," in *Proceedings of AAAI Workshop on Bridging the Gap between Task and Motion Planning*, pp. 33–42, Atlanta, GA, USA, July 2010.

[12] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'11)*, pp. 1470–1477, Shanghai, China, May 2011.

[13] A. Gaschler, R. P. Petrick, M. Giuliani, M. Rickert, and A. Knoll, "KVP: a knowledge of volumes approach to robot task planning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13)*, pp. 202–208, Tokyo, Japan, November 2013.

[14] J. Bidot, L. Karlsson, F. Lagriffoul, and A. Saffiotti, "Geometric backtracking for combined task and motion planning in robotic systems," *Artificial Intelligence*, vol. 247, pp. 229–265, 2017.

[15] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue, "Environment manipulation planner for humanoid robots using task graph that generates action sequence," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS'04)*, pp. 1174–1179, Sendai, Japan, September 2004.

[16] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 104–126, 2009.

[17] F. Lagriffoul, D. Dimitrov, J. Bidot, A. Saffiotti, and L. Karlsson, "Efficiently combining task and motion planning using geometric constraints," *The International Journal of Robotics Research*, vol. 33, no. 14, pp. 1726–1747, 2014.

[18] T. Lozano-Pérez and L. P. Kaelbling, "A constraint-based method for solving sequential manipulation planning problems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)*, pp. 3684–3691, Chicago, IL, USA, September 2014.

[19] R. Dearden and C. Burbridge, "An approach for efficient planning of robotic manipulation tasks," in *Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'13)*, pp. 55–63, Rome, Italy, June 2013.

[20] R. Dearden and C. Burbridge, "Manipulation planning using learned symbolic state abstractions," *Robotics and Autonomous Systems*, vol. 62, no. 3, pp. 355–365, 2014.

[21] D. Leidner and C. Borst, "Hybrid reasoning for mobile manipulation based on object knowledge," in *Proceedings of IROS Workshop on AI-based robotics*, Tokyo, Japan, November 2013.

[22] E. Erdem, K. Haspalamutgil, C. Palaz, V. Patoglu, and T. Uras, "Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'11)*, pp. 4575–4581, Shanghai, China, May 2011.

[23] G. Havur, K. Haspalamutgil, C. Palaz, E. Erdem, and V. Patoglu, "A case study on the tower of hanoi challenge: representation, reasoning and execution," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'13)*, pp. 4552–4559, Karlsruhe, Germany, March 2013.

[24] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'14)*, pp. 639–646, Hong Kong, China, May 2014.

[25] C. Phiquepal and M. Toussaint, "Combined task and motion planning under partial observability: an optimizationbased approach," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'19)*, pp. 9000–9006, Montreal, Canada, May 2019.

[26] R. Shome and K. E. Bekris, "Anytime multi-arm task and motion planning for pick-and-place of individual objects via handoffs," in *Proceedings of IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS'19)*, pp. 37–43, New Brunswick, NB, USA, August 2019.

[27] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 844–851, 2020.

[28] V. N. Hartmann, O. S. Oguz, D. Driess, M. Toussaint, and A. Menges, "Robust task and motion planning for long-horizon architectural construction planning," 2020, https://arxiv.org/abs/2003.13649.

[29] A. Akbari, Muhayyuddin, and J. Rosell, "Knowledge-oriented task and motion planning for multiple mobile robots," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 1, pp. 137–162, 2019.

[30] A. Akbari, F. Lagriffoul, and J. Rosell, "Combined heuristic task and motion planning for bi-manual robots," *Autonomous Robots*, vol. 43, no. 6, pp. 1575–1590, 2019.

[31] S. Thakar, A. Kabir, P. M. Bhatt et al., "Task assignment and motion planning for bi-manual mobile manipulation," in *Proceedings of IEEE International Conference on Automation Science and Engineering (CASE'19)*, pp. 910–915, Vancouver, Canada, June 2019.

[32] T. Ren, G. Chalvatzaki, and J. Peters, "Extended task and motion planning of long-horizon robot manipulation," 2021, https://arxiv.org/abs/2102.09066.

[33] C. R. Garrett, C. Paxton, T. Lozano-Pérez, L. P. Kaelbling, and D. Fox, "Online replanning in belief space for partially observable task and motion problems," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'20)*, pp. 5678–5684, Xi'an, China, May 2020.

[34] A. M. Kabir, S. Thakar, P. M. Bhatt et al., "Incorporating motion planning feasibility considerations during task-agent assignment to perform complex tasks using mobile manipulators," in *Proceeding of IEEE International Conference on Robotics and Automation (ICRA '20)*, pp. 5663–5670, Paris, France, September 2020.

[35] J. Motes, R. Sandström, H. Lee, S. Thomas, and N. M. Amato, "Multi-robot task and motion planning with subtask dependencies," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3338–3345, 2020.

[36] Y. Ding, X. Zhang, X. Zhan, and S. Zhang, "Task-motion planning for safe and efficient urban driving," 2020, https://arxiv.org/pdf/2003.03807.

[37] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Sample-based methods for factored task and motion planning," in *Proceedings of Robotics: Science and Systems*, Cambridge, MA, USA, April 2017.

[38] R. E. Fikes and N. J. Nilsson, "STRIPS: a new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 2, no. 3-4, pp. 189–208, 1971.

[39] M. Ghallab, A. Howe, C. Knoblock et al., "PDDL—the planning domain definition language," in *Proceedings of International Conference on Artificial Intelligence Planning Systems (AIPS'98)*, Pittsburgh, PA, USA, December 1998.

[40] K. Erol, J. Hendler, and D. S. Nau, "HTN planning: complexity and expressivity," in *Proceedings of 12th National Conference on Artificial Intelligence (AAAI'94)*, pp. 1123–1128, Seattle, WA, USA, August 1994.

[41] J. C. Latombe, *Robot Motion Planning*, Springer Science and Business Media, Boston, MA, USA, 2012.

[42] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," Technical Report TR 98-11, Iowa State University, Ames, IA, USA, 1998.

[43] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[44] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems: algorithms and applications," *European Journal of Operational Research*, vol. 119, no. 3, pp. 557–581, 1999.

[45] V. Kumar, "Algorithms for constraint-satisfaction problems: a survey," *Artificial Intelligence Magazine*, vol. 13, no. 1, pp. 32–44, 1992.

[46] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[47] J. Schulman, J. Ho, and A. Lee, "Finding locally optimal, collision-free trajectories with sequential convex optimization," *Robotics: Science and Systems*, vol. 9, no. 1, pp. 1–10, 2013.

[48] R. Diankov and J. J. Kuffner, "OpenRAVE: a Planning architecture for autonomous robotics," Tech. Rep. CMU-RI-TR-08-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2008.

[49] G. Niemeyer, "Python-constraint: solving constraint satisfaction problems in Python," 2017, https://pypi.org/project/python-constraint/.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5569652, 18 pages
https://doi.org/10.1155/2021/5569652

Research Article

Bayesian Analysis of Inverted Kumaraswamy Mixture Model with Application to Burning Velocity of Chemicals

Farzana Noor,[1] **Saadia Masood**,[2] **Mehwish Zaman**,[1] **Maryam Siddiqa**,[1] **Raja Asif Wagan**,[3] **Imran Ullah Khan**,[4] **and Ahthasham Sajid**[5]

[1]*Department of Mathematics & Statistics, International Islamic University, Islamabad 45320, Pakistan*
[2]*Department of Mathematics and Statistics, PMAS University of Arid Agriculture, Rawalpindi, Pakistan*
[3]*Department of Information Technology, Faculty of ICT,*
Baluchistan University of Information Technology Engineering and Management Sciences, Quetta, Pakistan
[4]*College of Underwater Acoustics Engineering, Harbin Engineering University, Harbin, Heilongjiang, China*
[5]*Department of Computer Science,*
Faculty of ICT Baluchistan University of Information Technology Engineering and Management Sciences, Quetta, Pakistan

Correspondence should be addressed to Imran Ullah Khan; khan@hrbeu.edu.cn

Received 29 January 2021; Revised 2 April 2021; Accepted 27 April 2021; Published 19 May 2021

Academic Editor: Dao B. Wang

Burning velocity of different chemicals is estimated using a model from mixed population considering inverted Kumaraswamy (IKum) distribution for component parts. Two estimation techniques maximum likelihood estimation (MLE) and Bayesian analysis are applied for estimation purposes. BEs of a mixture model are obtained using gamma, inverse beta prior, and uniform prior distribution with two loss functions. Hyperparameters are determined through the empirical Bayesian method. An extensive simulation study is also a part of the study which is used to foresee the characteristics of the presented model. Application of the IKum mixture model is presented through a real dataset. We observed from the results that Linex loss performed better than squared error loss as it resulted in lower risks. And similarly gamma prior is preferred over other priors.

1. Introduction

Mixture models appear as obvious candidates whenever datasets that consist of two or more heterogeneous populations are mixed together. Due to its modeling versatility, the finite mixture model has attracted a great deal of attention in the history of statistics. To analyze the heterogeneous nature of processes, the mixture models are comparatively more suitable than the simple models. A mixture model with finite components is suitable to use when data are overdispersed, to fit a zero-expansion model, to measure heavy-tailed density, and to test for heterogeneity in cluster analysis. Mixture models have been effectively used in many areas such as industrial engineering (Ali et al. [1]), biology (Bhattacharya [2]), social sciences (Harris [3]), economics (Jedidi et al. [4]), and reliability (Sultan et al. [5]). For more detail about the finite mixture models, see Everitt [6], Ali [7], Feroze and Aslam [8], Zhang and Huang [9], Fundi et al. [10], Tripathi et al. [11], Noor et al. [12], and Feroze and Aslam [13].

Many researchers have provided valuable literature on inverted distribution, for example, Aljuaid [14] studied inverse Weibull, Noor and Aslam [15] analyzed inverse Weibull mixture distribution, Abd EL-Kader et al. [16] analyzed inverted Pareto type I distribution, Basheer [17] proposed generalized alpha power inverse Weibull distribution, and Hassan and Zaky [18] present study on estimation of entropy for inverse Weibull distribution under multiple censored data. Kumaraswamy [19] proposed a distribution which has widespread applications, particularly in situations that are bounded from below and above, such as individual's height, test scores, atmospheric temperature, and hydrological data. AL-Fattah et al. [20] obtained the IKum distribution from the Kumaraswamy distribution using the transformation $X = (1/T) - 1$ when random variable T has Kumaraswamy distribution with α and β as shape parameters. They discussed important properties of inverted Kumaraswamy distribution and obtained parameters of the proposed model by using MLE and Bayesian technique. The IKum distribution has a long tail to the right; as a result, it can effectively be used for long-term reliability predictions and producing optimistic predictions as compared to other distributions.

Censoring is an important factor of experiments measuring life/failure times. Censored samples are encountered in life test whenever the experimenter has some obligations on the cost or available time for the experiment. Different censoring schemes are used for different experiments, but type I censoring is the most commonly used censoring scheme.

Our aim is to analyze inverted Kumaraswamy distribution in a different way as no other work after AL-Fattah et al. [20] is found on the IKum distribution. We propose a mixture model whose component densities are formed by IKum density and estimate the parameters and reliability function of the mixture model under study using Bayesian as well as frequentist method.

2. Methodology

2.1. Two-Component Mixture Model of IKum Distribution. A random variable X supposed to have a k component mixture model is defined as follows:

$$f(x) = \sum_{i=1}^{k} \delta_i f_i(X|\Theta), \tag{1}$$

where $\Theta = (\delta_i, \alpha_i, \beta_i)$, $i = 1, 2, \ldots, k$

The probability density function (pdf) and reliability function of the mixture model whose component densities are characterized by IKum distribution are given by

$$\begin{aligned} f(x;\Theta) &= \delta_1 f_1(x;\Theta_1) + (1-\delta_1) f_2(x;\Theta_2), \quad 0 < \delta_1 < 1, \\ R(x;\Theta) &= \delta_1 R_1(x;\Theta_1) + (1-\delta_1) R_2(x;\Theta_2), \quad 0 < \delta_1 < 1, \end{aligned} \tag{2}$$

where pdf and reliability function of i^{th} IKum density, respectively, are

$$\begin{cases} f_i(x;\Theta_i) = \alpha_i\beta_i(1+x_i)^{-(\alpha_i+1)}\{1-(1+x_i)^{-\alpha_i}\}^{\beta_i-1}, \\ R_i(x;\Theta_i) = \left[1-\{1-(1+x_i)^{-\alpha_i}\}^{\beta_i}\right], \\ x > 0, \alpha_i, \beta_i > 0, i = 1, 2. \end{cases} \tag{3}$$

2.2. Sampling and Likelihood Function under Type 1 Censoring. Suppose m items are taken from a population which is a mixture of two-component IKum model with prespecified termination time T_0. Let the test be conducted, s items are failed from m items, and $(m–s)$ items are still in working position. As per the work of Mendenhall and Hader [21], in many problems, only the futile (useless) items are easily marked as a family of first population and second population. For example, an engineer may divide failed items of electronic as a first population and second population on the basis of failure cause. From the whole population, s_1 units belong to the first population, s_2 are from the second population, and "$m - s$" items do not give us any information about the population to which they belong to. It is obvious that $s = s_1 + s_2$ are the number of uncensored items. Suppose that x_{ij} denote the failure times of j^{th} item which are belonging to the i^{th} subpopulation and that $x_{ij} \le t_0$, $i = 1$, 2 and $j = 1, \ldots, s_i$.

The likelihood function for the IKum mixture model using the above-discussed sampling scheme given by Mendenhall and Hader [21] is given by

$$\begin{aligned} L(\Theta|x) &\propto \prod_{j=1}^{s_1} \delta_1 f_1(x_{1j};\Theta_1) \prod_{j=1}^{s_2} (1-\delta_1) f_2(x_{2j};\Theta_2) \{\delta_1 R_1(x;\Theta_1) + (1-\delta_1) R_2(x;\Theta_2)\}^{m-s} \\ &= \prod_{j=1}^{s_1} \delta_1\alpha_1\beta_1 (1+x_{1j})^{-(\alpha_1+1)} \{1-(1+x_{1j})^{-\alpha_1}\}^{\beta_1-1} \prod_{j=1}^{s_2} (1-\delta_1)\alpha_2\beta_2 (1+x_{2j})^{-(\alpha_2+1)} \\ &\quad \{1-(1+x_{2j})^{-\alpha_2}\}^{\beta_2-1} \left[1-\delta_1\{1-(1+t_o)^{-\alpha_1}\}^{\beta_1} - (1-\delta_1)\{1-(1+t_o)^{-\alpha_2}\}^{\beta_2}\right]^{m-s}, \end{aligned} \tag{4}$$

where $x = (\mathbf{x}_{11}, x_{12}, \ldots, x_{1s_1}\mathbf{x}_{21}, x_{22}, \ldots, x_{2s_2})$ are uncensored observations for failure time.

2.3. Maximum Likelihood Estimation. Taking log of the likelihood function (4) and differentiating w.r.t. parameters result in five nonlinear equations. A solution of these nonlinear equations gives MLEs for the vector of parameters. We use the SAS package to compute ML estimates of the parameter and their MSEs.

$$\begin{aligned}
\frac{\partial Q(\Theta|\mathbf{x})}{\partial\delta_1} &= \frac{s_1}{\delta_1} - \frac{s_2}{(1-\delta_1)} - \frac{(m-s)(F_2(t)-F_1(t))}{R(t)},\\
\frac{\partial Q(\Theta|\mathbf{x})}{\partial\alpha_1} &= \frac{s_1}{\alpha_1} - \sum_{j=1}^{s_1}\ln\left(1+x_{1j}\right) - (\beta_1-1)w_1(x) - \frac{(m-s)\delta_1\beta_1\phi_1(t)F_1(t)}{R(t)\left(1-(1+t)^{-\alpha_1}\right)},\\
\frac{\partial Q(\Theta|\mathbf{x})}{\partial\alpha_2} &= \frac{s_2}{\alpha_2} - \sum_{j=1}^{s_2}\ln\left(1+x_{2j}\right) - (\beta_2-1)w_2(x) - \frac{(m-s)(1-\delta_1)\beta_2\phi_2(t)F_2(t)}{R(t)\left(1-(1+t)^{-\alpha_2}\right)},\\
\frac{\partial Q(\Theta|\mathbf{x})}{\partial\beta_1} &= \frac{s_1}{\beta_1} - \sum_{j=1}^{s_1}\ln\left(1-\left(1+x_{1j}\right)^{-\alpha_1}\right) - \frac{(m-s)\delta_1F_1(t)\ln\left(1-(1+t)^{-\alpha_1}\right)}{R(t)},\\
\frac{\partial Q(\Theta|\mathbf{x})}{\partial\beta_2} &= \frac{s_2}{\beta_2} - \sum_{j=1}^{s_2}\sum_{j=1}^{s_2}\ln\left(1-\left(1+x_{2j}\right)^{-\alpha_2}\right) - \frac{(m-s)(1-\delta_1)F_2(t)\ln\left(1-(1+t)^{-\alpha_2}\right)}{R(t)},
\end{aligned} \tag{5}$$

where

$$\begin{aligned}
F_i(t) &= \left\{1-(1+t)^{-\alpha_i}\right\}^{\beta_i},\\
w_i(x) &= \sum_{j=1}^{r_i}\frac{\left(1+x_{ij}\right)^{-\alpha_i}\ln\left(1+x_{ij}\right)}{\left(1-\left(1+x_{ij}\right)^{-\alpha_i}\right)},\\
\phi_i &= (1+t)^{-\alpha_i}\ln(1+t),\\
R(t) &= \Big[1-\delta_1\left\{1-(1+t_o)^{-\alpha_1}\right\}^{\beta_1} - \Big(1-\delta_1\left\{1-(1+t_o)^{-\alpha_2}\right\}^{\beta_2}\Big], \quad i=1,\ 2.
\end{aligned} \tag{6}$$

2.4. Bayes Estimation. The Bayesian approach is a powerful statistical tool used to reduce uncertainty in complex problems. Bayesian theory basically relies upon prior distribution and the use of loss functions. Loss function represents the loss incurred when the real parameter is derived from the estimated value. Square error loss function (SELF) used in the study is a symmetric loss function. In many situations, overestimation is more serious than underestimation, or vice versa. Asymmetric loss functions are those loss functions in which negative and positive errors of the same or different dimensions cause different losses. To compensate the situation, an asymmetric loss function is also used.

2.4.1. Posterior Density Assuming Informative (Gamma) Prior. It is assumed that α_i and β_i each have gamma prior distribution with (a_i, b_i) and (c_i, d_i) hyperparameters, respectively, and δ_1 assumes a uniform prior so joint prior density for $\alpha_1, \alpha_2, \beta_1, \beta_2$, and δ_1 is

$$g(\Theta) \propto \alpha_1^{a_1-1}e^{-b_1\alpha_1}\alpha_2^{a_2-1}e^{-b_2\alpha_2}\beta_1^{c_1-1}e^{-d_1\beta_1}\beta_2^{c_2-1}e^{-d_2\beta_2}, \quad a_i, b_i, c_i, d_i > 0, i = 1, 2. \tag{7}$$

Thus, posterior density using the likelihood function and joint prior in proportional form is as follows:

$$p(\Theta|\mathbf{x}) \propto \prod_{i=1}^{s1} \delta_1 \alpha_1 \beta_1 \left(1 + x_{1j}\right)^{-(\alpha_1+1)} \left\{1 - \left(1 + x_{1j}\right)^{-\alpha_1}\right\}^{\beta_1 - 1} \prod_{i=1}^{s2} (1-\delta_1)\alpha_2 \beta_2 \left(1 + x_{2j}\right)^{-(\alpha_2+1)} \left\{1 - \left(1 + x_{2j}\right)^{-\alpha_2}\right\}^{\beta 2 - 1}$$
$$\left[1 - \delta_1\{1 - (1 + t_o)^{-\alpha_1}\}^{\beta_1} - (1 - \delta_1)\{1 - (1 + t_o)^{-\alpha_2}\}^{\beta_2}\right]^{m-s} \alpha_1^{a_1 - 1} e^{-b_1 \alpha_1} \alpha_2^{a_2 - 1} e^{-b_2 \alpha_2} \beta_1^{c_1 - 1} e^{-d_1 \beta_1} \beta_2^{c_2 - 1} e^{-d_2 \beta_2}, \tag{8}$$
$$\alpha_i, \beta_i > 0, a_i, b_i, c_i, d_i > 0, 0 < \delta_1 < 1, i = 1, 2.$$

Integration of the posterior density does not produce estimators in compact and simple form; therefore, we use Lindley's approximation to obtain Bayes estimators, posterior risks, and reliability estimates for the shape parameters of the IKum mixture model.

2.4.2. Lindley's Procedure for Estimation of Parameters. Lindley [22] proposed an approximation known as Lindley's approximation used to conduct posterior analysis when posterior density involves a complex integral. In this approximation, Bayes estimator expands as a function that involves a posterior mode of $\widehat{\Theta}$. Lindley's approximation has been utilized by many authors for the estimation of the parameters for the simple as well as mixture models; see Jaheen [23], Ahmad et al. [24], Sultan et al. [25], etc.

Consider the following integral

$$\int U(\Theta) e^{Q(\Theta)} \mathrm{d}(\Theta), \tag{9}$$

where $\Theta = (\alpha_1, \alpha_2, \beta_1, \beta_2, \delta_1)$ is a vector of parameters, $U(\Theta)$ is an arbitrary function of Θ, and $Q(\Theta)$ is the logarithm of a posterior function for n observation. Lindley [22] suggested the following approximate Bayes estimator under SELF:

$$\widetilde{U}_{\mathrm{BL}}(\Theta) = E[U(\Theta)|x] \approx U(\widehat{\Theta}) + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\{U_{ij}(\widehat{\Theta})\tau_{ij} + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{s=1}^{m}\sum_{l=1}^{m} Q_{ijs}(\widehat{\Theta}) U_l(\widehat{\Theta})\tau_{ij}\tau_{sl}\right\}, \tag{10}$$

where $i, j, s = 1, 2, \ldots, m$

$$U_i(\widehat{\Theta}) = \frac{\partial U(\Theta)}{\partial \theta_i}\Big|_{\Theta = \widehat{\theta}},$$
$$U_{ij}(\widehat{\Theta}) = \frac{\partial^2 U(\Theta)}{\partial \theta_i \partial \theta_j}\Big|_{\Theta = \widehat{\theta}},$$
$$\tau_{ij} = (-Q)^{-1}_{m*m} = \sum_{m*m}, \tag{11}$$
$$Q_{ij}(\widehat{\Theta}) = \frac{\partial^2 Q(\Theta)}{\partial \theta_i \partial \theta_j}\Big|_{\Theta = \widehat{\theta}},$$
$$Q_{ijs}(\widehat{\Theta}) = \frac{\partial^3 Q(\Theta)}{\partial \theta_i \partial \theta_j \partial \theta_s}\Big|_{\Theta = \widehat{\theta}}.$$

All the functions on the right-hand side are to be obtained as the posterior mode. Q_{ijs} is given in Appendix B. Parameters of the proposed IKum mixture model using Lindley's approximation may be obtained as

$$\widetilde{U}(\Theta) = U(\widehat{\Theta}) + \frac{1}{2}\left(D + \sum_{i=1}^{5} B_i A_k\right), \quad i = 1, 2, \ldots, 5, \tag{12}$$

where

$$D = \sum_{j=1}^{5}\sum_{k=1}^{5} U_i(\Theta)\tau_{ik},$$
$$A_k = \sum_{i=1}^{5} U_i(\Theta)\tau_{ik},$$
$$B_i = \sum_{k=1}^{5} \tau_{kk} Q_{kki}(\Theta) + 2\left\{\sum_{k=2}^{5} \tau_{ik} Q_{iki}(\Theta) + E_i\right\}$$
$$E_i = \tau_{23}Q_{23i} + \tau_{24}Q_{24i} + \tau_{25}Q_{25i} + \tau_{34}Q_{34i} + \tau_{35}Q_{35i} + \tau_{45}Q_{45i}. \tag{13}$$

After equating $U(\Theta), U_{ik} = 0, i, k = 1, 2, \ldots, 5$, and $D = 0$, so the Bayes estimators of parameters $\alpha_1, \alpha_2, \beta_1, \beta_2$, and δ_1 of the IKum mixture model under SELF are given by

$$\widetilde{\delta}_{1(\mathrm{SELF})} = \widehat{\delta}_1 + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i1}, \tag{14}$$

$$\widetilde{\alpha}_{1(\mathrm{SELF})} = \widehat{\alpha}_1 + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i2}, \tag{15}$$

$$\widetilde{\alpha}_{2(\mathrm{SELF})} = \widehat{\alpha}_2 + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i3}, \tag{16}$$

$$\widetilde{\beta}_{1(\mathrm{SELF})} = \widehat{\beta}_1 + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i4}, \tag{17}$$

$$\tilde{\beta}_{2\,(\mathrm{SELF})} = \hat{\beta}_2 + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i5}, \tag{18}$$

where B_i, $i = 1, 2, \ldots, 5$ are given earlier and τ_{ik} are the elements of the inverse of the matrix Q_{ik}. Posterior risk under SELF is the variance which can be evaluated as $E[U(\Theta)|x]^2 - (E[U(\Theta)|x])^2$ by using Mathematica 12.

Under Linex loss function, equation (14) can be written as

$$E_{\Theta}\left(e^{-\eta\Theta}\right) = \int e^{(-\eta\Theta)} U(\Theta) e^{Q(\Theta)} \mathrm{d}(\Theta). \tag{19}$$

The Bayes estimators of IKum distribution under LLF are

$$\tilde{U}_{\mathrm{LLF}}(\Theta) = -\frac{1}{\eta}\ln\left\{e^{(-\eta\Theta)} + \frac{1}{2}\left(D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right)\right\}, \tag{20}$$

$$\tilde{\delta}_{1\,(\mathrm{LLF})} = -\frac{1}{\eta}\ln\left\{e^{-\eta p_1} + \frac{1}{2}\left(U_{11} + \sum_{i=1}^{5} B_i \tau_{i1}\right)\right\}, \tag{21}$$

$$\tilde{\alpha}_{1\,(\mathrm{LLF})} = -\frac{1}{\eta}\ln\left\{e^{-\eta\alpha_1} + \frac{1}{2}\left(U_{22} + \sum_{i=1}^{5} B_i \tau_{i2}\right)\right\}, \tag{22}$$

$$\tilde{\alpha}_{2\,(\mathrm{LLF})} = -\frac{1}{\eta}\ln\left\{e^{-\eta\alpha_2} + \frac{1}{2}\left(U_{33} + \sum_{i=1}^{5} B_i \tau_{i3}\right)\right\}, \tag{23}$$

$$\tilde{\beta}_{1\,(\mathrm{LLF})} = -\frac{1}{\eta}\ln\left\{e^{-\eta\beta_1} + \frac{1}{2}\left(U_{44} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i4}\right)\right\}, \tag{24}$$

$$\tilde{\beta}_{2\,(\mathrm{LLF})} = -\frac{1}{\eta}\ln\left\{e^{-\eta\beta_1} + \frac{1}{2}\left(U_{55} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i5}\right)\right\}, \tag{25}$$

and posterior risks under LLF are

$$\rho\left(\hat{\delta}_{1\,(\mathrm{LLF})}\right) = \ln\left\{e^{(-\eta\delta_1)} + \frac{1}{2}\left[D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right]\right\} + \eta\left(\delta_{1\,(\mathrm{MLE})} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i1}\right), \tag{26}$$

$$\rho\left(\hat{\alpha}_{1\,(\mathrm{LLF})}\right) = \ln\left\{e^{(-\eta\alpha_1)} + \frac{1}{2}\left(D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right)\right\} + \eta\left(\alpha_{1\,(\mathrm{MLE})} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i2}\right), \tag{27}$$

$$\rho\left(\hat{\alpha}_{2\,(\mathrm{LLF})}\right) = \ln\left\{e^{(-\eta\alpha_2)} + \frac{1}{2}\left[D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right]\right\} + \eta\left(\alpha_{2\,(\mathrm{MLE})} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i3}\right), \tag{28}$$

$$\rho\left(\hat{\beta}_{1\,(\mathrm{LLF})}\right) = \ln\left\{e^{(-\eta\beta_1)} + \frac{1}{2}\left[D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right]\right\} + \eta\left(\beta_{1\,(\mathrm{MLE})} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i4}\right), \tag{29}$$

$$\rho\left(\hat{\beta}_{2\,(\mathrm{LLF})}\right) = \ln\left\{e^{(-\eta\beta_2)} + \frac{1}{2}\left[D_{\mathrm{LLF}} + \sum_{i=1}^{5} B_i A_i\right]\right\} + \eta\left(\beta_{2\,(\mathrm{MLE})} + \frac{1}{2}\sum_{i=1}^{5} B_i \tau_{i5}\right). \tag{30}$$

B_i are defined earlier, τ_{ik} are the elements of the inverse of the matrix Q_{ik}, and D_{LLF} and l_i, k_i, m_i, z_i, θ_i where $i = 1, 2$ are given in Appendix B.

2.4.3. Posterior Density Assuming Informative (Inverse Beta) Prior. It is assumed that the shape parameters α_i and β_i have inverse beta prior with hyperparameters a_i, b_i, and $c_i d_i$, respectively,

$$\begin{aligned} g(\alpha_i) &\propto \alpha_i^{(a_i-1)} (1-\alpha_i)^{-(a_i+b_i)} \alpha_i > 0, \quad i = 1, 2, \\ g(\beta_i) &\propto \beta_i^{(c_i-1)} (1-\beta_i)^{-(c_i+d_i)} \beta_i > 0, \quad i = 1, 2. \end{aligned} \tag{31}$$

Joint prior density for α_i, β_i, and δ_i, is given by

$$g(\Theta) \propto \alpha_i^{(a_i-1)} (1-\alpha_i)^{-(a_i+b_i)} \beta_i^{(c_i-1)} (1-\beta_i)^{-(c_i+d_i)}. \tag{32}$$

Combining the likelihood function (4) and joint prior (32), the following joint posterior density of the IKum mixture model is obtained:

$$\begin{aligned} p(\Theta|x) \propto{} & \sum_{l=0}^{m-s} \sum_{n=0}^{l} \binom{m-s}{l} \binom{l}{n} (-1)^l \delta_1^{s1+l-n} (1-\delta_1)^{s2+n} \alpha_1^{(a_1-1)} (1-\alpha_1)^{-(a_1+b_1)} \beta_1^{(c_1-1)} (1-\beta_1)^{-(c_1+d_1)} \alpha_2^{(a_2-1)} \\ & (1-\alpha_2)^{-(a_2+b_2)} \beta_2^{(c_2-1)} (1-\beta_2)^{-(c_2+d_2)} \text{Exp}\left[-\left\{ 2\alpha_1 \sum_{j=1}^{s_1} \ln\left(1+x_{1j}\right) + \sum_{j=1}^{s_1} \ln\left(1-\left(1+x_{1j}\right)^{-\alpha_1}\right) \right\}\right] \\ & \text{Exp}\left[-\left\{ 2\alpha_2 \sum_{j=1}^{s_2} \ln\left(1+x_{2j}\right) + \sum_{j=1}^{s_2} \ln\left(1-\left(1+x_{2j}\right)^{-\alpha_2}\right) \right\}\right] \text{Exp}\left[\beta_1 \left\{ \sum_{j=1}^{s.} \ln\left(1-\left(1+x_{1j}\right)^{-\alpha_1}\right) + (j-n)\ln\left(1-(1+t)^{-\alpha_1}\right) \right\}\right] \\ & \text{Exp}\left[\beta_2 \left\{ \sum_{j=1}^{s2} \ln\left(1-\left(1+x_{2j}\right)^{-\alpha_2}\right) + n \ln\left(1-(1+t)^{-\alpha_2}\right) \right\}\right], \quad \alpha_1 > 0, \alpha_2 > 0, \beta_1 > 0, \beta_2 > 0, 0 < \delta_1 < 1. \end{aligned} \tag{33}$$

The posterior density in equation (33) again does not produce Bayes estimators in explicit form, so we use Lindley's approximation given in equation (14). The final form of Bayes estimators, posterior risks, and reliability function of IKum distribution under SELF and LLF assuming informative (inverse beta) prior is the same as given in (14)–(30).

2.5. Posterior Density Assuming Noninformative (Uniform) Prior. Noninformation priors are an important part of the Bayesian tool and are considered for Bayesian analysis when there is little or no prior information available.

Let $\alpha_i, \beta_i \sim U(0, \infty)$, and$\delta_i \sim U(0, 1)$, $i = 1, 2$

So the joint prior is

$$g(\Theta) \propto 1. \tag{34}$$

Assuming independence combining the prior with likelihood function (4). The joint posterior density of α_i, β_i, and δ_i is obtained as

$$\begin{aligned} p(\Theta|\mathbf{x}) \propto{} & \prod_{i=1}^{s1} \delta_1 \alpha_1 \beta_1 \left(1+x_{1j}\right)^{-(\alpha_1+1)} \left\{1-\left(1+x_{1j}\right)^{-\alpha_1}\right\}^{\beta_1-1} \prod_{i=1}^{s2} (1-\delta_1) \alpha_2 \beta_1 \left(1+x_{2j}\right)^{-(\alpha_2+1)} \left\{1-\left(1+x_{2j}\right)^{-\alpha_2}\right\}^{\beta 2-1} \\ & \left[1-\delta_1\left\{1-(1+t_o)^{-\alpha_1}\right\}^{\beta_1} - (1-\delta_1)\left\{1-(1+t_o)^{-\alpha_2}\right\}^{\beta_2}\right]^{m-s}, \quad \alpha_1, \alpha_2, \beta_1, \beta_2 > 0, 0 < \delta_1 < 1. \end{aligned} \tag{35}$$

Mathematical expressions of the Bayes estimators and posterior risk can be obtained by assigning hyperparameters a zero value in the expressions of Bayes estimators and posterior risks in Sections 2.4.1 and 2.4.3 under SELF and LLF.

2.6. Reliability Estimation. The objective of assessing the reliability of estimates is to determine how much of the variability in the data is due to errors in measurement. And how much is in the true parameters. Approximate Bayes estimator of reliability function of IKum at some value t can be obtained as

$$\widehat{R}(t) = R(x; \Theta) + \frac{1}{2}\left(D_R + \sum_{i=1}^{5} B_i A_i\right). \tag{36}$$

Here,

$$
\begin{aligned}
D_R = \{ & l_1(\tau_{12}+\tau_{21}) + l_2(\tau_{13}+\tau_{31}) + k_1(\tau_{14}+\tau_{41}) + k_2(\tau_{15}+\tau_{51}) + \delta_1 m_1(\tau_{24}+\tau_{42}) + \\
& (1-\delta_1)m_2(\tau_{35}+\tau_{53}) + \delta_1\theta_1\tau_{22} + (1-\delta_1)\theta_2\tau_{33} + \delta_1 z_1\tau_{44} + (1-\delta_1)z_2\tau_{55}\}, \\
A_i = [& \{F_2(t)-F_1(t)\}\tau_{i1} + \delta_1 l_1\tau_{i2} + (1-\delta_1)l_2\tau_{i3} + \delta_1 k_1\tau_{i4} + (1-\delta_1)k_2\tau_{i5}], \quad i = 1,2,3,4,5,
\end{aligned}
\tag{37}
$$

where B_i is defined earlier, τ_{ik} are the elements of the invers of the matrix Q_{ik}, and $l_i, k_i, m_i, z_i, \theta_i$ $(i = 1, 2)$ are given in Appendix B.

3. Results and Discussion

3.1. Monte Carlo Simulation. Simulation is performed to get insight into properties/trends of the obtained Bayes estimators. For this purpose, a Monte Carlo simulation is performed for 1000 samples of size $n = 30$, 50, and 100 for each selection of the vector of parameters $\Theta = (\delta_1, \alpha_1, \alpha_2, \beta_1, \beta_2)$ (0.4, 1, 1.5, 0.8, 0.5), (0.4, 2, 3, 1.3, 1.5) using the inverse transformation method as follows:

Generate a random sample of different selected sample sizes from the proposed mixture model using the inverse transformation method

If $u_1 < \delta_1$, then use u_1 to generate random variate x from the mixture of two-component IKum as $x = ((1-u_1^{1/\beta_1})^{-(1/\alpha_1)} - 1)$.

If $u_1 \geq \delta_1$, then use u_2 to generate random variate x from the mixture of two-component IKum as $x = ((1-u_2^{1/\beta_2})^{-(1/\alpha_2)} - 1)$.

Select a sample censored at a fixed test termination time t and only take censored observations.

For the different choice of parameters, hyperparameter for the informative priors (gamma) are selected for $i = 1, 2$ to satisfy $E(\beta_i) = c_i/d_i$ and inverse beta $E(\beta_i) = c_i/(d_i - 1)$,

$$
\begin{aligned}
E(\beta_1) &= \frac{c_1}{d_1} = \frac{5}{2}, \\
E(\beta_2) &= \frac{c_2}{d_2} = 2, \\
E(\beta_1) &= \frac{c_1}{d_1} = \frac{10}{2}, \\
E(\beta_2) &= \frac{c_2}{d_2} = \frac{6}{2}.
\end{aligned}
\tag{38}
$$

The above steps are repeated 1000 times. The Bayes estimates are computed over 1000 repetitions by averaging the estimate and the squared deviation, respectively. Estimates are computed using two informative (gamma and inverse beta) priors and uniform noninformative prior.

Results presented in Tables 1 and 2 (Appendix A) are obtained through simulation procedure which narrates the properties of the derived Bayes estimators and posterior risks of parameters and reliability estimate of the IKum mixture model. Different sample sizes, i.e., $n = 30$, 50, and 100 are taken to perform a simulation study. It is observed that as we increase the sample size, the estimate of parameters converges to a true parametric value. It is also observed that the use of LLF assuming gamma prior produces less posterior risk, hence can be thought of as a best loss function. An experimenter always tries to choose such a loss function for which he has to bear the minimum loss for estimation. In the same context, gamma prior resulted in smaller posterior risks as compared to other priors.

Bayes estimates are found overestimated for few cases and underestimated for few cases. It is all because the complex mixture model of IKum densities is considered and these Bayes estimates are obtained by using the approximate method. MLEs obtained in the simulation study are found to be somewhat inconsistent, and when compared with Bayes estimates, we found that PRs are lesser than MSEs of MLEs.

Results of simulation are also exhibited graphically which are given in Figure 1. From graphs of simulated Bayes estimates, we can see that we almost get the same results as obtained under simulation study numerically. It means we can obtain the same Bayes estimates through graphs.

3.2. Real Dataset Example. This dataset consists of 56 observations related to the burning velocity of different chemical materials. The burning speed/velocity is the laminar flame speed under the specified composition, temperature, and pressure conditions. It decreases as the inhibitor concentration increases and can be checked by analyzing the pressure distribution in the spherical vessel and by observing the flame propagation directly. Data related to burning velocity (cm/sec) of different chemical materials are given in Table 3 and are available at http://www.cheresources.com/mists.pdf.

The following information is extracted from the above real data to analyze the mixture of the IKum model when $T = 85$:

Table 1: BEs, PRs, and reliability estimates using UP, IBeta, and gamma prior under LLF and SELF for parameters $(\delta_1, \alpha_1, \alpha_2, \beta_1, \beta_2)$ (0.4, 1.5, 1, 0.8, 0.5) and $T = 5$.

			Bayes estimates					
n	Parameters	MLEs		SELF			LLF	
			UP	IBeta	Gamma	UP	IBeta	Gamma
30	$\widehat{\alpha}_1$	1.38324	1.33580	1.29420	1.28110	1.35901	1.30142	1.29010
		0.18960	0.02753	0.04292	0.04842	0.02675	0.04129	0.04720
	$\widehat{\alpha}_2$	2.91612	0.95686	0.95498	0.95418	0.96086	0.95680	0.95581
		0.60725	0.00188	0.00204	0.00212	0.00188	0.00187	0.00201
	$\widehat{\beta}_1$	0.74762	0.65793	0.68732	0.69473	0.65693	0.68702	0.69703
		0.04374	0.004022	0.004345	0.00451	0.00401	0.00424	0.00401
	$\widehat{\beta}_2$	0.49495	0.43708	0.43455	0.43332	0.43707	0.43454	0.43331
		0.01443	0.00402	0.00434	0.00451	0.00402	0.00434	0.00450
	$\widehat{\delta}_1$	0.429810	0.42820	0.428557	0.42874	0.41087	0.41087	0.41087
		0.00832	0.00084	0.00012	0.00087	0.00012	0.00012	0.00012
	$\widehat{R}(t)$	0.029982	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.001955	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367
50	$\widehat{\alpha}_1$	1.74450	1.61835	1.45847	1.45849	1.45921	1.45991	1.45990
		0.466704	0.0149**0**	0.002214	0.002210	0.002102	0.00108	0.00102
	$\widehat{\alpha}_2$	1.83449	1.01511	1.01534	1.01530	0.98625	0.98786	0.98788
		0.291776	0.000252	0.000258	0.000250	0.00044	0.00010	0.00010
	$\widehat{\beta}_1$	0.65516	0.84292	0.83686	0.83680	0.80439	0.80339	0.80337
		0.024720	0.002230	0.002021	0.002021	0.00025	0.000211	0.000202
	$\widehat{\beta}_2$	0.68191	0.54503	0.542902	0.542901	0.54978	0.54989	0.54980
		0.0174895	0.00223	0.00204	0.00203	0.00106	0.00010	0.00011
	$\widehat{\delta}_1$	0.40616	0.40920	0.40795	0.40755	0.40769	0.40755	0.40750
		0.006272	0.00103	0.00057	0.00057	0.00057	0.00049	0.00040
	$\widehat{R}(t)$	0.026996	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.002229	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367
100	$\widehat{\alpha}_1$	1.12980	1.47760	1.4775	1.47759	1.47825	1.4780	1.4785
		0.037469	0.00051	0.00050	0.000490	0.00472	0.00462	0.00453
	$\widehat{\alpha}_2$	2.39554	0.99279	0.99271	0.992701	0.99295	0.99290	0.99299
		0.1365480	0.00005	0.000051	0.00005	0.00049	0.00049	0.00049
	$\widehat{\beta}_1$	1.04398	0.78192	0.78197	0.78199	0.78180	0.78183	0.781868
		0.0243720	0.000161	0.000160	0.000159	0.00158	0.00151	0.00150
	$\widehat{\beta}_2$	0.61518	0.48740	0.48739	0.48749	0.48734	0.49735	0.49739
		0.0070329	0.00020	0.000161	0.000160	0.00193	0.001703	0.001702
	$\widehat{\delta}_1$	0.453810	0.40697	0.4066	0.40087	0.40652	0.40647	0.40087
		0.0025615	0.000301	0.00029	0.00012	0.002730	0.00271	0.00012
	$\widehat{R}(t)$	0.066993	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.000519	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367

$$\begin{aligned} \delta_1 &= 0.40, \\ m_1 &= 22, \\ m_2 &= 34, \\ s_1 &= 25, \\ s_2 &= 26, \\ \sum_{j=1}^{s_1} \ln\left(1 - \left(1 + x_{1i}\right)^{-\alpha_1}\right) &= -43.507, \\ \sum_{j=1}^{s_2} \ln\left(1 - \left(1 + x_{2i}\right)^{-\alpha_2}\right) &= -43.773. \end{aligned} \tag{39}$$

And we obtain the following required numerical quantities when $T = 70$:

$$\begin{aligned} \delta_1 &= 0.40, \\ m_1 &= 22, \\ m_2 &= 19, \\ s_1 &= 19, \\ s_2 &= 28, \\ \sum_{j=1}^{s_1} \ln\left(1 - \left(1 + x_{1i}\right)^{-\alpha_1}\right) &= -33.195, \\ \sum_{j=1}^{s_2} \ln\left(1 - \left(1 + x_{2i}\right)^{-\alpha_2}\right) &= -46.483. \end{aligned} \tag{40}$$

Similarly considering $T = 65$, we obtain

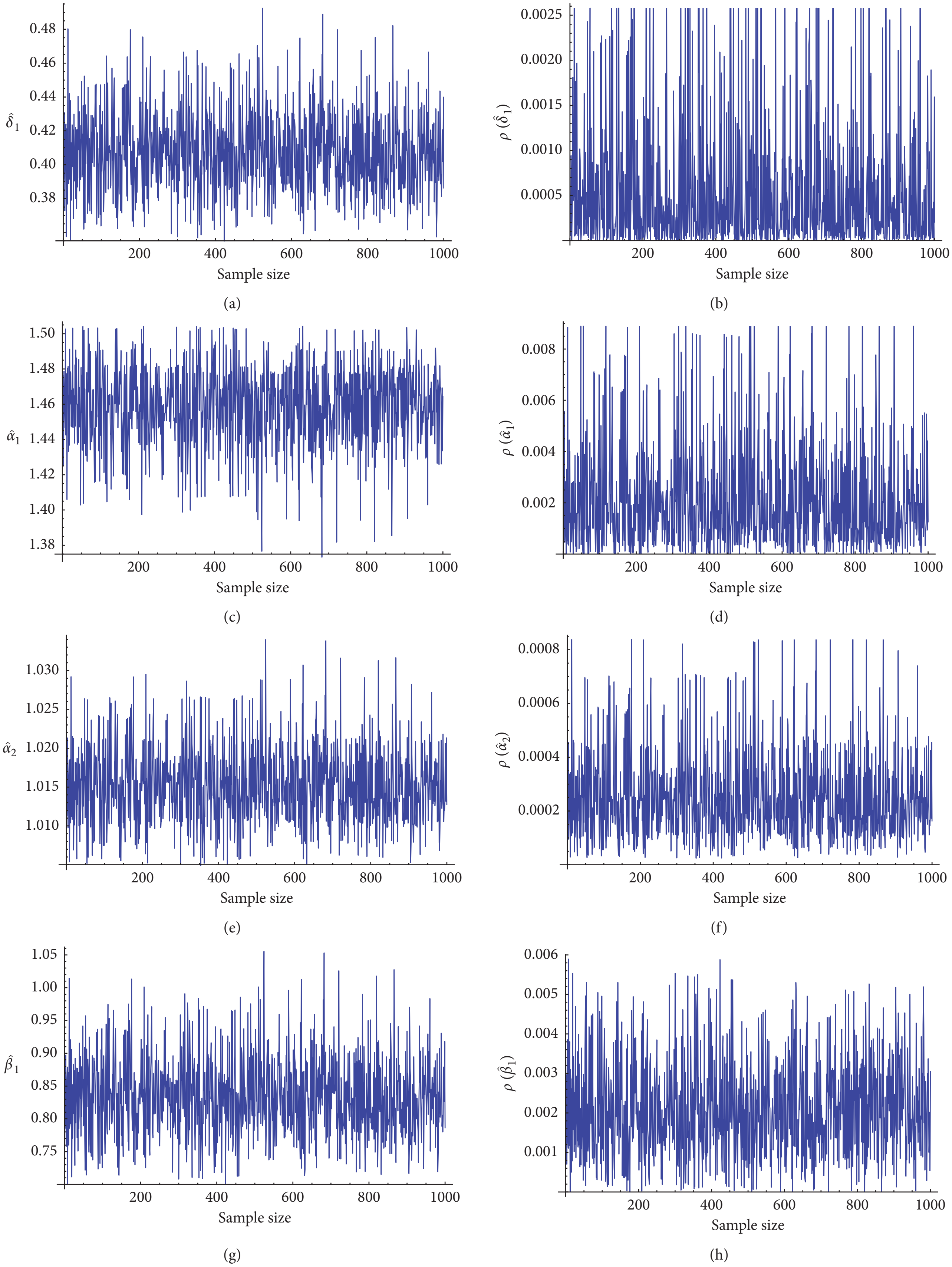
Table 2: BEs, PRs, and reliability estimates using UP, IBeta, and gamma prior under LLF and SELF parameters $(\delta_1, \alpha_1, \alpha_2, \beta_1, \beta_2)$ (0.4, 2, 3, 1.3, 1.5) and $T = 5$.

			Bayes estimates					
n	Parameters	MLEs		SELF			LLF	
			UP	IBeta	Gamma	UP	IBeta	Gamma
30	$\widehat{\alpha}_1$	2.320675	2.00356	2.00598	2.00398	2.00345	2.003098	2.00380
		0.601604	0.00001	0.00001	0.00001	0.00001	0.00000	0.00001
	$\widehat{\alpha}_2$	1.764759	2.9989	2.9979	2.9999	2.99591	2.99567	2.99589
		0.60725	0.00087	0.00077	0.00073	0.00001	0.00001	0.00001
	$\widehat{\beta}_1$	0.866185	1.23822	1.236987	1.23589	1.23519	1.23580	1.23518
		0.068320	0.00026	0.00020	0.00019	0.00026	0.00020	0.00026
	$\widehat{\beta}_2$	1.36015	1.51891	1.51633	1.51478	1.51676	1.51674	1.51601
		0.098695	0.00026	0.00021	0.00019	0.00025	0.00029	0.00026
	$\widehat{\delta}_1$	0.370755	0.42826	0.428555	0.42877	0.41089	0.41089	0.41089
		0.007937	0.00084	0.00012	0.00087	0.00012	0.00012	0.00012
	$\widehat{R}(t)$	0.029982	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.001955	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367
50	$\widehat{\alpha}_1$	1.63092	2.00310	2.00315	2.00300	2.0031	2.003001	2.00307
		0.196138	0.00001	0.00001	0.00001	0.00001	0.00000	0.00001
	$\widehat{\alpha}_2$	1.986596	2.9959	2.9969	2.9909	2.99590	2.99509	2.99599
		0.191953	0.00087	0.00077	0.00073	0.00001	0.00001	0.00001
	$\widehat{\beta}_1$	1.55010	1.2354	1.23674	1.23504	1.235103	1.2358	1.23511
		0.124060	0.00026	0.00020	0.00019	0.00026	0.00020	0.00026
	$\widehat{\beta}_2$	1.375517	1.51610	1.51610	1.51401	1.51698	1.51674	1.51601
		0.0653742	0.00026	0.00021	0.00019	0.00025	0.00029	0.00026
	$\widehat{\delta}_1$	0.40352	0.40920	0.40795	0.40755	0.40769	0.40755	0.40750
		0.005197	0.0010	0.00057	0.00057	0.00057	0.00049	0.00041
	$\widehat{R}(t)$	0.026996	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.002229	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367
100	$\widehat{\alpha}_1$	1.630920	2.00310	2.00315	2.00300	1.3590	1.30142	1.2901
		0.1961383	0.00001	0.0000	0.00003	0.02675	0.04129	0.04720
	$\widehat{\alpha}_2$	1.986596	2.9959	2.9969	2.9909	0.96086	0.95680	0.95581
		0.1919537	0.00085	0.00077	0.00073	0.00188	0.00187	0.00201
	$\widehat{\beta}_1$	1.550108	1.2354	1.23674	1.23504	0.65693	0.68702	0.69703
		0.1240605	0.000262	0.000202	0.000192	0.00401	0.00424	0.00450
	$\widehat{\beta}_2$	1.375517	1.5161	1.516101	1.51401	0.43707	0.43454	0.43331
		0.065374	0.00026	0.00021	0.00019	0.004021	0.004342	0.00450
	$\widehat{\delta}_1$	0.403526	0.40920	0.40795	0.40755	0.40769	0.40755	0.40750
		0.0051975	0.0010	0.00057	0.00057	0.00057	0.00049	0.00041
	$\widehat{R}(t)$	0.066993	0.037247	0.037346	0.037267	0.037312	0.037283	0.037266
		0.000519	0.001369	0.001361	0.001367	0.001364	0.001366	0.001367

$$
\begin{aligned}
\delta_1 &= 0.40, \\
m_1 &= 22, \\
m_2 &= 34, \\
s_1 &= 21, \\
s_2 &= 22, \\
\sum_{j=1}^{s_1} \ln\left(1 - \left(1 + x_{1i}\right)^{-\alpha_1}\right) &= -37.657, \\
\sum_{j=1}^{s_2} \ln\left(1 - \left(1 + x_{2i}\right)^{-\alpha_2}\right) &= -34.689.
\end{aligned} \tag{41}
$$

To analyze the data of the burning velocity of chemical materials, we consider three different censoring times. The data thus acquired are used to get MLEs and BEs of the parameter of the IKum mixture model that are given in Table 4. Bayes estimates are obtained assuming uniform, gamma, and inverse beta priors. However, two loss functions SELF and LLF are used for Bayesian estimation. It is found that MLEs are a bit lower than Bayes estimates. The mixture model comprises five parameters whose all four parameters are shape parameters except mixing weight. Mixing weight which is considered 0.40 in the mixture data is almost ideally estimated. Shape parameter α is estimated to be about on average 1.5 to 1.8 cm/sec for the first component, and for the second component density, it ranges from almost 0.90 to 1.00 cm/s.

3.3. Simulation Study of Real Data. In this section, using the estimates of the real dataset, we determine the estimates through a simulation study. It is observed that the Bayes estimators obtained through simulation are very close to the true values of estimators. One can easily observe that the suitable prior for these data is gamma prior and the best loss function is LLF because they provide less posterior risk. Results are given in Table 5.



(a)

(b)

(c)

(d)

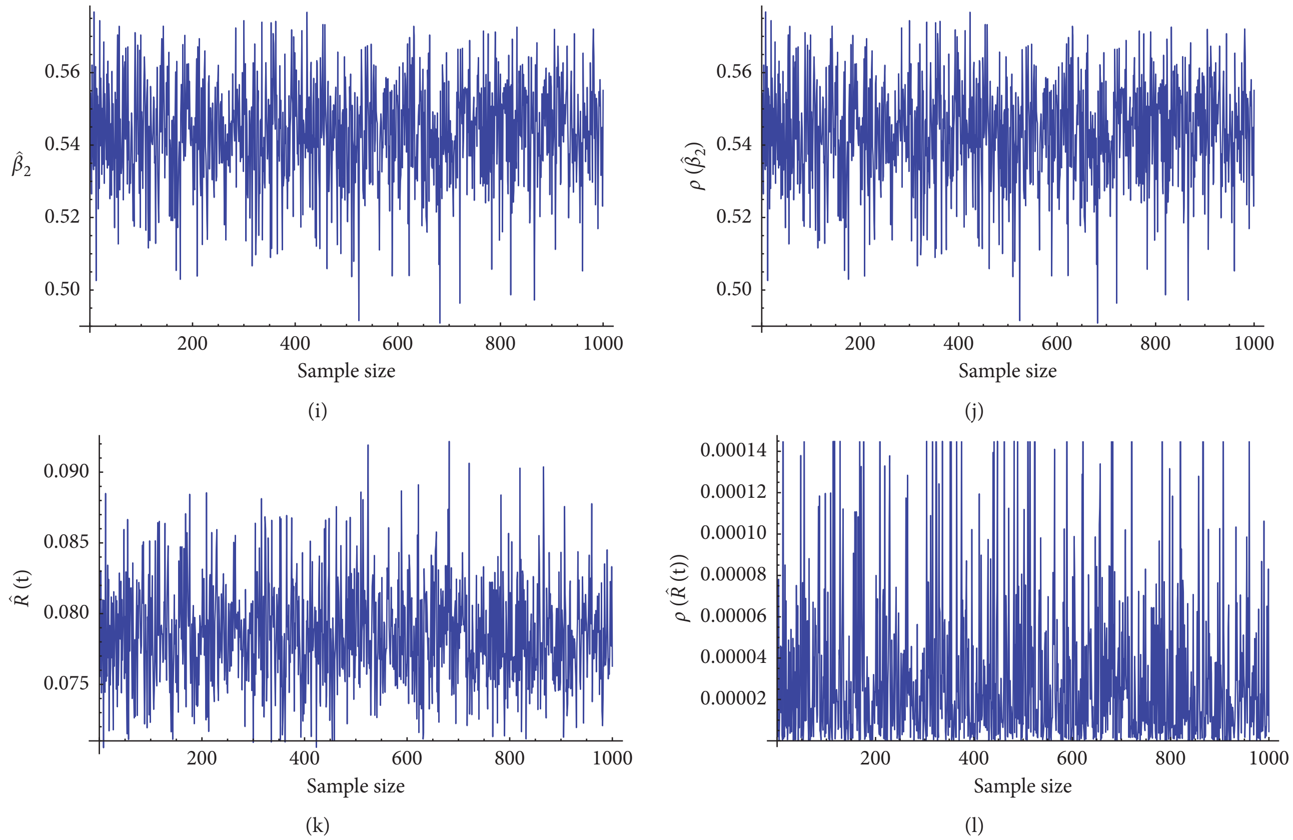
(e)

(f)

(g)

(h)

Figure 1: Continued.



(i)

(j)

(k)

(l)

Figure 1: Graphical representation of the Bayes estimators, posterior risk, and reliability function of the IKum mixture model based on simulation studies.

Table 3: Real dataset of the burning velocity of chemical materials.

68	61	64	55	51	68	44	50	82	60	89	61	54	166
66	50	87	48	42	58	46	67	46	46	44	48	56	47
54	47	89	38	108	46	40	44	312	41	31	40	41	40
56	45	43	46	46	46	46	52	58	82	71	48	39	41

Table 4: BEs and their PRs and reliability estimate using UP and IP (inverse beta and gamma) under LLF and SELF.

		Bayes estimates					
Parameters	MLEs	SELF			LLF		
		UP	IBeta	Gamma	UP	IBeta	Gamma
$T = 85,\ n = 56$							
$\hat{\alpha}_1$	1.37051	1.83988	1.83980	1.85330	1.78270	1.78143	1.54997
	0.08924	0.11546	0.11540	0.11538	0.01123	0.01120	0.00301
$\hat{\alpha}_2$	0.59800	0.92089	0.92087	0.92880	0.92446	0.92450	0.99108
	0.01446	0.00627	0.00626	0.00624	0.00034	0.00024	0.00007
$\hat{\beta}_1$	0.57709	0.85566	0.85560	0.86400	0.91727	0.91769	0.81050
	0.013332	0.00301	0.00309	0.00309	0.00025	0.00021	0.00009
$\hat{\beta}_2$	0.70302	0.51546	0.51544	0.57839	0.64110	0.64105	0.52460
	0.01606	0.00025	0.00024	0.00023	0.00055	0.00055	0.00035
$\hat{\delta}_1$	0.44828	0.42232	0.42202	0.42709	0.42662	0.42618	0.40190
	0.00446	0.00048	0.00048	0.00041	0.00010	0.00011	0.00001
$\hat{R}(t)$	0.94381	0.99511	0.99562	0.99403	0.99675	0.99820	0.99862
	0.00196	0.00001	0.00001	0.00006	0.00012	0.00013	0.00012

Table 4: Continued.

Parameters	MLEs	Bayes estimates					
			SELF			LLF	
		UP	IBeta	Gamma	UP	IBeta	Gamma
$T = 70,\ n = 56$							
$\hat{\alpha}_1$	1.06862	1.83988	1.80750	1.81851	1.78271	1.78120	1.54971
	0.06010	0.11546	0.09458	0.10148	0.01123	0.01120	0.00305
$\hat{\alpha}_2$	1.30658	0.92089	0.92659	0.93250	0.92446	0.92449	0.99104
	0.06097	0.00627	0.00539	0.00455	0.00034	0.00023	0.00006
$\hat{\beta}_1$	0.57246	0.85566	0.84716	0.85708	0.91727	0.91766	0.81049
	0.01725	0.00309	0.00220	0.00325	0.00026	0.00021	0.00009
$\hat{\beta}_2$	0.59864	0.51546	0.51452	0.57248	0.64110	0.64101	0.52459
	0.01280	0.00024	0.00021	0.00525	0.00055	0.00055	0.00035
$\hat{\delta}_1$	0.40425	0.42232	0.42029	0.42469	0.42662	0.42612	0.40197
	0.00520	0.00048	0.00041	0.00060	0.00011	0.00011	0.00001
$\hat{R}(t)$	0.93676	0.99516	0.99516	0.99412	0.99822	0.99820	0.99862
	0.00223	0.00001	0.00001	0.00002	0.00003	0.00003	0.00074
$T = 65,\ n = 56$							
$\hat{\alpha}_1$	1.18110	1.6711	1.67010	1.67012	1.52543	1.52441	1.52420
	0.06643	0.02896	0.02899	0.02894	0.00115	0.00116	0.00100
$\hat{\alpha}_2$	1.02660	0.95929	0.95923	0.95892	0.99544	0.99546	0.99540
	0.04790	0.00162	0.00626	0.00160	0.00002	0.00002	0.00002
$\hat{\beta}_1$	0.56466	0.82492	0.82492	0.82482	0.80514	0.80543	0.80501
	0.01518	0.00062	0.00309	0.00062	0.00005	0.00006	0.00005
$\hat{\beta}_2$	0.63268	0.50818	0.50819	0.50812	0.51218	0.51267	0.51215
	0.01819	0.00006	0.00023	0.00006	0.00017	0.00020	0.0001**0**
$\hat{\delta}_1$	0.48837	0.41121	0.41189	0.41120	0.40130	0.40124	0.40103
	0.00581	0.00012	0.00012	0.00012	0.00002	0.00002	0.00001
$\hat{R}(t)$	0.93409	0.99653	0.99654	0.99489	0.99663	0.99693	0.99899
	0.00052	0.00000	0.00000	0.00001	0.00000	0.00000	0.00000

Table 5: BEs, PRs, MLEs with MSEs, and reliability estimates using UP and IP (inverse beta and gamma) under LLF and SELF.

Parameters	MLEs	Bayes estimates					
			SELF			LLF	
		UP	IBeta	Gamma	UP	IBeta	Gamma
$\hat{\alpha}_1$	2.32066	1.51740	1.48950	1.85330	1.78270	1.78120	1.54970
	0.60160	0.00834	0.01369	0.11538	0.01123	0.01120	0.00304
$\hat{\alpha}_2$	1.76476	1.01512	1.01593	0.92883	0.92446	0.92449	0.99104
	0.60725	0.00071	0.00075	0.00624	0.00034	0.00023	0.00009
$\hat{\beta}_1$	0.86619	0.93457	0.93516	0.86400	0.91727	0.91766	0.81049
	0.06832	0.00122	0.00212	0.00309	0.00025	0.00021	0.00009
$\hat{\beta}_2$	1.36015	0.54724	0.56054	0.57839	0.64110	0.64101	0.52459
	0.09869	0.00127	0.00212	0.00023	0.00055	0.00055	0.00035
$\hat{\delta}_1$	0.37076	0.45179	0.45830	0.42709	0.42662	0.42612	0.40197
	0.00794	0.00420	0.00932	0.00048	0.00012	0.00011	0.00002
$\hat{R}(t)$	0.94176	0.95196	0.98027	0.98909	0.95199	0.98027	0.99693
	0.00196	0.000004	0.00000	0.00000	0.00000	0.00000	0.00000

4. Conclusion

In this paper, we conduct a Bayesian estimation of the unknown parameters and reliability function of the inverted Kumaraswamy mixture model under type 1 right censoring. For the choice of different sample sizes, $n = 30$, 50, and 100 are taken to perform a simulation study. It is observed as the sample size increases the parameters converge to their true parametric value. And it is also noted that LLF is found to be the best loss function assuming gamma prior because it has a less posterior risk.

Bayes estimates are found overestimated for some values and underestimated for few values. From the real dataset, it is observed that as the censoring times $T = 85$, 70, and 65 decrease, the posterior risk also decreases respectively. The BEs $\hat{\alpha}_1, \hat{\alpha}_2$, $\hat{\beta}_1$, and $\hat{\beta}_2$ represent the mean value of the burning velocity of the chemical material. And $\hat{R}(t)$ represents the reliability of the estimates. Simulation of real data is carried out to compare the parametric values. Graphical representation of parameters is also presented by taking the number of iterations on the x-axis and different parametric values on the y-axis. From these graphs, we come up with almost the same results of Bayes estimates. Bayes estimates are found better than MLEs as PRs are smaller as compared to MSEs.

Appendix

A. Numerical Results of Simulation Study

BEs, PRs, and reliability estimates using UP, IBeta, and gamma prior under LLF and SELF for parameters ($\delta_1, \alpha_1, \alpha_2, \beta_1, \beta_2$) (0.4, 1.5, 1, 0.8, 0.5) and $T = 5$ and (0.4, 2, 3, 1.3, 1.5) and $T = 5$ are shown.

B. Derivation for the Elements of Lindley's Approximation

Taking logarithm of posterior density given in equation (8)

$$\begin{aligned}Q(\Theta|x) = \ln\ p(\Theta|x) \propto\, & s_1 \ln\ \delta_1 + s_2 \ln(1-\delta_1) + (a_1 + s_1 - 1)\ln\ \alpha_1 + (a_2 + s_2 - 1)\ln\ \alpha_2 \\ & + (c_1 + s_1 - 1)\ln\ \beta_1 + (c_2 + s_2 - 1)\ln\ \beta_2 - b_1\alpha_1 - b_2\alpha_2 - c_1\beta_1 - c_2\beta_2 - (\alpha_1 - 1)\sum_{j=1}^{s_1}\ln\left(1 + x_{1j}\right) \\ & - (\alpha_2 - 1)\sum_{j=1}^{s_2}\ln\left(1 + x_{2j}\right) + (\beta_1 - 1)\sum_{j=1}^{s_1}\ln\left(1 - \left(1 + x_{1j}\right)^{-\alpha_1}\right) + (\beta_2 - 1)\sum_{j=1}^{s_2}\ln\left(1 - \left(1 + x_{2j}\right)^{-\alpha_2}\right) + \\ & (m-s)\ln\left(1 - \delta_1\left(1 - (1 + t_o)^{-\alpha_1}\right) - (1 - \delta_1)\left(1 - (1 + t_o)^{-\alpha_1}\right)\right).\end{aligned} \tag{B.1}$$

From equations (B.1) and (10), the elements $Q_{ijs}, Q_{ijs} = Q_{isj} = Q_{sij} = Q_{jis}$ and $t_o = T$ for $i, j, s = 1, 2, \ldots, 5$ are derived as follows:

$$\left\{\begin{aligned} \varphi_i &= (1+T)^{-\alpha_i}\ln(1+T), \\ \psi_i &= \sum_{j=1}^{m-s}\frac{\left(1 + x_{ij}\right)^{-\alpha i}\ln\left(1 + x_{ji}\right)}{\left(1 - \left(1 + x_{1i}\right)^{-\alpha_i}\right)}, \\ k_i &= -F_i\ln\left(1 - (1+T)^{-\alpha_i}\right), \\ l_i &= \frac{\beta_i F_i \varphi_i}{\left(1 - (1+T)^{-\alpha_i}\right)}, \\ m_i &= \frac{\delta_i F_i \varphi_i}{\left(1 - (1+T)^{-\alpha_i}\right)\left(1 + \beta_i\ln\left(1 - (1+T)^{-\alpha_i}\right)\right)}, \\ \theta_i &= \frac{\delta_i F_i \varphi_i \beta_i \ln(1+T) - (\beta_i - 1)\varphi_i}{\left(1 - (1+T)^{-\alpha_i}\right)^2}, \\ z_i &= -\delta_i\left(\ln\left(1 - (1+T)^{-\alpha_i}\right)\right)^2 F_i, \end{aligned}\right. \qquad i = 1, 2, \tag{B.2}$$

$$\begin{aligned} Q_{11} &= -\left[\frac{s_1}{\delta_1^2} + \frac{s_2}{(1-\delta_1)^2} + \frac{(m-s)(F_2 - F_1)^2}{R^2}\right], \\ Q_{12} &= \frac{(m-s)F_1\varphi_1\beta_1\left(p_1(F_2 - F_1) - R\right)}{\left(1 - (1+T)^{-\alpha_1}\right)R^2}, \\ Q_{13} &= \frac{(m-s)F_2\varphi_2\beta_2\left((1-p_1)(F_2 - F_1) + R\right)}{\left(1 - (1+T)^{-\alpha_2}\right)R^2}, \\ Q_{14} &= \frac{(m-s)F_1\ln\left(1 - (1+T)^{-\alpha_1}\right)\left((F_2 - F_1)p_1 - R\right)}{R^2}, \\ Q_{15} &= \frac{(m-s)F_2\ln\left(1 - (1+T)^{-\alpha_2}\right)\left((F_2 - F_1)(1-p_1) + R\right)}{R^2}, \end{aligned} \tag{B.3}$$

$$\begin{aligned}
Q_{22} &= -\frac{s_1}{\alpha_1^2} - (\beta_1 - 1)\psi_{11} + \psi_1^2 + \frac{(m-s)\delta_1\beta_1\varphi_1F_1}{(1-(1+T)^{\alpha_1})R}\left(\ln(1+T) - \frac{(\beta_1-1)\varphi_1}{(1-(1+T)^{\alpha_1})} - \frac{\delta_1\beta_1\varphi_1F_1}{(1-(1+T)^{\alpha_1})R}\right),\\
Q_{23} = Q_{32} &= \frac{(m-s)\delta_1(1-\delta_1)\beta_1\beta_2\varphi_1\varphi_2F_1F_2}{(1-(1+T)^{-\alpha_1})(1-(1+T)^{-\alpha_2})R^2},\\
Q_{24} = Q_{42} &= \psi_1 - \frac{(m-s)\delta_1\varphi_1F_1}{(1-(1+T)^{-\alpha_2})R^2}\left(\frac{1}{(1-(1+T)^{-\alpha_1})} + \frac{\ln(1-(1+T)^{-\alpha_1})\delta_1\beta_1F_1}{R} + \ln(1-(1+T)^{-\alpha_1})\beta_1\right),\\
Q_{25} = Q_{52} &= \frac{-(m-s)p_1(1-\delta_1)\beta_1\varphi_1F_1F_2\ln(1-(1+T)^{-\alpha_2})}{(1-(1+T)^{-\alpha_1})R^2},
\end{aligned} \tag{B.4}$$

$$\begin{aligned}
Q_{33} &= -\frac{s_2}{\alpha_2^2} - (\beta_2 - 1)(\psi_2^2 + \psi_{22}) - \frac{(m-s)(1-\delta_1)\beta_2\varphi_2F_2}{(1-(1+T)^{-\alpha_2})R}\left(\ln(1+T) + \frac{(\beta_2-1)\varphi_2}{(1-(1+T)^{-\alpha_2})} + \frac{(1-\delta_1)\beta_2\varphi_2F_2}{(1-(1+T)^{-\alpha_2})R}\right),\\
Q_{34} = Q_{43} &= \frac{(m-s)\delta_1(1-\delta_1)\beta_2\varphi_2F_1F_2\ln(1-(1+T)^{-\alpha_2})}{(1-(1+T)^{-\alpha_1})(1-(1+T)^{-\alpha_2})R^2},\\
Q_{35} = Q_{53} &= -\frac{(m-s)(1-\delta_1)\varphi_2F_2}{(1-(1+T)^{-\alpha_2})R}\left(1 + \beta_2\ln(1-(1+T)^{-\alpha_2}) + \frac{(1-\delta_1)\beta_2F_2\ln(1-(1+T)^{-\alpha_2})}{R}\right),\\
Q_{44} &= -\frac{s_1}{\beta_1^2} - \frac{(m-s)\delta_1F_2\ln(1-(1+T)^{-\alpha_1})}{R}\left(1 + \frac{\delta_1F_2}{R}\right),\\
Q_{45} = Q_{54} &= \frac{-(m-s)\delta_1(1-\delta_1)F_1F_2\ln(1-(1+T)^{-\alpha_1})\ln(1-(1+T)^{-\alpha_2})}{R^2},\\
Q_{55} &= -\frac{s_2}{\beta_2^2} - \frac{(m-s)(1-\delta_1)F_2\left(\ln\left(1-(1+T)^{-\alpha 2}\right)^2\right)}{R}\left(\frac{(1-\delta_1)F_2}{R} + 1\right),
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
Q_{111} &= \left(\frac{2s_1}{\delta_1^3} - \frac{2s_2}{(1-\delta_1)^3} + \frac{2(m-s)(F_2-F_1)^3}{R^3}\right),\\
Q_{112} = Q_{112} &= \frac{2(m-s)(F_2-F_1)F_1\beta_1\varphi_1}{(1-(1+T)^{-\alpha_1})R^2}\left(1 - \frac{\delta_1(F_2-F_1)}{R}\right),\\
Q_{113} = Q_{131} &= -\frac{2(m-s)(F_2-F_1)F_2\beta_2\varphi_2}{(1-(1+T)^{-\alpha_2})R^2}\left(1 + \frac{(1-\delta_1)(F_2-F_1)}{R}\right),\\
Q_{114} = Q_{141} &= \frac{2(m-s)(F_2-F_1)F_1\ln(1-(1+T)^{-\alpha_1})}{R^2}\left(1 - \frac{\delta_1(F_2-F_1)}{R}\right),\\
Q_{115} = Q_{151} &= \frac{2(m-s)(F_2-F_1)F_2\ln(1-(1+T)^{-\alpha_2})}{R^2}\left(1 - \frac{(1-\delta_1)(F_2-F_1)}{R}\right),
\end{aligned} \tag{B.6}$$

$$\begin{aligned}
Q_{122} = Q_{211} &= \frac{(m-s)F_1\beta_1\varphi_1}{(1-(1+T)^{-\alpha_1})R}\left(\begin{array}{c} -\dfrac{\delta_1(F_2-F_1)\ln(1+T)}{R} + \ln(1+T) + \dfrac{\beta_1\varphi_1\delta_1(F_2-F_1)\ln(1+T)}{(1-(1+T)^{-\alpha_1})R} - \dfrac{(\beta_1-1)\varphi_1}{(1-(1+T)^{-\alpha_1})}\\ +\dfrac{2\delta_1^2\beta_1\varphi_1(F_2-F_1)F_1}{(1-(1+T)^{-\alpha_1})R^2} - \dfrac{2\beta_1\varphi_1\delta_1F_1}{(1-(1+T)^{-\alpha_1})R}\end{array}\right),\\
Q_{123} = Q_{132} = Q_{312} &= \frac{(m-s)F_1F_2\beta_1\beta_2\varphi_2\varphi_1}{(1-(1+T)^{-\alpha_1})(1-(1+T)^{-\alpha_2})R^2}\left(\frac{2\delta_1(F_2-F_1)(1-\delta_1)}{R} + 2\delta_1 - 1\right),\\
Q_{124} = Q_{142} = Q_{412} &= \frac{(m-s)F_1\varphi_1}{(1-(1+T)^{-\alpha_1})R^2}\left(\begin{array}{c} \dfrac{\delta_1(F_2-F_1)}{R} - 1 + \dfrac{2\delta_1^2F_1\beta_1\ln(1-(1+T)^{-\alpha_1})}{R^2} - \dfrac{2\delta_1F_1\beta_1\ln(1-(1+T)^{-\alpha_1})}{R}\\ +\dfrac{\delta_1(F_1-F_2)\beta_1\ln(1-(1+T)^{-\alpha_1})}{R^2} - \beta_1\ln(1-(1+T)^{-\alpha_1})\end{array}\right),\\
Q_{125} = Q_{152} = Q_{251} &= \frac{(m-s)F_1F_2\beta_1\varphi_1\ln(1-(1+T)^{-\alpha_2})}{R^2(1-(1+T)^{-\alpha_1})}\left(\frac{2(1-\delta_1)\delta_1(F_2-F_1)}{R} + 2\delta_1 - 1\right),
\end{aligned} \tag{B.7}$$

$$
\begin{aligned}
Q_{133} = Q_{331} &= \frac{(m-s)F_2\beta_2\varphi_2}{R\left(1-(1+T)^{-\alpha_1}\right)}\left(-\ln(1+T) - \frac{(1-\delta_1)\ln(1+T)(F_2-F_1)}{R} + \frac{(1-\delta_1)(\beta_2-1)\varphi_2(F_2-F_1)}{R\left(1-(1+T)^{-\alpha_1}\right)} + \frac{(\beta_2-1)\varphi_2}{\left(1-(1+T)^{-\alpha_1}\right)}\right.\\
&\quad\left.+\frac{2(1-\delta_1)^2F_2\beta_2\varphi_2(F_2-F_1)}{R^2\left(1-(1+T)^{-\alpha_1}\right)} + \frac{2(1-\delta_1)\beta_2\varphi_2F_2}{R\left(1-(1+T)^{-\alpha_1}\right)}\right),\\
Q_{134} = Q_{341} = Q_{143} &= \frac{(m-s)F_2F_1\beta_2\varphi_2\ln\left(1-(1+T)^{-\alpha_1}\right)}{R^2\left(1-(1+T)^{-\alpha_1}\right)}\left(\frac{2\delta_1(1-\delta_1)(F_2-F_1)}{R} + 2\delta_1 - 1\right),\\
Q_{135} = Q_{351} = Q_{153} &= \frac{(m-s)F_2\varphi_2}{R\left(1-(1+T)^{-\alpha_1}\right)}\left(\begin{array}{c} 1+\beta_2\ln\left(1-(1+T)^{-\alpha_1}\right) + \dfrac{(1-\delta_1)(F_2-F_1)}{R} + \dfrac{2(1-\delta_1)^2\beta_2F_2(F_2-F_1)\ln\left(1-(1+T)^{-\alpha_1}\right)}{R^2}\\[2ex] +\dfrac{2\beta_2(1-\delta_1)n\left(1-(1+T)^{-\alpha_2}\right)F_2}{R} + \dfrac{(1-\delta_1)\beta_2(F_2-F_1)\ln\left(1-(1+T)^{-\alpha_1}\right)}{R}\end{array}\right),\\
Q_{144} = Q_{441} = Q_{414} &= \frac{(m-s)F_2n\left(1-(1+T)^{-\alpha_1}\right)n\left(1-(1+T)^{-\alpha_2}\right)}{R^2}\left(\frac{2\delta_1^2F_1(F_2-F_1)}{R} - 2\delta_1F_1 + \delta_1(F_2-F_1) - 1\right),\\
Q_{145} = Q_{451} &= \frac{(m-s)F_2F_1\ln\left(1-(1+T)^{-\alpha_1}\right)\ln\left(1-(1+T)^{-\alpha_2}\right)}{R^2}\left(\frac{2\delta_1(1-\delta_1)(F_2-F_1)}{R} + 2\delta_1 - 1\right),
\end{aligned}
\tag{B.8}
$$

$$
\begin{aligned}
Q_{155} = Q_{551} &= \frac{(m-s)F_2\ln\ \left(1-(1+T)^{-\alpha_2}\right)^2}{R^2}\left(1 + \frac{2(1-\delta_1)F_2}{R} + \frac{2(1-\delta_1)F_2}{R^2} + \frac{2(1-\delta_1)^2F_2(F_2-F_1)}{R} + \frac{(1-\delta_1)(F_2-F_1)}{R}\right),\\
Q_{222} &= -(\beta_1-1)\left(3\psi_{1111} + 2\psi_1^3 + \psi_{111}\right) + \frac{2s_1}{\alpha_1^3} + \frac{(m-s)\beta_1\delta_1\varphi_1F_1}{\left(1-(1+T)^{-\alpha_1}\right)R}\left(-\ln\ (1+T)^2\right) - \frac{2F_1^2\beta_1^2\delta_1^2\varphi_1^2}{\left(1-\ (1+T)^{-\alpha_1}\right)^2R^2} - \frac{F_1\beta_1(2\beta_1-2)\delta_1\varphi_1^2}{\left(1-\ (1+T)^{-\alpha_1}\right)^2R}\\
&\quad - \frac{2F_1^2\beta_1^2\delta_1^2\varphi_1^2}{\left(1-\ (1+T)^{-\alpha_1}\right)^2R^2} - \frac{F_1\beta_1(2\beta_1-2)\delta_1\varphi_1^2}{\left(1-\ (1+T)^{-\alpha_1}\right)^2R},\\
Q_{223} = Q_{332} &= \frac{(m-s)(1-\delta_1)\delta_1F_1F_2\beta_1\beta_2\varphi_2\varphi_1}{\left(1-(1+T)^{-\alpha_1}\right)\left(1-(1+T)^{-\alpha_2}\right)R^2}\left(\ln(1+T) - \frac{2\delta_1\beta_1\varphi_1F_1}{R\left(1-(1+T)^{-\alpha_1}\right)} - \frac{(\beta_1-1)\varphi_1}{\left(1-(1+T)^{-\alpha_1}\right)}\right),\\
Q_{224} = Q_{242} &= \frac{(m-s)(1-\delta_1)\delta_1F_1F_2\beta_1\varphi_1\ln\left(1-(1+T)^{-\alpha_2}\right)}{\left(1-(1+T)^{-\alpha_1}\right)R^2}\left(\ln(1+T) - \frac{2\delta_1\beta_1\varphi_1F_1}{R\left(1-(1+T)^{-\alpha_1}\right)} - \frac{(\beta_1-1)\varphi_1}{\left(1-(1+T)^{-\alpha_1}\right)}\right),\\
Q_{225} = Q_{252} &= \frac{(m-s)(1-\delta_1)\delta_1F_1F_2\beta_1\varphi_1\ln\left(1-(1+T)^{-\alpha_2}\right)}{\left(1-(1+T)^{-\alpha_1}\right)R^2}\left(\ln(1+T) - \frac{2\delta_1\beta_1\varphi_1F_1}{R\left(1-(1+T)^{-\alpha_1}\right)} - \frac{(\beta_1-1)\varphi_1}{\left(1-(1+T)^{-\alpha_1}\right)}\right),\\
Q_{234} = Q_{243} = Q_{342} &= -\frac{(m-s)(1-\delta_1)\delta_1F_1F_2\beta_2\varphi_1\varphi_2}{\left(1-(1+T)^{-\alpha_2}\right)\left(1-(1+T)^{-\alpha_1}\right)R^2}\left(1 + \beta_1\ln\left(1-(1+T)^{-\alpha_1}\right) + \frac{2\delta_1\beta_1\ln\left(1-(1+T)^{-\alpha_1}\right)F_1}{R}\right),\\
Q_{235} = Q_{253} = Q_{352} &= -\frac{(m-s)(1-\delta_1)\delta_1F_1F_2\beta_1\varphi_1\varphi_2}{\left(1-(1+T)^{-\alpha_2}\right)\left(1-(1+T)^{-\alpha_1}\right)R^2}\left(1 + \beta_2\ln\left(1-(1+T)^{-\alpha_2}\right) + \frac{2(1-\delta_1)\beta_2\ln\left(1-(1+T)^{-\alpha_2}\right)F_2}{R}\right),
\end{aligned}
\tag{B.9}
$$

$$\begin{aligned}
Q_{332} = Q_{233} = Q_{323} &= -\frac{(n-r)(1-\delta_1)\delta_1 F_1 F_2 \beta_1 \beta_2 \varphi_1 \varphi_2}{(1-(1+T)^{-\alpha_2})(1-(1+T)^{-\alpha_1})R^2}\left(\ln(1+T) - \frac{2(1-\delta_1)\beta_2\varphi_2 F_2}{R(1-(1+T)^{-\alpha_2})} - \frac{(1-\beta_2)\varphi_2}{(1-(1+T)^{-\alpha_2})}\right),\\
Q_{244} = Q_{442} = Q_{424} &= -\frac{(m-s)\delta_1 F_1 \varphi_1 \ln(1-(1+T)^{-\alpha_2})}{(1-(1+T)^{-\alpha_1})R}\left(2 + \beta_2 \ln(1-(1+T)^{-\alpha_2}) + \frac{2\delta_1\beta_1 F_1}{R} + \frac{2\delta_1^2\beta_1 \ln(1-(1+T)^{-\alpha_1})F_1^2}{R^2}\right.\\
&\quad \left.+\frac{3\delta_1\beta_1 \ln(1-(1+T)^{-\alpha_1})F_1}{R}\right),\\
Q_{245} = Q_{254} = Q_{452} &= -\frac{(m-s)(1-\delta_1)\delta_1 F_1 \varphi_1 \ln(1-(1+T)^{-\alpha_1})}{(1-(1+T)^{-\alpha_1})R^2}\left(1 + \beta_1 \ln(1-(1+T)^{-\alpha_2}) + \frac{2\delta_1\beta_1 \ln(1-(1+T)^{-\alpha_1})F_1}{R}\right),\\
Q_{255} = Q_{552} = Q_{525} &= -\frac{(m-s)(1-\delta_1)\delta_1 F_1 F_2 \varphi_1 \beta_1 \varphi_1 (\ln(1-(1+T)^{-\alpha_2}))^2}{(1-(1+T)^{-\alpha_1})R^2}\left(1 + \frac{2(1-\delta_1)F_2}{R}\right),\\
Q_{222} &= -(\beta_2 - 1)(3\psi_{2222} + 2\psi_2^3 + \psi_{222}) + \frac{2s_2}{\alpha_2^3} + \frac{(m-s)\beta_2(1-\delta_1)\varphi_2 F_2}{(1-(1+T)^{-\alpha_2})R}\\
&\quad \left(-\ln\ (1+T)^2 - \frac{2F_1^2\beta_1^2\delta_1^2\varphi_1^2}{(1-\ (1+T)^{-\alpha_1})^2R^2} - \frac{F_1\beta_1(2\beta_1-2)\delta_1\varphi_1^2}{(1-\ (1+T)^{-\alpha_1})^2R} - \frac{2F_1^2\beta_1^2\delta_1^2\varphi_1^2}{(1-\ (1+T)^{-\alpha_1})^2R^2} - \frac{F_1\beta_1(2\beta_1-2)\delta_1\varphi_1^2}{(1-\ (1+T)^{-\alpha_1})^2R}\right),
\end{aligned} \tag{B.10}$$

$$\begin{aligned}
Q_{334} = Q_{343} &= \frac{(m-s)(1-\delta_1)\delta_1 F_1 F_2 \beta_2 \varphi_2 \ln(1-(1+T)^{-\alpha_1})}{(1-(1+T)^{-\alpha_2})R^2}\left(\ln(1+T) - \frac{2(1-\delta_1)\beta_2\varphi_2 F_2}{R(1-(1+T)^{-\alpha_2})} - \frac{(\beta_2-1)\varphi_2}{(1-(1+T)^{-\alpha_2})}\right),\\
Q_{344} = Q_{443} = Q_{434} &= -\frac{(m-s)(1-\delta_1)\delta_1 F_1 F_2 \beta_2 \varphi_2 \ln\ (1-(1+T)^{-\alpha_1})^2}{(1-(1+T)^{-\alpha_2})R^2}\left(\frac{2\delta_1 F_1}{R} + 1\right),\\
Q_{345} = Q_{354} = Q_{453} &= -\frac{(m-s)(1-\delta_1)\delta_1 F_2 \varphi_2 \ln(1-(1+T)^{-\alpha_1})}{(1-(1+T)^{-\alpha_2})R^2}\ (1 + \beta_2 \ln(1-(1+T)^{-\alpha_2})\\
&\quad \left.+\frac{2(1-\delta_1)\beta_2 \ln(1-(1+T)^{-\alpha_2})F_2}{R}\right),\\
Q_{355} = Q_{553} = Q_{535} &= -\frac{(m-s)(1-\delta_1)F_2\varphi_2 \ln(1-(1+T)^{-\alpha_2})}{(1-(1+T)^{-\alpha_2})R}\left(2 + \beta_2 \ln(1-(1+T)^{-\alpha_2}) + \frac{2(1-\delta_1)F_2}{R}\right.\\
&\quad \left.+\frac{2(1-\delta_1)\beta_2 \ln(1-(1+T)^{-\alpha_2})F_2^3}{R^2} + \frac{3(1-\delta_1)\beta_2 \ln(1-(1+T)^{-\alpha_2})F_2}{R}\right),\\
Q_{444} &= \frac{2s_1}{\beta_1^3} - \frac{(m-s)\delta_1 F_1 \ln\ (1-(1+T)^{-\alpha_1})^2}{R}\left(\ln(1-(1+T)^{-\alpha_1}) + \frac{3\delta_1 F_1 \ln(1-(1+T)^{-\alpha_1})}{R} + \frac{2\delta_1^2 F_1^2 \ln(1-(1+T)^{-\alpha_1})}{R^2}\right),\\
Q_{445} = Q_{454} &= \frac{(m-s)\delta_1(1-\delta_1)F_1F_2 \ln(1-(1+T)^{-\alpha_2})\ln\ (1-(1+T)^{-\alpha_1})^2}{R^2}\left(\frac{2\delta_1 F_1}{R} + 1\right),\\
Q_{455} = Q_{554} = Q_{545} &= \frac{(m-s)\delta_1(1-\delta_1)F_1F_2 \ln\ (1-(1+T)^{-\alpha_2})^2 \ln(1-(1+T)^{-\alpha_1})}{R^2}\left(\frac{2(1-\delta_1)F_2}{R} + 1\right),\\
Q_{555} &= \frac{2s_2}{\beta_2^3} - \frac{(m-s)(1-\delta_1)F_2 \ln\ (1-(1+T)^{-\alpha_2})^2}{R}\left(\ln(1-(1+T)^{-\alpha_2}) + \frac{3(1-\delta_1)F_2}{R} + \frac{2(1-\delta_1)^2F_1^2}{R^2}\right),
\end{aligned} \tag{B.11}$$

$$\begin{aligned}
D_{\mathrm{LLF}} &= \frac{\partial U(\Theta)}{\partial \Theta} = U_{11}, U_{22}, U_{33}, U_{44}, U_{55}, \\
\text{if } \Theta = \delta_1 \text{ than}, U_{11} &= \frac{\partial}{\partial \delta_1}\left(e^{-\eta\delta_1}\right) = \delta_1^2 e^{-\eta\delta_1}, \\
\Theta = \alpha_1, U_{11} &= \frac{\partial}{\partial \alpha_1}\left(e^{-\eta\alpha_1}\right) = \alpha_1^2 e^{-\eta\alpha_1}, \\
\Theta = \alpha_2, U_{22} &= \frac{\partial}{\partial \alpha_2}\left(e^{-\eta\alpha_2}\right) = \alpha_2^2 e^{-\eta\alpha_2}, \\
\Theta = \beta_1, U_{33} &= \frac{\partial}{\partial \beta_1}\left(e^{-\eta\beta_1}\right) = \beta_1^2 e^{-\eta\beta_1}, \\
\Theta = \beta_2, U_{55} &= \frac{\partial}{\partial \beta_2}\left(e^{-\eta\beta_2}\right) = \beta_2^2 e^{-\eta\beta_2}.
\end{aligned} \tag{B.12}$$

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Ali, M. Aslam, D. Kundu, and S. M. A. Kazmi, “Bayesian estimation of the mixture of generalized exponential distribution: a versatile lifetime model in industrial processes,” *Journal of the Chinese Institute of Industrial Engineers*, vol. 29, no. 4, pp. 246–269, 2012.

[2] C. G. Bhattacharya, “A simple method of resolution of a distribution into Gaussian components,” *Biometrics*, vol. 23, no. 1, pp. 115–135, 1967.

[3] D. R. Harris, “The development of socio-legal studies in the United Kingdom,” *Legal Studies*, vol. 3, no. 3, pp. 315–333, 1983.

[4] K. Jedidi, H. S. Jagpal, and W. S. DeSarbo, “Finite-mixture structural equation models for response-based segmentation and unobserved heterogeneity,” *Marketing Science*, vol. 16, no. 1, pp. 39–59, 1997.

[5] K. S. Sultan, M. A. Ismail, and A. S. Al-Moisheer, “Mixture of two inverse Weibull distributions: properties and estimation,” *Computational Statistics and Data*, vol. 51, no. 1, pp. 5377–5387, 2007.

[6] B. S. Everitt, “Mixture distributions—I,” in *Encyclopedia of Statistical Sciences*John Wiley and Sons, Inc., New York, NY, USA, 2004.

[7] S. Ali, “Mixture of the inverse Rayleigh distribution: properties and estimation in Bayesian framework,” *Applied Mathematical Modelling*, vol. 39, no. 2, pp. 515–530, 2014.

[8] N. Feroze and M. Aslam, “Bayesian analysis of doubly censored lifetime data using two component mixture of Weibull distribution,” *Journal of the National Science Foundation of Sri Lanka*, vol. 42, no. 4, pp. 325–334, 2014.

[9] H. Zhang and Y. Huang, “Finite mixture models and their applications: a review,” *Austin Biometrics and Biostatistics*, vol. 2, no. 1, pp. 1–6, 2015.

[10] M. D. Fundi, E. G. Njenga, and K. G. Keitany, “Estimation of parameters of the two-parameter Rayleigh distribution based on progressive type-II censoring using maximum likelihood method via the NR and the EM algorithms,” *American Journal of Theoretical and Applied Statistics*, vol. 6, no. 1, pp. 1–9, 2017.

[11] Y. M. Tripathi, C. Petropoulos, F. Sultana, and M. K. Rastogi, “Estimating a linear parametric function of a doubly censored exponential distribution,” *Statistics*, vol. 52, no. 1, pp. 99–114, 2018.

[12] F. Noor, A. Sajid, S. B. H. Shah, M. Zaman, M. Gheisari, and V. Mariappan, “Bayesian estimation and prediction for Burr-Rayleigh mixture model using censored data,” *International Journal of Communication Systems*, vol. 32, no. 15, p. 4094, 2019.

[13] N. Feroze and M. Aslam, “Approximate Bayesian analysis of doubly censored samples from mixture of two Weibull distributions,” *Communications in Statistics - Theory and Methods*, vol. 48, no. 11, pp. 2862–2878, 2019.

[14] A. Aljuaid, “Estimating the parameters of an exponentiated inverted Weibull distribution under type-II censoring,” *Applied Mathematical Sciences*, vol. 7, no. 35, pp. 1721–1736, 2013.

[15] F. Noor and M. Aslam, “Bayesian inference of the inverse Weibull mixture distribution using type-I censoring,” *Journal of Applied Statistics*, vol. 40, no. 5, pp. 1076–1089, 2013.

[16] R. I. Abd ELs-Kader, G. R. AL-Dayian, and S. A. AL-Gendy, “Inverted Pareto type I distribution: properties and estimation,” *Journal of Faculty of Commerce AL-Azhar University*, vol. 21, pp. 19–40, 2003.

[17] A. M. Basheer, “Alpha power inverse Weibull distribution with reliability application,” *Journal of Taibah University for Science*, vol. 13, no. 1, pp. 423–432, 2019.

[18] A. S. Hassan and A. N. Zaky, “Estimation of entropy for inverse Weibull distribution under multiple censored data,” *Journal of Taibah University for Science*, vol. 13, no. 1, pp. 331–337, 2019.

[19] P. Kumaraswamy, “A generalized probability density function for double-bounded random processes,” *Journal of Hydrology*, vol. 46, no. 1-2, pp. 79–88, 1980.

[20] A. M. AL-Fattah, A. A. EL-Helbawy, and G. R. AL-Dayian, “Inverted Kumaraswamy distribution: properties and estimation,” *Pakistan Journal of Statistics (PJS)*, vol. 33, no. 1, pp. 37–61, 2017.

[21] W. Mendenhall and R. J. Hader, “Estimation of parameters of mixed exponentially distributed failure time distributions from censored life test data,” *Biometrika*, vol. 45, no. 3-4, pp. 504–520, 1958.

[22] D. V. Lindley, “Approximate bayesian methods,” *Trabajos de Estadistica Y de Investigacion Operativa*, vol. 31, no. 1, pp. 223–245, 1980.

[23] Z. F. Jaheen, “Bayesian estimations and predictions based on single Burr type XII models and their finite mixture,” Ph. D. thesis, Egypt University of Assiut, Assiut, Egypt, 1993.

[24] K. E. Ahmad, H. M. Moustafa, and A. M. Abd-Elrahman, “Approximate Bayes estimation for mixtures of two Weibull distributions under type-2 censoring,” *Journal of Statistical Computation and Simulation*, vol. 58, no. 3, pp. 269–285, 1997.

[25] K. S. Sultan and A. S. Al-Moisheer, “Approximate Bayes estimation of the parameters and reliability function of a mixture of two inverse Weibull distributions under type-2 censoring,” *Journal of Statistical Computation and Simulation*, vol. 83, no. 10, pp. 1900–1914, 2013.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9975700, 9 pages
https://doi.org/10.1155/2021/9975700

Research Article

Weapon Detection Using YOLO V3 for Smart Surveillance System

Sanam Narejo,[1] **Bishwajeet Pandey**,[2] **Doris Esenarro vargas**,[3] **Ciro Rodriguez**,[4] **and M. Rizwan Anjum**[5]

[1]*Department of Computer Systems Engineering, Mehran University of Engineering and Technology (MUET), Jamshoro, Pakistan*
[2]*Gran Sasso Science Institute, L'Aquila, Italy*
[3]*Universidad Nacional Federico Villarreal, Lima, Peru*
[4]*Universidad Nacional Mayor de San Marcos, Lima, Peru*
[5]*Department of Electronic Engineering, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan*

Correspondence should be addressed to Bishwajeet Pandey; dr.pandey@ieee.org

Received 4 March 2021; Revised 15 April 2021; Accepted 3 May 2021; Published 12 May 2021

Academic Editor: Zain Anwar Ali

Every year, a large amount of population reconciles gun-related violence all over the world. In this work, we develop a computer-based fully automated system to identify basic armaments, particularly handguns and rifles. Recent work in the field of deep learning and transfer learning has demonstrated significant progress in the areas of object detection and recognition. We have implemented YOLO V3 "You Only Look Once" object detection model by training it on our customized dataset. The training results confirm that YOLO V3 outperforms YOLO V2 and traditional convolutional neural network (CNN). Additionally, intensive GPUs or high computation resources were not required in our approach as we used transfer learning for training our model. Applying this model in our surveillance system, we can attempt to save human life and accomplish reduction in the rate of manslaughter or mass killing. Additionally, our proposed system can also be implemented in high-end surveillance and security robots to detect a weapon or unsafe assets to avoid any kind of assault or risk to human life.

1. Introduction

Violence committed with guns puts significant impact on public, health, psychological, and economic cost. Many people die each year from gun-related violence. Psychological trauma is frequent among children who are exposed to high levels of violence in their communities or through the media. Children exposed to gun-related violence, whether they are victims, perpetrators, or witnesses, can experience negative psychological effects over the short and long terms. Number of studies show that handheld gun is the primary weapon used for various crimes like break-in, robbery, shoplifting, and rape. These crimes can be reduced by identifying the disruptive behavior at early stage and monitoring the suspicious activities carefully so that law enforcement agencies can further take immediate action [1].

Levels of gun-related violence vary greatly among geographical locations and countries. The global death toll from use of guns may be as high as 1,000 dead each day [2]. According to statistics, 4.2 in 100000 people are killed in Pakistan every year in mass shootings. From street crimes to an individual institution attack, many precious lives suffered. This further indicates that manual surveillance system still needs human eye to detect the abnormal activities and it takes a sufficient amount of time reporting to security officials to tackle the situation.

Although the human visual framework is quick and precise and can likewise perform complex undertakings like distinguishing different items and recognizing snags with minimal cognizant idea, however, it is common truth that if an individual watches something very similar for quite a long time, there is an opportunity of sluggishness and lack of regard.

Nowadays, with the accessibility of huge datasets, quicker GPUs, advanced machine learning algorithms, and better calculations, we can now effectively prepare PCs and develop automated computer-based system to distinguish and identify numerous items on a site with high accuracy.

Recent developments indicate that machine learning [3–6] and advance image processing algorithms have played dominant role in smart surveillances and security systems [7, 8]. Apart from this, popularity of smart devices and networked cameras has also empowered this domain. However, human objects or weapon detection and tracking are still conducted at cloud centers, as real-time, online tracking is computationally costly. Significant efforts have been made in recent years to monitor robot manipulators that need high control performance in reliability and speed [9, 10]. The researchers have attempted to improve the response characteristics of the robotic system and to attenuate the uncertainties in [11]. The proposed developed robust model-free controller incorporates time delay control (TDC) and adaptive terminal sliding mode control (ATSMC) methods.

In this research work, we aim to develop a smart surveillance security system detecting weapons specifically guns. For this purpose, we have applied few compute vision methods and deep learning for identification of a weapon from captured image. Recent work in the field of machine learning and deep learning particularly convolutional neural networks has shown considerable progress in the areas of object detection and recognition, exclusively in images. As the first step for any video surveillance application, object detection and classification are essential for further object tracking tasks. For this purpose, we trained the classifier model of YOLO v3, i.e., "You Only Look Once" [12, 13]. This model is a state-of-the-art real-time object detection classifier. Furthermore, we are not just detecting the guns, rifles, and fire but also getting the location of the incident and storing the data for future use. We have connected three systems using socket programing as a demonstration for the real-life scenario as camera, CCTV operator, and security panels.

This work is an attempt to design and develop a system which can detect the guns, rifles, and fire in no time with less computational resources. It is evident from technological advancements that most of the human assisted applications are now automated and computer-based. Eventually, in future these computer-based systems will be replaced by more smart machines, robots, or humanoid robots. In order to provide visionary sense to robots, object detection plays fundamental part for understanding the objects and its interpretation. Thus, our proposed system can also be implemented in surveillance and security robots to detect any weapon or unsafe assets.

2. Literature Review

Reducing the life-threatening acts and providing high security are challenging at every place. Therefore, a number of researchers have contributed to monitoring various activities and behaviors using object detection. In general, a framework of smart surveillance system is developed on three levels: firstly, to extract low-level information like features engineering and object tracking; secondly, to identify unusual human activities, behavior, or detection of any weapon; and finally, the high level is about decision making like abnormal event detection or any anomaly. The latest anomaly detection techniques can be divided into two groups, which are object-centered techniques and integrated methods. The convolutional neural network (CNN) spatial-temporal system is only applied to spatial-temporal volumes of interest (SVOI), reducing the cost of processing. In surveillance videos of complex scenes, researchers in [14] proposed a tool for detecting and finding anomalous activities. By conducting spatial-temporal convolution layer, this architecture helps one to capture objects from both time domain and frequency domain, thereby extracting both the presence and motion data encoded in continuous frames. To do traditional functions to local noise and improve detection precision, spatial-temporal convolution layers are only implemented within spatial-temporal quantities of changing pixels. Researchers proposed anomaly-introduced learning method for detecting anomalous activities by developing multi-instance learning graph-based model with abnormal and normal bimodal data, highlighting the positive instances by training coarse filter using kernel-SVM classifier and generating improved dictionary learning known as anchor dictionary learning. Thus, abnormality is measure by selecting the sparse reconstruction cost which yields the comparison with other techniques including utilizing abnormal information and reducing time and cost for SRC.

Hu et al. [15] have contributed in detecting various objects in traffic scenes by presenting a method which detects the objects in three steps. Initially, it detects the objects, recognizes the objects, and finally tracks the objects in motion by mainly targeting three classes of different objects including cars, cyclists, and traffic signs. Therefore, all the objects are detected using single learning-based detection framework consisting of dense feature extractor and trimodal class detection. Additionally, dense features are extracted and shared with the rest of detectors which heads to be faster in speed that further needs to be evaluated in testing phase. Therefore, intraclass variation of objects is proposed for object subcategorization with competitive performance on several datasets.

Grega et al. presented an algorithm which automatically detects knives and firearms in CCTV image and alerts the security guard or operator [16]. Majorly, focusing on limiting false alarms and providing a real-time application where specificity of the algorithm is 94.93% and sensitivity is 81.18% for knife detection. Moreover, specificity for fire alarm system is 96.69% and sensitivity is 35.98% for different objects in the video. Mousavi et al. in [17] carried out video classifier also referred to as the Histogram of Directed Tracklets which identifies irregular conditions in complex scenes. In comparison to traditional approaches using optical flow which only measure edge features from two subsequent frames, descriptors have been developing over long-range motion projections called tracklets. Spatiotemporal cuboid footage sequences are statistically gathered on the tracklets that move through them.

Ji et al. developed a system for security footage which automatically identifies the human behavior using convolutional neural nets (CNNs) by forming deep learning model which operates directly on the raw inputs [18].

Therefore, 3D CNN model for classification requires the regularization of outputs with high-level characteristics to increase efficiency and integrating the observations of a variety of various models.

Pang et al. presented real-time concealed various object detection under human dress in [19]. Metallic guns on human skeleton were used for passive millimeter wave imagery which relies on YOLO algorithm on dataset of small scale. Subsequently, comparison is undertaken between Single MultiBox Detector algorithm, YOLOv3-13, SSD-VGG16, and YOLOv3-53 on PMMW dataset. Moreover, the weapon detection accuracy computed 36 frames per second of detection speed and 95% mean average precision. Warsi A et al. have contributed to automatically detecting the handgun in visual surveillance by implementing YOLO V3 algorithm with Faster Region-Based CNN (RCNN) by differentiating the number of false negatives and false positives [20], thus, taking real-time images and incorporating with ImageNet dataset then training it using YOLO V3 algorithm. They have compared Faster RCNN to YOLO V3 using four different videos and as a result YOLO V3 imparted faster speed in real-time environment.

3. Methodology

In this work, we have attempted to develop an integrated framework for reconnaissance security that distinguishes the weapons progressively, if identification is positively true it will caution/brief the security personals to handle the circumstance by arriving at the place of the incident through IP cameras. We propose a model that provides a visionary sense to a machine to identify the unsafe weapon and can also alert the human administrator when a gun or firearm is obvious in the edge. Moreover, we have programmed entryways locking framework when the shooter seems to carry appalling weapon. On the off chance conceivable, through IP webcams we can likewise share the live photo to approach security personals to make the move in meantime. Also, we have constructed the information system for recording all the exercises to convey impact activities in the metropolitan territories for a future crisis. This further ends up in designing the database for recording all the activities in order to take prompt actions for future emergency. Figure 1 presents the overall generalized approach of our research work divided into three parts.

The most important and crucial part of any application is to have a desired and suitable dataset in order to train the machine learning models. Therefore, we manually collected huge amount of images from Google. A few of the image samples are shown in Figure 2. For each weapon class, we collected at least 50 images. Using google-images-download is one of the best ways to collect images for constructing one's own dataset. We further saved those images to a folder called "images." One must save images in ".jpg" form; if the images are in different extensions, it will be a little troublesome and will generate errors when provided for training. Alternatively, since the images are processed in terms of batches, therefore prior to training, the sizes of all the images are transformed into the same width and height 416×416 pixels.

Object detection is primarily related to computer vision that includes distinguishing objects in computerized images. Object detection is a domain that has benefited immensely from the recent advancements in the realm of deep learning. YOLO is basically a pretrained object detector. It is a CNN model. A CNN is a deep learning algorithm which can take in a raw input image and assign learnable weights and biases to various aspects/objects in the image. A convolutional layer in CNN model is responsible of extracting the high-level features such as edges, from the input image. This works by applying kxk filter known as kernel repeatedly over raw image. This further results in activation maps or feature maps. These feature maps are the presence of detected features from the given input. Thus, the preprocessing required is much lower as compared to other classification algorithms, whereas in standard approach, filters are hand-engineered and in CNN these are learned through a number of iterations and training. Figure 3 indicates a basic CNN architecture as classification model for 10 different weapons. Subsequently, the next layer is Max-Pooling or Subsampling layer, which is responsible for reducing the spatial size of the convolved features. This is to decrease the computational power required to process the data through dimensionality reduction. ReLU is a rectified linear unit activation expressed in (1), which is related to the feature of non-saturating activation. It eliminates undesirable values from an activation map effectively by setting them to nil. Finally, the last layers are fully connected layers transforming the data into a 1-dimensional array. To create a particular long feature vector, the flattened output is fed to a feedforward neural network and backpropagation is applied to every iteration of training. These layers are liable to learn nonlinear combinations of the high-level features as represented by the output of the convolutional layer.

$$\text{ReLU:} \quad f(x) = \max(0, x). \tag{1}$$

As mentioned earlier that YOLO is a pretrained object detector, a pretrained model simply means that another dataset has been trained on it. It is extremely time consuming to train a model from scratch; it can take weeks or a month to complete the training step. A pretrained model has already seen tons of objects and knows how each of them must be classified. The weights in the abovementioned pretrained model have been obtained by training the network on COCO and Imagenet dataset. Thus, it can only detect objects belonging to the classes present in the dataset used to train the network. It uses Darknet-53 as the backbone network for feature extraction and uses three scale predictions. The DarkNet-53 is again convolutional neural network that has 53 layers as elucidated in Figure 4. DarkNet-53 is a fully convolutional neural network. Pooling layer is replaced with a convolution operation with stride 2. Furthermore, residual units are applied to avoid the gradient dispersion.

Initially, CNN architectures were quite linear. Recently, numerous variations are introduced, for example, middle

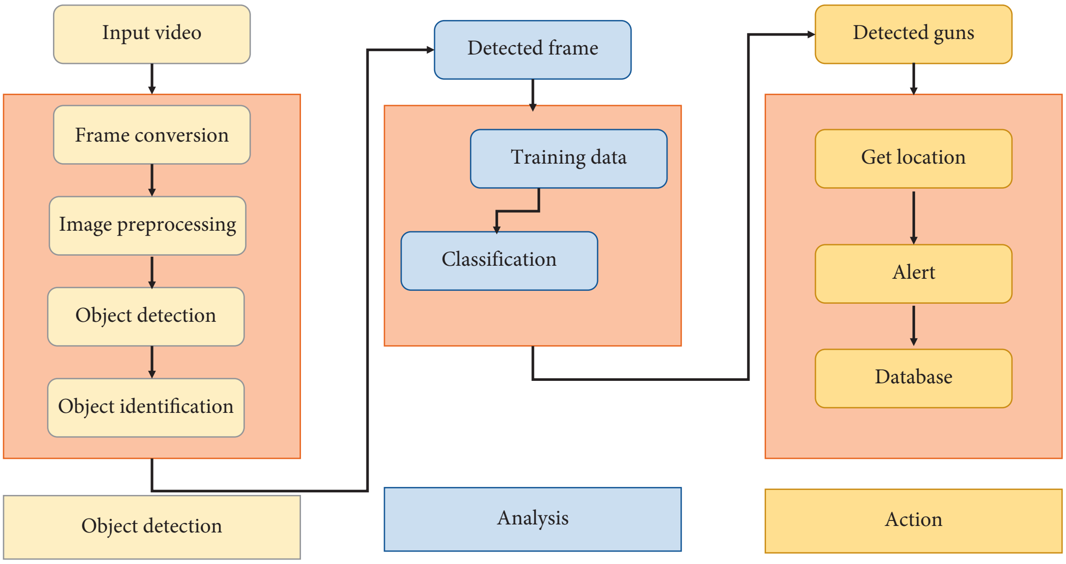


Figure 1: The flow of research methodology.

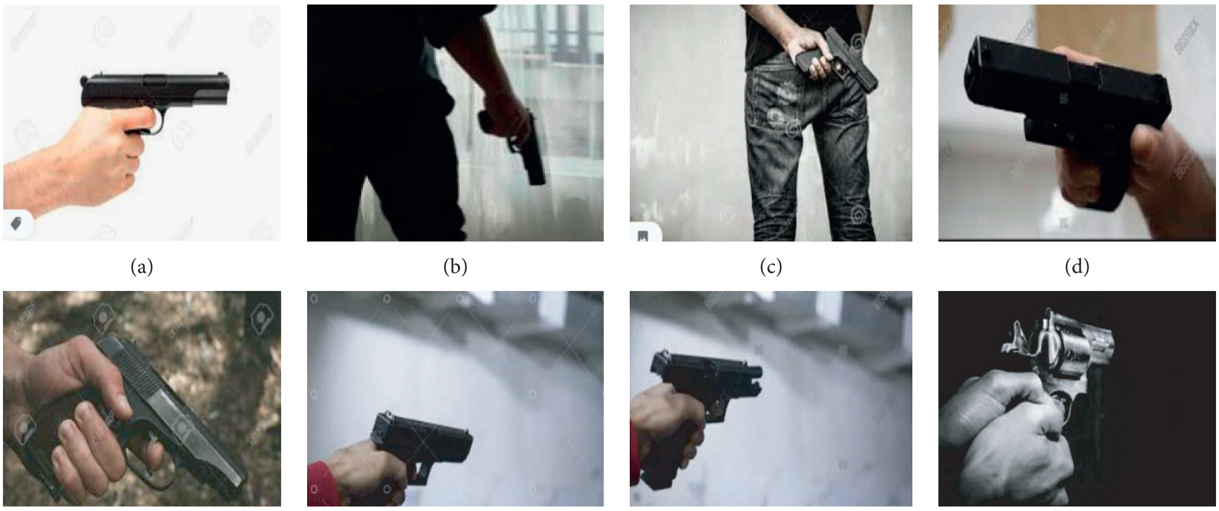

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2: Sample images from collected dataset.

blocks, skip connections, and aggregations of data between layers. These network models have already acquired rich feature representations by getting trained over a wide range of images. Thus, selecting a pretrained network and using it as a starting point to learn a new task is a concept behind transfer learning. In order to recognize the weapons, we took the weights of a pretrained model and trained another YOLO V3 model.

YOLO V3 is designed to be a multiscaled detector rather than image classifier. Therefore, for object detection, classification head is replaced by appending a detection head to this architecture. Henceforth, the output is vector with the bounding box coordinates and probability classes. YOLO V3 inherits Darknet-53 as its backbone, a framework to train neural networks with 53 layers as indicated in Figure 4. Moreover, for object detection task additional 53 layers are stacked over it, accumulating to a total of a 106-layer fully convolutional architecture. Due to its multiscale feature fusion layers, YOLO V3 uses 3 feature maps of different scales for target detection as shown in Figure 5.

4. Experimental Results

Image classification includes, for example, the class of one object in a picture. However, object localization is to recognize the area of at least one article in a picture and drawing a proliferating box around their degree as shown in Figure 6. Moreover, Figure 7 illustrates the detection of rifle from an animated video. The shape of the detection kernel is computed by $1 \times 1 \times (\text{bb x} (4 + 1 + \text{nc}))$. Hence, bb is the number of bounding boxes, "4" is for the 4 bounding box coordinate positions and 1 is object confidence, and nc is the number of

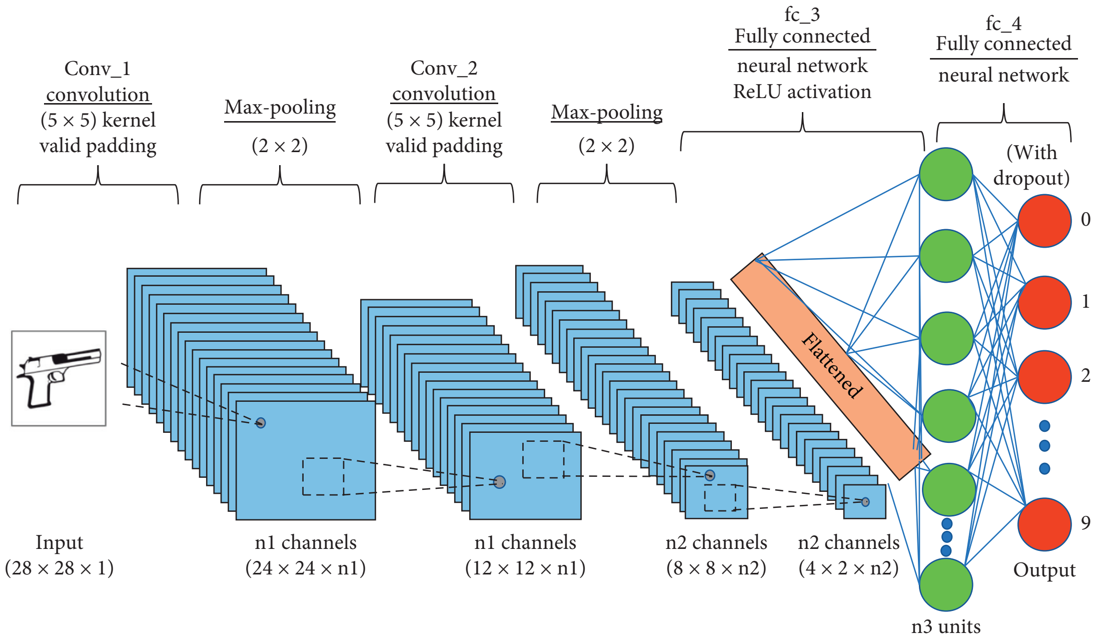


Figure 3: Feedforward convolutional neural network (CNN).

classes. The downsampling of the input image is for three scale predictions and is computed by strides 32, 16, and 8. The loss function over here is comprised on three sections, location error (L_{box}), confidence error (L_{cls}), and classification error (L_{obj}), as presented in (2).

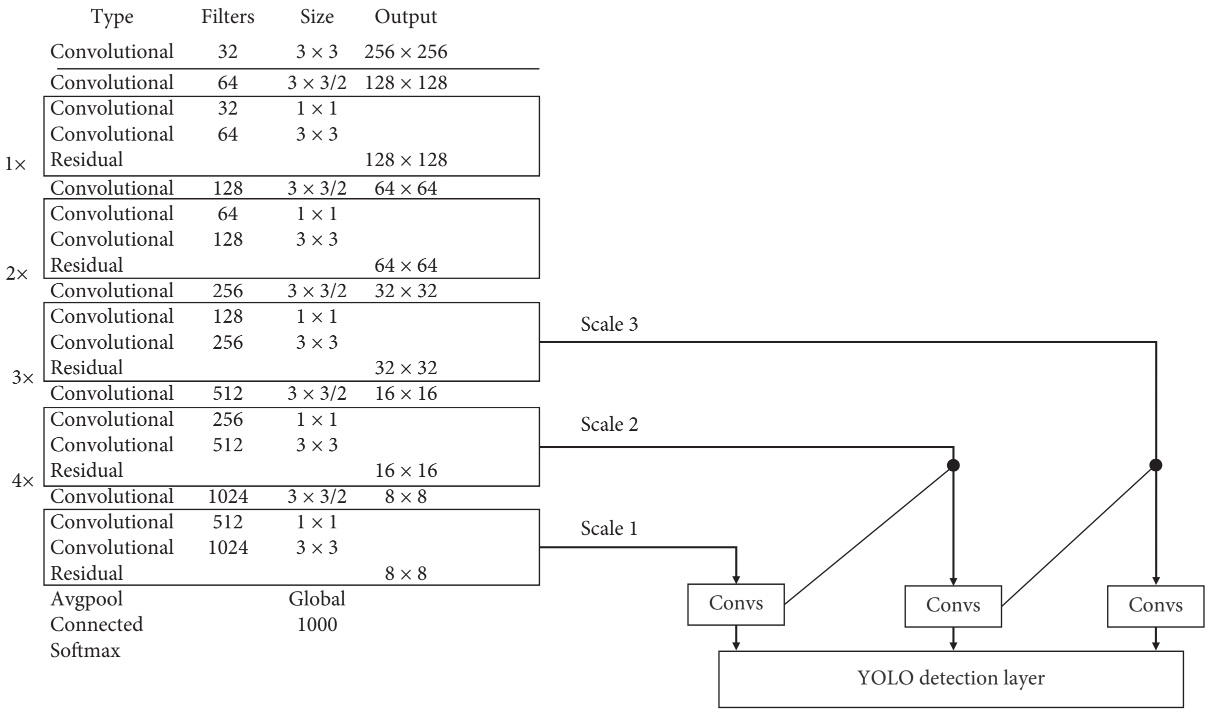
$$\text{Loss} = L_{\text{box}} + L_{\text{cls}} + L_{\text{obj}}. \tag{2}$$

Literature suggests that YOLO v2 often struggled with small object detections. This happened due to loss of fine-grained features as the layers downsampled the input. In conclusion, YOLO v2 applies an identity mapping, concatenating feature maps from a previous layer to capture low-level features. However, YOLO v2's architecture was lacking some of the influential essentials that are encapsulated in most of state-of-the-art algorithms. The early models were lacking in the residual blocks, skip connections, and upsampling. On the other hand, YOLO v3 incorporates all of these. The detection of smaller objects can be seen from cumulative results demonstrated in Figure 8. We retrained both YOLO V2 and YOLO V3. Alternatively, we also conducted comparative analysis of the models with traditional CNN which was trained from the very scratch with null weights. The obtained results are summarized in Table 1.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3/2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3/2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3/2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3/2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3/2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4: Architectural details of DARKNET-53 layers [10].

The subsequent part of our research is based on the recording of location where the weapon was detected so that the alarm is generated. For this purpose, at backend we have also created a Database. A desktop application is also developed in order to provide connectivity with the database system. There are four attributes that are collected from the site where an object like weapon was detected. The collected information needs to be translated into a geographical format of longitude and latitude. For this purpose, geocoding was performed. It is the method of translating addresses to geographical details, longitude, and latitude, to map their positions. As it can be seen from the relational table provided in Figure 9, the attributes are latitude, longitude, time, and location where weapons were seen or identified. At backend DAO (Data Access Object) layer is also available to show the user the data from the database. It is component of Java Foundation Classes (JFC), which is a



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3/2	128 × 128
	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
1×	Residual			128 × 128
	Convolutional	128	3 × 3/2	64 × 64
	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
2×	Residual			64 × 64
	Convolutional	256	3 × 3/2	32 × 32
	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
3×	Residual			32 × 32
	Convolutional	512	3 × 3/2	16 × 16
	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
4×	Residual			16 × 16
	Convolutional	1024	3 × 3/2	8 × 8
	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 5: Architectural description of YOLO V3.

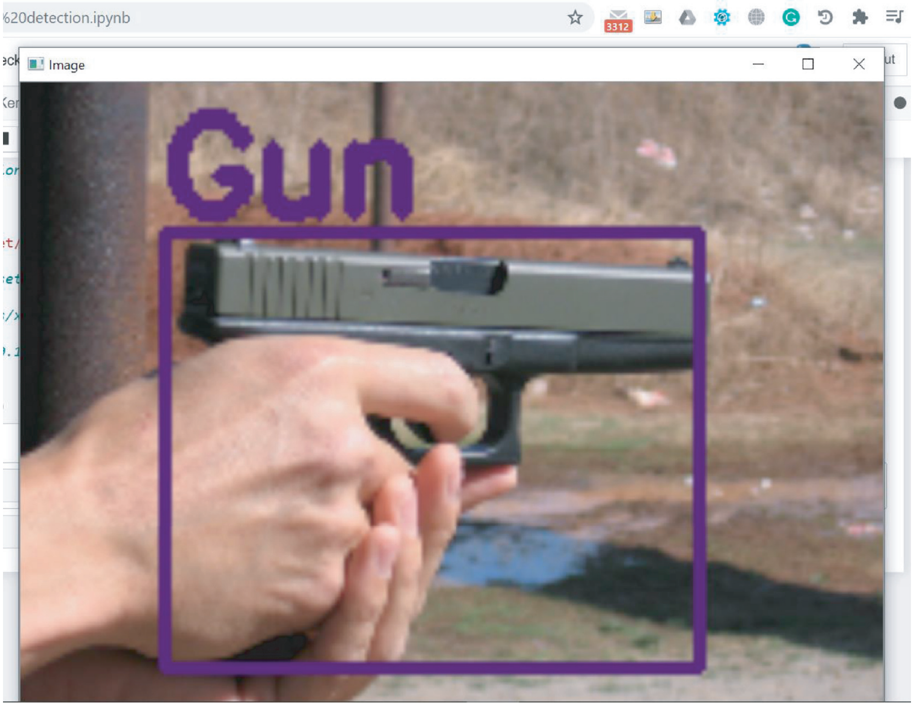


Figure 6: Bounding box around detected object; weapon category GUN.

GUI-providing API for Java programs. Swing provides packages that let us render our Java programs a complex collection of GUI components and it really is platform independent. Figure 10 presents the class diagram and implementation of DAO layer.

Our proposed system is further compared with the existing literature in Table 2. In [21], the proposed system includes CNN-based VGG-16 architecture as feature extractor, followed by state-of-the-art classifiers which are implemented on a standard gun database. The researchers

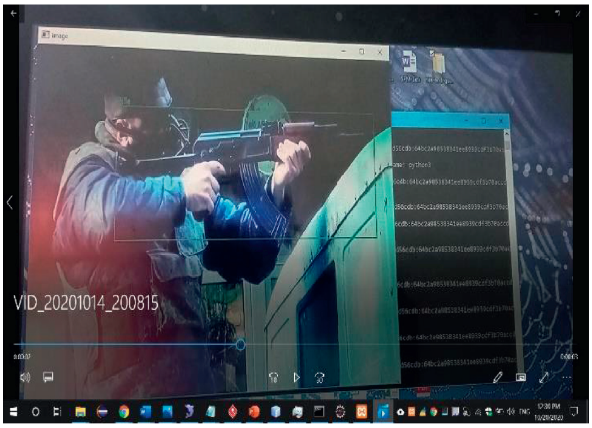


Figure 7: Real-time weapon detected from a video surrounded by bounding box. Weapon category rifle.

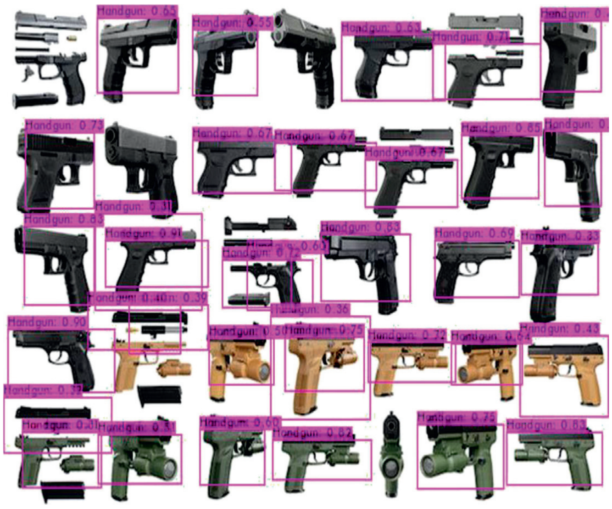
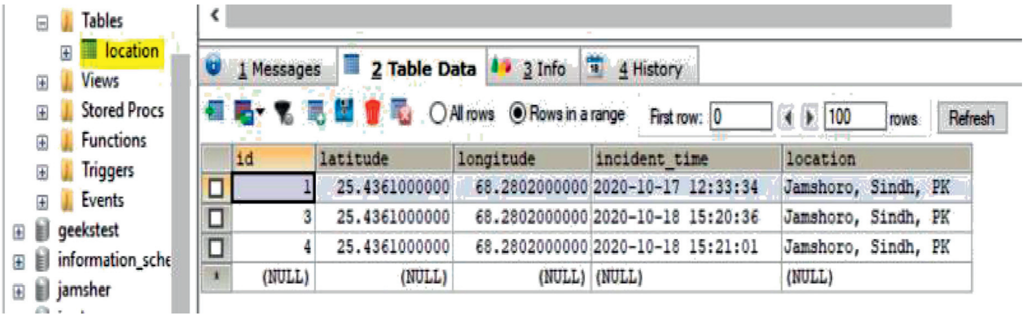


Figure 8: Cumulative result of detecting weapon with precision value.

Table 1: Experimental results for trained deep learning models.

S. no	Models	Accuracy
1	Traditional CNN	95
2	YOLO V2	96.76
3	YOLO V3	98.89



	id	latitude	longitude	incident_time	location
□	1	25.4361000000	68.2802000000	2020-10-17 12:33:34	Jamshoro, Sindh, PK
□	3	25.4361000000	68.2802000000	2020-10-18 15:20:36	Jamshoro, Sindh, PK
□	4	25.4361000000	68.2802000000	2020-10-18 15:21:01	Jamshoro, Sindh, PK
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Figure 9: Image presenting the recorded database.

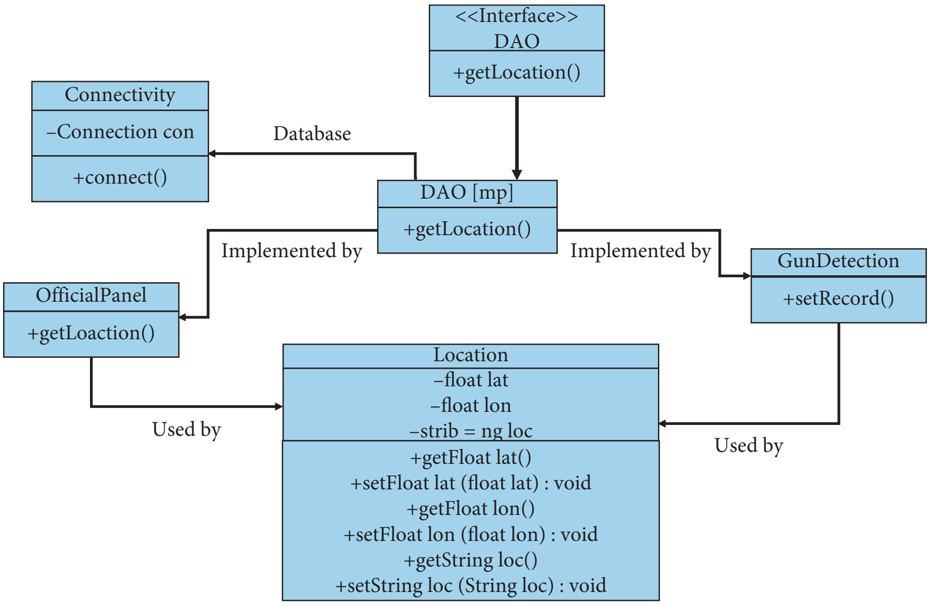


Figure 10: Class diagram for DOA layer.

Table 2: Comparison with the existing work.

S. no	Models	Dataset	Accuracy (%)
1	Our trained model YOLO V3	Image dataset collected for current research	98.89
2	Alexnet + SVM [22]	Gun video database [24]	95
4	Faster RCNN [23]	Streaming video	95.4
5	CNN VGG-16 [21]	IMDB	93.1

investigated four machine learning models, namely, BoW, HOG + SVM, CNN, and Alexnet + SVM, to recognize the firearms and knifes from a dataset of images [22]. Their work suggests that pretrained Alexnet + SVM performed the best. As it is evident from the previous studies, researchers have widely applied CNN and its variant for weapon or knife identification from CCTV videos [23]. It is obvious from Table 2 that the implemented YOLO v3 outperforms the rest of the other models.

5. Conclusion and Future Work

In this study, the state-of-the-art YOLO V3 object detection model was implemented and trained over our collected dataset for weapon detection. We propose a model that provides a visionary sense to a machine or robot to identify the unsafe weapon and can also alert the human administrator when a gun or a firearm is obvious in the edge. The experimental results show that the trained YOLO V3 has better performance compared to the YOLO V2 model and is less expensive computationally. There is an immediate need to update the current surveillance capabilities with improved resources to support monitoring the effectiveness of human operators. Smart surveillance systems would fully replace current infrastructure with the growing availability of low-cost storage, video infrastructure, and better video processing technologies. Eventually, the digital monitoring systems in terms of robots would fully replace current surveillance systems with the growing availability of cheap computing, video infrastructure, high-end technology, and better video processing.

Data Availability

The data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. A. Velastin, B. A. Boghossian, and M. A. Vicencio-Silva, “A motion-based image processing system for detecting potentially dangerous situations in underground railway stations,” *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 2, pp. 96–113, 2006.

[2] United Nations, *Office on Drugs and Crime, Report on “Global Study of Homicide”*, https://www.unodc.org/documents/data-and-analysis/gsh/Booklet1.pdf.

[3] P. M. Kumar, U. Gandhi, R. Varatharajan, G. Manogaran, R. Jidhesh, and T. Vadivel, “Intelligent face recognition and navigation system using neural learning for smart security in internet of things,” *Cluster Computing*, vol. 22, no. S4, pp. 7733–7744, 2019.

[4] V. Babanne, N. S. Mahajan, R. L. Sharma, and P. P. Gargate, “Machine learning based smart surveillance system,” in *Proceedings of the 2019 Third International Conference on*

I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), pp. 84–86, IEEE, Palladam, India, December 2019.

[5] A. Joshi, N. Jagdale, R. Gandhi, and S. Chaudhari, "Smart surveillance system for detection of suspicious behaviour using machine learning," in *Intelligent Computing, Information and Control Systems. ICICCS 2019. Advances in Intelligent Systems and Computing*, A. Pandian, K. Ntalianis, and R. Palanisamy, Eds., vol. 1039, Berlin, Germany, Springer, Cham, 2020.

[6] K.-E. Ko and K.-B. Sim, "Deep convolutional framework for abnormal behavior detection in a smart surveillance system," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 226–234, 2018.

[7] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, "Smart surveillance as an edge network service: from harr-cascade, SVM to a lightweight CNN," in *Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pp. 256–265, Philadelphia, PA, USA, April 2018.

[8] R. Xu, S. Y. Nikouei, Y. Chen et al., "Real-time human objects tracking for smart surveillance at the edge," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, May 2018.

[9] S. Ahmed, A. Ahmed, I. Mansoor, F. Junejo, and A. Saeed, "Output feedback adaptive fractional-order super-twisting sliding mode control of robotic manipulator," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, no. 1, pp. 335–347, 2021.

[10] S. Ahmed, H. Wang, and Y. Tian, "Adaptive fractional high-order terminal sliding mode control for nonlinear robotic manipulator under alternating loads," *Asian Journal of Control*, 2020.

[11] S. Ahmed, H. Wang, and Y. Tian, "Adaptive high-order terminal sliding mode control based on time delay estimation for the robotic manipulators with backlash hysteresis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 1128–1137, 2021.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[13] A. Farhadi and R. Joseph, "Yolov3: an incremental improvement," *Computer Vision and Pattern Recognition*, 2018.

[14] C. He, J. Shao, and J. Sun, "An anomaly-introduced learning method for abnormal event detection," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29573–29588, 2018.

[15] Q. Hu, S. Paisitkriangkrai, C. Shen, A. van den Hengel, and F. Porikli, "Fast detection of multiple objects in traffic scenes with a common detection framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1002–1014, 2015.

[16] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors*, vol. 16, no. 1, p. 47, 2016.

[17] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, "Analyzing tracklets for the detection of abnormal crowd behavior," in *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 148–155, IEEE, Waikoloa, HI, USA, January 2015.

[18] S. Ji, W. Xu, M. Yang, and K. Yu, "3 D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[19] L. Pang, H. Liu, Y. Chen, and J. Miao, "Real-time concealed object detection from passive millimeter wave images based on the YOLOv3 algorithm," *Sensors*, vol. 20, no. 6, p. 1678, 2020.

[20] A. Warsi, M. Abdullah, M. N. Husen, M. Yahya, S. Khan, and N. Jawaid, "Gun detection system using YOLOv3," in *Proceedings of the 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, pp. 1–4, IEEE, Kuala Lumpur, Malaysia, August 2019.

[21] G. K. Verma and A. Dhillon, "A handheld gun detection using faster r-cnn deep learning," in *Proceedings of the 7th International Conference on Computer and Communication Technology*, pp. 84–88, Kurukshetra, Haryana, November 2017.

[22] S. B. Kibria and M. S. Hasan, "An analysis of feature extraction and classification algorithms for dangerous object detection," in *Proceedings of the 2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE)*, pp. 1–4, IEEE, Rajshahi, Bangladesh, December 2017.

[23] A. Castillo, S. Tabik, F. Pérez, R. Olmos, and F. Herrera, "Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning," *Neurocomputing*, vol. 330, pp. 151–161, 2019.

[24] V. Gun, "Database," http://kt.agh.edu.pl/grega/guns/.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9927151, 23 pages
https://doi.org/10.1155/2021/9927151

Review Article

The Importance of Feature Processing in Deep-Learning-Based Condition Monitoring of Motors

Dileep Kumar,[1] **Jawaid Daudpoto**,[2] **Nicholas R. Harris**,[3] **Majid Hussain**,[1] **Sanaullah Mehran**,[1] **Imtiaz Hussain Kalwar**,[4] **Tanweer Hussain**,[1] **and Tayab Din Memon**[5,6]

[1]*National Centre of Robotics and Automation, HHCMS Lab, Mehran University of Engineering & Technology, Jamshoro 76020, Pakistan*
[2]*Department of Mechatronic Engineering, Mehran University of Engineering & Technology, Jamshoro 76020, Pakistan*
[3]*School of Electronics and Computer Science, Southampton University, SO32 1PH, Southampton, UK*
[4]*Department of Electrical Engineering, DHA Suffa University, Karachi, Pakistan*
[5]*Department of Electronic Engineering, Mehran University of Engineering & Technology, Jamshoro 76020, Pakistan*
[6]*School of Information Technology and Engineering (SITE), Melbourne Institute of Technology, Melbourne, Australia*

Correspondence should be addressed to Nicholas R. Harris; nrh@ecs.soton.ac.uk

Received 18 March 2021; Revised 14 April 2021; Accepted 25 April 2021; Published 8 May 2021

Academic Editor: Dao B. Wang

The advent of deep learning (DL) has transformed diagnosis and prognosis techniques in industry. It has allowed tremendous progress in industrial diagnostics, has been playing a pivotal role in maintaining and sustaining Industry 4.0, and is also paving the way for industry 5.0. It has become prevalent in the condition monitoring of industrial subsystems, a prime example being motors. Motors in various applications start deteriorating due to various reasons. Thus, the monitoring of their condition is of prime importance for sustaining the operation and maintaining efficiency. This paper presents a state-of-the-art review of DL-based condition monitoring for motors in terms of input data and feature processing techniques. Particularly, it reviews the application of various input features for the effectiveness of DL models in motor condition monitoring in the sense of what problems are targeted using these feature processing techniques and how they are addressed. Furthermore, it discusses and reviews advances in DL models, DL-based diagnostic methods for motors, hybrid fault diagnostic techniques, points out important open challenges to these models, and signposts the prospective future directions for DL models. This review will assist researchers in identifying research gaps related to feature processing, so that they may effectively contribute toward the implementation of DL models as applied to motor condition monitoring.

1. Introduction

Condition monitoring is described as a continuous process of diagnosis that allows prevention of unintended failure of a system. The basic principle of condition monitoring is to indicate the occurrence of deterioration by taking physical measurements at regular intervals. Subsequently, diagnosis procedures allow the planning of rectification strategies [1–4]. The primary reasons for the application of condition monitoring are increasing availability, prevention of damage, increased reliability, process optimization, increased time between outages, reduced production loss, and better operator information or insights. Its application leads to the development of prognostics, which allows for the estimation of the system's future health and the prediction of the remaining useful life of the system or system's components [5–8]. Motors are the backbone of industry; they start degrading due to different reasons such as long period of operation, variations of power supply, or harsh environment; which gradually lead to permanent damage [9–11]. Consequently, it becomes crucial to monitor the operation continuously.

In the past, there has been extensive research relating to anomaly detection, anomaly severity level detection, and detecting failing elements of the motor. Subsequently, efforts have been made to integrate these diagnosis and prognosis methods, which in turn increases the amount of data. Although condition monitoring system integration improves performance and increases the data volume (providing richer information), it poses different shortcomings such as increased complexity in the information correlating process and increased level of uncertainty [12]. Consequently, this requires novel approaches that can address these shortcomings leading to improved performance. AI-based approaches have been extensively used in the field of condition monitoring for many years [13–16]. With continuous progress in various models of AI, it progressed into machine learning (ML) and subsequently developed into deep learning (DL), which has driven significant impact in the development of modern industry, transportation, medical, and other domains.

The DL-fuelled condition monitoring of motors has set up new horizons in industry 4.0 and has been paving the way for industry 5.0. DL algorithms have impacted almost every area including business [17], medical sciences [18], natural language processing (NLP) [19], robotics [20], transportation [21], the power sector [22], and many other sectors of the modern world. The concept of DL was first coined as "deep belief networks" (DBN) in 2006 and was considered as one of the major breakthroughs in the world of technology [23]. The ability to learn data representation becomes significant with the application of DL models and makes it very attractive in the arena of intelligent diagnosis and prognosis [24, 25]. In comparison to conventional ML models, which can require significant effort in manual feature design and optimization, DL models can automatically extract the representations from the data.

In recent years, DL-based diagnosis and prognosis methods have outperformed conventional machine learning algorithms owing to their generalized nature and many other advantages such as end-to-end implementation, model upgradability, and representation learning using raw data. It does not require human knowledge or intervention in feature designing. Its performance improves with volume of data, but it requires high performance computing hardware such as graphics processing units (GPU) to inherently perform the intense number of necessary matrix multiplication operations. It advocates end-to-end problem-solving rather than dividing the problem into steps. DL-based diagnosis reduces cross-domain discrepancy by learning data representations with multiple levels of abstraction [26–29]. Compared to ML models, DL models can achieve superior performance and their classification accuracies have been tending to 100%. These benefits of DL models have attracted the attention of researchers and they have been extensively applying these models in their domains. Various DL-based condition monitoring methods for motors have been reported by researchers [30, 31]. Meanwhile, extensive adoption of DL methods has enabled industry to progress with better efficiency and sustainability [32]. DL algorithms have efficiently predicted the state of motors in industrial condition monitoring systems with a wide scope. In fact, some models such as GANs can generate data through learning a model of input distribution that is useful in cases when the dataset is small for diagnosis. These algorithms have been in the lime light owing to their merits relating to industry 4.0 and industry 5.0 [27, 33]. DL is reshaping itself through variations in architecture of models, which in turn is reshaping condition monitoring systems by adding more capabilities such as high reorganization accuracy, noise dealing capability, and end-to-end system implementation.

DL-based intelligent diagnosis of various industrial subsystems has witnessed remarkable improvement in performance [28, 34]. However, the disadvantage of DL models lies in feature processing and selection of its parameters such as the learning rate, momentum, and number of neurons in layers. Parameter selection is always a time-consuming and challenging task, which is often achieved through hit-and-trial methods [29, 35]. There have been different reviews and surveys conducted by the researchers relating to condition monitoring of motors and application of DL in this field [11, 36–40]. However, each review has been conducted in different contexts. For example, in [40] authors have surveyed applications of ML and DL in condition monitoring of various machines in the context of vibration data as a key factor for the surveyed studies. Meanwhile, Choudhary et al. [11] have conducted a review of various faults, which occur in induction motors based on various input data types. Authors have also briefly reviewed the techniques used in condition monitoring of induction motors such as Fuzzy Logic, Artificial neural networks, Neuro-Fuzzy inference systems, and support vector machines. They found that noninvasive techniques such as thermal imaging are overcoming the conventional condition monitoring methods. On the other hand, in [41] authors have briefly reviewed various condition monitoring techniques particularly based on current, vibration, and acoustic signals. There is therefore still a need for a comprehensive review on the applications of DL models in condition monitoring of motors. Furthermore, the article presents open challenges and future research directions to promote the application of DL models in engineering scenarios. Compared to the existing reviews, this review focuses on input data and feature-processing techniques used for effective fault diagnosis in the field of DL-based condition monitoring of motors. It surveys and summarizes the recent developments in actual applications of various feature-processing techniques in DL-based condition monitoring of motors. The contribution of this article can be summarized as follows:

(i) Provides an integrated overview of current trends and consolidates the recent work of various researchers related to feature processing in DL-based condition monitoring of motors

(ii) Endeavours to provide an in-depth analysis approach and a valuable road map to engineers and researchers working in this field, which may assist them in signposting the direction of future research

(iii) Presents the merits and limitations of DL for motor condition monitoring methods based on state-of-the-art contributions reported in the literature

The remainder of this article is constructed as follows. Section 2 reviews feature processing methods used with DL models in condition monitoring of motors, and the techniques used to resolve problems posed by such methods, Section 3 summarizes and discusses performance aspects of models, highlights challenges related to DL, and presents future directions of this field, and Section 4 provides concluding remarks on this review article.

2. Feature Processing for DL-Based Condition Monitoring of Motors

In DL-based motor condition monitoring, data are acquired using various sensors and stored in a database. Typically, vibration, acoustic emission, current and temperature signals are used to monitor the condition of a machine [42]. Each diagnostic method has different capabilities for detecting various types of faults in motors. In the next step, data are preprocessed, and a model is built. Subsequently, the model is trained on data. The end goals or targets are typically defined as fault detection, failure prediction, and remaining useful life (RUL) estimation but are not limited to these cases. Figure 1 illustrates the generalized concept of the DL model pipeline and its comparison with the machine learning (ML) model pipeline. Feature processing is applied to extract buried features in noisy input data through feature extraction and feature selection techniques. After performing feature processing, data are fed to various ML models such as support vector machine (SVM), decision trees (DT), random forest (RF), and K-nearest neighbour (KNN). Moreover, feature processing assists in identifying and removing outliers and redundant information from the dataset and converting raw data to more manageable groups for processing. In addition, it reduces the dimensionality of large datasets, which in turn speeds up the learning process [43–45]. Feature processing, which includes both feature extraction and selection techniques, is very important for deploying reliable ML-based solutions for industry. It not only simplifies the entire process of tackling a domain-specific problem but also makes the process understandable for human experts. Reducing the number of features, or enabling the selection of useful ones, greatly reduces the hardware dependence and the need of highly nonlinear function mapping by the ML models.

Feature extraction techniques include: (i) time-domain techniques, such as statistical parameters, including mean, standard deviation, root mean square (RMS), covariance, kurtosis, and skewness. (ii) Frequency domain features, including fast Fourier transform (FFT), spectral kurtosis, root variance frequency, and mean frequency. (iii) Time-frequency domain features, including wavelet transform (WT), short-time Fourier transform (STFT), Hilbert-Huang transform (HHT), Hilbert transform (HT), and empirical model decomposition (EMD). These techniques basically provide DL models with meaningful representation of raw data and aid in the simplification of decision boundary definition identified by classification models. In the case of regression-based models, these techniques help lower the order of function to be mapped by DL models, which result in computational efficiency and robust real-time deployment. Popular feature selection techniques are principal component analysis (PCA), linear discriminant analysis (LDA), and independent component analysis (ICA) [41, 46]. These techniques are particularly known for their ability to reduce the dimensionality of the dataset. PCA produces an estimated representation of a bulk of features in a concise manner and reduces the feature space. This helps not only in optimization but also aids in enhancing the reliability and explainability of DL models by visualizing the feature space possible.

DL models can learn representations from raw input data and can process accordingly for supervised or unsupervised tasks as shown in Figure 1. Generally, it does not involve steps such as feature extraction and feature selection. Hence, DL-based algorithms have been extensively used in the field of condition monitoring of motors. The various architectures of DL that have been used in motor condition monitoring include deep belief networks (DBN), multilayer perceptron (MLP), autoencoders (AE), deep Boltzmann machine (DBN), convolutional neural networks (CNNs), recurrent neural networks (RNN), and generative adversarial neural networks (GANs). Each of these architectures was developed keeping in mind the ways data could be presented to them. For example, an MLP is designed to learn from tabular data, a CNN is traditionally used to map a 2D field (which in most cases is an image) to an output variable, RNN in the same way was designed to handle sequences of data, making it an attractive paradigm for time-series forecasting or speech-processing tasks. However, these architectures do allow a certain level of flexibility and have been used for alternative tasks. For example, an image could be broken into a sequence of pixels and presented as tabular data to an MLP, but this comes at the cost of explainability. Visualization of a row of image pixels into feature space would be incomprehensible. This lays more emphasis on feature-processing techniques to be used along with a given architecture because proper feature formation is necessary for even considering the trade-off between accuracy and explainability of the models. Researchers have put efforts for optimizing the performance of these methods through different feature-processing techniques which play a key role in the DL-based condition monitoring of motors. In this section, each of these techniques is reviewed with respect to problems, which are addressed by these methods, and how successful they were at the task.

2.1. Multilayer Perceptron (MLP).

2.1. Multilayer Perceptron (MLP). MLP is one of the most utilised DL model topologies, and also one of the oldest. It is a fully connected neural network consisting of one or more hidden layers. It can easily perform classification tasks with its simple structure. However, it becomes difficult to train the model as tasks become complex owing to the increase in

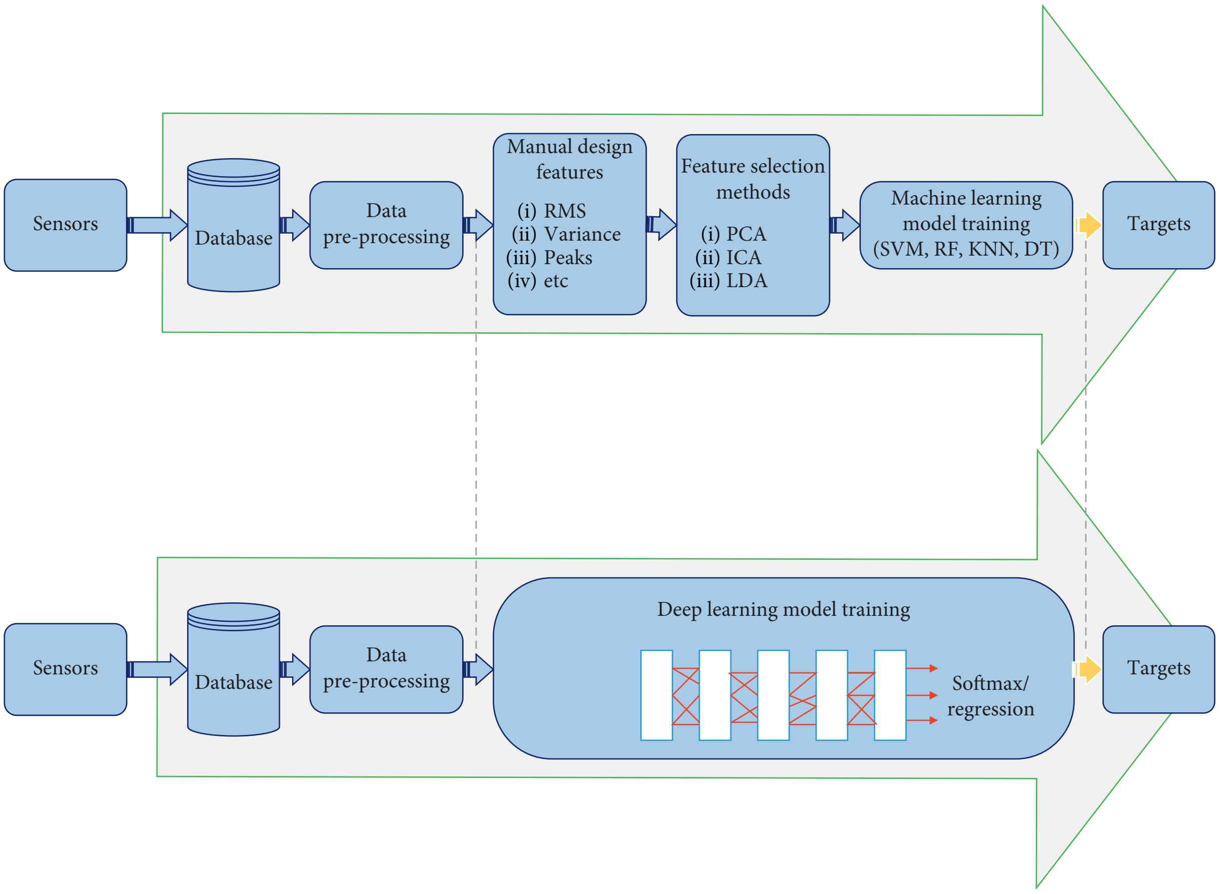


Figure 1: Comparison between ML and DL pipelines.

datasets. It is trained through a supervised learning approach called back-propagation (BP). Figure 2 shows the basic structure of the MLP model.

The MLP has been used in motor condition monitoring applications for a long time owing to its easy training process. Various authors have employed it for motor fault classification tasks. Vieira et al. [47] have employed multiple MLPs with single layer to classify stator winding short-circuit faults of the induction motor using the current data. The dataset was collected by varying the operational conditions of the motor/load/converter. The authors have investigated multiple hypotheses related to the conditions and their effects on the classifier's accuracy. The MLP model was able to achieve 92.6% accuracy at zero load and 76.9% accuracy at full load. It was observed in the results that variations in the operating frequency did not affect the classifier's accuracy. However, it was harder to correctly classify the faults with low severity, which led to the observation that the accuracy of the classifier increases as the fault severity level increases.

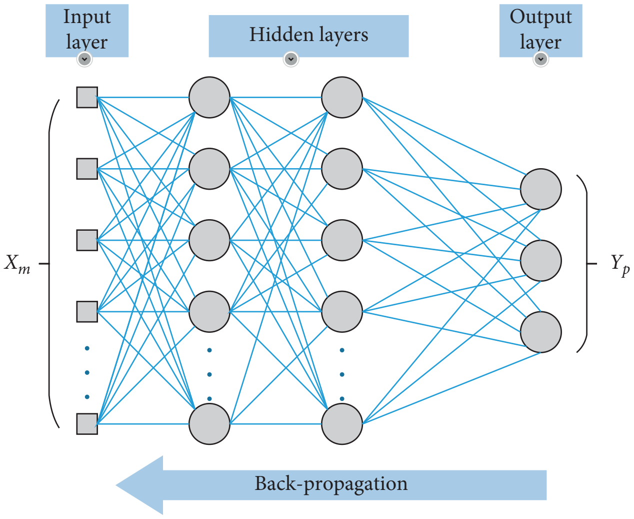


Figure 2: MLP structure.

Palacios et al. [48] have used MLP to carry out motor fault classification using current and voltage amplitudes in induction motors. The authors have used discretized time domain signals for classifying faults such as rotor broken-bar fault (1/2/3/4 broken-bars), bearing faults (inner race grooves, outer race grooves, and rolling element defect), and stator short-circuit fault. Results confirmed a better performance of the method, which achieved 94.6% accuracy compared to other conventional methods including SVM and KNN. It was also observed that the proposed method was computationally more efficient than the comparative methods. In [49], authors have used MLP with mutual information (MI) for fault classification of the induction motor. MI was employed for feature extraction, and it describes the similarity between time-series data simultaneously acquired from the setup. The method was applied to detect the stator winding short-circuit fault using the features extracted from the current signature data. Experimental results confirmed that the MLP performed better than the radial basis function (RBF) in terms of classifying healthy and faulty conditions of the motor with accuracy over 99%, whereas RBF achieved 96–97% accuracy.

Zolfaghari et al. [50] have used an MLP-based classifier for broken bar fault detection. They have input motor current data after performing FFT and wavelet packet transform, for superior feature extraction, which in turn reduced the burden on the classifier. Furthermore, they have extracted WP-based statistical parameters for intelligent broken-bar severity level detection (one/two/three broken bars). The investigation results confirmed that the method yields promising results in terms of rotor broken-bar severity-level detection, even in the no-load condition with accuracy of 98.8%. In [51], authors have used an MLP model for real-time fault classification of the induction motor. Authors have employed a nonlinear manifold technique called curvilinear component-based analysis (CCA) for feature extraction. The method was employed to classify two major faults: stator interturn fault and rotor broken-bar fault. Results revealed that the MLP model effectively classified the faults in the induction motor with an accuracy of 95.2%.

Bazan et al. [52] used an MLP model with the MI feature extraction method to classify induction motor bearing faults. MI shows the reduction in uncertainty associated with one random variable when combined with information from another variable in simultaneously acquired time-series data. Features were extracted from multiple current signatures under different operating conditions. The results confirmed the robustness of the method in the bearing fault classification compared to the conventional methods, including SVM and KNN. The MLP network achieved 90.5% accuracy with 10% voltage unbalance condition compared to SVM and KNN, which achieved 84% and 83.3%, respectively. Authors have suggested future research work in terms of investigating the real-time implementation of the approach and the potential accuracy trade-off with the speed of the motor. Recent applications of MLP for condition monitoring of motors are summarized in Table 1.

2.2. Autoencoders (AE).

AEs are frequently employed as an unsupervised feature extraction technique. They have the capability of reducing the dimensions of input data whilst retaining most of the information. Figure 3 illustrates a deep AE structure and its operation. It has two blocks: the first block is the AE encoder and the second block is the decoder. The encoder network generates low-dimension representations while the decoder network reconstructs the input data from these low-dimension representations. In addition, it uses the reconstruction error as a loss function. Deep AE is initially trained using an unsupervised method that is known as pretraining. The pretraining process allows better convergence as it reduces overfitting and optimises the layers. The process of pretraining of a deep AE model is explained as follows:

(i) A single layer extracts an initial parameter for the following hidden layer and predicts itself using the input vector. Through this technique, it learns about the data without any feedback or labeled data. Subsequently, it stores the learned features as weights for the hidden layers of the network.

(ii) Similarly, the following layer learns about features for succeeding hidden layers and the process is continuous for all the remaining layers.

(iii) Eventually, learned information reaches the output layer typically through a softmax activation function to give a probability distribution.

After this pretraining process, both the networks can be trained simultaneously for classification. Many researchers have argued that AE can be trained without pretraining, but this requires some additional effort in training process or initialization [53].

The recent developments in AE by researchers have presented various derivatives of AE for 1-D and 2-D data such as denoising autoencoders (DAE) [54], variational autoencoders (VAE) [55], and sparse autoencoders (SAE) [56]. Although DAE and SAE have the same structure as a simple AE, they differ in loss function and inputs. Some research has been carried out to improve the performance of AEs using different techniques and have been employed for diagnosis and prognosis of motors. AE and its derivatives have been used in motor condition monitoring with different goals. Sun et al. [57] have employed SAE with DNN for unsupervised feature extraction. The model was fed with vibration data to classify different faults of the induction motor. They employed the "dropout" regularization method to avoid overfitting during the training process. The SAE remained inactive during the testing process. The results confirmed that the approach provided better performance with 97.6% accuracy compared to conventional models such as SVM and linear regression (LR), which achieved 96.4% and 92.7% accuracy, respectively. Liu et al. [58] have used STFT and SAE to extract features from the sound/acoustic emissions signals of a rolling bearing. Considering barriers to feature extraction in the DNN, they have used STFT for fast and effective feature extraction. This enabled fault classification with fewer training data samples and resulted in increased classifier accuracy of the DNN compared to the model without STFT features. The SAE model with the STFT-based input achieved average accuracy of 96.2%.

In [59], authors have performed bearing fault classification using AE and extreme learning machines (ELM). ELM is employed with AE owing to its advantage of high training speed. No explicit feature extraction technique was involved with the employed method. The AE-ELM model input data were frequency features extracted from vibration data. This approach yielded better classification performance

Table 1: Application of MLP in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2016	Current signature	MLP [47]	Induction motor fault classification	Operating frequency of the motor did not affect the accuracy of the classifier. However, increase in the accuracy was observed as severity of the fault increased
2017	Current signature and voltage signal amplitudes	MLP [48]	Induction motor fault classification	The method yielded better classification results and it was computationally efficient compared to the traditional methods
	Current signature + MI	MLP [49]	Induction motor fault classification	MI feature extraction method allowed MLP to produce promising results compared to RBF
2018	Current signature + FFT + WPT + statistical parameters	MLP [50]	Rotor broken bar fault classification	Superior feature extraction allowed to detect the fault severity level even with no-load condition
	Current signature + CCA	MLP [51]	Induction motor fault classification	The method allowed to correctly classify the faults with higher accuracy owing to reduced dimensions of the data by CCA
2019	Current signature + MI	MLP [52]	Bearing fault classification	MI feature extraction method allowed to classify the bearing faults more effectively compared to conventional methods

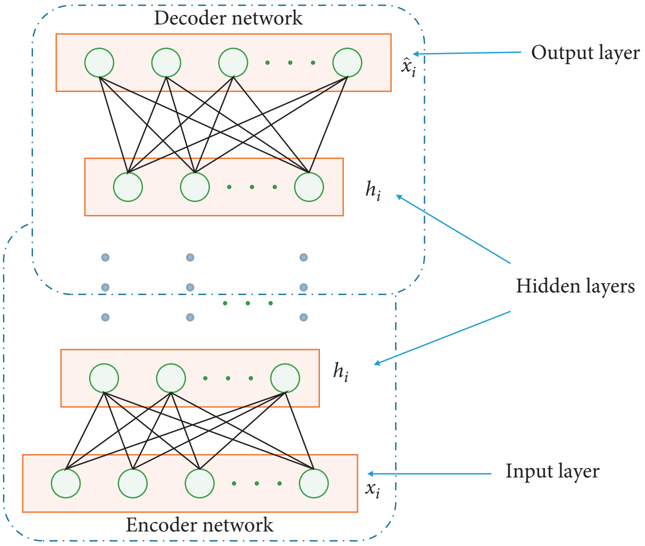


Figure 3: Deep AE structure.

than SAE and results also revealed that it was faster than SAEs. Both the AE-ELM and SAE had an accuracy of 100%. However, the SAE model had a slower response time of 20s in the online learning compared to AE-ELM, which took 0.6s. Thus, it was concluded that AE-ELM can be applied for real-time fault diagnosis owing to its faster response and higher accuracy. Chen et al. [60] have used deep SAE with noise added vibration data for rolling bearing fault severity level classification and life stage prediction. The input data along with added noise were fed to the deep SAE, constraining the overfitting problem caused by limited training data. They have used the two different vibration datasets for classifying different bearing severity levels and life stages. The investigation was also carried out with and without noise data. The comparative analysis between the deep SAE with and without noise revealed that the model with noisy data effectively overcame the overfitting problem and achieved 98.3% accuracy, whereas deep SAE without noisy data achieved 93.7% accuracy. The authors in [61] employed stacked SAE with hybrid features for feature extraction and DNN for classification of bearing fault severity level. This approach allowed extracting more discriminative information, which in turn raised the classification accuracy. In addition, the input to the DAE was a combination of time domain features, wavelet energy features, and power spectrum features extracted from the vibration data. This hybrid feature pool was used to overcome nonlinearity in the data. The results of the investigation revealed that the proposed method outperformed the vibration spectral-imaging-based DNN method with an accuracy of 99.1%. Lu et al. [62] have used a stacked denoising autoencoder (SDA) for rolling bearing fault classification using raw vibration data. They have compared the performance of SDA with SAE by inputting the data with varying signal-to-noise ratio (SNR) values. From the comparative investigation results, it was observed that the SDA outperformed the SAE, SVM, RF, and AE with maximum accuracy of 99.8% at the SNR of 20 dB owing to its learning capacity for complex and nonlinear mapping relationships. Sun et al. [63] have employed SAE and DNN for rolling bearing fault classification. They have input vibration data to the model after compressing it. Data compression was used for handling the significant amount of data more efficiently. The investigation of the method confirmed the higher classification accuracy of 97.4% with data compression compared to the DNN model without data compression, which achieved 96.7% accuracy.

Shao et al. [64] have used ensemble deep AEs for the classification of various roller bearing faults. They used 15 standard AEs, each with different activation functions. This approach overcame the manual feature extraction and limitations of the individual model. These AEs were used for extracting features from vibration data. The diagnosis results revealed that the proposed method performed more robustly and produced more effective classification results than the individual AE models with different activation functions, DBN, and CNN, with an accuracy of 97.1%. In [65], authors have used an unsupervised deep AE to predict the bearing decay state in an inverter-fed induction motor. They input stator current data as a compact representation of the bearing state. The acquired data of the artificially introduced bearing faults with varying loads was segmented using a sliding window of 24s. Subsequently, they have extracted time and frequency domain features and then fed them to the models. The investigation results demonstrated that the deep AE produced clearer clusters of different bearing conditions with higher classification accuracy than a shallow neural network (SNN).

In [66], the authors have used ensemble stacked autoencoders (ESAE) for bearing fault classification. They have applied FFT to raw vibration data and fed to the model. The proposed method was compared with time-domain features' extraction methods. The comparative results confirmed the superior performance of the ESAE with FFT technique compared to the ESAE with other feature extraction techniques such as average, kurtosis, and variance with a minimum mean absolute error (MAE) of 0.0672. Skylvik et al. [67] have employed deep AE to classify different faults in induction motors such as interturn faults, bearing faults, and broken rotor bar faults. For preprocessing, they employed Welch's method to estimate spectral density. Subsequently, they have applied FFT to current data and then fed that to the model. The performance comparison with SVM and KNN models confirmed that the proposed scheme performed much better than the traditional schemes with an accuracy of 96.1%. Zhao et al. [68] have constructed an optimal hybrid DL model, which consists of SAE and GRU, to classify the rolling bearing faults more accurately. In addition, they have used the gray-wolf optimising algorithm to enhance the performance of the model. The proposed model was capable of extracting features from raw vibration data. The results confirmed that the model achieved robust and effective results with an accuracy of 97.1%, which is higher than other traditional models, including ANN, SVM, SAE, and gated recurrent units (GRU).

Zhu et al. [69] have presented a novel DL model called stacked pruning denoising autoencoder (SPDAE) for rolling bearing fault diagnosis. The model reduces information loss by introducing new channels to interconnect the layers. The pruning operation was carried out on nonsuperior units in the model to reduce the number of training parameters, which in turn enhanced training efficiency and precision. To ensure the uniqueness of the dimensions extracted from SPDAE, the features from the same layer were fused. Raw vibration data with noise was fed to the model for the bearing fault classification. Comparative results confirmed the superiority of the proposed model compared with the conventional models, such as ANN, SAE, and DAE. The SPDAE model achieved 100% accuracy on the bearing dataset. The authors have suggested future studies investigating noise addition, pruning operations, and feature fusion operations, which may improve the performance of the model. State-of-the-art research related to condition monitoring of motors using AEs is summarized in Table 2.

2.3. Restricted Boltzmann Machines (RBM). The RBM are Markov-based special-type models, consisting of two layer neural networks [70]. They employ an unsupervised training method based on a greedy layer-wise process. They have the capability to learn missing data patterns. RBM-based networks pose difficulty in the training process and also in tracking the loss function. There are two types of RBM-based DL models, which are Deep Boltzmann Machine (DBM) and Deep Belief Network (DBN). DBN is a semidirected model, while DBM is a completely undirected graph model, as shown in Figure 4(b). The RBM-based DL models are reported as follows:

2.3.1. Deep Boltzmann Machine (DBM). The DBM can be considered as a stacked RBM, which comprises multiple hidden and visible layers rather than only RBM layers as shown in Figure 4(a), where the blue dotted line shows the separation of the layers [71]. Each layer is composed of symmetrically coupled stochastic units. DBM has the capacity to learn features from complex data and capture higher-order correlations among the hidden features. The DBM model is trained as a combined model, in comparison to the other type of RBM, a deep belief network (DBN) (Section 2.3.2), which is trained layer by layer [72]. Hence, the training process of the DBM requires more computational power than for DBN models.

2.3.2. Deep Belief Networks (DBN). Like DBM, DBN is formed by stacking RBM layers in such a way that output of the n-th layer becomes input to the $(n+1)$-th layer. Figure 4(b) shows the structure of the DBN. It is a mixed directed and undirected graphical model. It can process large amount of nonlinear data [73]. As an RBM, it is also trained in a greedy layer-wise unsupervised fashion [74]. Fine-tuning is required after the pretraining process, which is performed either on training data label or on a proxy for DBN log-likelihood. Targets are achieved by adding a softmax layer on top of the DBN architecture.

Both the DBN and DBM have been used in various condition monitoring systems for motors. In [75], the authors have employed DBM for roller bearing fault classification. Three types of features (time-domain features, frequency-domain features, hybrid features) were extracted from vibration data and fed to the model. Among these features, hybrid features yielded the better classification performance with an accuracy of 99.5%. Tao et al. [76] have used DBN and a multisensor information fusion technique for bearing fault detection. They have inputted time-domain features from multiple vibration sensors to the model. It was observed in the results that DBN not only adaptively fused the multisensory data but also increased the accuracy up to 97.5%, that is, 10% greater than that of the model with single sensor data. The DBM model was able to achieve highest accuracy among the comparative models, including SVM, KNN, and ANN. In [77], authors have classified bearing degradation states using DBN and the Weibull distribution. Bearing degradation states were classified based on fitted RMS of the vibration data by Weibull distribution, which

Table 2: Application of AE in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2016	Raw vibration	SAE + DNN [57]	Motor fault classification	Integrated "dropout" regularization allowed to avoid overfitting, which in turn improved the classification performance
	Acoustic emissions/sound signals + STFT	SAE [58]	Rolling bearing fault classification	STFT allowed for fast and effective feature extraction, which in turn increased accuracy of the classifier
2017	Vibration + FFT	AE + ELM [59]	Bearing fault classification	AE-ELM approach increased the speed of the training process. The approach was much faster than SAE. However, the model did not perform well for online sequential learning
	Raw vibration	SAE [60]	Rolling bearing fault severity level classification	Deep SAE inputted with added noise vibration data has effectively overcome the overfitting problem posed by small training datasets
	Vibration + time domain features + wavelet energy features + power spectrum features	SAE + DNN [61]	Rolling bearing fault severity level classification	Hybrid features pool was able to tackle the nonlinearity in the vibration data and resulted in more effective classification results
	Raw vibration	DAE and SAE [62]	Rolling bearing fault classification	The comparative analysis performed between DAE and SAE classifiers showed that the DAE outperformed SAE classifier due to its potentiality of learning complex nonlinear mapping relationships
	Vibration + compression	SAE + DNN [63]	Rolling bearing fault classification	Data compression technique with SAE enables effective fault classification, with huge datasets through an easy approach
2018	Raw vibration	Ensemble AE [64]	Rolling bearing fault classification	A novel ensemble of 15 AEs, each with different activation function overcome the limitations of individual AE models and removed manual feature extraction
	Current + time domain and frequency domain features	Deep AE [65]	Unsupervised bearing fault prediction	Results showed effectiveness of the technique in terms bearing fault prediction with clear clustering and high accuracy
2019	Vibration + FFT	ESAE [66]	Rolling bearing fault classification	FFT with ESAE has revealed superior performance of the proposed model in comparison with traditional models
	Current + FFT	AE [67]	Induction motor fault classification	FFT with the AE yielded superior performance compared to the existing models
	Raw vibration	SAE + GRU [68]	Rolling bearing fault classification	An optimal hybrid DL mode constructed using SAE and GRU can extract rich features from raw vibration data. Results confirmed the superiority of the proposed model
2020	Raw vibration	Stacked pruning DAE [69]	Rolling bearing fault classification	A novel model called SPADE with pruning operation increased efficiency and precision of the model by decreasing the training amount of the model

was employed to avoid fluctuation in statistical parameters. The results confirmed the effectiveness of the technique through run-to-failure experiment. The model effectively classified the bearing degradation states with an accuracy of 89.9%.

Shao et al. [21] have used a DBN model for the fault classification of induction motors in manufacturing. Different classes of the motor faults included normal motor, unbalanced rotor, stator winding defect, defective bearing, bowed rotor, and broken bar. They utilised frequency domain features extracted from vibration data investigated the effect of the depth of the model on the classification accuracy. The investigation results confirmed the effectiveness of the method for automatic fault classification in manufacturing. In [78], the authors have investigated an improved version of the DBM by varying its energy function. The proposed method addressed the problem of the DBM, which can only process the binary data; by replacing the binary visible units with the Gaussian units, the variation allowed the DBM to process real-value data. This model was employed for bearing fault classification using raw vibration data. Supervised training was performed followed by greedy unsupervised training to initialize the model parameters. Results confirmed the effectiveness of the method in fault classification using real-value data with manual feature extraction with a classification error of 1.9%.

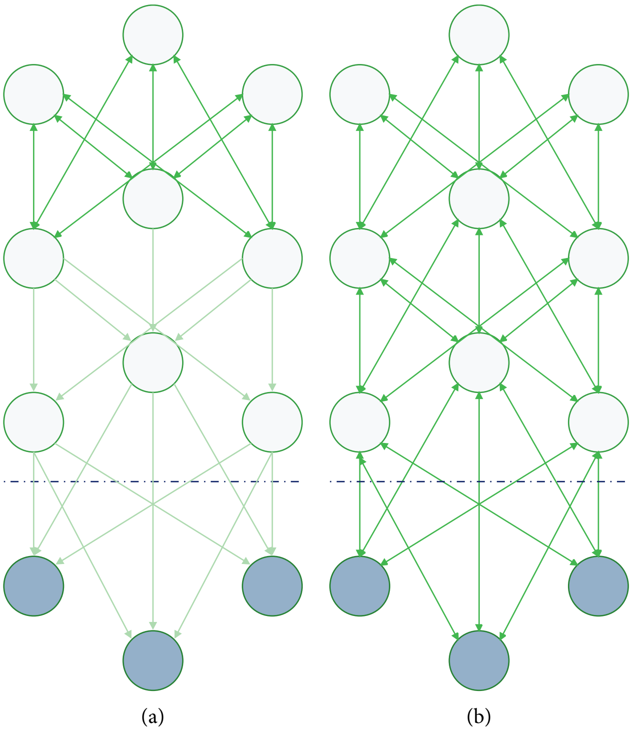


Figure 4: (a) Deep belief network. (b) Deep Boltzmann machine.

In [79], researchers have presented a method to classify rolling bearing faults using DBM, principal component analysis (PCA), and a least square support vector machine (LS-SVM). DBM was used for feature extraction, PCA was used to reduce the dimensionality of the data, and the LS-SVM was applied for classifying the bearing faults. Acoustic emission signals were used as input to the model owing to its higher sensitivity compared to vibration. The combination of DBM and PCA identified better features, which in turn increased classification accuracy of the model. Experimental results confirmed the effectiveness of the model with an accuracy of 95.4%. Zhao el al. [80] presented a variation mode decomposition- (VMD-) and Hilbert transform- (HT-) based DBN (VHDBN) for rolling bearing fault classification. Bearing vibration signals are decomposed into intrinsic mode functions (IMFs) through VMD. Subsequently, HT extracted the instantaneous frequency and amplitude of the IMFs and constructed a feature matrix that was fed to the DBN model. A combination of VMD and HT allowed the extraction of improved features and achieved diagnosis accuracy of 98%. The investigation results confirmed that the VHDBN algorithm has great advantage over conventional algorithms such as SVM and the DBN with time domain signals.

In [81], the authors have presented a novel architecture-multiscale cascaded DBN (MCDBN) for automatic fault classification in motors. This variation to DBN added parallel learning capability by introducing a multiscale coarse-grained method, which in turn improved the feature extraction performance. The vibration signal was split into subsignals of equal window size using sliding window with data overlap. The technique allowed DBN to learn features from the data at multiscale rather than inherent information. Subsequently, coarse-grained time-series data at different scales was obtained through the coarse-grained process. Experimental results confirmed superiority of the method compared to standard DBM with an accuracy of 99.8%. The authors have suggested that the investigation has provided a promising tool for condition monitoring of industrial motors. Yu et al. [82] have used a two-stage approach by combining DBN and the Dempster–Shafer (D-S) theory for bearing faults and their severity level classification. D-S theory was employed for fusing vibration data from multiple sensors (horizontal and vertical vibration). A genetic algorithm (GA) and a particle swarm optimization- (PSO-) based hybrid algorithm was used to optimize the parameters of DBN during the training. The results revealed that the fused hybrid GA-PSO algorithm has not only upgraded capabilities of the DBN classifier but also enhanced computational efficiency with an overall accuracy of 99.6%. Authors have also used wavelet package decomposition (WPD) for extracting energy features from the vibration data of bearings. Table 3 summarizes applications of DBN and DBM in condition monitoring of motors.

2.4. Convolutional Neural Networks (CNN). A neural network layer that employs a convolution operation includes convolution layers in its structure. Figure 5 illustrates the structure of a CNN. It consists of two blocks; the first block comprises convolution layers and pooling layers that extract features from the data. Multiple stacks of convolutional layer

Table 3: Application of DBN and DBM in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2016	Vibration + hybrid features	DBM [75]	Rolling bearing fault classification	Among three features, hybrid features allowed the model to perform much better than other models
	Multisensor vibration + time domain features	DBN [76]	Bearing fault classification	Multivibrations fusion technique with DBN outperformed the model with single sensor
	Vibration + time domain features	DBN [77]	Bearing degradation states classification	Weibull distribution dealt with fluctuations in statistical features, which in turn enhanced the ability to classify the bearing degradation states
2017	Vibration + FFT	DBN [21]	Induction motor fault classification	The developed automatic fault classifier effectively detected faults of motors in manufacturing
2018	Raw vibration	DBM [78]	Rolling bearing fault classification	Binary units of DBM were replaced with Gaussian units so that DBM can process real-value data. Results confirmed the usefulness of the method
2019	Acoustic emission signals	DBN + PCA + ls-SVM [79]	Rolling bearing fault classification	Combination of DBM and PCA enabled for learning better features, which in turn increased the accuracy of the model
	Vibration + VMD + HT	VHDBN [80]	Rolling bearing fault classification	Employed VMD and HT for better feature extraction. The results of proposed method confirmed great advantages in classification accuracy
2020	Raw vibration + sliding window + coarse-grained method	MSDBN [81]	Various mechanical fault classification	Multiscale feature extractor enabled for learning better features, which in turn increased the accuracy of the method than standard DBN
	Vibration + wavelet package decomposition (WPD)	DBN + D-S theory + GA + PSO [82]	Bearing fault and severity level classification	GA-PSO algorithm improved learning capability and computation efficiency of the DBM classifier. In addition, the D-S theory-based vibration data fusion increased the accuracy of the model

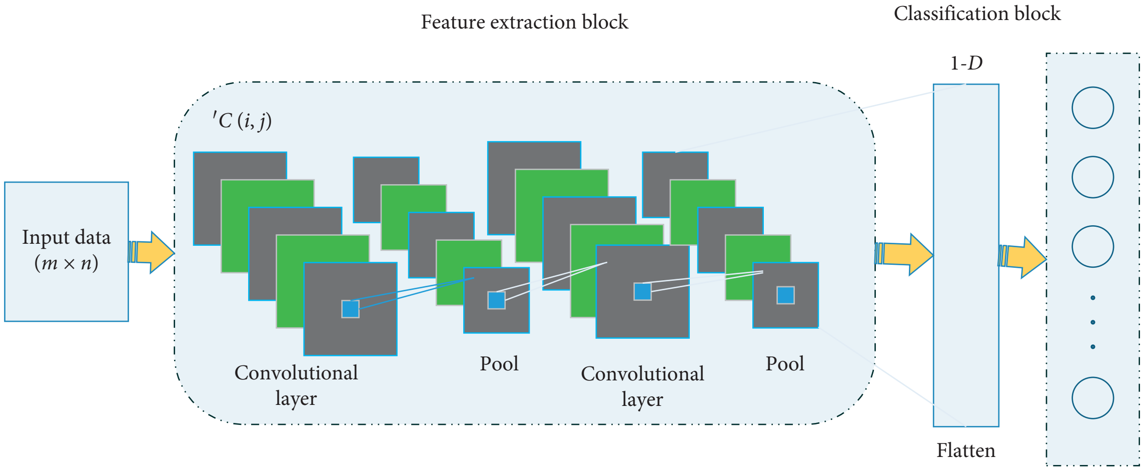


Figure 5: CNN structure.

and pooling layer are employed to extract rich features from data. The second block comprises fully connected layers that predict target variables through learning representations from training data.

The discrete convolution networks are employed to extract representation from 1D or 2D data through learnable filters. The convolutional operation results in output C are as given in equation (1), and the output of the convolutional layer is known as the feature map.

$$C(i, j) = \sum_{m} \sum_{n} I(m.n)K(i - m,\ j - n), \qquad (1)$$

where I and K are the input and filter, respectively. The pooling layer reduces the size of the feature map and assists in avoiding overfitting. Following the pooling operation, the feature map is flattened (1D array) and fed to the classification block [83]. The classification block generates output based on the extracted features.

CNN and its derivatives have been widely used in condition monitoring systems. In [84], the authors have used CNN with DFT for classifying motor faults. The method was used to classify motor faults such as bearing faults and lubricant degradation levels. Compared to traditional methods that rely on manual feature extraction, this method allowed the automatic extraction of features from scaled vibration data. Then, nonoverlapping windows are separated, each containing measurement samples of one second.

Subsequently, the DFT of the windows are calculated. At the end, these frequency decompositions are fed to the deep CNN model. Overall, this method yielded better performance than classical feature engineering techniques such as kurtosis, skewness, mean, and standard deviation. Liu et al. [85] have proposed a novel method called dislocated time-series CNN (DTS-CNN) for fault classification of electric motors. The dislocation layer in the model can extract the relationship between periodic vibration signals with different intervals. This modification of CNN performs more effectively under nonstationary conditions. This method extracts features from the raw data. The model mitigates the overfitting problem through weight sharing and sparse connectivity. The model was used to predict nine different motor faults, and it was confirmed through the results that this model performed better than the standard CNN with an accuracy of 96.3%.

In [86], authors have employed a 1D-CNN for real-time classification of the bearing faults. The method did not require any feature extraction technique, which in turn made it fast and computationally efficient. Motor current signals were fed to the model under constant speed condition. Experimental results confirmed the effectiveness of the method compared to the conventional models including MLP, SVM, and RBF. Guo et al. [87] have investigated a novel adaptive deep CNN (ADCNN) for bearing fault classification and their severity levels. It avoided training process failure due to unsuitable learning rate by the addition of adaptive learning rate and momentum. It also enabled the automatic extraction of features from raw vibration signals. The investigation results confirmed the superiority of the method compared to existing methods such as support vector regression machine (SVRM).

Ding et al. [88] have proposed a novel method called energy-fluctuated multiscale feature (EFMF) learning with deep CNN for the spindle bearing fault classification. A multiscale deep CNN was constructed using different layers such as convolution and pooling layers with sigmoid function. It combined the skipping layer with the last convolution layer, which provided input to the multiscale layer. Meanwhile, wavelet packet energy images (WPI) were constructed using wavelet packet transform (WPT) and phase space reconstruction. This 2D image was able to reveal energy fluctuations of the vibration signals and reconstruct local relationships among the WP nodes. Furthermore, representations were learned by the deep CNN architecture through brightness (frequency energy) variations of the energy-fluctuated images. Taking advantage of local and global features, the model was able to effectively classify the ten different spindle-bearing faults with a maximum accuracy of 98.8%. The model showed outstanding performance compared to the traditional methods, including standard CNN, BPNN, PCA, and LDA. In [89], authors have used an ensemble deep CNN model with the improved Dempster–Shafer (D-S) evidence theory (IDSCNN) for bearing fault classification under varying load conditions. The improved D-S evidence theory was employed for data fusion from the two vibration sensors. It was implemented using a similarity matrix and a modified Gini index. It addressed the two problems of traditional D-S evidence theory: objective evaluation of the basic function of the evidence body, and solving conflicting evidence from different sensors. Deep CNN was fed with the RMS value from FFT of the vibration signals. The experimental results confirmed higher performance of the model compared to the other existing models such as MLP, SVM, deep convolutional neural networks with wide first-layer kernels (WDCNN), and DSCNN.

Zhang et al. [90] have employed a novel method called CNN with training interference (TICNN), which can detect bearing faults with noisy data and under varying load conditions. The authors have used two techniques to introduce antinoise and domain adaptation abilities, which allowed the model to extract features from raw vibration signals. This was done by adding "dropout" layers and very small batch training. In addition, they have initially used wide convolution kernels for suppressing the noise, which is followed by small convolutional kernels that extracted rich representations from the data. It was confirmed through the results that the model performed stable and accurate classification due to ensemble learning with an average accuracy of 95.5%, whereas the comparative methods of SVM, MLP, and DNN only achieved poor performance with average accuracy of 65%, 80%, and 80%, respectively. The authors have suggested that this model would be useful in industrial environments. In [91], the authors have employed CNN as machinery health indicator. The method automatically learned features from the vibration data and constructed health indicators (HIs). HIs were constructed using a specific degradation process. However, the HIs faced a problem of outlier region deviation that was referred to as the trend burr, which negatively affects the performance of the HIs. The method was effectively able to detect and correct the outlier regions. The investigation results confirmed the effectiveness of the method in feature extraction.

Jia et al. [92] have employed a model called deep normalized CNN (DNCNN) to classify the bearing faults using the vibration data. The study addressed two problems of CNN. First the imbalanced fault classification problem was addressed using normalized CNN with weighted softmax, which allowed learning of better features and avoided misclassification. Another problem is that it was not clear what the CNN had actually learned as DL methods are treated as black box models. This problem was addressed using a neuron activation maximization (NAM) technique, which suppressed the useless information by analysing the kernels of the normalized CNN. The model produced effective results by overcoming these two problems and had a maximum accuracy of 99.2%.

Hoang et al. [93] have employed vibration image CNN (VI-CNN) to classify the rolling bearing faults. Authors have converted the vibration signals into gray-scale images by normalizing into range of [−1, 1]. Consequently, the amplitude of each sample becomes the intensity of pixels of the vibration image. It was confirmed that the model effectively classified the faults even in varying load conditions with an accuracy of 97.7%. The authors have suggested that the method can robustly perform in industrial environments. Li et al. [94] have employed a novel method by combining CNN and S-transforms to classify the bearing faults. The S-transform was combined with a CNN as an S-layer, which allowed extraction of features from the two accelerometers data. The S-layer automatically converted the vibration data into a 2D time-frequency matrix. The investigation results confirmed superior performance compared to the existing methods, namely, SVM, KNN, linear discriminant (LD), and bagged trees (BT).

In [95], the authors have used CNN based on a capsule network (ICN) for bearing fault classification. The method was intended for strong generalization and used a dynamic routing capsule net with an inception block. The Inception block removed the nonlinearity of the capsule. Before applying these steps, vibration data were converted to time-frequency graph using STFT and fed to the model. Subsequently, the model classified the faults through varying lengths of the capsule. The investigation results confirmed the higher generalization power of the model with an accuracy of 82%, compared to the standard CNN. Hoang et al. [96] have employed CNN with an information fusion technique to classify bearing faults of a motor. They have employed two phase current signatures, which further split into equal samples using a sliding window. Then, 1D signals are converted into 2D matrix by rearranging the array signal. The transformed 2D current signal is effectively a gray-scale image and this is fed into the model. Experimental results confirmed the effectiveness of the technique in detecting the faults with acceptable performance with a maximum accuracy of 99.4%.

Li et al. [97] have used CNN with WPT for rolling bearing fault classification. WPT extracted features from raw vibration signals, which were further transformed again into gray-scale images. The investigation results showed that the model achieved superior fault detection with individual fault detection accuracy of up to 100% because of the richness of the input features. In [98], the authors have used enhanced LeNet-5 CNN for bearing fault classification. STFT was employed to convert vibration data into 2D images, and hierarchical regularization was used to speed up the training process. In addition, the scaled exponential linear unit (SELU) function was employed to avoid excessive dead-nodes during the training process, which in turn enhanced the robustness of the model learning. It was observed through results that the model accelerated the training process by selecting sensitive features even under varying load conditions. Moreover, the model effectively classified the faults with higher accuracy up to 100% compared to the models with individual time or frequency features. In [99], authors have used CNN to diagnose the stator winding faults of induction motors. They have normalized the raw current data and then converted it into a three-dimensional matrix. The method was able to detect the number of faulty stator-winding shorted turns without extensive preprocessing. The investigation results confirmed the applicability of the method in real-time, regardless of motor operating conditions. Furthermore, results showed a high accuracy of up to 100% in the individual fault detection. Research studies on motor diagnosis and prognosis using CNN are summarized in Table 4.

2.5. Recurrent Neural Networks (RNN).

RNNs yield better performance with sequential or time-series data, which makes them the most suitable candidate for condition monitoring of motors. According to ref. [100], RNNs are the deepest among all neural networks, and their architecture is as shown in Figure 6. They differ from MLP, which only maps input data to target vectors, while RNNs map the entire history of past inputs to target vectors owing to their memory capability. For supervised tasks, RNNs can be trained by employing back-propagation through time. The transition function of a basic RNN can be defined as in

$$h_t = \mathbb{H}(x_t, h_{t-1}), \tag{2}$$

where $\mathbb{H}$ is a nonlinear and differentiable transfer function, x_t is current time information, and h_{t-1} is the previous time information.

Although RNNs perform better on time-series data, they face the gradient vanishing problem. This problem was addressed through the advent of long short-term memory (LSTM) in 1997 [101], which has outperformed in various fields. LSTMs have the capability to memorize and forget representations of data. Moreover, gated recurrent units (GRU) and bidirectional LSTMs can enhance model flexibility and capacity. As shown in Figure 7, deep LSTM can increase the representation learning capabilities through propagating the output of one layer as an input to the next layer.

RNN and its derivatives have been extensively used in motor condition monitoring owing to its memorizing power and robust performance. Zhao et al. [102] have employed deep LSTM with time-series data to classify motor faults. The “dropout” layers were used for model regularization. Raw vibration data were used as input to the model. Comparative results confirmed the effectiveness of the LSTM model with minimum root mean error (RMSE) of 10.2% compared to conventional techniques such as MLP and basic RNN. In [103], authors have employed local feature-based GRU (LFGRU) for motor fault classification again using raw vibration data. Local features were extracted from synchronized windows of the multisensory data, and then fed to the model. Experimental results confirmed the robustness of the model in classifying the motor faults with the maximum accuracy of 99.6%. In [104], authors have used RNN for rolling bearing fault prediction. They have investigated the model with two types of features: time domain and frequency domain. FFT was used to convert vibration data into frequency domain features. Experimental results

Table 4: Application of CNN in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2016	Raw vibration	CNN + DFT [84]	Motor fault classification	Owing to the automatic feature extraction, the method produced promising results compared to classical feature engineering methods
	Raw vibration	DTS-CNN [85]	Motor fault classification	Dislocation layer yielded better performance than standard CNN owing to rich feature extraction
	Raw current	1D-CNN [86]	Bearing fault classification	The simple architecture of the model allowed real-time detection of the faults
	Raw vibration	ADCNN [87]	Bearing fault classification and their severity level classification	Adaptive architecture of the CNN allowed to learn rich features from the data, which in turn increased its performance
2017	Vibration + WPI	Multiscale deep CNN [88]	Spindle bearing fault classification	Novel feature learning techniques with multiscale deep CNN allowed the model to outperform other traditional models through learning local and global features simultaneously
	Raw vibration data from two sensors	IDSCNN [89]	Bearing fault classification	Improved data fusion technique produced better classification results by addressing the problems in the conventional feature extraction techniques
2018	Raw vibration	TICCN [90]	Bearing fault classification	Dropout layers with the wide and small convolutional layers supressed the noise of data. Meanwhile, ensemble model allowed to attain high classification accuracy
	Raw vibration	CNN [91]	Motor health indication	The method was able to overcome outlier regions and effectively learned features from the data
	Raw vibration	DNCCN [92]	Bearing fault classification	The model produced effective classification results by overcoming data imbalance problem
2019	Raw vibration	VI-CNN [93]	Rolling bearing fault classification	2D vibration image allowed the method to effectively classify the faults without any feature learning and denoising technique
	Raw vibration	ST-CNN [94]	Bearing fault classification	The method produced higher performance than existing methods owing to automatic feature extraction by S-layer
	Vibration + STFT	ICN [95]	Bearing fault classification	Inception block resulted in better generalization than CNN
	Raw current signatures	CNN + IF [96]	Bearing fault classification	Promising results were obtained owing to information fusion
2020	Raw vibration + WPT	CNN [97]	Rolling bearing fault classification	Classification performance improved owing to the gray-scale vibration images
	Vibration + STFT	CNN + SELU function [98]	Rolling bearing fault classification	The method effectively classified the fault owing to the 2D-images of the data and regularization
	Raw current signatures	CNN [99]	Stator winding fault detection	The method can effectively detect stator winding faults from raw current data without any preprocessing

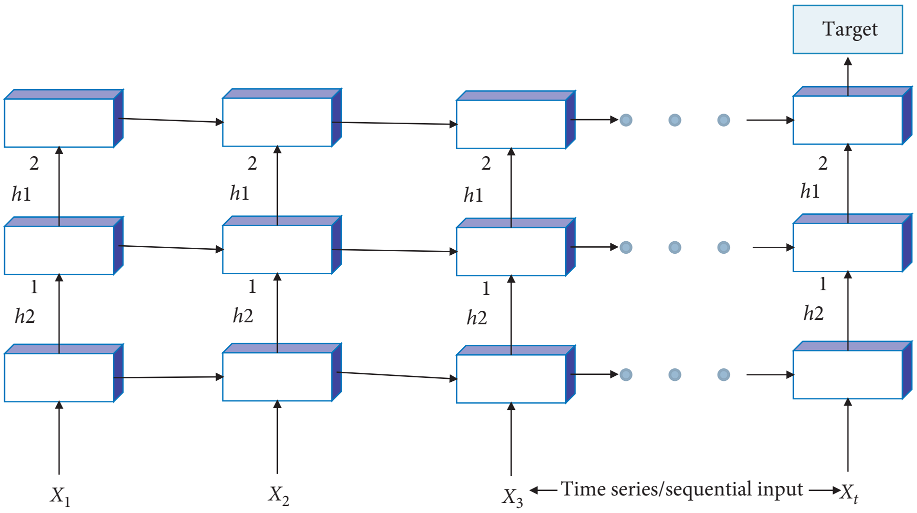


Figure 6: RNN structure.

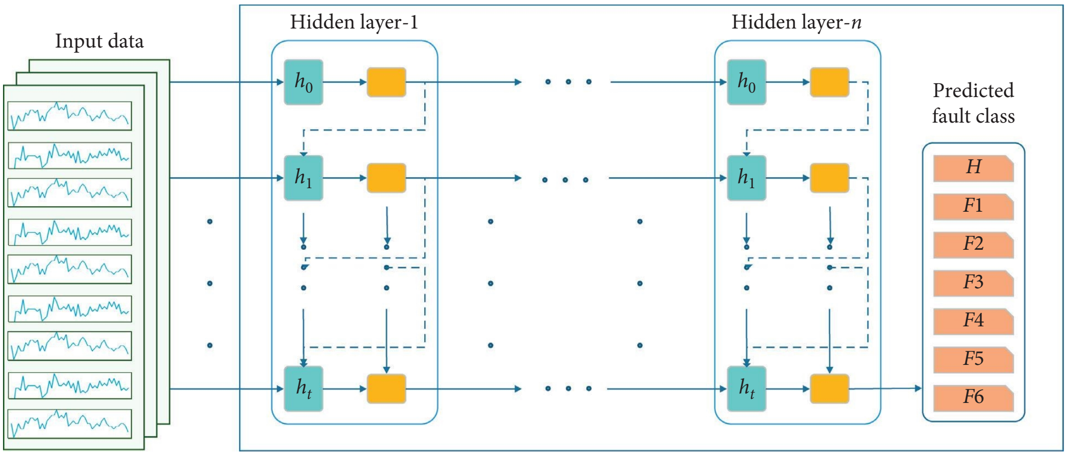


Figure 7: Deep LSTM classifier structure.

demonstrated that the model predicts the conditions of bearing better with frequency domain features.

Xiao et al. [105] have used deep LSTM to classify various motor faults using vibration data. The authors have not used any feature learning technique for classifying motor faults such as broken-bar, bowed-bar, bowed-rotor, faulty bearing, and voltage imbalance using raw vibration data. Results confirmed the superiority of the model performance with an accuracy of 98.6% compared to the conventional models including SVM, MLP, and standard RNN. In [106], the authors have predicted bearing performance degradation using bottleneck features based on LSTM. A wavelet threshold denoising (WTD) technique eliminated noise from the vibration data. Then, statistical features were extracted and finally fed to the stacked autoencoder (SAE) to obtain bottleneck features. The bottleneck features take depth and nonlinearity of the signals into account. Experimental results showed the effectiveness and superiority of the method in predicting the bearing's degradation with a resulting minimum RMSE of 0.0891 which compares well to existing methods including SVM and MLP.

Xiao et al. [107] have employed LSTM with weighted batch normalization (BN) for detecting different faults in the induction motor. They have employed the two manual feature extractions, namely, empirical statistical parameters and recurrent quantification analysis (RQA) to add antinoise capability in the model. The weighted BN allowed the evaluation of the contribution of each feature learning technique, and validated noise reduction. The results confirmed the robustness and effectiveness of the model in fault classification of the induction motor with an accuracy of 99.3% compared to the other DL models such as SVM, MLP, CNN, and standard LSTM. In [108], authors have classified bearing faults using a hierarchical stacked LSTM. The input layer received the vibration data, and then two stacked LSTMs were able to effectively learn representations from the data without any preprocessing. The experimental results confirmed that the model produced promising results owing to its deep structure with an accuracy of 99% and the model outperformed the state-of-the-art models such as SVM, MLP, 1-layer LSTM, and CNN.

Zhang et al. [109] have employed LSTM to assess the bearing performance degradation and predict the RUL using vibration and temperature data. The authors have used waveform entropy (WE) to identify the running condition of the bearing by computing the local mean of logarithmic vibration energy. In addition, a particle swarm optimization (PSO) method was used to optimize parameters of the LSTM, which in turn improved its learning performance. The authors have divided degradation states into different stages by time. WE has performed effectively under various degradation states while it lagged in some conditions owing to window length. However, it did not reflect any negative impact on the performance of the model. Experimental results showed the effectiveness of the method in indicating the degree of degradation in assessing the degradation states with an accuracy of 93.1%. In [110], the authors have used deep gated recurrent units (DGRU) with ELM to classify the faults of an adaptive rolling bearing. In addition, the authors have used an artificial fish swarm algorithm (AFSA) for GRU parameter optimization. The model consists of two stacked layers of GRU, which learned features from the raw vibration data. Lastly, ELM is applied for accurate classification of the faults based on learned features. The model was able to achieve 94.5% testing accuracy and the results demonstrated robust performance of the model compared to the conventional models such as CNN, DBN, and SAE.

Enshaei et al. [111] have used bidirectional deep LSTM (BiD-LSTM) to classify bearing faults. BiD-LSTM takes sequential data into account in both the forward and backward directions. It was confirmed through experimental results that the deeper BiD-LSTM performed better than the single-layered network and achieved 100% testing accuracy. The model effectively classified faults with high accuracy with raw vibration data. In [112], the authors have employed a hierarchical GRU network (HGRUN) for predicting future health index (HI) and RUL of the bearing. The kernel principle component analysis (KPCA) and exponentially weighted moving average (EWMA) were used to design a modified HI. Firstly, statistical features are extracted (time domain, frequency domain, and hybrid domain), then the KPCA fused these features and transformed it as a unified

HI. Subsequently, EWMA further modified the HI, which can depict the bearing degradation process. Lastly, HGRUN was developed by stacking multiple GRU layers and inputting with the modified HI. Experimental results confirmed that the method effectively depicted the degradation process of the bearing and can predict the future HI and RUL of the bearing. Comparison showed the superiority of the technique to the existing techniques with the minimum of 13.8 ± 2.8%. In [113], the authors have used the enhanced deep GRU and complex wavelet packet energy moment entropy for early bearing fault classification. Complex wavelet packet energy moment entropy as a monitoring index allows reduction in aliasing and the detection of dynamic changes in the vibration data. Subsequently, deep GRU allows to learn the complex mapping relationships from the vibration data. Lastly, the learning capability of the model was enhanced using a modified training algorithm based on learning rate decay strategy. Experimental results confirmed the effectiveness of the method in early fault detection compared to the other prognosis methods. These recent studies related to motor condition monitoring using RNN and its variants are summarized in Table 5.

2.6. Generative Adversarial Networks (GAN). A Generative Adversarial Network (GAN) is a binomial zero-sum game-theory-based learning model. It comprises two models: a generator model (G) and a discriminator model (D). The structure of a GAN is shown in Figure 8(a). Both the models can have a different type of neural network architecture such as RNN, AE, or CNN. The D-model tries to increase the probability of collected true data (x) and decrease the probability of samples generated by the G-model. The G-model tries to cheat the D-model by generating a sample training set using a noise input (z), gradually improving its performance until the D-model can no longer discriminate between the true data and the generator data. Thus, by employing an antagonism training process, the capacity of both the models is improved simultaneously [114, 115]. The optimization of this two-player game is calculated as given in

$$\begin{aligned}\min{}_G\max{}_D V(D,G) = {} & E_{x-P_{\text{data}}(x)}[\log D(x)] \\ & + E_{z-P_z(z)}[\log(1-D(G(z)))].\end{aligned} \tag{3}$$

Considering a supervised learning approach, the GAN models can generate fake labels which are like real data. Figure 8(b) gives a visual illustration of a GAN-based classifier. The classifier receives samples from G-model and the classification error back-propagates through G-model and classifier.

In recent years, researchers have employed GANs and its derivatives in motor condition monitoring. It is often employed to address the data imbalance problem through data augmentation. Shao et al. [116] have developed an auxiliary classifier GAN (ACGAN)-based framework to learn and generate realistic one-dimensional vibration data. Both the generator and discriminator consisted of 1D-CNN, which was allowed to learn representations and generate high-quality artificial data samples. In addition, batch normalization was employed to address the gradient vanishing problem during the training process of the GAN, which in turn assisted in avoiding overfitting. They have employed statistical methods to evaluate the quality of the generated data. Experimental results demonstrated the effectiveness of the model in data augmentation. The model robustly classified the faults by addressing the data unbalance problem with an accuracy of 99.1%. In [117], the authors have employed deep GAN for bearing fault diagnosis using an imbalanced dataset. The authors have used a two-stage approach: the first stage augmented the data through the GAN model; then the second stage classified the faults using a deep CNN model. They have introduced GAN with multiple generators in which each generator is dedicated to the specific bearing conditions. Investigation results confirmed the robustness of the approach in data augmentation and fault classification with a maximum accuracy of 99.9%. Authors have verified the approach by applying it on two different datasets. Applications of GANs in motor condition monitoring are summarized in Table 6.

3. Summary, Challenges, and Prospects of DL Models

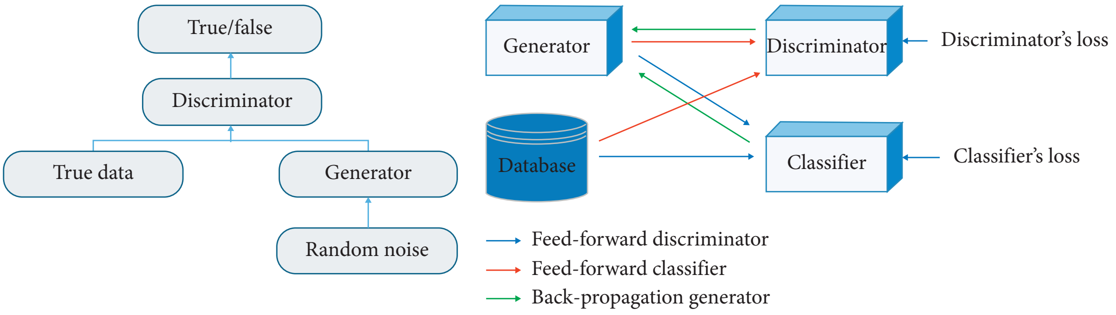
The effective application of DL models in condition monitoring systems extensively rely on data acquisition, data labeling, feature processing, and model parameter optimization. However, these processes are challenging, time-consuming, and may require domain expert knowledge. In the previous section, we have comprehensively reviewed each DL model and their application in condition monitoring of motors. Although, the reviewed research demonstrated promising leads in condition monitoring of motors, there are various open challenges that are yet to be completely solved. Figure 9 shows a heat map of DL models that have been used along with a variety of feature-processing techniques. From the heat map, it is apparent that the DL models have been effectively performing on raw input data owing to their deep hierarchical architectures. However, time domain features also have been used in various researches with less complex models that show the importance of feature-processing techniques in simplifying the training and testing process of DL Models. On the other hand, Figure 10 shows a 3D map of the number of publications using the type of input data with different DL models for motor fault diagnosis. The majority of the current literature has focused on using vibrational analysis for motor condition monitoring tasks.

The following section presents some challenges in the application of DL models and future directions to improve the performance of these models. It also summarizes the strengths and drawbacks of these models:

3.1. Class Data Imbalance. During the data extraction process, most of the time healthy samples of the data outnumber the ones representing fault conditions. Hence, while

Table 5: Application of RNNs and its variants in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2016	Raw vibration	LSTM [102]	Motor fault detection	The model outperformed the existing models owing to its deep architecture
2017	Raw vibration from multiple sensors	LFGRU [103]	Motor fault classification	The model was able to effectively classify motor faults owing to deep structure without any extensive feature extraction
	Vibration + time domain features + frequency domain features	RNN [104]	Rolling bearing fault prediction	Model showed better prediction results with frequency domain features
2018	Raw vibration	LSTM [105]	Motor fault classification	The results confirmed effectiveness of the technique compared to existing methods owing to gated functions of the model
	Vibration + WTD + time domain features	SAE + LSTM [106]	Prediction of bearing performance degradation	Results confirmed effectiveness of the method in achieving the task compared to the conventional methods
2019	Vibration + hybrid feature + RQA	LSTM [107]	Induction motor fault classification	Method effectively classified the motor faults due to the antinoise feature extraction techniques
	Raw vibration	LSTM [108]	Rolling bearing fault classification	The results demonstrated efficiency in the bearing fault classification owing to inherent mechanism and deep architecture
	Vibration + temperature + WE	LSTM + PSO [109]	Bearing degradation assessment and RUL prediction	Results confirmed effectiveness of the method in estimating the bearing life states and degree of degradation
	Raw vibration	GRU + AFSA + ELM [110]	Adaptive rolling bearing fault classification	Results revealed robust performance of the model owing to optimized parameter selection by AFSA and accurate classification by ELM
	Raw vibration	BiD-LSTM [111]	Bearing fault classification	Results demonstrated effectiveness of the method owing to its long-term dependency in time series data
	Vibration + time, frequency, hybrid features	HRGUN + KPCA + EWMA [112]	Future HI prediction and RUL prediction of the bearing	KPCA and EWMA allowed to effectively track the degradation process and promisingly predicted the HI and RUL of the bearing
2020	Raw vibration + complex wavelet packet energy moment entropy	Enhanced deep GRU [113]	Early bearing fault classification	The monitoring health index technique and the modified training algorithm allowed to detect fault at an early stage



(a)

(b)

Figure 8: (a) GAN structure. (b) GAN-based classifier structure.

Table 6: Application of GAN in condition monitoring of motors.

Year	Input data and features	DL model	Application	Remarks
2019	Raw vibration	ACGAN + batch normalization [116]	Bearing fault classification and data augmentation	Results revealed effectiveness of the model in classification due to data augmentation and batch normalization
2020	Raw vibration	GAN + CNN classifier [117]	Bearing fault classification and data augmentation	The two-stage approach yielded promising results in terms of data augmentation and fault classification

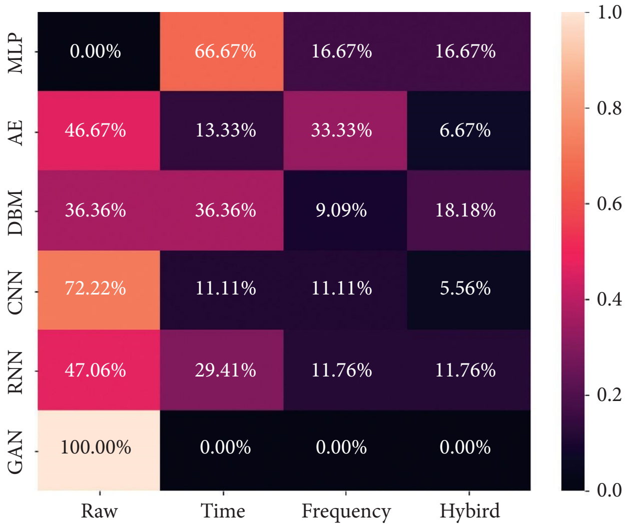


Figure 9: Heatmap of DL models versus type of input data features.

designing a classification model for anomaly detection or fault identification, there is a clear bias toward classes that have more samples of data. There are multiple ways to deal with this bias in classification models; examples include undersampling which is reducing the samples of oversampled classes, and oversampling which is increasing samples of minority classes by synthesizing new examples. A prebiased model can also be used with weights, which is a clever way to deal with imbalanced data, where individual class weights are calculated, and higher weights would lower their effect on model prediction. Most DL models have performed effectively in different applications but these models have some drawbacks which require more investigation for producing optimal results. The generative models such as AEs and GANs, although harder to train, can provide a way to synthesize authentic data.

3.2. Feature Processing. Feature processing reducing the number of features greatly reduces the hardware dependence and the need for a highly nonlinear function mapping by DL models. However, the bulk of the research is conducted on raw input data with DL models as highlighted in Figure 9, which truly exploits the potential of these models. However, performance evaluation of these models with current evaluation matrices is vague. Either the trade-off between raw and formulated features versus explainability of the models needs to be taken in account or new evaluation matrices need to be developed.

3.3. Model Selection. Model selection for addressing domain-specific problems heavily depends upon the way the problem is formulated. Anomaly detection problems are mostly formulated as classification problems. Another factor that plays a role in selecting a proper model for the problem is the type of data available. The right choice for the model also depends upon the way data is formulated.

3.4. Hyperparameter Tuning. This is one of the most important parts of employing a DL algorithm and includes model tuning in terms of (a) number of neurons per layer, (b) number of layers, and (c) choices regarding initialization, activations, optimizers, learning rate, loss calculation, etc. Although most of the time this is done through multiple trial and error experiments, the gradient vanishing and exploding, idle points, and nonconvex optimizations are currently generating research interest. The development of new frameworks like Keras, Tensorflow, Theano, and Pytorch have stimulated the process of experimentation, and the community of researchers addressing these issues is increasing with increasing progress.

3.5. Generalization Power. The generalization power of a DL model is basically its ability to identify the correct sample and is often defined in terms of overfitting or underfitting. The complexity of a model should be increased only when needed. However, there are multiple approaches by which the overfitting problem of the DL models can be addressed. L1 and L2 regularizations can smooth the training process; a “dropout layer” between fully connected layers works in the same way to reduce the complexity of the model in a controlled fashion; and batch-normalizations techniques have the capacity to reduce overfitting by reducing the effect of some dominant neurons in the network.

3.6. Interpretability of DL Models. DL models, which are inherently black box models, do not provide much insight into their inner workings. For humans to trust these models, they need to be interpretable and explainable. Multiple publications [26, 118] have reported that a neural network could be fooled easily into choosing a wrong category by making minor changes to pixels and neither discriminative nor generative models are an exception. Interpretability alone might not be enough for humans to trust these black box models; they will need explainability. Explanatory Artificial intelligence (XAI) is one of the frontiers of DL research.

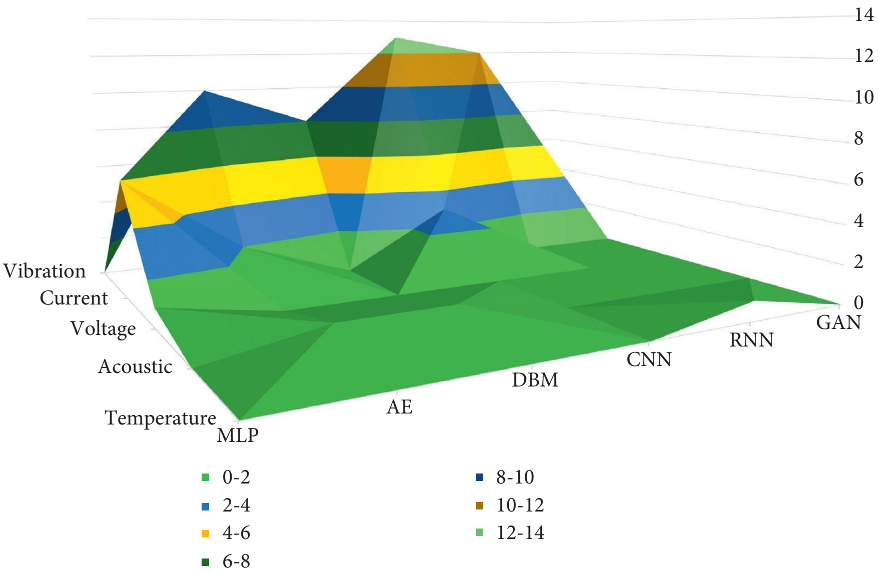


Figure 10: Three-dimensional map of the number of publications of DL models in terms of type of input data.

Table 7: Strengths and drawbacks of DL models.

DL model	Strengths	Drawbacks
MLP [47, 83]	(i) Can work with clean, balanced, and scaled data, regardless of the data type (ii) Integration in real-time systems is easy and allows one-dimensional data analysis	(i) Requires a lot of tuning to work on dirty or unscaled data
AE [118, 127]	(i) Allows to learn rich representations and reduces dimensionality (ii) Can work as a denoising technique to get cleaner data (iii) Easy implementation	(i) Requires massive training data and training time (ii) Poses difficulty in discriminating relevant data
DBN [76, 77]	(i) Can achieve higher level of generalization on one-dimensional raw data	(i) Slow training and inefficient
DBM [75, 128]	(i) Allows for one-dimensional data analysis (ii) Combined optimization of all the layer parameters	(i) Slower training than DBN and inefficient (ii) Combined optimization becomes impractical for large data
CNN [10, 83]	(i) Fits well for multidimensional data analysis (ii) Enables for feature extraction from raw data	(i) Complex architecture (ii) Requires large datasets and takes long training time (iii) Estimations of continuous data are poor
RNN/LSTM/GRU [83, 113, 129]	(i) Performs well with time series or sequential data (ii) Better forecasting ability in time series and sequential data	(i) Without proper constraints on weights and gradient clipping, might suffer from the gradient either vanishing or becoming unbounded
GAN [115, 116, 130]	(i) Learns underlying representation of data well (ii) Seems to achieve a discriminator with less generalization error where its generator output (fake data) provides a regularization effect (iii) Only algorithm that can work in semisupervised or even unsupervised setting to identify under observation clusters (iv) Model with high fidelity	(i) Very complex architecture and implementation is even difficult (ii) Difficult to model discrete data (iii) Perhaps not suitable for real-time implementation

3.7. Hardware Challenges. DL models, regardless of their efficiency at solving the task on hand, invite hardware challenges, especially when their deployment in a standalone or embedded system is considered. Deep models due to their complex topology require higher computing power, energy, and memory. Training and optimization of these models is an iterative process during which multiple high-dimensional matrix multiplications are performed. Therefore, much of the research has focused on reducing the cost of multiply and accumulate (MAC) [119]. Key factors taken into consideration while integrating these models with hardware are accuracy, energy consumption, throughput/latency, and cost [120]. Designing an efficient DL architecture that incorporates all these factors is done through optimizing and compressing the DL models through algorithmic techniques or designing application-specific hardware. Algorithmic techniques usually focus on retaining the accuracy of a DL architecture after performing pruning/compression on it. On the other hand, hardware is designed to make it energy-efficient and reduce the latency. However, codesign of algorithm and hardware has also been explored [121]. Hardware solutions for deploying DL architectures range from general purpose solutions (GPUs) to application-specific solutions (FPGAs, ASICs). The optimum choice of hardware depends upon the application. Since CPUs are least used due to their limited throughput, GPUs are currently the preferred option for training and inference of DL models. Nvidia has produced solutions like CUDA and cuDNN for easy and fast implementation and inference from DL models. FPGAs-based solutions particularly focus on efficient data routing and benefit from sparsity and reuse of data points already fetched in the computation network [122]. Moreover, hardware accelerators using variable bandwidth and reconfigurable data flow paths also have been developed [123–126]. Efficient deployment of DL architectures on edge is an active area of research and many paradigms are yet to be explored for effective performance.

The strengths and drawbacks of these models are summarized in Table 7.

4. Concluding Remarks

In this paper, a comprehensive review is presented related to the usage of various feature-processing techniques with DL models for condition monitoring of motors. This paper has reviewed the literature in terms of input data and feature types used in effective fault diagnosis of motors using DL models. More specifically, this work summarizes the applications of DL models in terms of various feature-processing techniques and how these techniques have solved various problems that allow to achieve high generalization. It was observed that the usage of feature-processing techniques has raised the efficiency of DL models and reduced hardware dependency. Moreover, usage of feature-processing techniques with DL models allowed faster execution owing to the feature extraction and removal of redundant information. It was found from the literature review that the AEs, DBM, DBNs, and MLP architectures have been in the focus of researchers for fault diagnosis and prognosis. On the other hand, CNN and RNN also have received attention for their application in condition monitoring of motors. However, their complex architectures demand expert human knowledge and feature processing for successful implementation. Furthermore, GANs have solved the class imbalance problem in data to some extent. However, more effort is still required for acceptable results.

Moreover, it has been observed that most of the available work focuses on mechanical faults diagnosis using DL models, specifically bearing faults. However, very limited work is conducted related to electrical fault diagnosis and prediction using DL models. Researchers may be facing difficulties in introducing electrical faults owing to danger associated with these faults. However, it is crucial to explore this area because motors also get damaged owing to various electrical faults. A large portion of the DL-based condition monitoring of motors have been conducted in relation to fault classification. However, a few researchers have conducted research related to health index (HI) prediction and RUL estimation of the motor and its components. Meanwhile, data fusion techniques have been successfully used with various models, which allowed the improvement of model classification accuracy. More work is required in this area; through fusing various diagnostic methods such as current and vibration, model performance could be improved further. Consequently, it will uplift condition-based maintenance of motors by employing DL models.

Meanwhile, DL models have also exhibited some deficiencies, which can be viewed as prospective future opportunities for researchers and engineers in this domain. This review has demonstrated that the DL-based diagnosis mostly involve a supervised learning approach; however, the practicality of such an approach is highly challenging and time-consuming in real engineering scenarios. From a future perspective, DL models need to be employed for automatic end-to-end diagnosis, which includes feature learning from data acquisition to motor fault classification or prediction. Moreover, acquiring more data does not necessarily mean that the DL models will produce better results. Thus, feature-processing techniques are essential for better generalization. It was noticed that the majority of research in this domain was focused on fault diagnosis in certain components such as the bearing. Very little research has been conducted on root cause analysis, degradation, and RUL prediction. Considering future opportunities, there is an urgent need for advanced feature-processing techniques that can assist in analysing the huge amounts of data and yield effective diagnosis and prognosis results. Furthermore, there is great demand for research on explainable AI (XAI), which will overcome the problem of the vague operation of DL algorithms. The authors of this review believe that the practitioners working in this domain would find this article very useful in solving their problems and evaluating the methods. Meanwhile, this review gives valuable pointers to a road map for future research in this domain. The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed subsections.

Data Availability

The references and data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The authors thank the “Haptics, Human Robotics, and Condition Monitoring Lab” established in Mehran University of Engineering and Technology, Jamshoro, under the umbrella of the National Center of Robotics and Automation funded by the Higher Education Commission (HEC), Pakistan, and the support of the EPSRC under grant no. EP/

S005463/1 (better FITT early detection of contact distress for enhanced performance monitoring and predictive inspection of machines).

References

[1] A. Hmidet and O. Boubaker, “Real-time low-cost speed monitoring and control of three-phase induction motor via a voltage/frequency control approach,” *Mathematical Problems in Engineering*, vol. 2020, 2020.

[2] W. Zhang, M.-P. Jia, L. Zhu, and X.-A. Yan, “Comprehensive overview on computational intelligence techniques for machinery condition monitoring and fault diagnosis,” *Chinese Journal of Mechanical Engineering*, vol. 30, no. 4, pp. 782–795, 2017.

[3] D. Goyal and B. S. Pabla, “The vibration monitoring methods and signal processing techniques for structural health monitoring: a review,” *Archives of Computational Methods in Engineering*, vol. 23, no. 4, pp. 585–594, 2016.

[4] M. Deighton, *Facility Integrity Management: Effective Principles and Practices for the Oil, Gas and Petrochemical Industries*, Gulf Professional Publishing, London, UK, 2016.

[5] G. Yao, B. Fan, W. Wang, and H. Ma, “A novel median-point mode decomposition algorithm for motor rolling bearing fault recognition,” *Mathematical Problems in Engineering*, vol. 2020, 2020.

[6] P. Gangsar and R. Tiwari, “Comparative investigation of vibration and current monitoring for prediction of mechanical and electrical faults in induction motor based on multiclass-support vector machine algorithms,” *Mechanical Systems and Signal Processing*, vol. 94, pp. 464–481, 2017.

[7] S. E. Pandarakone, Y. Mizuno, and H. Nakamura, “Distinct fault analysis of induction motor bearing using frequency spectrum determination and support vector machine,” *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 3049–3056, 2016.

[8] N.-H. Kim, D. An, and J.-H. Choi, *Prognostics and Health Management of Engineering Systems: An Introduction*, Springer, Berlin, Germany, 2016.

[9] H. Chen, B. Jiang, W. Chen, and H. Yi, “Data-driven detection and diagnosis of incipient faults in electrical drives of high-speed trains,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4716–4725, 2018.

[10] J.-H. Han, D.-J. Choi, S.-K. Hong, and H.-S. Kim, “Motor fault diagnosis using CNN based deep learning algorithm considering motor rotating speed,” 2019.

[11] A. Choudhary, D. Goyal, S. L. Shimi, and A. Akula, “Condition monitoring and fault diagnosis of induction motors: a review,” *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 1221–1238, 2019.

[12] A. K. S. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[13] S. Yin, X. Li, H. Gao, and O. Kaynak, “Data-based techniques focused on modern industry: an overview,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 657–667, 2014.

[14] T. Han, D. Jiang, Q. Zhao, L. Wang, and K. Yin, “Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery,” *Transactions of the Institute of Measurement and Control*, vol. 40, no. 8, pp. 2681–2693, 2018.

[15] Z. Xu, C. Hu, F. Yang et al., “Data-driven inter-turn short circuit fault detection in induction machines,” *IEEE Access*, vol. 5, pp. 25055–25068, 2017.

[16] C. T. Kowalski and T. Orlowska-Kowalska, “Neural networks application for induction motor faults diagnosis,” *Mathematics and Computers in Simulation*, vol. 63, no. 3–5, pp. 435–448, 2003.

[17] A. Habib, F. K. Saddozai, A. Sattar, A. Khan, I. A. Hameed, and F. M. Kundi, “User intention mining in bussiness reviews: a review,” 2018.

[18] D. Shen, G. Wu, and H. I. Suk, “Deep learning in medical image analysis,” *Information Fusion*, vol. 19, pp. 221–248, 2017.

[19] S. Sun, C. Luo, and J. Chen, “A review of natural language processing techniques for opinion mining systems,” *Information Fusion*, vol. 36, pp. 10–25, 2017.

[20] H. A. Pierson and M. S. Gashler, “Deep learning in robotics: a review of recent research,” *Advanced Robotics*, vol. 31, no. 16, pp. 821–835, 2017.

[21] H. Shao, H. Jiang, H. Zhang, and T. Liang, “Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2727–2736, 2017.

[22] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, “Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets,” *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203–1215, 2018.

[23] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] R. Roberts, G. Giancontieri, L. Inzerillo, and G. Di Mino, “Towards low-cost pavement condition health monitoring and analysis using deep learning,” *Applied Sciences*, vol. 10, no. 1, p. 319, 2020.

[26] D.-T. Hoang and H.-J. Kang, “A survey on Deep Learning based bearing fault diagnosis,” *Neurocomputing*, vol. 335, pp. 327–335, 2019.

[27] J. Villalba-Díez, M. Molina, J. Ordieres-Meré, S. Sun, D. Schmidt, and W. Wellbrock, “Geometric deep lean learning: deep learning in industry 4.0 cyber-physical complex networks,” *Sensors*, vol. 20, no. 3, p. 763, 2020.

[28] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, “A survey of deep learning and its applications: a new paradigm to machine learning,” *Archives of Computational Methods in Engineering*, vol. 23, pp. 1–22, 2019.

[29] Z. Wang, K. Liu, J. Li, Y. Zhu, and Y. Zhang, “Various frameworks and libraries of machine learning and deep learning: a survey,” *Archives of Computational Methods in Engineering*, vol. 23, pp. 1–24, 2019.

[30] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, “Machine remaining useful life prediction via an attention based deep learning approach,” *IEEE Transactions on Industrial Electronics*, vol. 23, 2020.

[31] J. Wang, S. Li, Z. An, X. Jiang, W. Qian, and S. Ji, “Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines,” *Neurocomputing*, vol. 329, pp. 53–65, 2019.

[32] A. Suresh, R. Udendhran, and M. Balamurugan, “Integrating IoT and machine learning - the driving force of industry 4.0,” *Internet of Things for Industry 4.0*, vol. 23, pp. 219–235, 2020.

[33] S. Nahavandi, “Industry 5.0-A human-centric solution,” *Sustainability*, vol. 11, no. 16, p. 4371, 2019.

[34] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: a survey," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 144–163, 2019.

[35] S. Shao, R. Yan, Y. Lu, P. Wang, and R. Gao, "DCNN-based multi-signal induction motor fault diagnosis," 2019.

[36] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: a review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020.

[37] D. K. Soother, J. Daudpoto, and A. Shaikh, "Vibration measurement system for the low power induction motor," *Engineering Science and Technology International Research Journal*, vol. 2, pp. 53–57, 2018.

[38] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[39] Z. Zhao, "Deep learning algorithms for rotating machinery intelligent diagnosis: an open source benchmark study," *ISA Trans*, vol. 23, 2020.

[40] G. Toh and J. Park, "Review of vibration-based structural health monitoring using deep learning," *Applied Sciences*, vol. 10, no. 5, p. 1680, 2020.

[41] D. K. Soother and J. Daudpoto, "A brief review of condition monitoring techniques for the induction motor," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 43, no. 4, pp. 499–508, 2019.

[42] A. Choudhary, S. Shimi, and A. Akula, "bearing fault diagnosis of induction motor using thermal imaging," 2018.

[43] X. Zhang, Q. Zhang, M. Chen, Y. Sun, X. Qin, and H. Li, "A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method," *Neurocomputing*, vol. 275, pp. 2426–2439, 2018.

[44] Y. Lei, Z. He, Y. Zi, and Q. Hu, "Fault diagnosis of rotating machinery based on multiple ANFIS combination with GAs," *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2280–2294, 2007.

[45] V. Stojanovic and N. Nedic, "Robust identification of OE model with constrained output using optimal input design," *Journal of the Franklin Institute*, vol. 353, no. 2, pp. 576–593, 2016.

[46] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and Information Systems*, vol. 34, no. 3, pp. 483–519, 2013.

[47] R. G. Vieira, C. M. Medeiros, and E. T. Silva, "Classification and sensitivity analysis to detect fault in induction motors using an MLP network," 2016.

[48] R. Palácios, W. Godoy, A. Goedtel, I. da Silva, D. Moríñigo-Sotelo, and O. Duque-Perez, "Time domain diagnosis of multiple faults in three phase induction motors using inteligent approaches," 2017.

[49] G. H. Bazan, P. R. Scalassara, W. Endo, A. Goedtel, W. F. Godoy, and R. H. C. Palácios, "Stator fault analysis of three-phase induction motors using information measures and artificial neural networks," *Electric Power Systems Research*, vol. 143, pp. 347–356, 2017.

[50] S. Zolfaghari, S. B. M. Noor, M. Rezazadeh Mehrjou, M. H. Marhaban, and N. Mariun, "Broken rotor bar fault detection and classification using wavelet packet signature analysis based on fourier transform and multi-layer perceptron neural network," *Applied Sciences*, vol. 8, no. 1, p. 25, 2018.

[51] R. Kumar, G. Cirrincione, M. Cirrincione, A. Tortella, and M. Andriollo, "A topological and neural based technique for classification of faults in induction machines," 2018.

[52] G. H. Bazan, P. R. Scalassara, W. Endo, and A. Goedtel, "Information theoretical measurements from induction motors under several load and voltage conditions for bearing faults classification," *IEEE Transactions on Industrial Informatics*, vol. 23, 2019.

[53] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering & System Safety*, vol. 121, pp. 43–60, 2014.

[54] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," 2008.

[55] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[56] N. K. Verma, V. K. Gupta, M. Sharma, and R. K. Sevakula, "Intelligent condition based monitoring of rotating machines using sparse auto-encoders," 2013.

[57] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, 2016.

[58] H. Liu, L. Li, and J. Ma, "Rolling bearing fault diagnosis based on STFT-deep learning and sound signals," *Shock Vibration*, vol. 2016, 2016.

[59] W. Mao, J. He, Y. Li, and Y. Yan, "Bearing fault diagnosis with auto-encoder extreme learning machine: a comparative study," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 231, no. 8, pp. 1560–1578, 2017.

[60] R. Chen, S. Chen, M. He, D. He, and B. Tang, "Reliability, rolling bearing fault severity identification using deep sparse auto-encoder network with noise added sample expansion," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 231, no. 6, pp. 666–679, 2017.

[61] M. Sohaib, C. H. Kim, and J. M. Kim, "A hybrid feature model and deep-learning-based bearing fault diagnosis," *Sensors (Basel)*, vol. 17, no. 12, p. 2876, 2017.

[62] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[63] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 185–195, 2017.

[64] H. Shao, H. Jiang, Y. Lin, and X. Li, "A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders," *Mechanical Systems and Signal Processing*, vol. 102, pp. 278–297, 2018.

[65] F. Cipollini, L. Oneto, A. Coraddu, S. Savio, and D. Anguita, "Unintrusive monitoring of induction motors bearings via deep learning on stator currents," *Procedia Computer Science*, vol. 144, pp. 42–51, 2018.

[66] P. Lin and J. Tao, "A novel bearing health indicator construction method based on ensemble stacked autoencoder," 2019.

[67] A. J. Skylvik, K. G. Robbersmyr, and H. Van Khang, "Data-driven fault diagnosis of induction motors using a stacked autoencoder network," 2019.

[68] K. Zhao, H. Jiang, X. Li, and R. Wang, "An optimal deep sparse autoencoder with gated recurrent unit for rolling bearing fault diagnosis," *Measurement Science Technology*, vol. 31, no. 1, 2019.

[69] H. Zhu, J. Cheng, C. Zhang, J. Wu, and X. Shao, "Stacked pruning sparse denoising autoencoder based intelligent fault diagnosis of rolling bearings," *Applied Soft Computing*, vol. 88, 2020.

[70] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.

[71] F. AlThobiani and A. Ball, "An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4113–4122, 2014.

[72] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," 2009.

[73] Z. Chen, C. Li, and R.-V. Sánchez, "Multi-layer neural network with deep belief network for gearbox fault diagnosis," *Journal of Vibroengineering*, vol. 17, no. 5, pp. 2379–2392, 2015.

[74] C. Zhang, J. H. Sun, and K. C. Tan, "Deep belief networks ensemble with multi-objective optimization for failure diagnosis," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 23, 2015.

[75] S. Deng, Z. Cheng, C. Li, X. Yao, Z. Chen, and R.-V. Sanchez, "Rolling bearing fault diagnosis based on Deep Boltzmann machines," 2016.

[76] J. Tao, Y. Liu, and D. Yang, "Bearing fault diagnosis based on deep belief network and multisensor information fusion," *Shock and Vibration*, vol. 2016, 2016.

[77] M. Ma, X. Chen, S. Wang, Y. Liu, and W. Li, "Bearing degradation assessment based on weibull distribution and deep belief network," 2016.

[78] D. Liu, Q. Wang, J. Tao, G. Li, and J. Wu, "fault diagnosis method based on improved deep Boltzmann machines," 2018.

[79] G. Niu, B. Zhang, P. Ziehl, F. Ferrese, and M. Golda, "Rolling element bearing fault diagnosis based on deep belief network and principal component analysis," *Proceedings of the Annual Conference of the PHM Society*, vol. 11, p. 1, 2019.

[80] H. Zhao, H. Liu, J. Xu, C. Guo, and W. Deng, "Research on a fault diagnosis method of rolling bearings using variation mode decomposition and deep belief network," *Journal of Mechanical Science and Technology*, vol. 33, no. 9, pp. 4165–4172, 2019.

[81] X. Yan, Y. Liu, and M. Jia, "Multiscale cascading deep belief network for fault identification of rotating machinery under various working conditions," *Knowledge-Based Systems*, vol. 193, p. 105484, 2020.

[82] K. Yu, T. R. Lin, and J. Tan, "A bearing fault and severity diagnostic technique using adaptive deep belief networks and Dempster-Shafer theory," *Structural Health Monitoring*, vol. 19, no. 1, pp. 240–261, 2020.

[83] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[84] O. Janssens, V. Slavkovikj, B. Vervisch et al., "Convolutional neural network based fault detection for rotating machinery," *Journal of Sound and Vibration*, vol. 377, pp. 331–345, 2016.

[85] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, "Dislocated time series convolutional neural architecture: an intelligent fault diagnosis approach for electric machine," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, 2016.

[86] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.

[87] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490–502, 2016.

[88] X. Ding and Q. He, "Energy-fluctuated multiscale feature learning with deep convnet for intelligent spindle bearing fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 1926–1935, 2017.

[89] S. Li, G. Liu, X. Tang, J. Lu, and J. Hu, "An ensemble deep convolutional neural network model with improved D-S evidence fusion for bearing fault diagnosis," *Sensors*, vol. 17, no. 8, p. 1729, 2017.

[90] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439–453, 2018.

[91] L. Guo, Y. Lei, N. Li, T. Yan, and N. Li, "Machinery health indicator construction based on convolutional neural networks considering trend burr," *Neurocomputing*, vol. 292, pp. 142–150, 2018.

[92] F. Jia, Y. Lei, N. Lu, and S. Xing, "Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization," *Mechanical Systems and Signal Processing*, vol. 110, pp. 349–367, 2018.

[93] D.-T. Hoang and H.-J. Kang, "Rolling element bearing fault diagnosis using convolutional neural network and vibration image," *Cognitive Systems Research*, vol. 53, pp. 42–50, 2019.

[94] G. Li, C. Deng, J. Wu, X. Xu, X. Shao, and Y. Wang, "Sensor data-driven bearing fault diagnosis based on deep convolutional neural networks and S-transform," *Sensors (Basel)*, vol. 19, no. 12, p. 2750, 2019.

[95] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, 2019.

[96] D. T. Hoang and H. J. Kang, "A motor current signal based bearing fault diagnosis using deep learning and information fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 23, 2019.

[97] G. Li, C. Deng, J. Wu, Z. Chen, and X. Xu, "Rolling bearing fault diagnosis based on wavelet packet transform and convolutional neural network," *Applied Sciences*, vol. 10, no. 3, p. 770, 2020.

[98] Y. Zhang, K. Xing, R. Bai, D. Sun, and Z. Meng, "An enhanced convolutional neural network for bearing fault diagnosis based on time–frequency image," *Measurement*, vol. 10, 2020.

[99] M. Skowron, T. Orlowska-Kowalska, M. Wolkiewicz, and C. T. Kowalski, "Convolutional neural network-based stator current data-driven incipient stator fault diagnosis of inverter-fed induction motor," *Energies*, vol. 13, no. 6, p. 1475, 2020.

[100] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[101] S. Hochreiter and J. Schmidhuber, "Long short-term memory neural computation 9," 1997.

[102] R. Zhao, J. Wang, R. Yan, and K. Mao, "Machine health monitoring with LSTM networks," 2016.

[103] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, 2017.

[104] Q. Cui, Z. Li, J. Yang, and B. Liang, "Rolling bearing fault prognosis using recurrent neural network," 2017.

[105] D. Xiao, Y. Huang, X. Zhang, H. Shi, C. Liu, and Y. Li, "fault diagnosis of asynchronous motors based on LSTM neural network," 2018.

[106] G. Tang, Y. Zhou, H. Wang, and G. Li, "Prediction of bearing performance degradation with bottleneck feature based on LSTM network," 2018.

[107] D. Xiao, Y. Huang, C. Qin, H. Shi, and Y. Li, "Fault diagnosis of induction motors using recurrence quantification analysis and LSTM with weighted BN," *Shock and Vibration*, vol. 2019, 2019.

[108] L. Yu, J. Qu, F. Gao, and Y. Tian, "A novel hierarchical algorithm for bearing fault diagnosis based on stacked LSTM," *Shock and Vibration*, vol. 2019, 2019.

[109] B. Zhang, S. Zhang, and W. Li, "Bearing performance degradation assessment using long short-term memory recurrent network," *Computers in Industry*, vol. 106, pp. 14–29, 2019.

[110] K. Zhao and H. Shao, "Intelligent fault diagnosis of rolling bearing using adaptive deep gated recurrent unit," *Neural Processing Letters*, vol. 23, pp. 1–20, 2019.

[111] N. Enshaei and F. Naderkhani, "Application of deep learning for fault diagnostic in induction machine's bearings," 2019.

[112] X. Li, H. Jiang, X. Xiong, and H. Shao, "Rolling bearing health prognosis using a modified health index based hierarchical gated recurrent unit network," *Mechanism and Machine Theory*, vol. 133, pp. 229–249, 2019.

[113] S. Haidong, C. Junsheng, J. Hongkai, Y. Yu, and W. Zhantao, "Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing," *Knowledge-Based Systems*, vol. 188, p. 105022, 2020.

[114] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, and D. Warde-Farley, "Generative adversarial nets in advances in neural information processing systems (NIPS)," 2014.

[115] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of generative adversarial networks (GANS): an updated review," *Archives of Computational Methods in Engineering*, vol. 23, pp. 1–28, 2019.

[116] S. Shao, P. Wang, and R. Yan, "Generative adversarial networks for data augmentation in machine fault diagnosis," *Computers in Industry*, vol. 106, pp. 85–93, 2019.

[117] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, p. 107377, 2020.

[118] H. Sagha, N. Cummins, B. Schuller, and K. Discovery, "Stacked denoising autoencoders for sentiment analysis: a review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 5, p. e1212, 2017.

[119] V. Sze, Y. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine Learning: challenges and opportunities," 2016.

[120] B. Moons, D. Bankman, and M. Verhelst, "Embedded deep learning," 2019.

[121] H. Song, *Efficient Methods and Hardware for Deep Learning*, Cadence, San Jose, UK, 2017.

[122] M. Capra, B. Bussolino, A. Marchisio, M. Shafique, G. Masera, and M. Martina, *An Updated Survey of Efficient Hardware Architectures for Accelerating Deep Convolutional Neural Networks*, Future Internet, London, UK, 2020.

[123] H. Sharma, "Bit fusion: bit-level dynamically composable architecture for accelerating deep neural network," 2018.

[124] S. Ryu, H. Kim, W. Yi, and J. J. Kim, "Area and energy-efficient precision-scalable neural network accelerator with bitwise summation," 2019.

[125] H. Kwon, A. Samajdar, and T. Krishna, "Maeri," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 461–475, 2018.

[126] W. Lu, G. Yan, J. Li, S. Gong, Y. Han, and X. F. Li, "A flexible dataflow accelerator architecture for convolutional neural networks," 2017.

[127] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 441–451, 2019.

[128] T. Pan, J. Chen, J. Pan, and Z. Zhou, "A deep learning network via shunt-wound restricted Boltzmann machines using raw data for fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 23, 2019.

[129] H. F. R. dos Santos, L. W. C. da Silva, and A. P. B. Sobral, "Forecast of multivariate time series sampled from industrial machinery sensors," *Brazilian Journal of Operations and Production Management*, vol. 17, no. 1, pp. 1–12, 2020.

[130] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5596301, 12 pages
https://doi.org/10.1155/2021/5596301

Research Article

FPGA-Based Time-Domain Channel Estimation in Gaussian Mixture Model

Muhammad Khalid,[1] **Abid Muhammad Khan**,[2] **Muhammad Rauf**,[3] **Muhammad Taha Jilani**,[4] **and Sheraz Afzal**[2]

[1]*Electrical Engineering Department, Bahria University, Karachi Campus, Karachi, Pakistan*
[2]*Electronic Engineering Department, Sir Syed University of Engineering and Technology, Karachi, Pakistan*
[3]*Department of Electronic Engineering, Dawood University of Engineering and Technology, Karachi, Pakistan*
[4]*Graduate School of Science and Engineering, PAF Karachi Institute of Economics and Technology, Karachi, Pakistan*

Correspondence should be addressed to Muhammad Khalid; mkhalid.bukc@bahria.edu.pk and Muhammad Rauf; muhammad.rauf@duet.edu.pk

Received 4 February 2021; Revised 30 March 2021; Accepted 19 April 2021; Published 3 May 2021

Academic Editor: Dao B. Wang

The performance of time-domain channel estimation deteriorates due to the presence of Gaussian mixture model (GMM) noise, which results in high mean squared error (MSE) as a challenging issue. The performance of the estimator further decreases when the complexity of the estimator is high due to the high convergence rate. In this paper, an optimized channel estimation method is proposed with low complexity and high accuracy in the GMM environment. In this channel estimation, an improved Gauss-Seidel iterative method is utilized with a minimum number of iterations. The convergence rate of the Gauss-Seidel method is improved by estimating an appropriate initial guess value when no guard bands are used in the orthogonal frequency-division multiplexing (OFDM) symbol. Simulation results provide an acceptable MSE for GMM environments, up to the probability of 5% impulsive noise component. This paper also presents the design and implementation of the proposed estimator in the NEXYS-2 FPGA platform that provides resources allocation, reconfigurability, schematic, and the timing diagram for detailed insight.

1. Introduction

In wireless communication, the performance of the system is often limited because it undergoes many unfavorable effects when the signal is transmitted to the receiver. The transmitted signals are normally scattered and reflected and arrived at the receiver through multiple paths [1]. Accordingly, this time-varying channel poses a serious challenge when it comes to estimating it in an efficient manner using the least complex method. Thus, the estimation of the channel is one of the most challenging and core issues in wireless communication. The additive white Gaussian noise (AWGN) has always been the dominant noise model in wireless communication systems, mainly because of two reasons: the first reason is that analytical manipulation is simple and the second one is due to central limit theorem. However, the presence of the impulsive nature of noise in some communication channels gives rise to non-Gaussian characteristics. In general, impulsive noise occurs randomly in the form of a sharp increase in the magnitude [2].

Research efforts, exploited for channel estimation techniques, are with the assumption of AWGN noise models [3, 4]. It is well known that this assumption is appropriate for many applications but some practical environments exist, which are incorrectly modelled by the AWGN noise model [5]. When the noise deviates from Gaussian, the performance of the existing channel estimators may deteriorate significantly because Gaussian-based estimators are sensitive to the impulsive noise [6]. The purpose of this paper is to develop a robust channel estimator that could perform well in the Gaussian noise environment, as well as in the GMM environment. The non-Gaussian noise is modelled by the well-known GMM model. This noise model characterizes a broad class of non-Gaussian noise distributions that include mainly the impulsive noise which is present in several wireless communication systems [7].

Previous studies [8–10] discuss the simplicity and robustness of the least mean square (LMS) and minimum mean square error (MMSE) algorithms based on the AWGN model. However, the performance of these estimators deteriorates due to interpolation error when the density of the pilots is decreased. Furthermore, some more renowned techniques are available in the literature, in which DVB-T2 systems are discussed. The OFDM-based second-generation terrestrial standard DVB-T2 is considered as a flexible approach for system parameters of the mobile and portable television broadcasting. The performance of DVB-T/T2 systems is investigated in [11] and compared with two mobile channel models such as VU30 (Vehicular Urban) and MR100 (Motorway Rural). In order to reduce the energy consumption and implementation complexity, a new standard is defined as DVB-T2-Lite. In paper [12], the performance of DVB-T2 is investigated and compared with DVB-T2-Lite for mobile receivers in typical urban (TU6) mobile channel. It is reported that at new lower values of code rate DVB-T2-Lite outperforms DVB-T2. The performance of the above-discussed estimators is limited by the presence of impulsive noise in a wireless communication system.

To address the impulsive nature of noise, transform domain methods are preferred. Therefore, the authors of papers [13–15] present that the transform domain approach can be helpful to recognize the variations in channel impulse response (CIR). However, the transformed domain channel estimators have relatively large computational complexity [16, 17]. First and foremost, it requires DFT and then it is followed by zero padding and IDFT operation. Besides, it requires knowledge of channel length for zero padding, which is normally not known to the receiver.

The authors in [18] presented the GMM model and GMM including censored and dropped observation (CD-GMM) for improving the performance of the Wi-Fi fingerprinting based indoor positioning system and GMM parameters are estimated. To deal with the censored and dropped data the combination of Bayesian Information Criterion (BIC) to estimate the best number of components of the CD-GMM and EM algorithm is used. The authors achieved improved positioning with the least computational cost; however, the hardware of the proposed estimator is not presented. Literature also reveals some popular state-of-the-art channel estimators based on the Bayesian learning technique [19–21]. Another state-of-the-art method is the iterative-based technique [22–28]. However, Bayesian learning techniques are complex and require prior knowledge of the initial values of estimated parameters. Similarly, the iterative based techniques suffer from high complexity problem due to high convergence rate. Consequently, these methods need a large number of hardware resources.

Hence, it is more fruitful to design a channel estimator with low convergence rate and it does not depend upon the density of the pilots in GMM environment. Accordingly, the proposed work focuses on time-domain channel estimation method, which is based on iterative technique that utilizes cyclic cross-correlation between received signal and no guard band subpilot sequence to estimate the channel impulse response. The convergence rate of the Gauss-Seidel method can be improved by selecting an appropriate value as an initial guess.

The main contributions to this article are summarized as follows:

(1) The time-domain channel estimator is presented, which utilizes an appropriate initial guess and improves the convergence rate of iterative Gauss-Seidel method in the GMM environment

(2) The FPGA implementation of the proposed channel estimation is investigated with its hardware resource requirements

(3) Simulation results provide an acceptable MSE, up to the probability of 5% impulsive noise component in GMM environment

The remainder of the manuscript is organized as follows: Section 2 discusses the system model. Section 3 presents the detailed discussion of an improved Gauss-Seidel method. Simulation parameters and performance evaluation are explained in Section 4. Section 5 covers the FPGA implementation of the proposed estimator. Section 6 presents the results and discussion. Finally, Section 7 concludes this paper.

2. System Model

Consider a baseband equivalent, band-limited system model; Figure 1 shows the OFDM Transceiver Block Diagram. In Figure 1, $D[k]$ and $P[k]$ are the frequency-domain signals, represented as follows:

$$P[k] = \sum_{N=0}^{N_{FFT}-1} p(n)e^{\left(-j2\pi kn/N_{FFT}\right)}, D[k] = \sum_{N=0}^{N_{FFT}-1} d(n)e^{\left(-j2\pi kn/N_{FFT}\right)}, \tag{1}$$

where n is the time-domain index; $n = 0, 1, \ldots, N_{FFT} - 1$ of an OFDM symbol. To reduce the correlation error in time-domain channel estimation, $P[k]$ is obtained as follows:

$$P[k] = P_1[k] + P_2[k]. \tag{2}$$

The frequency-domain signal $S[k]$ is the sum of pilot and data signal, given as

$$S[k] = P[k] + D[k]. \tag{3}$$

After the IFFT operation and adding of cyclic prefix, the signal $s\prime(n)$ is transmitted through the multipath fading channel $h(n)$, represented as

$$h(n,\tau) = \sum_{i=0}^{L-1} h_i e^{j\left(2\pi/N_{FFT}\right)Tn}\delta\left(\tau - \tau_i\right), \tag{4}$$

where h_i is the i^{th}complex path gain, L is total number of channel taps, and τ_i is the normalized path delay. Without loss of generality, it is possible to use the low-pass system

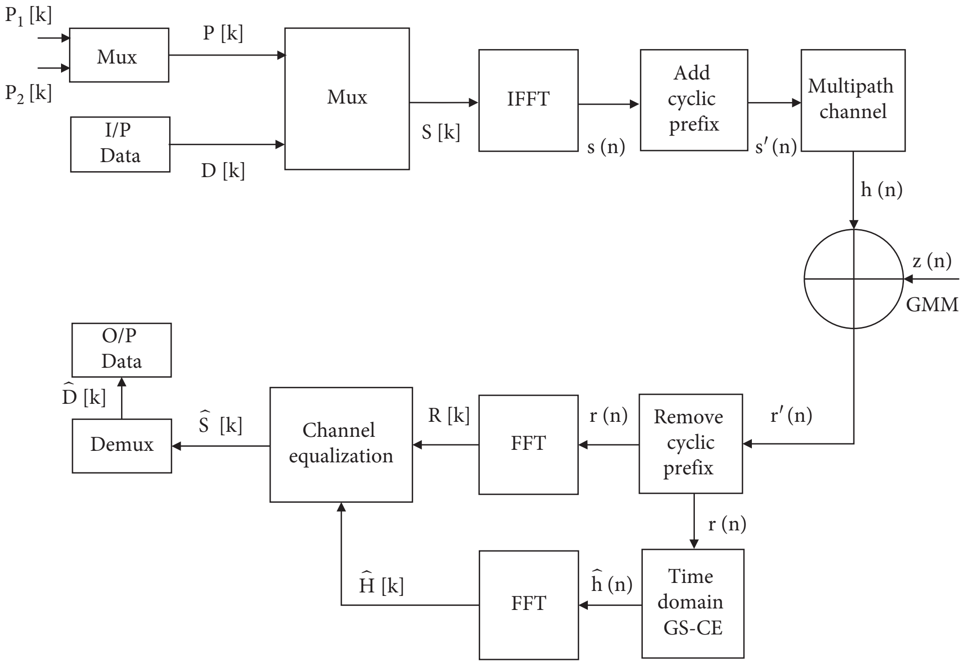


Figure 1: OFDM Transceiver Block Diagram.

model. Thus, the received signal $r(n)$ excluding cyclic prefix can be written as [29]

$$r(n) = \sum_{m=0}^{N_{FFT}-1} h(m)\left[\mathrm{d}(n-m)_{N_{FFT}} + p(n-m)_{N_{FFT}}\right] + z(n), \tag{5}$$

where m is the indexing, which is used in time domain, $(.)_{N_{FFT}}$ represents the modulo of N_{FFT}, and $z(n)$ represents the GMM noise [19], given as

$$p(z(n)) = (1-\varnothing).\mathscr{CN}\left(0,\sigma_n^2\right) + \varnothing.\mathscr{CN}\left(0,T\sigma_n^2\right), \tag{6}$$

where $T \gg 1$ represents impulsive-noise strength and $\mathscr{CN}(0,T\sigma_n^2)$ represents the Gaussian distributions with zero mean and variance σ_n^2, and $\varnothing$ is the controlling parameter of the impulsive noise level. According to equation (6), variance of GMM is obtained as

$$\sigma_z^2 = E\left\{z^2(n)\right\} = (1-\varnothing)\sigma_n^2 + \varnothing T\sigma_n^2. \tag{7}$$

Note that if $\varnothing = 0$, then $z(n)$ becomes the AWGN model. The cyclic correlation of received signal $r(n)$ with the locally generated $p_1(n)$ is given by

$$C_{rp_1}(n) = h(n)\otimes\left[C_{dp_1}(n) + C_{p_1p_1}(n) + C_{p_2p_1}(n)\right] + C_{zp_1}(n). \tag{8}$$

In general, $C_{xy}(n)$ is the cyclic correlation between x and y in equation (8). It is shown in [29] that $C_{\mathrm{d}p_1}(n) = 0$. Thus, the output of cyclic correlation becomes

$$C_{rp_1}(n) = h(n)\otimes\left[C_{p_1p_1}(n) + C_{p_2p_1}(n)\right] + C_{zp_1}(n). \tag{9}$$

The cross correlation of noise $z(n)$ with $p_1(n)$ is $C_{zp_1}(l)$, which can be represented as Gaussian mixture noise with zero mean and variance σ_z^2. Also $C_{zp_1}(l) = z(l)$. Therefore, equation (9) can be expressed as

$$C_{rp_1}(l) = h(l)\otimes\left[C_{p_1p_1}(l) + C_{p_2p_1}(l)\right] + z(l), \quad 0 \le l \le N_P - 1. \tag{10}$$

It is noted that, over the length of N_P, $C_{p_1p_1}(0)$ and $C_{p_2p_1}(0)$ act like delta functions. Thus, equation (10) can be rewritten as

$$C_{rp_1}(l) = h(l)\left[C_{p_1p_1}(0) + C_{p_2p_1}(0)\right] + \left[\sum_{\substack{n_1=0,\\ n_1\neq l}}^{L-1} h(n_1).C_{p_2p_1}(N_P + l - n_1)\right] + z(l). \tag{11}$$

To estimate the channel impulse response $h(l)$, equation (11) can be written as

$$h(l) = \frac{C_{rp_1}(l)}{C_{p_1p_1}(0) + C_{p_2p_1}(0)} - \frac{\begin{array}{c}\sum^{L-1}_{n_1 = 0,} h_{n_1}.C_{p_2p_1}(N_P + l - n_1) - z(l) \\ n_1 \neq l\end{array}}{C_{p_1p_1}(0) + C_{p_2p_1}(0)}. \tag{12}$$

It is clear from equation (12) that $h(l)$ is the difference of two components. The first component is free from windowing attribute due to having no guard bands and the second component contains windowing attribute. If transmitted OFDM signal does not contain guard band subcarriers, then $h(l)$ can be simplified as

$$h(l) = \frac{C_{rp_1}(l)}{C_{p_1p_1}(0) + C_{p_2p_1}(0)}, \tag{13}$$

where equation (13) is termed as initial guess for $h(l)$. However, in most practical systems, the transmitted frequency-domain signal contains guard band subcarriers at their respective positions. Therefore, $h(l)$ contains both components. The most straightforward approach for solving equation (13) for $h(l)$, is by using deconvolution. But the problem with the deconvolution is that not only is it computationally complex but also it requires matrix inversion that can have null values in denominator. Consequently, it increases hardware implementation cost and hence the performance of receiver deteriorates [30]. Due to these reasons, the channel estimation using deconvolution is not practically desirable. Herein, the following method is proposed.

3. Improved Gauss-Seidel Method

In this section, channel impulse response is estimated by using equation (12). For this purpose, Gauss-Seidel iterative method is utilized. Moreover, the convergence rate of Gauss-Seidel method is improved by selecting an appropriate initial guess, such as using component which is without windowing attribute. Therefore, we have the following:

(1) The estimated initial guess $\widehat{h}_l^{(k)}$ (with no guard bands) is given by

$$\widehat{h}_l^{(k)} = \frac{C_{rp_1}(l)}{C_{p_1p_1}(0) + C_{p_2p_1}(0)}, \quad \text{where } k = 0. \tag{14}$$

(2) To estimate the next iterative value $\widehat{h}_l^{k+1}$, substitute the values of $\widehat{h}_l^{k}$ in equation (12).

$$\widehat{h}_l^{(K+1)} = \widehat{h}_l^{k} - \frac{\sum^{L-1}_{\substack{n_1=0,\\ n_1\neq l}} \widehat{h}_{n_1}^{k}.C_{p_2p_1}\left(N_p - l - n_1\right)}{C_{p_1p_1}(0) + C_{p_2p_1}(0)}. \tag{15}$$

(3) Similarly, for further iterative value,

$$\widehat{h}_l^{(k+2)} = \widehat{h}_l^{(k+1)} - \frac{\sum^{L-1}_{\substack{n_1=0,\\ n_1\neq l}} \widehat{h}_{n_1}^{k+1}.C_{p_2p_1}\left(N_p - l - n_1\right)}{C_{p_1p_1}(0) + C_{p_2p_1}(0)}. \tag{16}$$

(4) After each iteration, it keeps checking whether the difference in error is less than the tolerance.

$$\frac{\widehat{h}_l^{(k+1)} - \widehat{h}_l^{(k)}}{\widehat{h}_l^{(k+1)}} = \epsilon_r. \tag{17}$$

(5) Iteration process continues till the difference in errors is not smaller than the tolerance ϵ_r.

4. Performance Evaluation

This section evaluates the performance of the proposed estimator DVB-T2 system in fixed reception scenario over Rayleigh (RL20) channel model, where MSE and BER are estimated for AWGN and GMM channel environments. In order to estimate the channel estimation alone, a perfect synchronization can be assumed, and a rectangular window is implemented by using N_P number of null pilot subcarriers in the frequency domain as a guard band. In OFDM systems, quadrature amplitude modulation (QAM) is used extensively, due to having capability of acquiring higher spectral efficiency by setting suitable constellation size. For simulation, data constellation is chosen to be 4-QAM and 16-QAM, while pilot subcarriers are unipolar BPSK modulated and boosted with a factor of 16/9. The average transmitted power of symbol is normalized to 1. For fairly accurate communication, the range of SNR is selected from 0 dB to 30 dB for MSE and from 0 to 25 dB for BER performance. Simulation is performed 1000 times for each SNR. The simulation parameters are shown in Table 1.

5. FPGA Implementation

This section presents the design and implementation of the proposed channel estimator by employing the GS method in NEXYS-2 FPGA that provides reconfigurability, resource allocation, and the timing diagram. A method of implementation of the FPGA scheme has been presented in a recent study [32, 33]. The hardware implementation of the proposed estimator is implemented in two phases as shown in Figure 2.

Phase I starts with coding and verification of the proposed estimator in MATLAB 9.2 R2017a. It is then translated

Table 1: Simulation parameters taken from [31].

Parameters		DVB-terrestrial
FFT points (N_{FFT}) mode		2048
Symbol duration		224 us
Guard interval (N_G)		1/16, 14 us
Pilots spacing (N_F)		16
Number of pilots (N_P)		128
Signal constellation $D[k]$		4 QAM, 16 QAM
Code rate		1/2, 1/3
FFT sampling frequency F_s		9.14 MHz
Channel bandwidth		8 MHz
Nonzero channel taps of Rayleigh fading channel		14
Pilot pattern		16 (3) = 48
FPGA inputs	Pilot subcarrier	2
	Received signal	1

into VHDL code by using HDL code generation. In VHDL code generation, the HDL coder assigns a workflow advisor that inquires two files: one is function file (contains the main algorithm) and the second one is test bench file (contains inputs). The workflow advisor first defines the input type and then translates the input values into the fixed point from floating point. Finally, the MATLAB code is translated into HDL code as shown in Figure 2. The estimator is translated into HDL using HDL coder in MATLAB, and then this code is further checked and verified by Xilinx 14.2.

Phase II explains the design process execution of FPGA as shown in Figure 2. In Xilinx, the code endures the four main steps. In the first step, design functionality has been verified and checked through a test bench waveform. Simulation results also help to transform into RTL abstraction which describes the algorithm to its circuit-level concept describing the components at the gate-level abstraction that can be useful in hardware design. If the complex operation at the circuit level is not justified and efficient enough in terms of multipliers and address, then the circuit description is changed and simulations are performed again to ensure that the conceptualized approach does function appropriately. In the second step, the HDL code is synthesized, which provides the translation of the code into a gate-level net-list. Furthermore, the postsynthesis simulation verifies and confirms the design results at the gate and flip-flop level to prove the transparency of gate-level abstraction translation. In the third step, the generated gate-level abstraction undergoes route and placement procedure. Finally, a programming file is generated, which implements the proposed design on the FPGA board.

Experimental setup of the system is shown in Figure 3. The results are verified on the FPGA NEXY-2 board, in which the following parameters are analyzed.

5.1. Resource Allocation. The computation complexity of the estimator is analyzed by using the mathematical expression. The complex mathematical operation is defined in terms of fixed point generation by using floating point to fixed point conversion in MATLAB HDL coder. This real conversion of complex operation has various levels of complexity when defining in terms of hardware design. Table 2 defines the hardware complexity of circuit at gate level of abstraction by using different mathematical operations such as multiplication/division and addition/subtraction.

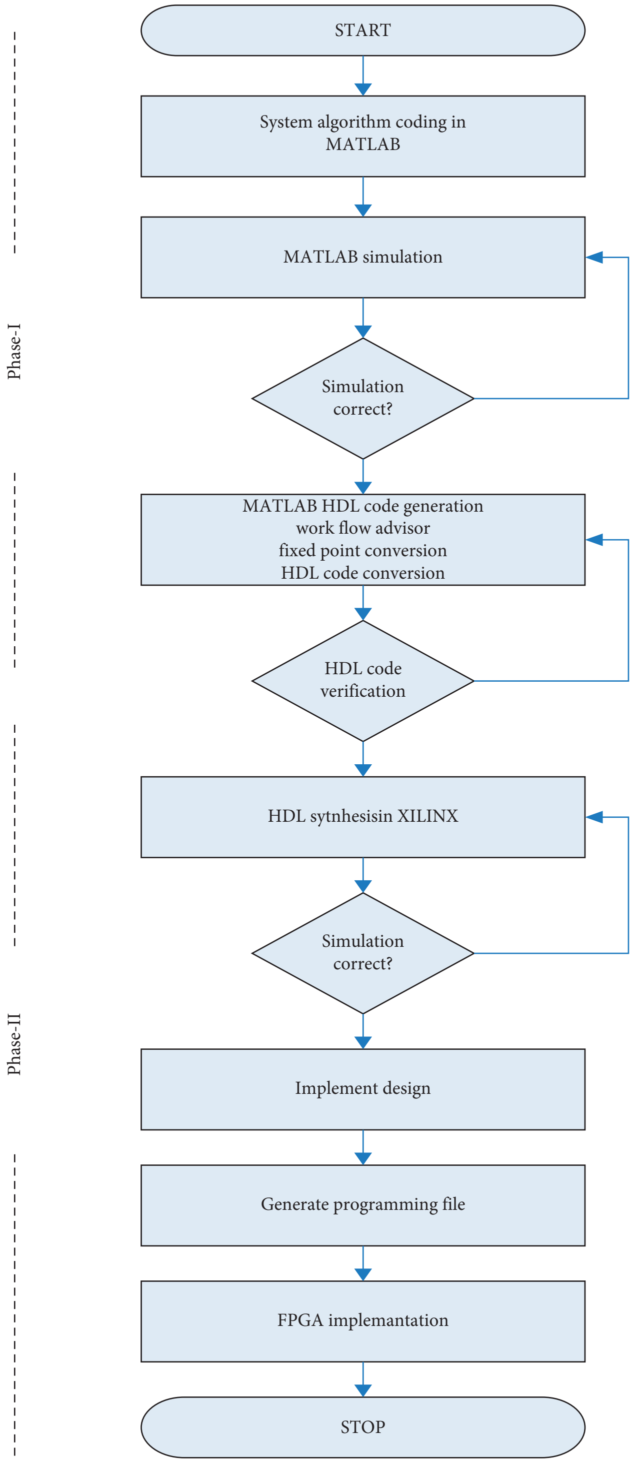


Figure 2: Design flow of the system.

5.2. Schematic Diagram. Figure 4 shows the schematic diagram for GS estimator, in which three input variable are

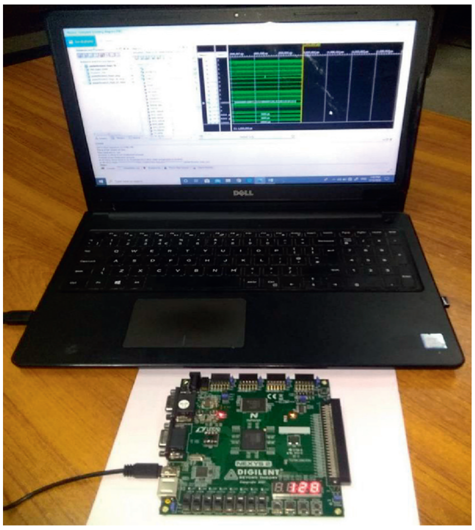


Figure 3: Experimental setup on FPGA implementation.

Table 2: Hardware recourses allocation.

Real operations	Operation counts
Multiplication/division	5
Addition/subtraction	112

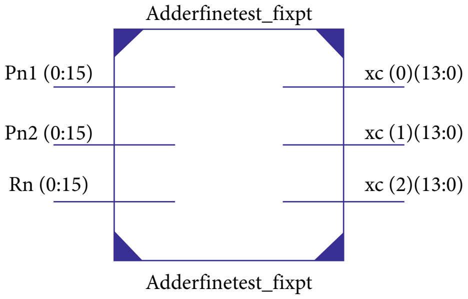


Figure 4: Schematic diagram for the GS estimator.

used: two for pilot subcarriers $P_1(n), P_1(n)$ and one for received signal $R(n)$. However, the three output variables (nonzero channel coefficients) are estimated as shown in Table 1.

5.3. Timing Diagram. Figure 5 represents the timing simulation for the estimator in the GMM environment. It shows the closest emulation that is actually downloaded to device. It allows inspecting all functional and timing requirements for the device. Random bit-oriented input variables for both pilot and received signals are taken on which the outputs are verified.

6. Results and Discussion

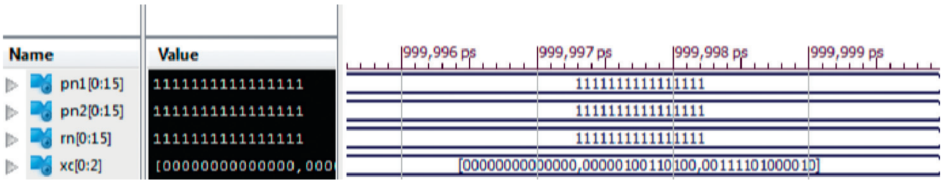
In this section, first of all, the Gauss-Seidel based time-domain channel estimation method is tested at 13 dB SNR. Figure 6 depicts the MSE of the CIR, which is plotted against the number of iterations. The legends AWGN pilot Amp. = 1 and GMM pilot Amp. = 1 represent the MSE performance with a unity pilot amplitude in AWGN and GMM, respectively. The general behaviour of two legends shows that the MSE converges in one iteration. This is due to the selection of appropriate initial guess as discussed in the last two paragraphs of Section 2. In other words, the larger part of the actual CIR solution is present in an initial guess (no guard bands) term. Therefore, the actual solution of channel estimation is the small correction to the solution with an initial guess. The two legends AWGN pilot Amp. = 2 and GMM pilot Amp. = 2 represent the MSE performance with a pilot amplitude of boosted factor 2. In this case, it is analyzed that the number of iterations increases, and the MSE converges in three iterations as represented by circles. This is due to the higher difference between the correlation error terms arising from $C_{p_2 p_1}(l)_{l\neq 0}$ and $C_{p_1 p_1}(l) + C_{p_2 p_1}(l)_{l=0}$ in equation (12).

Figures 7–9 depict the MSE performance of the proposed channel estimator for the GMM channel with three different values of $\varnothing$ (controlling parameter). Three different values of $\varnothing$ are considered for analyzing the effect of higher probabilities of the impulsive noise component. In Figures 7–9, the MSE performances of existing channel estimators (DFT and MMSE) are compared with the proposed methods (initial guess and GS estimator) with 16-QAM modulation scheme. It is evident that the performance of the GS estimator outperforms those of the other estimators. It can be further seen that the difference of MSE is dominant at the high SNR regime. The comparison of the three figures depicts that the MSE difference between an initial guess and GS estimator reduces at higher values of $\varnothing$. In GMM, there are two error components present in the estimation of CIR. The first one is due to the AWGN and the second one is due to the impulsive noise as discussed in equation (7). At a high value of $\varnothing$, the impulsive noise component is relatively dominant and the MSE of GS estimator is increased.

To observe the effect of the M-order modulation scheme in the GMM environment, the performance of the proposed estimator is evaluated with the 4-QAM in Figures 10 and 11. A similar trend is found with the 4-QAM modulation scheme as observed with 16-QAM. It is also evident that the lower-order modulation scheme improves MSE performance. Furthermore, in the presence of GMM noise, the MSE performance degraded when the value of the controlling parameter $\varnothing$ is increased from 0.01 to 0.05.

Figures 12–14 show the bit error rate (BER) curves for GMM environments. In BER performance, the convolution code with the code rates of 1/2 and 1/3 is used. The Viterbi algorithm is utilized to decode the convolutional code. Results show that the BER performance of the proposed estimator is improved and a similar trend of MSE is found as discussed in Figures 7–9, respectively.

The low code rate not only increases the error correction capabilities but also increases overhead. The code rate is represented by a number (k/n) and it measures the efficiency of the code. The performance of the proposed estimator is also evaluated with 1/3 code rate in Figures 15 and 16. Simulation results show that BER decreases with a lower code rate.



Name	Value
pn1[0:15]	1111111111111111
pn2[0:15]	1111111111111111
rn[0:15]	1111111111111111
xc[0:2]	[00000000000000,000

Figure 5: Timing diagram.

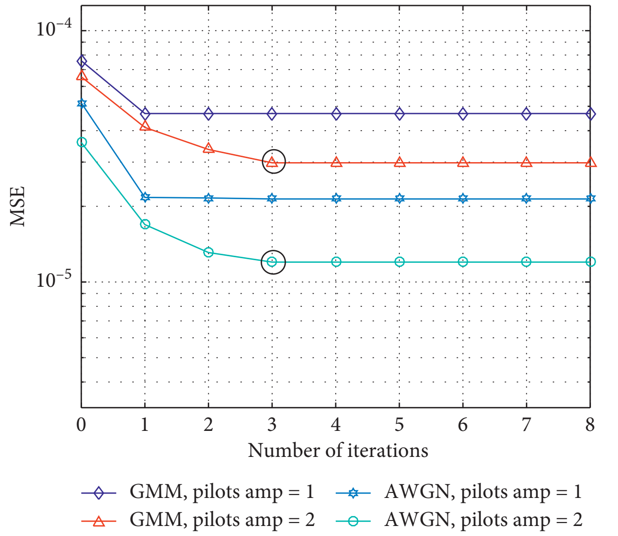


Figure 6: Number of iterations at different pilot's amplitudes.

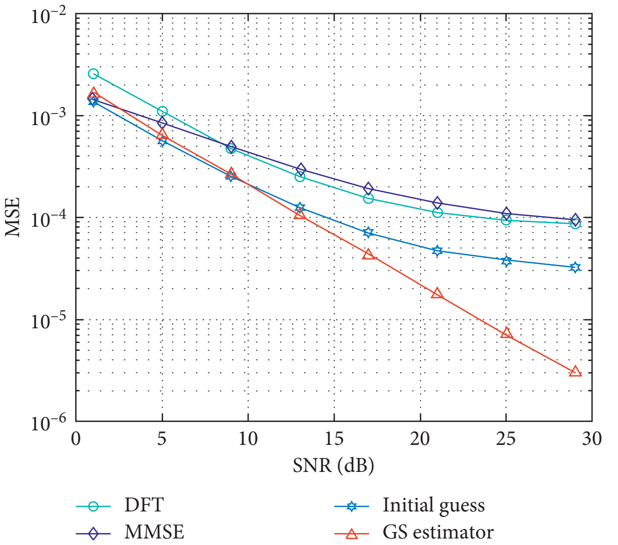


Figure 7: Performance of the estimator in AWGN channel with $\varnothing = 0, T = 0$, 16-QAM.

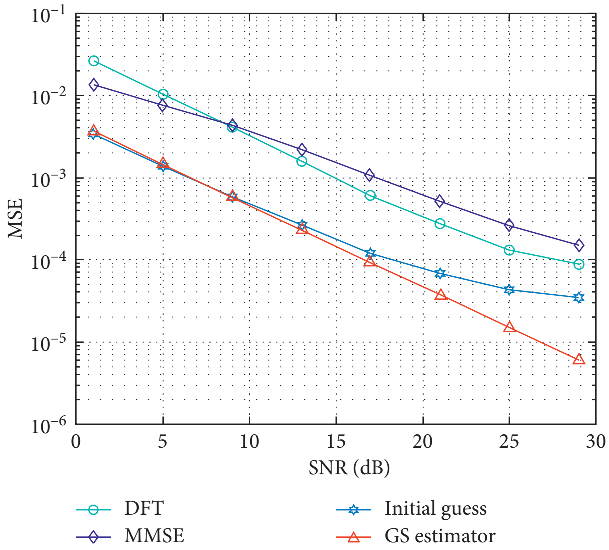


Figure 8: Performance of the estimator in GMM channel with $\varnothing = 0.01, T = 100$, 16-QAM.

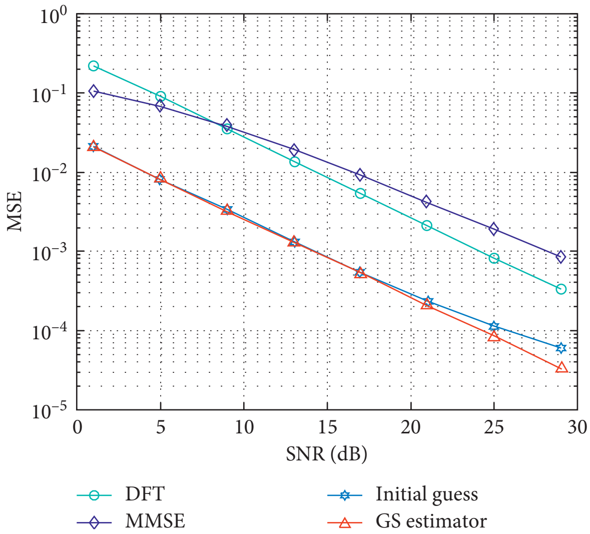


Figure 9: Performance of the estimator in GMM channel with $\varnothing = 0.05, T = 100$, 16-QAM.

In Figures 17 and 18, the effects of the boosting factor of the pilot subcarrier for the proposed time-domain Gauss–Seidel based estimator and the conventional frequency-domain MMSE channel estimator are analyzed, respectively. In both figures, the numbers of pilot subcarrier vary from 32 to 128. It can be seen that the MSE improves when the number of pilot subcarrier increases; however, spectral efficiency deteriorates from 98.4% to 93.8% with the unity boost factor of the pilot subcarrier. It is also observed

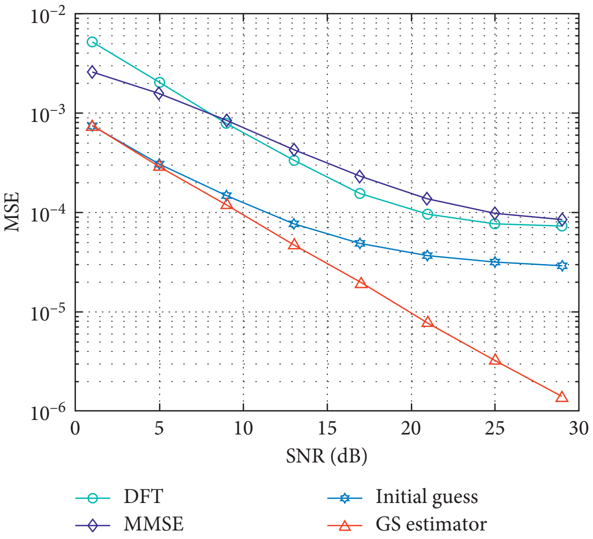


FIGURE 10: Performance of the estimator in GMM channel with $\varnothing = 0.01, T = 100$, 4-QAM.

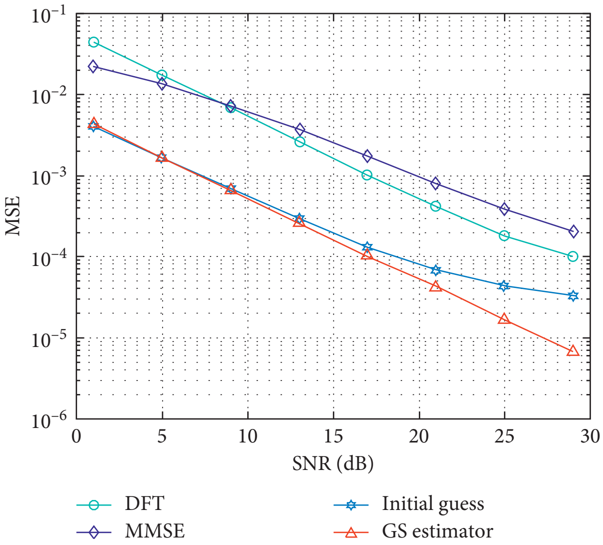


FIGURE 11: Performance of the estimator in GMM channel with $\varnothing = 0.05, T = 100$, 4-QAM.

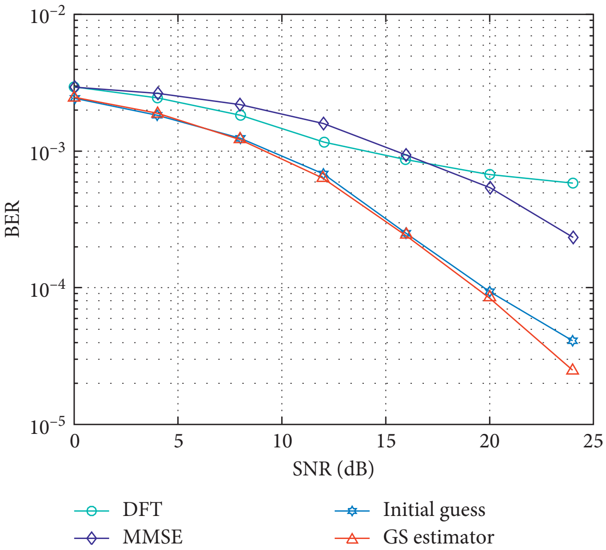


FIGURE 12: BER performance of the estimator in an AWGN channel with $\varnothing = 0, T = 0$ and 16-QAM, code rate = 1/2.

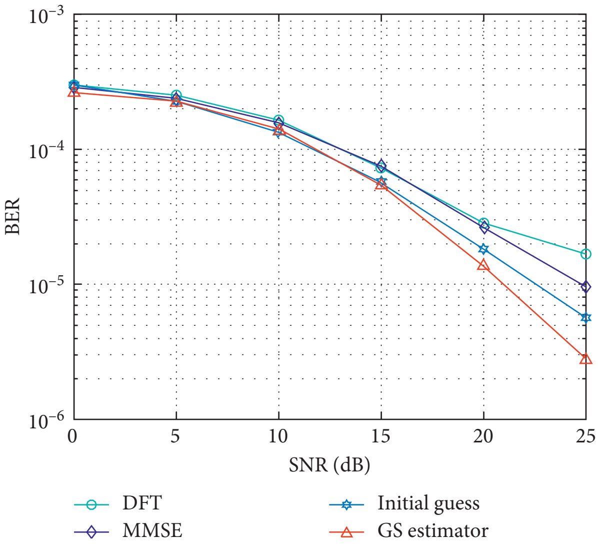


Figure 13: BER performance of the estimator in GMM channel with $\varnothing = 0.01, T = 100$ and code rate = 1/2.

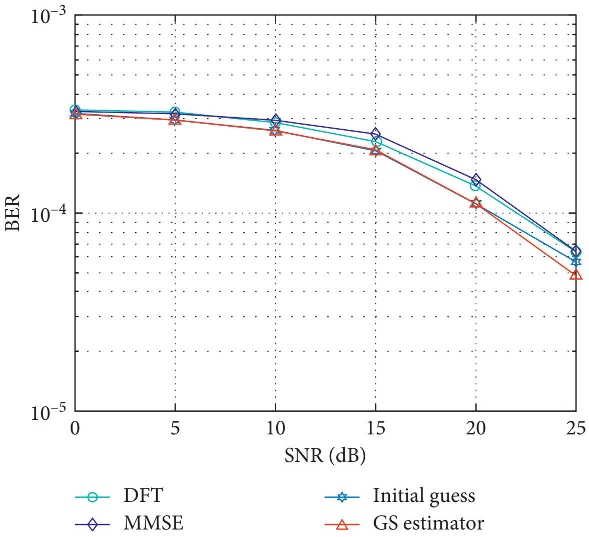


Figure 14: BER performance of the estimator in GMM channel with $\varnothing = 0.05, T = 100$ and 16-QAM, code rate = 1/2.

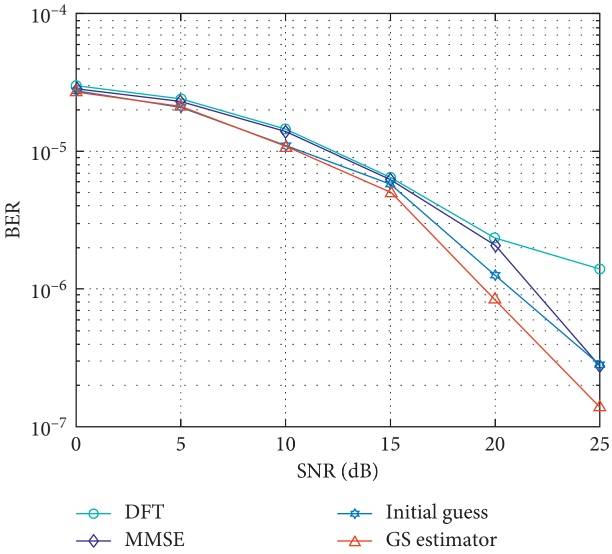


Figure 15: BER performance of the estimator in GMM channel with $\varnothing = 0.01, T = 100$ and code rate = 1/3.

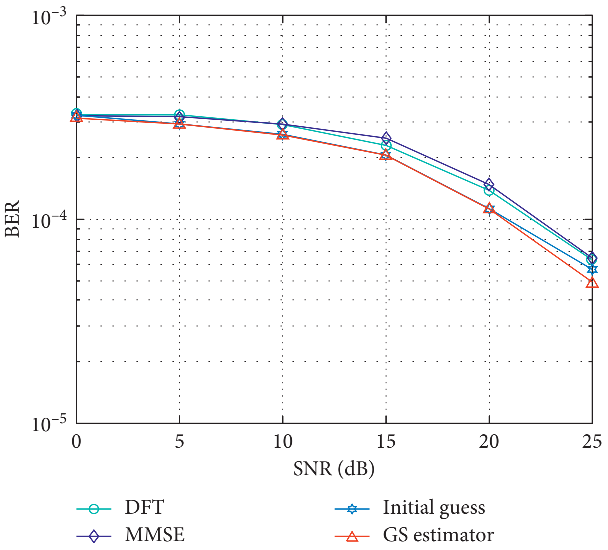


Figure 16: BER performance of the estimator in GMM channel with $\varnothing = 0.05, T = 100$ and code rate = 1/3.

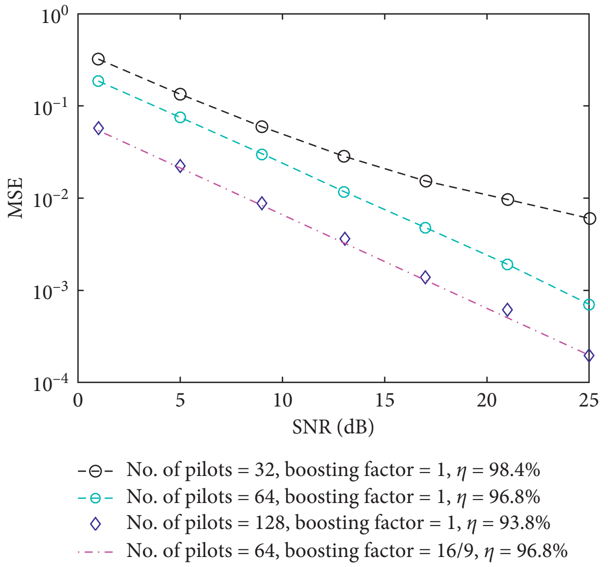


Figure 17: Effect of boosting factor of pilot subcarrier for the proposed time-domain estimator.

that the minimum MSE is achieved when 128 pilot subcarriers are used. In Figure 17, it can be seen that a similar MSE and higher spectral efficiency can be achieved when 16/9 boosted power level of pilot subcarriers is used with 64 pilot subcarriers for the proposed estimator. However, the impact of the boosting factor of pilot subcarriers is not prominent in frequency-domain channel estimator, especially at the high SNR regime in Figure 18. The performance of frequency-domain estimators mainly depends upon interpolation, which decreases with the density of the pilots. Hence, the impact of the boosting factor of the pilot subcarriers can be achieved in the time-domain channel estimation schemes only.

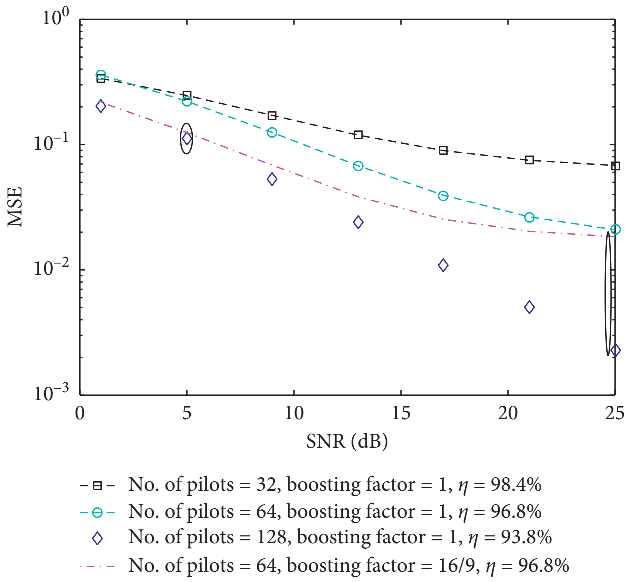


Figure 18: Effect of boosting factor of pilot subcarrier for the frequency-domain MMSE estimator.

7. Conclusion

The presented work investigates an efficient and low-complexity Gauss–Seidel based time-domain channel estimation method for AWGN and GMM environments, where the Gauss–Seidel method is utilized to obtain a high convergence rate by estimating an appropriate initial value. The simulation results show that the proposed estimator achieves much better accuracy compared to DFT and MMSE algorithms. Moreover, the effect of the boosting factor of the pilot subcarriers is analyzed in the proposed time-domain estimator. The computation complexity of the presented estimator is analyzed on FPGA, which shows the number of hardware recourses' utilization, schematic diagram, and timing diagram. Furthermore, it provides acceptable results for GMM environments, up to the probability of impulsive component of 5%. Therefore, one would like to extend the presented work by considering higher probabilities of the impulsive noise component. Another possible extension of this work is to investigate the SNR estimation in a similar environment.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This experimental work was supported by Higher Education Commission (HEC), Pakistan, under Grant 1710/SRGP/R&D/HEC for Sir Syed University of Engineering and Technology, Karachi.

References

[1] L. Gordon, *Stuber,"Principles of Mobile Communication"*, Kluwer Academic Publication, New York, NY, USA, 2nd edition, 2002.

[2] T. Zhang and G. Gui, “Sign function based sparse adaptive filtering algorithms for robust channel estimation under non-Gaussian noise environments,” *Algorithms*, vol. 9, no. 3, p. 54, 2016.

[3] A. M. Khan and V. Jeoti, “On time-domain OFDM channel estimation: use of pilots correlation for digital video broadcasting (DVB) cable receiver,” *Optoelectronics and Advanced Materials- Rapid Communications*, vol. 9, no. 5, pp. 891–897, 2015.

[4] Y. Wang, Z. Zhang, and Y. Chen, “Pilot-aided joint channel estimation for OFDM based cooperative multi-cell networks,” *Telkomnika Indonesian Journal of Electrical Engineering*, vol. 12, pp. 6860–6867, 2014.

[5] R. J. Kozick and B. M. Sadler, “Maximum-likelihood array processing in non-Gaussian noise with Gaussian mixtures,” *IEEE Transactions on Signal Processing*, vol. 48, no. 12, pp. 3520–3535, 2000.

[6] D. Middleton, “Non-Gaussian noise models in signal processing for telecommunications: new methods an results for class A and class B noise models,” *IEEE Transactions on Information Theory*, vol. 45, no. 4, pp. 1129–1149, 1999.

[7] G. Gui, Li Xu, W. Ma, and B. Chen, “Robust adaptive sparse channel estimation in the presence of impulsive noises,” in *Proceedings of the 2015, IEEE International Conference on Digital Signal Processing (DSP)*, pp. 628–632, Singapore, Singapore, July 2015.

[8] E. Hari Krishna, K. Sivani, and K. Ashoka Reddy, “OFDM channel estimation using novel LMS adaptive algorithm,” in *Proceedings of the 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, January 2017, Chennai, India.

[9] D. N. Bhange and C. G. Dethe, “Performance analysis of channel estimation techniques for proposed pilot aided 3D-OFDM DVB-T2 system,” *International Journal of Computer Science and Information Security*, vol. 14, no. 9, 2016.

[10] M. Aparna, P. V. Vineetha, and S. Kirthiga, “Channel modeling and estimation of polarized mimo for land mobile satellite systems,” in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, September 2017.

[11] L. Polak and T. Kratochvil, “Exploring of the DVB-T/T2 performance in advanced mobile tv fading channels,” in *Proceedings of the 2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, Rome, Italy, July 2013.

[12] D. A. Samo, M. Slimani, G. Baruffa, and L. Rugini, “A performance study of DVB-T2 and DVB-T2-Lite for mobile reception,” *Digital Signal Processing*, vol. 37, pp. 35–42, 2015.

[13] A. T. Muharrem and E. Perrins, “Pilot based channel estimation and transform domain analysis in broadband PLC for smart grid,” in *Proceedings of the 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 283–288, Vancouver, Canada, October 2013.

[14] Y. Zhao and A. Huang, “A novel channel estimation method for ofdm mobile communication systems based on pilot signals and transform-domain processing,” in *Proceedings of the 1997 IEEE 47th Vehicular Technology Conference Technology in Motion*, pp. 2089–2093, Phoenix, AZ, USA, May 1997.

[15] M. Zourob and R. Rao, “2×1-D fast fourier transform interpolation for lte-a ofdm pilot-based channel estimation,” in *Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, UAE, November 2017.

[16] H.-M. Wang, J.-C. Jiang, and Y.-N. Wang, “Model refinement learning and an example on channel estimation with universal noise model,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 31–46, 2021.

[17] Y. Qiang, X. Shao, and X. Chen, “A model-driven deep learning algorithm for joint activity detection and channel estimation,” *IEEE Communications Letters*, vol. 24, no. 11, pp. 2508–2512, 2020.

[18] V. U. Trung Kien, K. H. Manh, and L. E. Hung Lan, “Performance enhancement of wi-fi fingerprinting-based ips by accurate parameter estimation of censored and dropped data,” *Radio Engineering*, vol. 28, no. 4, pp. 740–748, 2019.

[19] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, “Channel estimation for massive MIMO using Gaussian-mixture bayesian learning,” *IEEE Transactions on Wireless Communications*, vol. 20, pp. 1356–1368, 2014.

[20] J. Lin, M. Nassar, B. L. Evans, and B. L. Evans, “Impulsive noise mitigation in powerline communications using sparse bayesian learning,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1172–1183, 2013.

[21] T. Zhang and G. Gui, “Reweighted factor selection for SLMS-RL1 algorithm under Gaussian mixture noise environments,” *Algorithms*, vol. 8, no. 4, pp. 799–809, 2015.

[22] G. Gui, Li Xu, and N. Shimoi, “Stable sparse channel estimation algorithm under non-Gaussian noise environments,” in *Proceedings of the 2015 21st Asia-Pacific Conference on Communications (APCC)*, pp. 561–565, Kyoto, Japan, October 2015.

[23] W. Kim, H. John, and L. Hansen, “Advanced parallel gaussian mixture model based feature compensation integrated with iterative channel estimation,” *Speech Communication*, vol. 73, pp. 81–93, 2015.

[24] R. Liu, T.-L. Kung, and K. K. Parhi, “Impulse noise correction in OFDM systems,” *Journal of Signal Processing Systems*, vol. 74, no. 2, pp. 245–262, 2014.

[25] Z.-Q. He, H. Li, Z.-P. Shi, J. Fang, and L. Huang, “A robust iteratively reweighted l2 approach for spectral compressed sensing in impulsive noise,” *IEEE Transactions on Broadcasting*, vol. 46, pp. 215–220, 2017.

[26] K. Ogura, A. Nakamura, and M. Itami, “A study on class an impulsive noise cancellation and channel estimation under Rayleigh fading environment,” in *Proceedings of the 2018, IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-2, Las Vegas, NV, USA, January 2018.

[27] K. Yan, H. Zhang, and H.-C. Wu, “Robust multipath channel estimation in the presence of impulsive noise,” *IET Communications*, vol. 12, no. 2, pp. 228–235, 2018.

[28] B. Partap and P. Singh, “Hybrid approach for channel estimation using iterative compensation and LMS algorithm,” *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 34–37, 2018.

[29] L. Mingqi, T. Jianguo, and Z. Wenjun, “A channel estimation method based on frequency-domain pilots and time-domain processing for OFDM systems,” *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 1049–1057, 2004.

[30] W. Luo, Q. Ou, Y. Fei, L. Cui, and J. Jin, “Analysis of a new hidden attractor coupled chaotic system and application of its weak signal detection,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 8849283, 15 pages, 2020.

[31] ETSI TS 102 831 V1.2.1 (2012-08), *Digital Video Broadcasting (DVB); Implementation Guidelines for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*, ETSI, Cannes, French, 2012.

[32] R. Wang, C. Li, S. Çiçek, K. Rajagopal, and X. Zhang, "A memristive hyperjerk chaotic system: amplitude control, FPGA design, and prediction with artificial neural network," *Complexity*, vol. 2021, Article ID 6636813, 17 pages, 2021.

[33] N. Wu, A. Zain Anwar, M. Mujtaba Shaikh, M. Rehan Yahya, and M. Aamir, "Compact and high speed architectures of KASUMI block cipher," *Wireless Personal Communications*, vol. 106, no. 4, pp. 1787–1800, 2019.

Hindawi
Mathematical Problems in Engineering
Volume 2023, Article ID 9825281, 1 page
https://doi.org/10.1155/2023/9825281

Retraction

Retracted: Internet of Things (IoT)-Enabled Unmanned Aerial Vehicles for the Inspection of Construction Sites: A Vision and Future Directions

Mathematical Problems in Engineering

Received 13 September 2023; Accepted 13 September 2023; Published 14 September 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] A. Israr, G. E. M. Abro, M. Sadiq Ali Khan, M. Farhan, and S. u. A. Bin Mohd Zulkifli, "Internet of Things (IoT)-Enabled Unmanned Aerial Vehicles for the Inspection of Construction Sites: A Vision and Future Directions," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9931112, 15 pages, 2021.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 9931112, 15 pages
https://doi.org/10.1155/2021/9931112

Review Article

Internet of Things (IoT)-Enabled Unmanned Aerial Vehicles for the Inspection of Construction Sites: A Vision and Future Directions

Ambar Israr,[1] Ghulam E. Mustafa Abro,[2] M. Sadiq Ali Khan,[3] Muhammad Farhan,[4] and Saif ul Azrin Bin Mohd Zulkifli[2]

[1]*Electronic Engineering Department, Sir Syed University of Engineering & Technology, Karachi, Pakistan*
[2]*Department of Electrical and Electronic Engineering, University Teknologi Petronas, Seri Iskandar Perak 32610, Malaysia*
[3]*Department of Computer Science, University of Karachi, Karachi, Pakistan*
[4]*Department of Electrical Engineering and Technology, Government College University, Faisalabad, Pakistan*

Correspondence should be addressed to Ghulam E. Mustafa Abro; ghulam.engr@gmail.com and Muhammad Farhan; mfarhan@gcuf.edu.pk

Received 11 March 2021; Revised 12 April 2021; Accepted 19 April 2021; Published 28 April 2021

Academic Editor: Dao B. Wang

Today people are witnessing the rapid evolvement in every area. This is because of the emerging trends in communication technology and autonomous unmanned vehicles. These trends have led us towards the high standards of health, energy, transportation, monitoring, and surveillance of huge domestic and industrial projects. Thus, this review paper presents the integration of the latest trend in communication technology, i.e., Internet of things (IoT) with unmanned aerial vehicles (UAVs). This manuscript not only reviews the use of IoT-enabled unmanned aerial vehicles for inspecting the several construction sites but also emphasizes the utilization of such IoT-enabled autonomous aerial vehicles for ensuring the health and safety measures at the site. It discusses the major limitations and shortcomings of state-of-the-art techniques for the same purpose, i.e., optimization issues in path planning, lightweight artificial intelligence (AI) and computer vision algorithms, coordination in communication using IoT, and scalability of IoT network. Thus, this paper shall help the reader to explore different open research problems in-depth.

1. Introduction

It is observed that the population of this planet is rapidly increasing, and it is an intellectual guess; this number is likely to double by the 2060s. Today's modern facilities of metropolitan cities have attracted and even compelled many of the families and individuals to migrate from their native rural places to these urban and smart cities. This movement from rural areas to smart cities has brought several challenges for the local city government to ensure the standard and quality of life. In such conditions, one of the options for administrative staff is the effective and smart use of the Internet of robotic things (IoRT) [1]. Apart from managing the resources, these IoRT strategies will be used effectively in monitoring the progress of various online and offline construction sites. The terms online and offline construction sites illustrate the nature of site such that the sites where material for development is purchased from other out-sources that are far, are known or labelled as online construction sites whereas the sites where the material is produced, i.e., concrete, are known as offline construction sites. Such utilization of IoT-enabled drones or IoRT will enhance the level of updating the progress to the supervisor or site manager within a short time. There are several IoT-oriented applications that one may find in the previous research work [2]. These applications have been proposed to improve several domains, i.e., public health sector, mobility of goods, water resource management, waste management, smart environment, and government [3].

In this paper, the authors performed a deep review of the published works regarding IoT-enabled drones and shall discuss the technical shorting comings of the topic. Furthermore, this manuscript covered different strategies (latest and futuristic) that will support the use of IoT-enabled UAVs in applications. While studying the different research papers, it has been observed that these UAVs are utilized in several applications as shown in Figure 1. Historical background shares that such drones were initially used by military services, but nowadays these UAVs with IoT add-on also used a lot in civil applications such as infrastructure inspection, public safety, traffic control, agriculture, and crop health surveillance, leading us to enjoy several perks. The usage of UAVs in the construction industry will be a cost-effective solution [3] because of the flexible manuovers of UAVs at most of the dull, dirty, difficult, and even dangerous locations. This makes the proposed solution one of its kinds. These IoT-enabled drones can fly over the objects with accurate altitude and acquire the best quality pictures using a high-resolution camera to further process it via machine algorithm and IoT like applications to update the progress of any construction site either it is connected or not connected. For nonconnected sites, the number of drones can be increased and may work in master-slave combination [4] to update the results whereas for connected sites, the update can be obtained using the proper flight schedule [4].

The modern standards for the quality of life demand improvement in utilizing the resources smartly, increasing sustainability, and reducing the negative effects on the environment. This led us to the innovations in the field of information and communication technologies, which includes smart environment, smart living, smart economy, smart people, smart mobility, smart governance, and smart tourism. Such trends are witnessed in various cities of the world, i.e., Malta in the Republic of Malta, Dubai in UAE, Kochi in India, San Jose in California, the United States (US), and several others in Europe. Many other countries are also focusing ongoing with the flow; for example, Dubai has planned the same way of evolution [4]. A similar type of project is also initiated by the government of Saudi Arabia named "Vision 2030" [5]. In such modernization, IoT-enabled drones have got lots of popularity except for the construction site where the usage of such drones can be exceptionally fruitful in detecting the issues that matter for construction site performance. A site engineer always wishes to have sustainability of site work while managing the number of resources to get the maximum efficiency [6]. Certain defects can never be estimated quickly and with conventional methods, i.e., defects in the envelopes of pillars of a building, air leakages, moisture/humidity, and damage of thermal insulations that lead to degradation of site performance [7] and increasing the greenhouse gases. Currently, it has been observed that approximately 65% of energy is dissipated from modern housing schemes [8] and 72% from the industrial area in central Europe [9]. Thus, a novel method should be introduced to monitor and control this loss. Drones in this regard may provide comprehensive results such that with the help of thermal cameras, they can detect infrared (IR) radiations [10]. This detection of IR radiations is known as thermography which is prescribed in several research works for identifying the energy-efficient buildings within the site [11].

One may see several research works based on thermographic mapping and imaging techniques where a dense image is modelled to conclude the thermal energy loss from any site or building [12]. Such algorithms produce quality thermal resolution-oriented images, as shown in Figure 2 to minimize the long hours that are required to check the quality and assess the defects, specifically the CO_2 emission as illustrated in Figures 2(b) and 2(c) whereas Figure 2(a) is the normal figure. The only constraint technique is the time consumption in acquiring the 3D model of the site due to the terrestrial scanning method. Apart from this limitation, the technique can be utilized for monitoring the damages, i.e., missing insulation in building, types of cracks, moisture defects, and heat losses.

In order to present a comprehensive way to monitor the construction sites, this paper shows the support for IoT platform and addresses the shortcomings of computer vision techniques specifically the huge time consumption for thermal image acquisition, and researchers derived various digital image processing algorithms, i.e., elimination of noise interference using the Butterworth low pass filter. This leads to fast processing and fine enhancement of image quality with any external noise. After studying these research contributions, the paper suggests the amalgamation of such techniques with unmanned aerial vehicles to provide speedy and accurate surveillance of construction sites. Various papers have highlighted the same need [11, 13–16]. Despite all, the proposed review paper presents the major limitations of already derived techniques for smart surveillance at the construction site. In addition to this, it emphasizes the combination of Internet of things, computer vision, and an area of unmanned aerial vehicles to evaluate the progress of any offline or online construction. The paper provides a feasibility study to integrate the multidisciplinary and trending areas to inspect the progress, and, in this regard, many research manuscripts were studied and highlighted here for readers. This paper shall be surely one of the ways to transform conventional surveillance methods.

The paper has been divided into several sections such that computer vision techniques for the inspection of construction sites in modern era. The next section is about the utilization of UAVs for the safety of workers. This paper also focuses on Internet of things and its boom in today's world. The next immediate section addresses the brief survey of IoT-enabled unmanned aerial vehicles used so far in the inspection, and it seems that their usage is still at the stage of infancy. This paper provides a broader picture and future directions for using IoT-enabled UAVs for inspection purpose. Lastly, this also motivates the research community to see the research issues and findings within the same area for future provisions.

The organization of the paper is divided into mainly six sections. In Section 1, reader may find the introduction to the paper theme and some significance of recent research contributions. The study of current inspecting methods and techniques are discussed under Section 2 along with major

Figure 1: IoT-enabled UAVs for different applications.

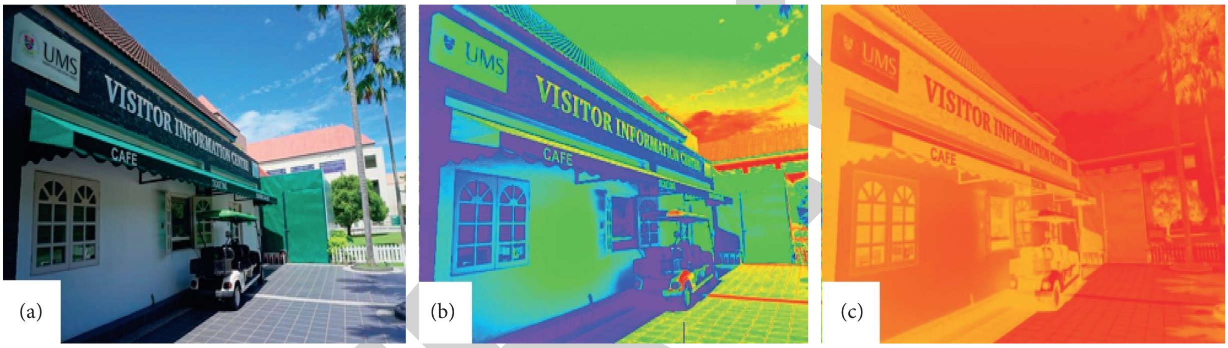


Figure 2: (a) RGB Image, (b) thermographic image, and (c) orthothermogram image.

limitations or constraints. Similarly, in Section 3, one may study the strategies followed and proposed for the surveillance of worker's safety at various construction sites. The diversity and importance of IoT-enabled drones and other applications have been highlighted under Section 4. The reader may study the limitations of IoT-enabled drones for the surveillance of construction sites under Section 5. The theme for this vision along with future directions is mentioned in Section 6. Last but not the least, the comprehensive conclusion is mentioned as Section 7 of this review manuscript.

2. Inspection Techniques

This section will discuss the techniques and modern computing algorithms that are utilized for the inspection of either construction or any building site. The proper advantages and shortcomings will be discussed in this section. The idea of discussing these techniques is to motivate researchers to use them with an IoT-enabled UAV to have better update from the site in less time.

2.1. Model Construction via Laser Scan. This technique is among the most popular ones in the literature whereas the 3D model is obtained using thermographic imaging of a specific building or site. This provides an easy way to see the exact defects of the site. For a decade, researchers have proposed optimized mapping using the LiDAR sensing method. This LiDAR-based optimization strategy involves the reproduction of accurate elevation of several buildings and structures [17]. Moreover, the farmlands and waterways can also be identified using the technique. The images derived from such a technique are presented in Figure 3 where Figure 3(a) is the normal image and Figure 3(b) is the thermographic image. The author has taken the thermographic images of more than 85 positions using the terrestrial laser scan method, and this took more than five working days.

This terrestrial scan needs huge processing time due to preprocessing; thus, to reduce this computation, researchers have proposed the creation of geometry using the thermal simulation method [18] before the actual results. A similar strategy has been followed in [19] where more focus was set on deriving efficient results with a smaller number of scans known as light detection and ranging (LiDAR). Furthermore, the time constraint for laser scan had been addressed in [20] where authors proposed parametric-based 3D model reconstruction [20] whereas they extended the same technique to acquire the volumetric 3D model. So far here, the scans were of normal resolution; thus, some of the researchers also put

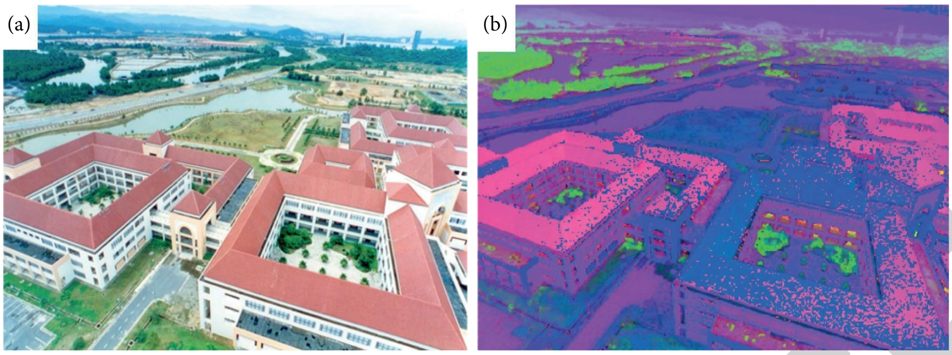


Figure 3: (a) LiDAR sensing image; (b) images of the entire site based on LiDAR scan.

some efforts to improve the resolution by using the optical diffraction element (DOE)-structured light [21]. This DOE-oriented technique is based on laser patterns and over the application of a convolution neural network (CNN). This approach improved the resolution, but still, it consumes huge processing time. To reduce the motion distortion, one may see [22] where continuous-time simultaneous localization and mapping (CTSLAM) is proposed. This research provided a cost-effective solution with less but efficient continuous scans. Laser scan-based identification and monitoring of construction site involves the high-density outcomes [23] and high demand in real-world for providing efficient and accurate results. The only drawback is the time consumption during scans of the positions. Thus, for identifying something, i.e., during the energy audit of a large-scale area that requires a speedier and reduced memory-oriented solution, this technique is preferred a lot by many researchers.

2.2. Model Reconstruction through Photogrammetry. The procedure that involves certain recordings, measuring photographic information, i.e., patterns, is known as photogrammetry [24]. In the last few years, several advancements occurred in computer vision and control area. This provided ease in three-dimensional model reconstruction for construction sites [24]. Structuring from motion (SFM), stitching, and overlapping of images are the most popular among the low-cost and high-resolution 3D model reconstruction techniques. Photogrammetry is divided further into three types depending on the type of method, i.e., RGB-D, monocular, and stereophotogrammetry. Mostly, the RGB-D photogrammetry technique was performed using Kinect RGB-D sensors [25–27], where a real-time Kinect fusion algorithm is processed to merge the maps into one. This strategy resolves multiple issues and provides an optimized solution by integrating the machine learning algorithms in real-time. This technique is most of the cases, observed in the mapping of large indoor environments using simultaneously localization and mapping (SLAM)-based mapping [28]. In this entire procedure, the important thing is the graphical processing unit (GPU) that performs the SLAM algorithm, and it requires the support of a thermal camera for monitoring the large-scale indoor building or construction site.

The 3D model reconstruction is performed using a regular camera, and later combining the photos from different angles [29] is known as monocular photogrammetry. The most common software used for this purpose is a two-DOF turntable. In this technique, several steps are executed such that masking photos from the respective background, photo mesh analysis, and lastly texturing step. This was so far static image and object approach but later in [30], and image-based model reconstruction was introduced using a single moving camera that led to the real-time 3D mapping of surroundings as shown in Figure 4. In this figure, Figure 4(a) is the original RGB image whereas Figure 4(b) is the mapped image based on thermal 3D model reconstruction methodology.

This monocular photogrammetry does not consist of any extra sensor, i.e., infrared or LiDAR; therefore, it provides efficient results with less memory usage and faster processing. These features have high potential to be equipped within any unmanned aerial vehicle with an on-board processor, i.e., National Instrument's myRIO controller to perform the computational tasks and provide the right outcome that is essential for the surveillance of any construction. This entire work is speedy enough as it can be executed using the general processor unit (GPU) of a smart mobile phone in real-time [31]. The literature shares that such algorithms and strategies are well enough for the indoor sites, whereas for the outdoor adaptability, it still produces 3D orthothermogram image showing the extensive inspection but not practically demonstrated still.

The construction of a 3D model based on stereo images is known as stereophotogrammetry. This technique is used for creating several models such that the human face [30]. This approach involves with spatial-temporal stereo-matching algorithms with noise rejection. It provides more promising results so far [31]. These techniques must be embedded with unmanned aerial vehicles for time-efficient results [32] with SLAM-based approach.

The approach for deploying a camera, inertial measurement unit (IMU), and integrating SLAM [33] algorithm may provide better time-efficient and cost-effective results for monitoring any sort of large ongoing construction site [34]. The images based on stereo reconstruction are comparatively better understood, as shown in Figure 5. These images are compared with the ground truth images, and the

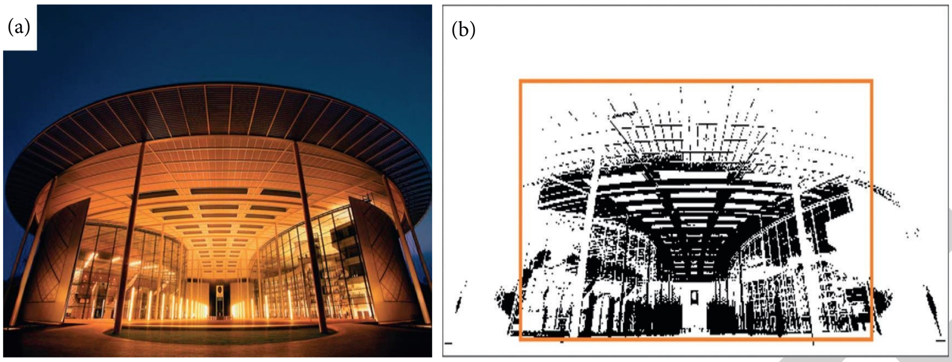


Figure 4: (a) Ori0067inal RGB image; (b) mapping based on thermal imagery onto the reconstructed 3D model.

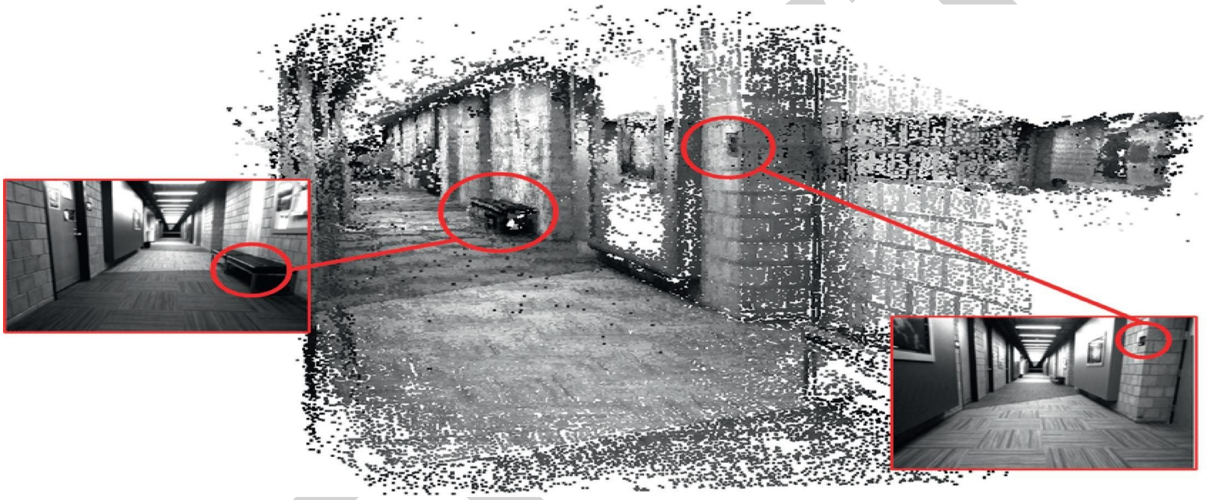


Figure 5: Dense 3D model reconstruction of a building model [35].

results are validated with an experimental setup too [36]. Moreover, the mentioned experimental technique and this high-resolution disparity maps along with the algorithm are applicable even for both indoor and outdoor environments, no matter for any online or offline site.

Concluding the entire photogrammetry topic, the monocular strategy is the most cost-effective approach [35] but consumes lots of postprocessing time and computing [37]. However, the stereo imaging technique can reduce this heavy computation and processing time. This means stereo imaging can provide the same results as produced by photogrammetry in various real-world applications. Since this approach is limited by nonmatching pairs within a nonplanar environment; thus, to resolve this shortcoming, RGB-D sensor technique is used [38]. By integrating the RGB-D sensor technique, the results are not only improved, but this is the only low-cost solution so far 3D mapping and may replace most of the techniques in the construction industry [27, 32].

2.3. Simultaneous Localisation and Mapping (SLAM). The use of camera and active sensors integrated within the autonomous robots to locate itself in any space is known as simultaneous localization and mapping (SLAM). This technique has been frequently utilized in the 3D model mapping of indoor as well as outdoor environments [39]. Mostly the sensors used in this field are laser scanner, global positioning satellite, and thermal sensor. The SLAM algorithm embedded with such autonomous mobile robots helps us to locate such areas that are dangerous or difficult [39]. In the literature, researchers have proposed multiple camera vision-based SLAM techniques that integrate multiple images and track the navigation of micro aerial vehicles (MAVs) [40]. This SLAM algorithm embedded with autonomous vehicles consumes some time hence to have high-speed navigation and to map a low-drift odometry technique is proposed in [33] known as inertial SLAM. To boost up the accuracy at high-speed, researchers added more cameras and included the geometric features [41]. Simply, after going through the detailed evaluation of SLAM-based frameworks, one can use it for acquiring promising results, i.e., 3D mapping, producing thermographic 3D models, and defect analysis of any site.

One of the variants of SLAM is introduced as point and lines using the RGB-D algorithm where 3D point constraint errors are also tackled and even lines are constructed [42]. In addition to this, one may see the S4-SLAM algorithm used for outdoor ground and water surface multiple scenario-based applications. This algorithm is mostly proposed for unmanned vehicles and ships and is amalgamated with odometry function [43].

2.4. Surveillance Using Drones and UAVs. The aerial surveys were introduced in the early 1980s [44]. This was the first time when fixed-wing aircraft and helicopters were used to monitor the sites in case of a natural disaster. In today's era, people from various fields opt for multirotor drones known as unmanned aerial vehicles (UAVs) for several applications due to their flexible and aggressive maneuvers [10, 45–50].

The main constraint of using drones for the inspection of such sites is the unawareness of the environment; thus, this review paper focuses on computer vision and other mapping techniques too. Apart from this, there are several research contributions, i.e., use of infrared technology-based inspection using quadrotor craft [15]. Almost more than 100 literature resources from the last 35 years have witnessed and emphasized the use of drones for several techniques, i.e., using a drone for inspection of site work on the local street [51] and drones for auditing the energy drawbacks [52]. One of the major reasons to recommend drones for such surveillance is because they can fly over a large area and may hover at one place too. Several quadrotor-type UAVs can perform vertical take-off and landing (VTOL) that may help in capturing the high-quality images of any site, i.e., bridge inspection [52]. Furthermore, it was seen that an abutment in Dakota was monitored using the DJI Phantom drone and provided an effective analysis of overall defects in real-time. To the best knowledge of authors, drones can identify the damages of very high-rise buildings and the scanning of hidden areas using thermography [40]. Understanding this requirement, various researchers proposed low-cost approaches, i.e., in [53] where orthoimages of the site are produced using multiview photogrammetry with a resolution of 0.066 of the digital elevation model using UAV. The results provided great help in identifying the faults more accurately.

In addition to this, one may see [54] where UAV is equipped with real-time thermography and some thermal sensors for the assessment of different building structures. As far as the shortcoming of such techniques is concerned, then one can see two major shortcomings as stated below:

- The battery timings of UAVs restrict them to monitor or inspect the large complex building structures.
- While identifying the defect, UAV cannot remember the exact location of fault; hence, the operator must remember the exact location. This is because of the lack of 3D mapping. Hence, for unknown targeted buildings, the defects and flaws are not relocatable.

Summarizing it all, the surveys and inspections based on drones/UAVs are highly time-efficient and productive. However, despite having such perks of drone-based surveillance and inspection, there is a huge lack in the utilization at construction industries. This strategy not only inspects the progress but also monitors for energy audits on large complex construction sites, as shown in [16].

2.5. 3D Modelling via Thermography. The inspection of any construction site, whether any bridge or building, must be performed time efficiently as well as nondestructively. This is the major reason researchers proposed several techniques such that virtual building geometry and thermography based on infrared and orthothermogram [12]. Such thermographic techniques help in identifying water leakages and heat losses in a very accurate manner. In this section, 3D modeling based on thermography and mapping is covered for large-scale construction sites. The framework discussed in [12, 55, 56] shares the mapping of infrared thermal images onto point clouds of the building structure. Some researchers had refined the image processing algorithm for both thermographic and laser-scanned point clouds to acquire better results. Furthermore, some researchers suggest a hybrid strategy such that the merge of thermal modeling with the LiDAR system. Though this hybrid technology provides the best results, due to huge postprocessing concurrency, this can never opt for real-time implementation. To address this issue, one may replace the LiDAR with stereo depth sensors, as discussed previously in this manuscript [57]. This technique has high accuracy, but still, the technique was time-consuming for large complex construction sites.

For mapping the indoor environment, there had been various wheeled mobile robots (WMRs) are proposed that amalgamate both infrared thermography and laser scans to generate the three-dimensional orthothermogram [58]. This approach produces fine radiometric results of an indoor environment as shown in Figure 6. To acquire the qualitative defect recognition and analysis, this review manuscript proposes the integration of such techniques within unmanned aerial vehicles (UAVs).

2.6. Damage Detection through Computer Vision and Machine Learning. Computer vision or image processing is the most emerging field that is opted to render an image for machine perception [59]. This section will review the most trending image processing-related investigations made for the inspection of buildings or any construction site, i.e., indoor or outdoor environments. Since last two decades, there has been a lot of advancement in the domain of computer vision technique design for the inspection of offline and online construction sites [60]. During this epoch, various algorithms had been introduced so far, i.e., convolution neural network (CNN) in [61, 62], used for the identification of cracks in metal surfaces. In the same way, one can see the research contribution of [63] where a nondestructive image processing algorithm is proposed to monitor the defects of historical buildings. The research was mainly focused on erosion on the surface of building structures. Furthermore, this research concluded that it is very difficult to automate the defect identification without considering the physical variables such that environmental factors such as temperature and humidity.

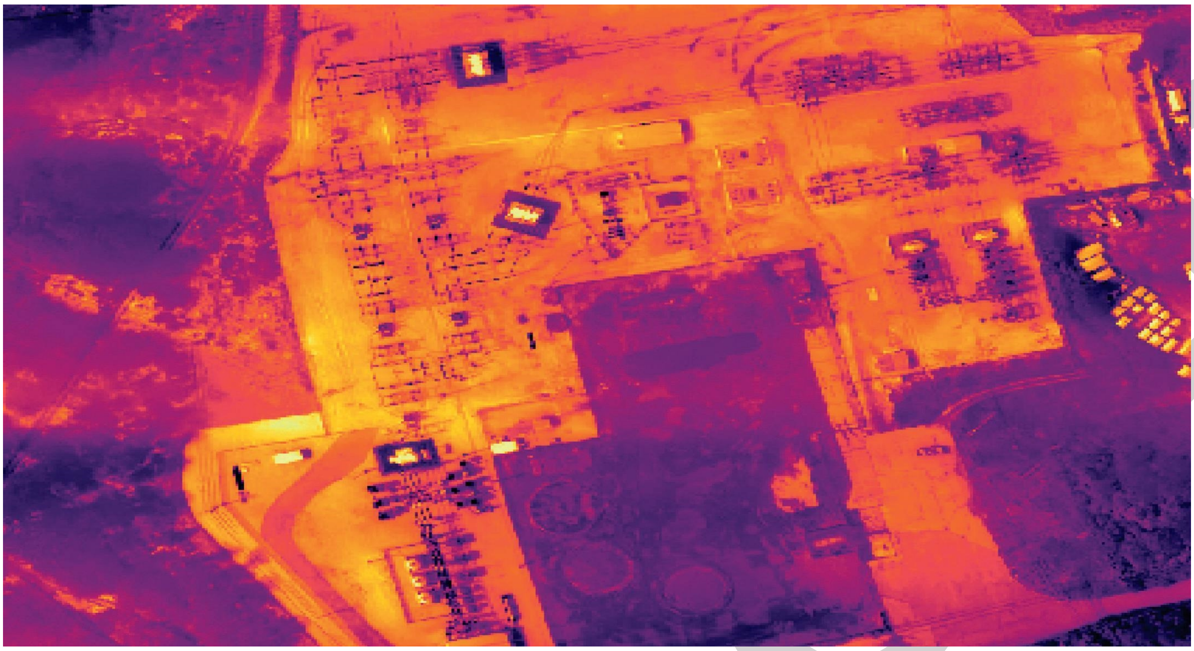


Figure 6: A radiometric result.

Thus, there is an immediate need to have an extraordinary approach (machine learning tactics) to automate the defect analysis and identification with these physical parameters. In the literature, one may find mobile-enabled cost-effective and deep learning-oriented damage analysis strategies [64]. In this approach, the CNNs had been trained for more than 30,000 iterations and 3 hours with more than 600 images of various cases. This proposed approach provided high-accuracy outcomes with consistent identification. The unknown fault or irregular defect must be trained first, and this was the only shortcoming for this algorithm. A similar approach had been introduced to identify the earthquake, i.e., spalling and corrosion [65]. It has been seen that the results had more than 86% accuracy rate in real-time. The contributors of this work suggested embedding the same algorithm within unmanned aerial vehicles to reduce both time and cost while covering the entire large-scale inspection. This is not an end yet; the same approach had been seen in visualizing the energy auditing too [66].

If one is looking for fault/defect analysis and identification with high-accuracy and time efficiency, then the only way is to hybridize the machine learning algorithms with the image processing strategies. So far, the algorithms such as artificial neural networks and CNN can produce a meaningful outcome in various fields [67]; hence, they are suggesting being hybridized with image processing algorithms and deployed within the UAVs for better timely response in real-time [68].

From Sections 2.1 to 2.6, there were the several trending techniques that can be useful for the inspection and surveillance of any construction sire with large-scale efficiency. Moreover, Table 1 shows the summary of entire techniques along with their perks and shortcomings.

3. UAVs for Worker's Safety

This section presents the usage of unmanned aerial vehicles at construction sites ensuring the proper mitigation of potential risks. In addition to this, it also highlights the most significant role of UAVs in health and safety assurance at the site so that every professional may perform their tasks at their workplace safely. The utilization of robotics and autonomous systems in every field is common today; thus, this manuscript emphasizes the usage of IoT-enabled UAVs within the construction site too. Before this, the UAVs are commonly used in many areas such as military, recreational activities, fire/emergency units, traffic management aerial photography, and surveillance. Regarding the use of UAVs for worker's or workplace's safety, very rare literature is available. There are several uncertain incidents that have been occurred at various construction sites [65]. The use of UAVs will surely decrease this number by close and continuous monitoring of the site [69, 70]. Amalgamating the same computer vision algorithm within UAVs, one can monitor the dress-up of labours and professionals that whether they have worn safety helmets and shoes or not. These drones can fly with variable altitudes and thus can be used to monitor the person who is doing welding or steel bending. Later, the people who are not following the health and safety rules may be penalized too with photographic proof or evidence [71].

4. Internet of Things and Its Diversity

The Internet of things (IoT) is defined as the intelligent connection of things by using some active sensors to the Internet [72]. The paper suggests the IoT-enabled drone to achieve better inspection method for construction sites. In

Table 1: Trending techniques for the inspection of the construction site.

S. no.	Techniques	Constraints
1	Machine learning with image processing algorithms	(i) Greater number of iterations are required to derive a reliable outcome. (ii) Testing data must be updated for high-accuracy results.
2	Thermographic modelling	(i) The proposed approach is heavy due to computational work and time-consuming because of the superimposition of thermal imaging to point clouds.
3	Simultaneous localization and mapping (SLAM)	(i) The existing method is depending on the laser scans or laser communication which is not good for aerial survey. (ii) In the existing approach, the switching of normal mode to direct photogrammetric mode will distort the reliability of the derived outcomes and localization.
4	RGB-D type photogrammetry	(i) The approach consumes computing power a lot. (ii) In addition to this, it has a noise effect due to IR depth sensors. (iii) It consumes time also because of postprocessing.
5	Stereo imaging-based photogrammetry	(i) Encountered with parallax effect that leads to an inaccurate model reconstruction.
6	Laser scan technique	(i) The hardware setup is much expensive and not recommended for aerial mapping.
7	Monocular photogrammetry	(i) The results were of low resolution. (ii) The plain walls were reconstructed inaccurately in this approach and the algorithm was light depended.
8	Unmanned aerial vehicles (UAVs)	(i) The only shortcoming is the less flight time, especially in the presence of inertial measurement unit and other additional sensors, i.e., depth sensor/camera or GPS.

this way, one can also address the constraints of IoT also, i.e., scalability of the network, computing constraints, and diverse provisions for diverse applications. By proposing IoT-enabled UAVs, one may not only address these constraints but also enhance these three critical expectations of Internet of things. In the past few years, there is a boom of IoT, and it is used significantly in various areas. In simple words, the IoT concept is related to connecting the system with varying capabilities of the Internet. Thus, it most frequently opts for a machine-to-machine communication [73]. The paper is therefore suggesting communicating the drone with a base station system where a supervisor of the entire construction site is evaluating the progress of work [74].

Since a decade, it is observed that there is a lot of emphasizes on the amalgamation of IoT with AI and computer vision algorithms to derive the better results [75–78]. One of the reasons to discuss IoT-enabled drones in this manuscript is that the Internet of things has great capability to tackle the extracted information from the image data capture by drone and analyse it for better measurements of entire progress at the site. This capability is usually known as automated intelligence [79]. This concept of automated intelligence has been introduced for many years ago, but at that time, the realization of the construction site, it is still in its early days. The primary focus for using IoT is to process and analyse the massive image data derived by UAV-enabled camera. To achieve reliable results with automated intelligence, one must ensure the three constraints as mentioned below:

One should establish the scalable network architecture with ubiquitous coverage known as scalability of the network.

Ensuring the best computing algorithm that extracts the data and concludes the outcome easily, this is termed as intelligence.

Support of diverse sensors to acquire the data known as diversity.

These three constraints are illustrated in a better way in Figure 7. These three constraints are defined under this section briefly.

If one is establishing the Internet of thing-based system, then scalability is one of the most important things to ensure. Scalability is the working area in which the network can communicate efficiently. In the case of monitoring the large and complex construction sites, it is very much necessary to integrate such communication techniques that may continuously bound the system within the communication; otherwise, the results are not easy to derive. Since the outcome is to identify the faults and defects, hence in this regard, artificial intelligence and computer vision algorithms are required. The IoT feature will gather all data, i.e., images of construction sites from a scalable network over a complex and large site to enable these algorithms so that the defects and progress can easily be identified. The term “diversity” indicates the different applications based on several sensors and geographic visual display techniques [80] that automate any procedure of IoT. If the sensors do not perform well or have time delay in sensing the physical data or capturing images with low-resolution in our case, then the outcome will not be reliable enough. This section concludes that one should need to achieve these expectations and so far, the best solution is to deploy the multiple nodes and design a complete network within the construction site. For such deployment, IoT-enabled nodes that will retrieve the data will be based on power source as well as storage space too.

5. Inspection Using IoT-Enabled UAVs

This section offers the benefits of UAVs along with technical as well as nontechnical issues related to the utilization of UAVs. The technical issues and shortcomings are discussed below in the context of prototyping, algorithm design, and flight operations of UAV systems.

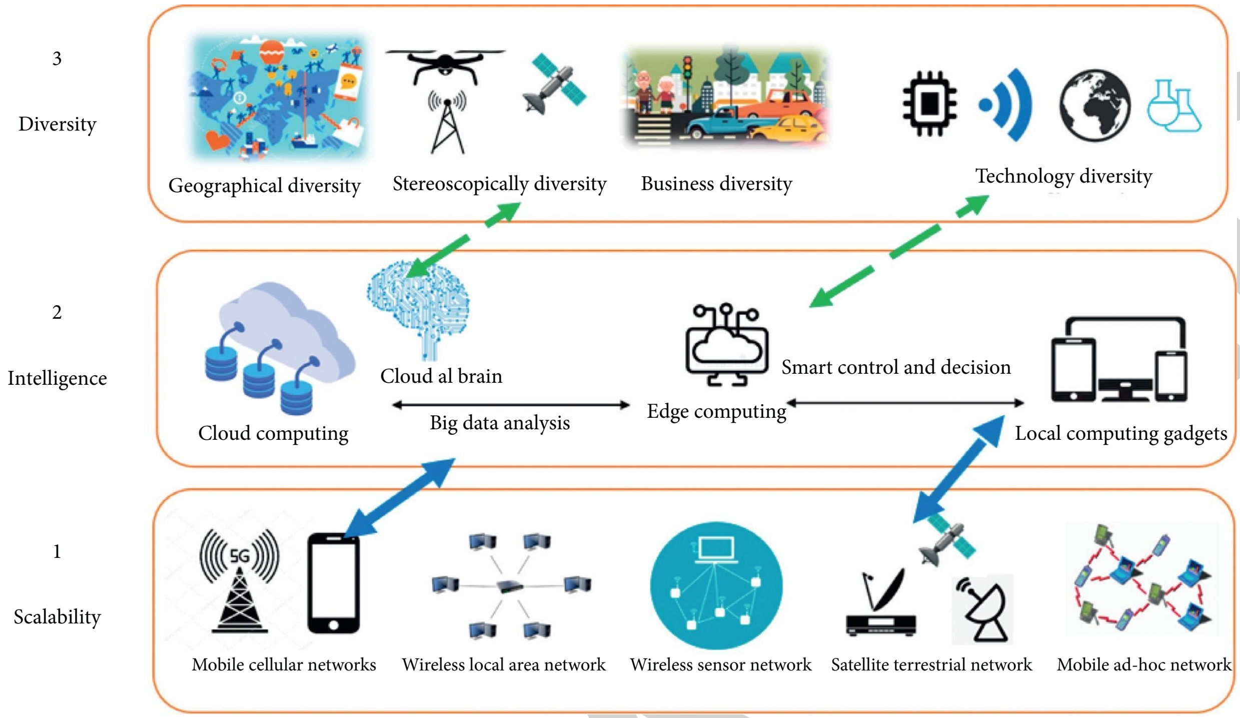


Figure 7: Three constraints of Internet of things (IoT).

5.1. Technical and Nontechnical Issues in Utilization of IoT-Enabled UAVs. One of the concerning issues for utilizing the IoT-enabled UAVs is security. The first issue is the privacy issue. During the flight of an unmanned aerial vehicle, especially with an embedded camera, this may infringe on the privacy of people around [81]. In many countries, there are some of the regions which are restricted areas; hence, the UAVs must not fly over them, and in this regard, the communication between UAV and ground control room (GCR) must be enough protected and secured. If these drones are utilized in smart cities, then one must consider the type of all security attacks [82]. These issues are of various types, i.e., malicious-prototype attacks, where the design of drone assembly is done to harm others. The second type of attack is concerned with the mission configuration parameters where one may access the communication system and amend it to change the mission settings, i.e., wrong path planning. It is also suggested that one should secure the computers that are involved in communicating these unmanned IoT-enabled systems [83]. While studying the literature, one may come across many security and privacy strategies recommended for such scenarios, i.e., vehicular ad hoc networks (VANETs), wireless LAN (WLAN), voice over IP (VoIP), radio RFID, wireless sensor networks (WSN), and Internet of things (IoT) [84, 85]. The second most trending issue related to IoT-enabled UAVs is the safety issues as they are executing various missions and flights from one point to the second point. This is because, during flight and aggressive maneuvers, this drone will pass by buildings and even nearby people too. Thus, safety is among the best issues in this domain. Looking over these issues, researchers have proposed the hybrid algorithms and integration of advanced sensors within UAVs. Researchers proposed certain regulations, i.e., speed, height at which these drones must be operated, and last but not least is the proximity [86–90].

Reliability is also one of the technical issues where one may look for fewer chances for hardware failure or the loss of the rotor's effectiveness loss. This happens because of any malfunctions within the mechatronic components. Researchers have studied this point and have proposed various approaches. In one of the most frequent approaches, one may find the adaptive-robust fault-tolerant control designs to detect the system faults and balance it accordingly [91]. These hardware or prototype faults can be compensated, but the failure in communication between the ground control room and UAVs is the most major issue. This failure can be in the form of time delay also, and thus, it cannot deliver for the mission properly. Researchers have proposed multichannel-oriented communication. However, most of these failures can also be detected using systematic verification, as indicated in research [91].

The communication failure issue has been studied in detail, as this is the only bridge between UAVs and GCR. Researchers proposed multiple channels and nodes to establish a stable communication [92]. In establishing such techniques, many of the parameters have been dictated such that quality of service for bandwidth, time delay in communication, and data transfer rate. In order to ensure security, safety, and reliability, one should first think up for this issue. Many manuscripts already initiated solutions such that by recommending the direct link, and cellular and mesh topologies [93]. It has been observed that various topologies have different constraints such that direct link is one of the smart techniques, but the signal strength can be weakened in case of obstruction, and it does not support longer distance-based communication. For having long-distance communication, it needs high-powered transmitters and modified antennas. This is not yet simple because if there are multiple drones in an area, this may lead to high communication delays. The cellular topology though has several advantages, but it can never be recommended for UAVs as this will degrade the cellular networks within the region. The possible and a bit expensive solution for this is to propose a separate cellular infrastructure within an area. Moreover, the last technique is mesh topology used to synchronize the GCR with the UAVs, which is still an attractive topology, but this requires high security as mentioned in the research [94].

In designing such smart and IoT-enabled UAVs, one must concern about the factor of cost. One must design the UAVs in a manner that it can be cost-effective, but simultaneously it can execute major functions too. For such a cost-effective hardware design, the majority of the researchers go for the underactuated systems [95]. Some of the research contributions have suggested the modular type of prototyping too, along with various but suitable and flexible software architectures that can be changed from application to application [96, 97]. It has been seen that there had been a rapid growth in the field of designing autonomous controls for the stability of such underactuated systems [98, 99] that can surely reduce the cost either on the hardware or software algorithm.

Integrating or amalgamating the UAVs with other subsystems is one of the major steps, and there are many constraints too as highlighted in the above section of the literature. This type of integration is essential and may lead to exploring various dimensions of the drone in several other fields but challenging at the same point. To tackle such things such that manipulating some payload from one place to another, researchers have proposed some middle software framework; this integrates two applications such that flight mode of UAV, and it can also accommodate the global position of UAV [100]. In addition to this, when any design is implemented first, it is tested too, which means it requires a specific testing environment for validating the software simulation results. In most of the cases, it has been observed that the prototypes developed for the first time are tested in the limited development environment and help in providing close results as well. So far, there is not any comprehensive environment that helps in producing many effective results. One of the nontechnical issues one may find is ethics. Flying drones and UAVs in the sky may infringe privacy, and this is not as per the ethical norms because of the multiple public and private areas. There is a great chance of spying the spots using UAVs. This can be one of the constraints in utilizing an IoT-enabled drone [101]. Thus, the manuscript suggests a code of ethics for implementing the drones within any territory. This code of ethics will define the boundary and procedures for proper uses of UAVs.

In addition to this, one may set a focus on proposing an appropriate transportation system for smart cities with network scalability to enforce such drones to monitor the construction sites such that proposed in [102]. There are several solutions that are based on blockchain which addresses the limitations such that high transaction storage, latency in response, communication with ground control room (GCR), and lastly the bandwidth. Keeping this in mind, researchers have proposed unmanned aerial vehicles (UAVs) with secure communication protocol in compliance with the recent architecture of 5G communication and the power of artificial intelligence [103]. The techniques such as InterPlanetary File System (IPFS) facilitate nowadays the storage limitation and provide better network performance and security. These techniques already proved their robustness and flexibility on decision taking measures via amalgamating the 5G and artificial intelligence (AI)-based strategies.

6. Research Findings, Vision, and Future Directions

After discussing the techniques for inspecting the site, technical and nontechnical issues related to the UAVs, one may see the integration of such strategies into one another may provide the time and cost-effective solution to inspect the online or offline construction site. After studying several research contributions, one may come across with some of the challenges that are associated only with the methodology by which one is going to implement the work. Discussing these research findings, one may see the on-board controller deployed on UAV such that underactuated quadrotor will be performing so many things such that the efficient inspecting algorithm with high-quality visual feed and thermal facility, real-time data monitoring that maps up the data from a number of sensors embedded with this board, visual-based odometry, and live transmission and may also tackle the glitch-free communication in between the UAV and ground control room (GCR). Thus, a powerful board is therefore required that surely consume more power sources, i.e., battery. This will surely reduce the flight duration and one may not inspect the large area of the site easily with these IoT-enabled UAVs [104].

Furthermore, one may see the complex and hybrid machine learning algorithm for fault identification at either real-time or offline consumes huge processing time and in addition to this, they require multiple data set with several training iterations and experimentations. Till now, there is an efficient algorithm produced because of qualitative analysis and one should work on quantitative analysis too for better fault identification [105]. It has been noticed that the average flight duration for the UAVs proposed is in between 20 and 35 minutes. Hence, this will be one of the major limitations in inspecting the large area of the construction site. However, researchers have proposed some of the advanced mobility options [7, 39, 58, 106]. One of the most suitable options is to propose the scheduling for UAVs to land at one of the nearby jurisdictions and get themselves self-charged to improve the time and speed, but in this approach, one may see the discontinuity in the mapping results and ultimately in the reconstruction of a superimposed 3D thermographic model. This section offers the main research findings of IoT-enabled UAVs as summarized in Table 2.

Table 2: Summary of all research findings.

Sr.	Research findings	References
1	On-board controller for IoT-enabled UAVs requires high power consumption because of several tasks.	[104]
2	One should come up with hybrid predictive learning algorithms based on quantitative and qualitative analysis for fault identification.	[105]
3	UAVs with usually have 20–30 minutes to fly in the sky which is very less time for monitoring any construction site. Thus, we need some smart mobility strategies.	[7, 106, 107]
4	For LiDAR-based reconstruction techniques, there is always huge time requirement, and thus, this is not recommended for large construction sites.	[108, 109]
5	There is also an immediate need to improve the communication between ground control room and UAVs.	[110]

Table 2 defines the main findings that are retrieved by the author after evaluating the literature review. The techniques used before for the same purpose using drones and some computing algorithms have some limitations. These limitations include the high-power controller or general-purpose processor [104], the necessity of efficient computer algorithm that assesses the quantitative faults such that the progress of one building or storey, and qualitative fault analysis too such that the cracks in the bridge pillars or wall. This table also states that there is an immediate need to establish a good communication network between ground control room (GCR) and the IoT-enabled drones.

The idea behind this manuscript is to provide vision and future directives to improve the construction industry by introducing the IoT-enabled unmanned aerial vehicles that are equipped with advanced computer vision and artificial intelligence-based algorithms as discussed in this manuscript. Not only does this type of amalgamated system accelerate the construction of megaprojects with an easy update, but also, in addition to this, one may visualize and predict the way the construction progresses. There had been more than 50 plus megacities in the year 2017 in Asia where it is reported that more than 1.5 billion people have been moved to these cities and yet to travel by the next 20 years. In these cities, there are various mega projects going on. Thus, the government of each state has invested a lot of their resources and budget. To monitor the right progress, the idea presented in this paper is so far the best suitable fit for it. This is an admitted fact that the potential of IoT, computer vision, and AI-enabled applications is limitless and there are various advantages, i.e., network agility, easy to deploy, orchestrate, and secure at hyper-scale. Many mega construction projects in metropolitan cities can be completed on time by utilizing this approach:

Integration of mapping techniques and AI algorithms needs a powerful controller board and proper battery resources [111].

The LiDAR-based approach is much reliable, but for the small-scale construction site, whereas it is not recommended for a huge area of construction because of its time consumption while scanning [107, 108, 109]. It has been noted that while performing high definition thermal imaging, it requires real-time data processing, and this is still a potential restriction that can be unrestrained using 5G-based data communication [112].

While studying the LiDAR-based SLAM algorithm, it has been observed that in real-time, it does not reconstruct the dense map that results in a less accurate outcome. To cope up, one may opt for an artificial algorithm and its utilization to improve the mapping outcomes as proposed in [110, 113].

7. Conclusion

The main aim of this manuscript is to present the major opportunities for inspecting any online or offline construction sites using the application of autonomous UAVs enabled dully with the feature of the Internet of things (IoT).

In this comprehensive survey paper, a variety of inspection techniques and applications of UAVs with technical and nontechnical issues have been discussed. Furthermore, one may also explore the limitations and constraints of amalgamating these techniques together, which is the most trending thing nowadays. This is because of reducing the inspecting construction sites mainly where one is mainly concerned to monitor the progress and reduce the emission of carbon and improve the execution of any mega construction project.

The discussion regarding the inspection, i.e., LiDAR-based approach, SLAM, and thermographic methods, with an amalgamation of artificial intelligence and IoT plays a very vital role in deriving the quality results. The key findings of this entire study are mentioned in the below postulates: there is an immediate need to explore and develop the hybrid solution to amalgamate the inspecting techniques with artificial intelligence and IoT features for such smart UAVs.

One must go for both types of analysis such that quantitative as well as qualitative when it comes to thermographic 3D model reconstruction. The advent of artificial intelligence so far improved the methodology for fault detection, and they can be incorporate with IoT-enabled drones for qualitative and quantitative inspection for online and offline construction sites over a large scale.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

For the paper, Ghulam E Mustafa Abro, Muhammad Farhan, and Amber Israr were responsible for idea and conceptualization; literature review was performed equally by all the authors; data analysis was done by Saif ul Azrin Bin Mohd Zulkifli and Sadiq Ali Khan; and the manuscript was drafted by Ghulam E Mustafa Abro and critically revised by Saif ul Azrin Bin Mohd Zulkifli and Muhammad Farhan.

Acknowledgments

The authors are thankful to Prof. Dr. Wang Dao bo, who invited us to submit the article in his special issue. In addition to this, the authors wish to extend their sincere thanks to the support from the Center of Graduate Studies, Universiti Teknologi PETRONAS, Malaysia, for providing the research facilities to carry on this collaborated research work.

References

[1] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153, Article ID 119293, 2020.

[2] Z. Khan, A. Anjum, and S. L. Kiani, "Cloud based big data analytics for smart future cities," in *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pp. 381–386, Dresden, Germany, December 2013.

[3] N. Mohamed, J. Al-Jaroodi, I. Jawhar, and S. Lazarova-Molnar, "A service-oriented middleware for building collaborative UAVs," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 309–321, 2014.

[4] F. Mohammed, I. Ahmed, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Opportunities and challenges of using UAVs for Dubai Smart city," in *Proceedings of the 2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–4, Dubai, UAE, April 2014.

[5] D. Moshashai, A. M. Leber, and J. D. Savage, "Saudi Arabia plans for its economic future: vision 2030, the National Transformation Plan and Saudi fiscal reform," *British Journal of Middle Eastern Studies*, vol. 47, no. 3, pp. 381–401, 2020.

[6] E. R. Jimson, K. Nisar, and M. H. B. A. Hijazi, "Bandwidth management using software defined network and comparison of the throughput performance with traditional network," in *Proceedings of the 2017 International Conference on Computer and Drone Applications (IConDA)*, pp. 71–76, Kuala Lumpur, Malaysia, November 2017.

[7] Y. Liu, M. Zhu, and H. Zhang, "Processed RGB-D slam using open-source software." in *Proceedings of the IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 623–626, Guangzhou, China, July 2017.

[8] Energy Consumption by End-Use | Energy n.D (accessed April 21, 2020), https://ec.europa.eu/energy/content/energy-consumption-end-use_en?redir¼1.

[9] Heating and Cooling | Energy n.D. https://ec.europa.eu/energy/topics/energyefficiency/heating-and-cooling_en?redir¼1.

[10] S. Gallardo-Saavedra, L. Hernandez-Callejo, and O. Duque-Perez, "Technological review of the instrumentation used in aerial thermographic inspection of photovoltaic plants," *Renewable and Sustainable Energy Reviews*, vol. 93, 2018.

[11] W. Cai, X. Wen, S. Wang, and L. Wang, "A real-time detection method of building energy efficiency based on image processing," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 295–304, 2019.

[12] S. Lagüela, J. Armesto, P. Arias, and J. Herráez, "Automation of thermographic 3D modelling through image fusion and image matching techniques," *Automation in Construction*, vol. 27, pp. 24–31, 2012.

[13] A. Kirimtat and O. Krejcar, "A review of infrared thermography for the investigation of building envelopes: advances and prospects," *Energy and Buildings*, vol. 176, pp. 390–406, 2018.

[14] H. Huang, A. V. Savkin, and X. Li, "Reactive autonomous navigation of UAVs for dynamic sensing coverage of mobile ground targets," *Sensors*, vol. 20, no. 13, p. 3720, 2020.

[15] E. Lucchi, "Applications of the infrared thermography in the energy audit of buildings: a review," *Renew Sustain Energy Rev*, vol. 35, 2018.

[16] D. Popescu, F. Stoican, G. Stamatescu, L. Ichim, and C. Dragana, "Advanced UAV-WSN system for intelligent monitoring in precision agriculture," *Sensors*, vol. 20, no. 3, p. 817, 2020.

[17] T. L. Garwood, B. R. Hughes, D. O'Connor, J. K. Calautit, M. R. Oates, and T. Hodgson, "A framework for producing gbXML building geometry from point clouds for accurate and efficient building energy modelling," *Applied Energy*, vol. 224, pp. 527–537, 2018.

[18] H. Son, S. Lee, and C. Kim, "Automated 3D model reconstruction to support energy-efficiency," *Procedia Engineering*, vol. 145, pp. 571–578, 2016.

[19] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Computers & Graphics*, vol. 54, pp. 94–103, 2016.

[20] S. Ochmann, R. Vock, and R. Klein, "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 151, pp. 251–262, 2019.

[21] Z. Song, S. Tang, F. Gu, C. Shi, and J. Feng, "DOE-based structured-light method for accurate 3D sensing," *Optics and Lasers in Engineering*, vol. 56, 2019.

[22] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic LiDAR fusion:dense map-Centric continuous-time SLAM," *IEEE International Conference on Robotics and Automation*, vol. 32, 2018.

[23] H. El-Din Fawzy, "3D laser scanning and close-range photogrammetry for buildings documentation: a hybrid technique towards a better accuracy," *Alexandria Engineering Journal*, vol. 58, no. 4, pp. 1191–1204, 2019.

[24] P. Tang, S. Vick, J. Chen, and S. German Paal, "Surveying, geomatics, and 3D reconstruction," in *Infrastructure Computer Vision*Butterworth-Heinemann, Oxford, UK, 2020.

[25] N. Namitha, S. M. Vaitheeswaran, V. K. Jayasree, and M. K. Bharat, "Point cloud mapping measurements using kinect RGB-D sensor and kinect fusion for visual odometry," *Procedia Computer Science*, vol. 89, pp. 209–212, 2016.

[26] Kinect X. https://www.xbox.com/en-US/kinect.

[27] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia | IEEE Computer Society Digital Library*, vol. 19, 2009.

[28] O. Guclu, “Can AB. k-SLAM: a fast RGB-D SLAM approach for large indoor environments,” *Comput Vis Image Understand*, vol. 32, 2019.

[29] J. Kaufman, A. E. Rennie, and M. Clement, “Single camera photogrammetry for reverse engineering and fabrication of ancient and modern artifacts,” *Procedia International Registration Plan*, vol. 36, pp. 223–229, 2015.

[30] O. Krutikova, A. Sisojevs, and M. Kovalovs, “Creation of a depth map from stereo images of faces for 3D model reconstruction,” *Procedia Computer Science*, vol. 104, pp. 452–459, 2017.

[31] K. Fu, Y. Xie, H. Jing, and J. Zhu, “Fast spatial-temporal stereo matching for 3D face reconstruction under speckle pattern projection,” *Image and Vision Computing*, vol. 85, pp. 36–45, 2019.

[32] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, and J. Civera, “S-PTAM: stereo parallel tracking and mapping,” *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.

[33] N. U. Jacobo Berlles and K. Kumar, “Implementation of an autonomous path planning & obstacle avoidance UGV using SLAM,” in *Proceedings of the 2018 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–5, Lahore, Pakistan, February 2018.

[34] R. A. Hamzah, A. F. Kadmin, M. S. Hamid, S. F. A. Ghani, and H. Ibrahim, “Improvement of stereo matching algorithm for 3D surface reconstruction,” *Signal Processing: Image Communication*, vol. 65, pp. 165–172, 2018.

[35] A. Jalandoni, I. Domingo, and P. S. C. Taçon, “Testing the value of low-cost Structure-from-Motion (SfM) photogrammetry for metric and visual analysis of rock art,” *Journal of Archaeological Science: Reports*, vol. 17, pp. 605–616, 2018.

[36] https://www.youtube.com/watch?v¼QTKd5UWCG0Q REMODE: Probabilistic. Monocular Dense Reconstruction in Real Time – YouTube n.D.

[37] F. Bandini, T. P. Sunding, J. Linde, I. K. Jensen, C. J. Köppl, and O. Smith, “Unmanned Aerial System (UAS) observations of water surface elevation in a small stream: comparison of radar altimetry, LIDAR and photogrammetry techniques,” *Remote Sensing of Environment*, vol. 237, Article ID 111487, 2020.

[38] F. Dornaika and R. Chung, “Mosaicking images with parallax,” *Signal Processing: Image Communication*, vol. 19, no. 8, pp. 771–786, 2004.

[39] P. Kim, J. Chen, and Y. K. Cho, “SLAM-driven robotic mapping and registration of 3D point clouds,” *Automatic ConStructure*, vol. 89, 2018.

[40] S. Yang, S. A. Scherer, X. Yi, and A. Zell, “Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles,” *Robotics and Autonomous Systems*, vol. 93, pp. 116–134, 2017.

[41] J. Zhang and S. Singh, “Laser-visual-inertial odometry and mapping with high robustness and low drift,” *Journal of Field Robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[42] C. Zhang, “RGB-D SLAM with a novel 2D and 3D geometric constraint model of point and line features,” *IEEE Access*, vol. 9, no. 2021, pp. 9958–9971.

[43] B. Zhou, H. Yi, K. Qian, X. Ma, and X. Li, “S4-SLAM: a real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications,” *Autonomous Robots*, vol. 32, pp. 1–22, 2020.

[44] M. Fox, D. Coley, S. Goodhew, and P. De Wilde, “Thermography methodologies for detecting energy related building defects,” *Renewable and Sustainable Energy Reviews*, vol. 40, pp. 296–310, 2014.

[45] J.-L. Jean-Luc, L. Hostettler, L. Zulliger et al., “Lake imaging and monitoring aerial drone,” *HardwareX*, vol. 3, pp. 146–159, 2018.

[46] I. Gomes, L. Peteiro, J. Bueno-Pardo et al., “What’s a picture really worth? On the use of drone aerial imagery to estimate intertidal rocky shore mussel demographic parameters,” *Estuarine, Coastal and Shelf Science*, vol. 213, pp. 185–198, 2018.

[47] D. R. A. Almeida, E. N. Broadbent, A. M. A. Zambrano et al., “Monitoring the structure of forest restoration plantations with a drone-lidar system,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 68, 2019.

[48] D. Zorbas, L. Di Puglia Pugliese, T. Razafindralambo, and F. Guerriero, “Optimal drone placement and cost-efficient target coverage,” *Journal of Network and Computer Applications*, vol. 75, pp. 16–31, 2016.

[49] P. Urbanová, M. Jurda, T. Vojtíšek, and J. Krajsa, “Using drone-mounted cameras for on-site body documentation: 3D mapping and active survey,” *Forensic Science International*, vol. 281, pp. 52–62, 2017.

[50] N. M. Kumar, K. Sudhakar, M. Samykano, and V. Jayaseelan, “On the technologies empowering drones for intelligent monitoring of solar photovoltaic power plants,” *Procedia Computer Science*, vol. 133, pp. 585–593, 2018.

[51] C. Eschmann, C.-M. Kuo, C.-H. Kuo, and C. Boller, “Unmanned aircraft systems for remote building inspection and monitoring,” in *Proceedings of the 6th European Workshop on Structural Health Monitoring*, Dresden, Germany, July 2012.

[52] J. Seo, L. Duque, and J. Wacker, “Drone-enabled bridge inspection methodology and application,” *Autom ConStruct*, vol. 94, 2018.

[53] M. Gao, X. Xu, Y. Klinger, J. Van Der Woerd, and T. Paul, “High-resolution mapping based on an Unmanned Aerial Vehicle (UAV) to capture paleoseismic offsets along the Altyn-Tagh fault, China,” *Scientific Reports*, vol. 7, no. 1, pp. 1–11, 2017.

[54] A. G. Entrop and A. Vasenev, “Infrared drones in the construction industry: designing a protocol for building thermography procedures,” *Energy Procedia*, vol. 132, pp. 63–68, 2017.

[55] S. Lagüela, L. Díaz-Vilariño, J. Armesto, P. Arias, and C. Lagoas, “Marcosende vigo. “Thermographic 3D models as the foundation for building information models,” in *Proceedings of the 11th International Conference on Quantitative InfraRed Thermography*, pp. 1–10, Naples, Italy, June 2012.

[56] S. Lagüela, L. Díaz-Vilariño, J. Martínez, and J. Armesto, “Automatic thermographic and RGB texture of as-built BIM for energy rehabilitation purposes,” *Automation in Construction*, vol. 31, pp. 230–240, 2013.

[57] D. González-Aguilera, P. Rodriguez-Gonzalvez, J. Armesto, and S. Lagüela, “Novel approach to 3D thermography and energy efficiency evaluation,” *Energy and Buildings*, vol. 54, pp. 436–443, 2012.

[58] L. López-Fernández, S. Lagüela, D. González-Aguilera, and H. Lorenzo, “Thermographic and mobile indoor mapping for the computation of energy losses in buildings,” *Indoor and Built Environment*, vol. 26, no. 6, pp. 771–784, 2017.

[59] G. E. M. Abro and K. Kumar, “Implementation of fruit grading & sorting station using digital image processing techniques,” *Sir Syed University Research Journal of Engineering & Technology*, vol. 7, no. 1, p. 6, 2017.

[60] A. M. Paterson, G. R. Dowling, and D. A. Chamberlain, "Building inspection: can computer vision help?" *Automation in Construction*, vol. 7, no. 1, pp. 13–20, 1997.

[61] V. Hoskere, Y. Narazaki, and Y. Narazaki, "Advances in computer vision-based civil infrastructure inspection and monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, 2019.

[62] C. Xu, J. Xie, G. Chen, and W. Huang, "An infrared thermal image processing framework based on superpixel algorithm to detect cracks on metal surface," *Infrared Physics & Technology*, vol. 67, pp. 266–272, 2014.

[63] R. A. Galantucci and F. Fatiguso, "Advanced damage detection techniques in historical buildings using digital photogrammetry and 3D surface anlysis," *Journal of Cultural Heritage*, vol. 36, pp. 51–62, 2019.

[64] N. Wang, X. Zhao, P. Zhao, Y. Zhang, Z. Zou, and J. Ou, "Automatic damage detection of historic masonry buildings based on mobile deep learning," *Automation in Construction*, vol. 103, pp. 53–66, 2019.

[65] J. Howard, V. Murashov, and C. M. Branche, "Unmanned aerial vehicles in construction and worker safety," *American Journal of Industrial Medicine*, vol. 61, no. 1, pp. 3–10, 2018.

[66] Y. Guo, Z. Tan, H. Chen et al., "Deep learning-based fault diagnosis of variable refrigerant flow air-conditioning system for building energy saving," *Applied Energy*, vol. 225, pp. 732–745, 2018.

[67] H. Goyal, M. Hanmandlu, and D. P. Kothari, "An artificial intelligence based approach for control of small hydro power plants," *Centre for Energy Studies, Indian Institute of Technology*, vol. 37, 2014.

[68] Y. Narazaki, V. Hoskere, T. A. Hoang, Y. Fujino, A. Sakurai, and B. F. Spencer, "Vision-based automated bridge component recognition with high-level scene consistency," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 5, pp. 465–482, 2020.

[69] J. Irizarry and D. B. Costa, "Exploratory study of potential applications of unmanned aerial systems for construction management tasks," *Journal of Management in Engineering*, vol. 32, no. 3, Article ID 05016001, 2016.

[70] R. Ashour, T. Taha, F. Mohamed et al., "Site inspection drone: a solution for inspecting and regulating construction sites," in *Proceedings of the 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, Abu Dhabi, UAE, October 2016.

[71] W. Cao, J. Zhang, C. Cai et al., "CNN-based intelligent safety surveillance in green IoT applications," *China Communications*, vol. 18, no. 1, pp. 108–119, 2021.

[72] G. E. M. Abro, "Shoaib ahmed shaikh, safeeullah soomro, Ghulam abid, kundan kumar, and fiaz ahmed. "Prototyping IOT based smart wearable jacket design for securing the life of coal miners."" in *Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 134–137, Abu Dhabi, UAE, August 2018.

[73] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[74] D. Evans, https://blogs.cisco.com/digital/internet-of-everything-harnessingan- exponentially-more-powerful-internet-ioe-infographic#comments Internet of everything: Harnessing an exponentially more powerful internet, 2012.

[75] M. H. Miraz and P. S. Maaruf Ali, "Excell, and rich picking. "A review on internet of things (IoT), internet of everything (IoE) and internet of nano things (IoNT)," in *Proceedings of the 2015 Internet Technologies and Applications (ITA)*, pp. 219–224, Wrexham, UK, August 2015.

[76] J. L. Holland and S. Lee., "Internet of everything (IoE)," *Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities*, vol. 21, pp. 215–245, 2019.

[77] F. Hussain, *Internet of Things: Building Blocks and Business Models*, Springer International Publishing, Berlin, Germany, 2017.

[78] M. Miraz, M. Ali, P. Excell, and R. Picking, "Internet of nano-things, things and everything: future growth trends," *Future Internet*, vol. 10, no. 8, pp. 68–8, 2018.

[79] Y. Liu, H.-N. Dai, Q. Wang, and K. Mahendra, "Shukla, and Muhammad Imran. "Unmanned aerial vehicle for internet of everything: opportunities and challenges," *Computer Communications*, vol. 155, 2020.

[80] A. Lilian Wei, K. Nisar, Z. I. Awang Ismail, and I. Welch, "Survey on geographic visual display techniques in epidemiology: taxonomy and characterization," *Journal of Industrial Information Integration*, vol. 18, no. 2, pp. 01–14, 2020.

[81] E. Vattapparamban, G. . İsmail, A. İ. Yurekli, K. Akkaya, and S. Uluağaç, "Drones for smart cities: issues in cybersecurity, privacy, and public safety," in *Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Paphos, Cyprus, September 2016.

[82] N. M. Rodday, R. D. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 993-994, Istanbul, Turkey, April 2016.

[83] T. D. Chung, R. B. Ibrahim, V. Sagayan Asirvadam, N. B. Saad, and S. M. Hassan, "Simulation of WirelessHART networked control system with packet dropout," in *Proceedings of the 2015 10th Asian Control Conference (ASCC)*, pp. 1–6, Sabah, Malaysia, April 2015.

[84] P. Vijayakumar, V. Chang, L. Jegatha Deborah, B. Balusamy, and P. G. Shynu, "Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks," *Future Generation Computer Systems*, vol. 78, pp. 943–955, 2018.

[85] D. Liao, G. Sun, H. Li, H. Yu, and V. Chang, "The framework and algorithm for preserving user trajectory while using location-based services in IoT-cloud systems," *Cluster Computing*, vol. 20, no. 3, pp. 2283–2297, 2017.

[86] P. Angelov, *Sense and Avoid in UAS: Research and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[87] R. Carnie, R. Walker, and C. Peter, "Image processing algorithms for UAV" sense and avoid," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 2848–2853, Orlando, FL, USA, May 2006.

[88] B. Korn and C. Edinger, "UAS in civil airspace: demonstrating "sense and avoid" capabilities in flight trials," in *Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, p. 4, St. Paul, MN, USA, April 2008.

[89] S. Ramasamy, R. Sabatini, and A. Gardi, "Avionics sensor fusion for small size unmanned aircraft sense-and-avoid." in *Proceedings of the 2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, pp. 271–276, Benevento, Italy, December 2014.

[90] O. Shakernia, W.-Z. Chen, and V. Raska, *Passive Ranging for UAV Sense and Avoid Applications*, Infotech@ Aerospace, Arlington, VA, USA, 2005.

[91] Y. S. Brar, J. S. Dhillon, and D. P. Kothari, "Multiobjective load dispatch based on genetic-fuzzy technique," in

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5581788, 10 pages
https://doi.org/10.1155/2021/5581788

Research Article

Semantic Optimization of Feature-Based SLAM

Peng Li, Lili Yin, Jiali Gao, and Yuezhongyi Sun

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, Heilongjiang, China

Correspondence should be addressed to Peng Li; printing3d@126.com

Received 30 January 2021; Revised 10 March 2021; Accepted 24 March 2021; Published 13 April 2021

Academic Editor: Zain Anwar Ali

The purpose of this paper is to provide reasonable recommendation and removal of inappropriate information for SLAM (Simultaneous Localization and Mapping) technology based on feature method. The methodology is to propose a semantic recognition of environment objects in the natural scene through object detection, which is a kind of bag of word method in SLAM problem between the key frames and object level, the method of establishing key frames, and the relationship between the target object levels, through the practical significance of the target object level to judge the merits of the target object level information, and then combined with key frames in the visual SLAM relations with relevant information, so as to get object level targets in each key frame and the relationship between the relevant information, so as to achieve through the object level semantic information to judge the merits of the key frames and screening, as well as to the key frames to judge the merits of the relevant information and screening purpose. The finding of the study is the above method can retain the information of high reliability and good stability for visual SLAM and process the key frames with poor reliability or low stability and the information related to key frames.

1. Introduction

For visual SLAM, in order to ensure the stability of positioning and mapping, key frames with good stability and relevant information should be retained as far as possible [1]. The relevant information here refers to the information used for mapping and map-ping correlation calculation in visual SLAM, since the front-end of visual SLAM (visual odometry) is divided into direct front-end and indirect front-end. For the former, this relevant information, namely, feature points, such as SIFT, SURF, and ORB, are effective feature points to be extracted. For indirect visual SLAM, the relevant information is the bright ness represented by a single pixel. In these SLAM methods, it is assumed that all motion estimation is carried out under a relatively ideal premise; that is, the information has good invariability with the change of time and space. However, in the actual production or living environment, neither the object target at the macro level nor the brightness of a single pixel at the micro level is constant. For the direct method, some researchers have proposed that the automatic calibration of luminosity can provide good pretreatment results for the visual odometer. However, in the indirect method, since the feature points have good rotation and scale invariance, and the brightness change in a certain range will not have a great impact on them, there is no need to preprocess for their constant luminosity. However, in actual activities, any object is not immutable, such as pedestrians on the roadside and books on the desk, which are possible moving visual targets. When the feature points collected in the process of positioning fall on these targets, the results calculated by the system will be abnormal.

The method proposed in this paper hopes to carry out semantic understanding of relevant information in the image through the object detection method and filter the key frames and relevant information obtained to eliminate highly dynamic objects and unstable objects in semantic concept, so as to improve the positioning robustness of SLAM system.

2. Related Work

In WACV2018, Zhong et al. used the method of target detection to SLAM remove dynamic points. In order to synchronize the target detection process with SLAM process, only the key frames were detected, and then the method of feature matching probability was extended to the moving process to remove the influence of dynamic points. In the SLAM process, only the static points were tracked, and the slam mapping was projected onto the image. In order to improve the effect of target detection, the samples which are difficult to mine are used as training data [2]. In the research results of IROS2018, Yu et al. proposed a method of realizing semantic SLAM for dynamic environment, ds-slam, which combined semantic segmentation with motion consistency detection to remove dynamic points in the environment. In this method, the boundary of the object is obtained by semantic segmentation. If the dynamic points are located in the object according to the motion consistency detection, all the points in the object are denoted as dynamic points [3].

ECVV2018 port of articles, Shen Shao jieren team was proposed based on stereo vision the semantics of the 3D objects and track the autopilot, under the background of autonomous study the SLAM problem in dynamic environment, the Faster-R-CNN method for object detection using binocular camera, and the semantic information fusion to the solution of the unified optimization framework. ORB feature points can be divided into the background and object, first, use the background to estimate the maximum likelihood of the position of the camera and the position of the landmark point. After entering the posture tracking task of the camera, the trajectory is transformed into a priori object size information and semantic information; it is guided by the prior information. Next, the maximum posterior probability estimation method is used to estimate the position of the target, and finally the maximum posterior probability is transformed into a least squares problem to solve [4]. InIEEE2019, Wang et al. proposed SalientDSO, which changed the strategy of DSO [5] on the uniform selection of tracking points. First, the significance graph of the image was extracted through SalGAN [6] network, which was related to the attention of each pixel. Then, the image was semantically segmented by PSPNet [7], and the significance score of each pixel was adjusted by the semantic segmentation results to reduce the significance score of the region without information. Finally, the image was divided into k×k grids, and the median value of significance was calculated for each grid as the basis for each grid to be selected, and then the points of concern were further selected according to DSO in each grid. After the improvement, the change of point to light and angle of view in the significance region is more robust [8]. In 2018, on issues related to vehicle navigation, Ganti et al. proposed visual SLAM, where feature selection is determined by network uncertainty. In this paper, information entropy is calculated to determine whether the observed data is used to update the estimated state quantity, and the uncertainty of semantic segmentation is integrated into the information entropy and calculation. While calculating the entropy change of pose, if the category uncertainty of a feature point in semantic segmentation is higher, the entropy change of pose is lower and the feature is less easy to be selected. Through this method, the feature points are screened, the feature points with less information are removed, and the scale of the map is reduced without losing much accuracy [9]. In the research of ITSC2017, Murali et al. proposed a method of semantic landmarks to assist accurate vehicle navigation. By adding threshold signals to the factor graph, object categories were distinguished, and corresponding observation data were added to determine whether the corresponding observation data should be added, so as to remove relevant feature points of fixed objects [10].

In IEEE2018 robotics and automation journal, Bescos et al. proposed dylam, which meticulously processed the dynamic points of rgb-d input images, not only removing dynamic objects, but also restoring the background blocked by dynamic objects. For objects with motion characteristics, the method of no motion problem is adopted. The overlapping degree selected is the first five key frames of each current frame. When the difference between the projection point and depth value of the feature point in the current frame is calculated, when the difference between the feature point and the corresponding position in the depth map exceeds a certain threshold, the characteristic point will be determined as the dynamic point. In addition, the pixels with the same depth value around the dynamic pixel are also set as dynamic pixels. Finally, in order to prevent the accuracy of edge segmentation from causing the classification of points in the background as dynamic points, points with large variance in depth around dynamic pixels were set as static points [11]. Brasch IROS2018 article puts forward the monocular semantic SLAM for high dynamic environment, considering the high dynamic environment with a large number of dynamic objects, so this article does not directly point out the potential motion feature points but uses the idea of SVO depth filtering to estimate the dynamic rate of change of punctuation [12], constantly adding new observation data, and the depth of the right punctuation is updated. On the premise of the depth convergence of landmarks, the map is added. On this basis, a priori value is assigned to the static rate according to the output of the semantic segmentation network, and then the static rate of the landmark points is updated when new observation data is introduced to realize the smooth transition of the landmark points between dynamic and static [13]. In ICRA2018, Stenborg et al. explored the stability of long-term positioning using semantic segmentation while participating in vehicle positioning projects. It solves the problem that the time span of the current detected features and the saved map features in the application scene of automatic driving is large, and the robustness of the features is high. A localization algorithm based on semantic tag and 3D position is proposed. A unified observation model based on SIFT features and semantic features is defined, and the pixel category and corresponding landmark category in the image are consistent as much as possible by adjusting the pose [14]. In ICRA2017, Sean et al. proposed the probabilistic data association of semantic SLAM, unified optimization of geometric information, semantic information, and IMU data in an optimization framework, and solved the problem through EM algorithm, realizing a SLAM system with higher accuracy [15, 16]. Suwoyo et al. proposed the adaptive development of svsf for a feature-based slam algorithm using maximum likelihood estimation and expectation maximization, which is designed to solve the online problem of Simultaneous Localization and Mapping (SLAM) [17].

The existing research in the field cannot provide reasonable recommendation and removal of inappropriate information for SLAM technology based on feature method. The overall idea of this project is to combine semantic information with ORB-SLAM2 and improve the traditional visual SLAM scheme through acquisition of object level semantic information, for higher efficiency of the loop closure testing provides a good database. So this paper proposed a semantic recognition of environment objects in the natural scene through object detection, and specific content is a key frame and target layer between the SLAM problem, establish a relation between key frames and target layer method, through the practical significance of the target layer to judge the merits of the target layer information, and combining with the SLAM relationship of key frames on the vision and the related information, the relationship between the object level targets in each key frames and related information are obtained, so as to realize through the object-level semantic information to judge the pros and cons of key frames. The above methods can retain the information of high reliability and good stability for visual SLAM and process the key frames with poor reliability or low stability and the information related to key frames.

The paper is structured as follows: Section 3 addresses the semantic optimization, which includes four parts: local mapping based on object level information filtering, semantic bags of words, and key frame filtering and landmark point filtering. Section 4 demonstrates the benefits of the proposed SR in a real scenario. This paper ends in Section 5 with the conclusions.

3. Semantic Optimization

In this paper, the object detection algorithm YOLO (You Only Look Once) is used to process the RGB (Red, Green, and Blue) images corresponding to the key frames obtained by the tracking thread in the local map construction thread, obtain the category and position information of objects in these RGB images, and establish the data association between the key frames and objects through the word bag model. The probability that an object is a dynamic object is calculated according to the category and probability of the object output by YOLO. Then for each key frame corresponding to the RGB image through dynamic object corresponding to the influential factors and dynamic map point within the region accounted for calculating the factors influencing the dynamic score, thus to screening of key frames, the dynamic road signs point to filter, improve the stability of the process of building a local punctuation, and improve the matching relation between the visual framework.

3.1. Local Mapping Based on Object Level Information Filtering. The tracking thread only determines whether the current frame needs to be added with a key frame. It does not really add the map. The main function is local positioning. The local build thread of ORB-SLAM2 is primarily tasked with maintaining the local map and managing key frames. The local mapping thread screens the key frames obtained by the tracking thread, fuses the landmark points in the key frames, eliminates redundant key frames and landmark points, maintains a stable global map, and provides the filtered key frames for loop detection. The thread of the local graph modifies the position of the key frame and the location of the signpost by determining more constraint relations. Specific tasks include the addition of key frames, elimination of landmark points, creation and fusion of landmark points, local BA, and local key frame elimination.

This article fuses semantic information with ORB-SLAM2 to help ORB-SLAM2 better complete the task of locating and mapping. After receiving the key frame from the tracking thread, the local map is processed with YOLOv3 to obtain the semantic category label of the object in the image. Then, according to the dynamic probability of the given category label, the dynamic probability of the object in the image is presented. After obtaining accurate semantic labels and dynamic probabilities [18–20], combining with the position of objects output by YOLO, feature filtering is carried out for key frames and landmark points, so as to minimize the impact of dynamic objects on map construction and location.

In order to better complete the above process, this paper constructs the relationship between key frames and objects by analogy with the association between key frames and landmark points in ORB-SLAM2, so as to further analyze whether a key frame is redundant and to eliminate landmark points that are dynamic objects.

3.2. Semantic Bags of Words. In the process of inserting a key frame by the thread of local map construction, the common view needs to be updated for each key frame that is added; that is, a new node is added for the new key frame k_i, and the edge between the key frames that have a co-visual relationship with it is updated. The co-visual relationship is judged by the number of common landmarks. The spanning tree is then updated to link k_i to the key frame with which it views the most, and finally to calculate the word bag representation of the key frame k_i.

In order to find the corresponding relationship between the key frame and the object and facilitate the processing of dynamic feature points in the feature filtering stage, this paper USES yolov3-tiny to process the RGB image corresponding to the key frame passed in by the local map construction thread to obtain the category and position information of the object. After the classification and position information of the object is confirmed, the classifiers marked object category and the probability that the object is a dynamic object are used to classify the object. The information obtained after classification includes the position information, size information, category information, and the probability that the object belongs to a dynamic object.

According to the requirements of this paper, after obtaining object level information through YOLO, the relationship between key frames and objects is built through the word bag model. During the construction of the relationship between key frames and objects, O_i is stored for each object:

(1) Coordinates of landmark points located on the object

(2) Class label of object

(3) Object location and size information

K_i is stored for each key frame:

(1) The corresponding RGB image is used for detecting the object

(2) Corresponding depth image for generation point cloud

(3) The object observed in this key frame is recorded with a BoW, which is marked as BoW_O in this article to distinguish it from the BoW native to ORB-SLAM2

From the above description, the relationship between the key frame and the object can be established so that the key frame can find its related object, and vice versa.

The previous chapter of this paper has made a simple introduction to the word bag model and the special matching based on the word bag model, the generation of ORB-SLAM2 dictionary using the classical k-means algorithm for feature clustering [21, 22]; clustering problems often use the method of unsupervised machine learning, through the machine to find the law of data to complete the clustering. For the problem of establishing a dictionary of k words with N feature points in the image, each word can be regarded as a set of local feature points. In the following studies, on the basis of k-means, the hierarchical clustering method and k-means++ [23, 24] were proposed to improve k-means. Considering the efficiency of search, the K fork tree is used to express the dictionary on the basis of k-means. There are N feature points. The algorithm steps to construct a K fork tree with a depth of d and bifurcation into K are as follows:

(1) At the node of root, to ensure uniform clustering, k-means++ is used to cluster all samples into k class.

(2) Cross k to each node in the first layer of the tree. The samples belonging to this node are regrouped into k class, and the second layer is obtained.

(3) The leaf layer is the word by analogy.

The structure of the k fork tree is a k branch, and the tree with a depth of d can contain k^d words. Nodes outside the leaf layer are used for quick search. The leaf layer is used to construct words. The dictionary schematic diagram of k-fork tree is shown in Figure 1. ORB-SLAM2 puts the extracted ORB features of the image into a container and then mobilizes DBoW2's build interface to generate the word bag model.

In this paper, a word bag model between key frames and objects is established. Due to the limited number of object level objects and the classification basis of category labels, clustering is also unnecessary. For key frame A, use object level objects as words to build A word bag model such as 1. η represents the weight of the corresponding word, which is represented by the dynamic probability of the object corresponding to the word, and N represents the sum of the number of detected objects. Then, the corresponding depth of the k-fork tree is 1, and the value of k is N. Then, the dictionary schematic diagram of the object level word bag model is shown in Figure 2.

$$A = \{(w_1, \eta_1), (w_2, \eta_2), \ldots, (w_N, \eta_N)\} \triangleq v_A. \qquad (1)$$

3.3. Key Frame Filtering. Considering that there are two kinds of key frames containing dynamic objects, one is that the picture area occupied by dynamic objects is small, and the number of dynamic landmark points corresponding to dynamic objects is small. This paper considers to keep the key frames in this case and only remove the landmark points related to dynamic objects. The other case is that dynamic objects occupy a larger area of the picture, and there are more relevant dynamic landmark points. Such key frames contain limited information that can be used as effective landmark points. This paper considers eliminating such key frames. The object corresponding to each frame and the probability that the object is a dynamic object can be corresponding through the word bag model. The size, position, and relevant landmark points of the object can be searched through the corresponding dictionary.

First, according to the object's dynamic influence factors to selection of key frames, this paper designs a screening function key frames for screening, screening function is shown in formula (2) select, where *awh* for the area of the influence factors of dynamic object, a is the weight of the parameter, w, h by YOLO output calculation area factor, the second dfrac$M_s M_h$ as belonging to the influence factors of dynamic number of feature points of objects, including M_s for this belongs to the dynamic object in the key frames of mappoints number, M_h is the total number of key frame middle punctuation, d is the weight parameter of the item, C_i and D_i are, respectively, the probability that the object belongs to the ith category, and the probability that the object of the ith category is a dynamic object, $C_i D_i$ is the probability product, and sum$_{I=1}^{N} C_i D_i$ is the probability product sum of the object category of N.

Above take into account the dynamic object and the area of dynamic relative map points which accounted for two factors, the type of calculation on key frames of an object selection function, set in each key frames can identify an object, a total of count, key frame selection function calculating formula is shown in formula (3), *f* parameter, used to adjust the output in the range of [0, 1], setting out the key frames of threshold value is S_{com}, calculated judgment S_{sum} is greater than S_{com}, if formed, delete the corresponding key frames, and make the local graph thread receive the new key frame.

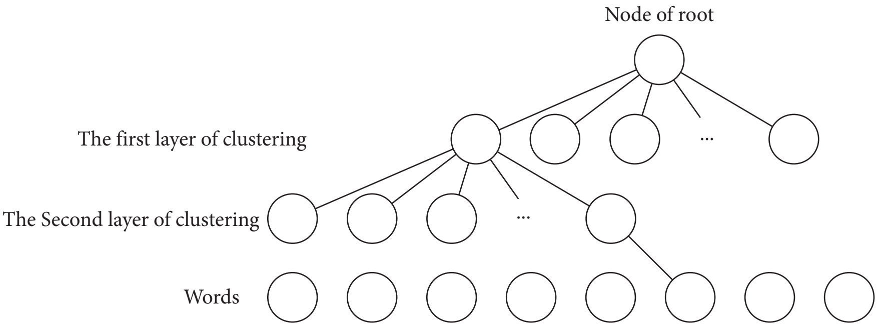


Figure 1: Semantic diagram of k-tree dictionary.

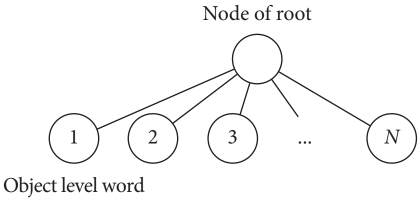


Figure 2: Dictionary diagram.

$$S=\sum_{i=1}^{N}C_iD_i\left(awh+d\frac{M_s}{M_h}\right), \tag{2}$$

$$S_{\text{sum}}=f\sum_{j=1}^{\text{count}}S_j. \tag{3}$$

3.4. Landmark Point Filtering. After adding key frames, the first thing you need to do is get rid of the bad signposts. A signpost in ORB-SLAM2 that wants to stay on the map must pass a rigorous test to determine if it can be traced to three consecutive key frames after it has been created. YOLO, this paper provides semantic information and adds a filtering wheel combining semantic information before ORB- SLAM2 original map points to filter the correct markers, filtering the features and placing key frames on the map points relative to the dynamic objects after filtering the step to calculate each object corresponding to the four formulas of key frames, each product of the probability of the corresponding N object category, and comparing with the corresponding threshold, in order to exceed the threshold map points associated with this object, optimize the total view of the key frames. Auxiliary key frames and landmark points are free from the interference of dynamic objects. The most important thing is that the landmark points in the local map are removed and the retained information is more meaningful.

$$\sum_{i=1}^{N}C_iD_i. \tag{4}$$

After the first round of screening, the landmarks associated with dynamic objects have been removed. While maintaining ORB-SLAM2's original screening process, in order to ensure a good relationship between the signposts and key frames, the following two conditions are required for good signposts to be screened:

(1) More than 25% frames can be observed in theory

(2) After the punctuation is created, it can be tracked by at least three consecutive key frames

According to the above conditions, the map points do not meet the conditions. Even if the road marking points meet the above conditions, the map points cannot be guaranteed not to be deleted. When the corresponding key frame is deleted, the map points are regarded as local BA external points, and the map points still belong to dynamic objects and will continue to be filtered.

The above two rounds of operations have preserved the good signpost points, and based on this, some new signpost points have been recovered by solving the ORB feature points connected to the key frames in the common view using PnP. Similarly, traverse the current key frame corresponding to the level of adjacent and secondary adjacent key frames, the current frame with the corresponding map points fusion, and finally, the current key frame corresponding to the property of the map points updates; these properties include the average direction of observation, observation distance, and best descriptor, such as the new signs point of view, the spanning tree.

4. Experiments

After the key frame is inserted, firstly, the object detection task is performed on the RGB image corresponding to each inserted frame through YOLOv3 to obtain the corresponding object category and size information in the frame, and the dynamic rate of the object is added according to the object category, and then according to the above information according to the method provided in this article to the key frame and landmark point selective filtering. This section conducts experiments for these two parts.

4.1. Filtering of Key Frames Based on Object Level Information. To filtering, key frames in the thread of local embedding key frame selection, based on YOLOv3 object categories, dynamic rate, size, and location information, get relevant dynamic waypoint information, comprehensive overall dynamic keyframe information judgment; the influencing factors of high dynamic key frames are: avoid the dynamic

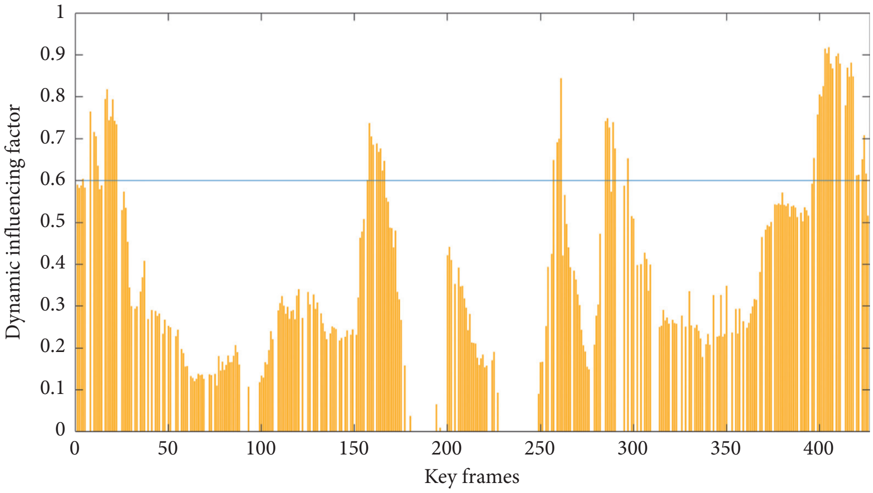


Figure 3: Key frame filtering.

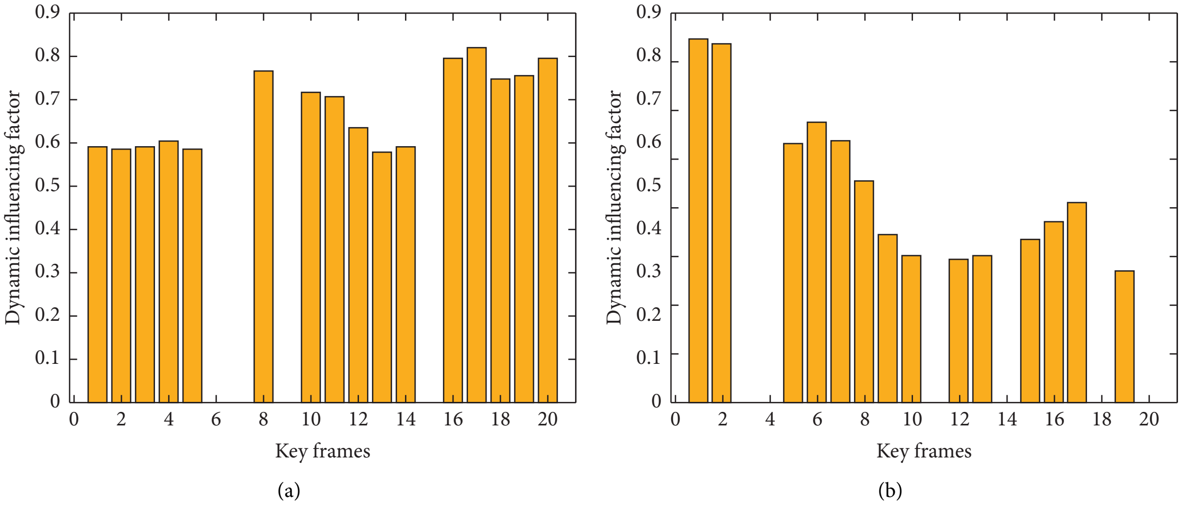
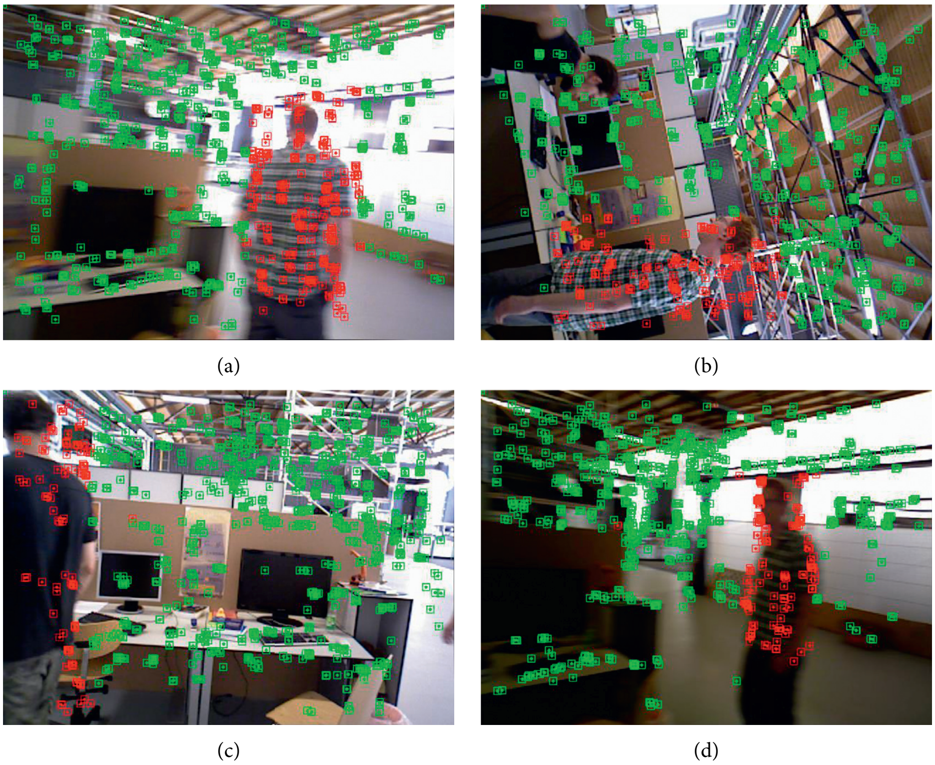


Figure 4: Key frame filtering of the first 40 key frames.

object, the more the greater the dynamic object, or the more dynamic mapping point retain key frames, key frames and the influence of the loopback detection obviously. In this paper, the dynamic influencing factors of key frames are calculated by formulas (2) and (3).

This section conducted experiments on the above process on the TUM dataset, which retained 426 key frames after tracking the thread. Each key frame was taken as the horizontal axis, and the dynamic influencing factors of the calculated key frames were taken as the vertical axis. The threshold value $S_{\text{com1}} = 0.6$ was taken to obtain the bar statistical graph as shown in Figure 3. The horizontal line of $y = 0.6$ in the figure is the cut-off line represented by the threshold value, and the corresponding key frame with the ordinate value higher than this line is removed.

For more convenient explanation, the data of 1–20 frames and 21–40 frames are enlarged and displayed. See Figure 4 for details. The left figure shows the data of 1–20 frames and the right figure shows the data of 21–40 frames. In this figure, it can be seen that the dynamic influencing factors of frames 6, 7, 15, 23, 24, 31, 34, 38, and 40 are zero, indicating that there is no dynamic object in these key frames. To remove frames 8, 10, 11, 12, 16, 17, 18, 19, 20, 21, 22, etc., in which the dynamic factor is high, the removal of these key frames can eliminate the unreliability of the total view, depending on the relationship, largely avoiding the concentration of many dynamic feature points on dynamic objects which will provide wrong action orientation, from the entire data set key frames as shown in Figure 4, It can be seen from Figure 20 that 10 frames and around 170–160 frames can be observed. Frames 400–420 are the continuous key frames of the concentrated activity of dynamic objects, and the key frames of these positions are selectively eliminated by the judgment of dynamic influencing factors. The RGB images corresponding to these position key frames are selectively posted in Figure 5. Two people can be observed in


(a)

(b)

(c)

(d)

FIGURE 5: The RGB images corresponding to the key frames filtered out.

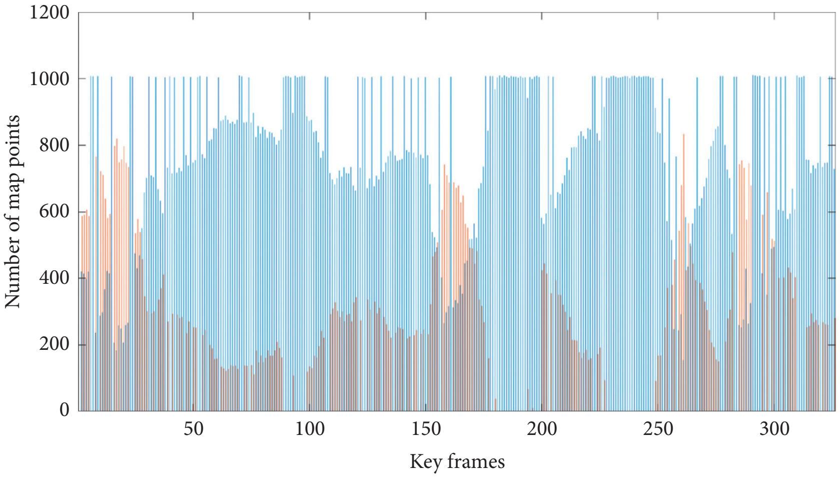


FIGURE 6: Map point filtering.

the key frames at these positions, which occupy a large area of the whole RGB image and have extremely high activity.

Through the above experiments, it can be seen that the improved ORB-SLAM2 has accurately filtered the extremely high key frames of dynamic influencing factors from the tracking thread in the local map construction thread, providing a good foundation for the stability of loop detection.

4.2. Filtering of Landmark Points Based on Object Level Information. Through the above process, the total area distribution is larger, and there are more dynamic object keyframes, but some dynamic objects still occupy less area, and the related dynamic map points have fewer keyframes, so the corresponding dynamic map points are also retained. Some buildings, roads, and signs affect the accuracy. Therefore, in this paper, through formula (4), the dynamic

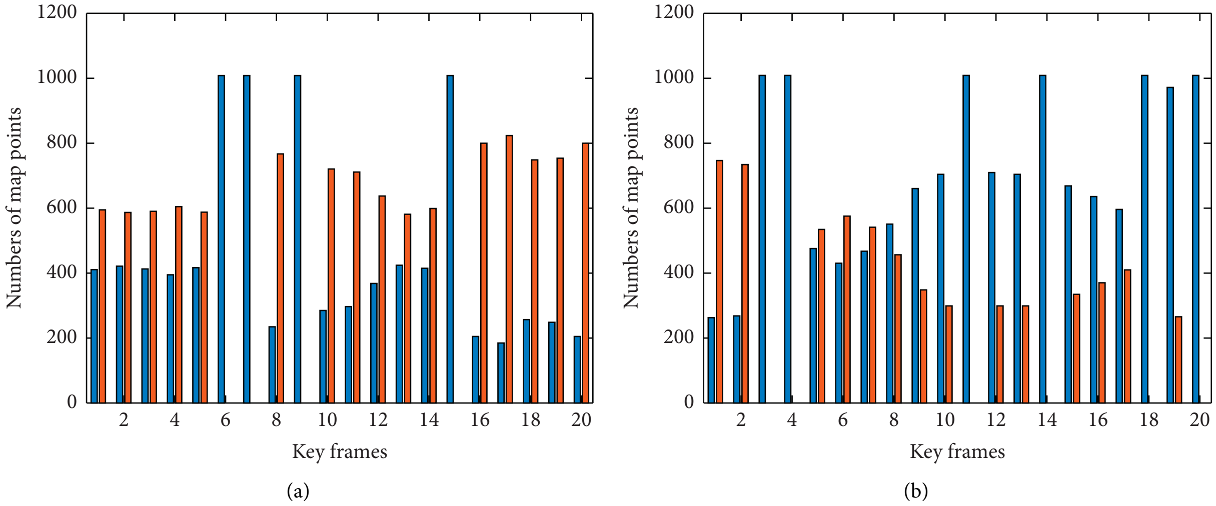


Figure 7: Map point filtering of the first 40 key frames.

influencing factors of dynamic objects related to dynamic landmark points are determined, and the corresponding key frames are retained, but the object-related landmark points with relatively large dynamic influencing factors are removed.

In this section, experiments were conducted on the TUM dataset on the above process. Each key frame was taken as the horizontal axis, and the number of signposts was taken as the vertical axis. The filtering of signposts is represented by Figure 6. In order to facilitate the explanation, the data of 1–20 frames and 21–40 frames in the remaining key frames after key frame filtering were enlarged and displayed. See Figure 7 for details. The left figure shows the data of 1–20 frames and the right figure shows the data of 21–40 frames. Among them, frames 7, 15, 23, 24, 31, 34, 38, and 40 have no corresponding orange bars, indicating that there is no dynamic object in these key frames, or the probability of dynamic object is very small. The landmark points are not considered to be eliminated, and the road punctuation is eliminated to different degrees in other frames. The landmarks correspond to the feature points obtained by ORB detection, and the elimination of the corresponding landmarks in 3D space can be expressed by the feature points, which can be represented by displaying the feature points on the RGB image corresponding to the key frame.

Extract four keys to weed out corresponding map points corresponding to RGB images, as shown in Figure 5. To weed out the 3D map points corresponding feature points shows red, not out of the 3D map points corresponding feature points according to green. People can be seen in the figure; the dynamic related feature point is marked in red.

As can be seen from the above experiments, this paper improved the selection of punctuation in the middle of ORB-SLAM2 by filtering signposts based on object level information. Dynamic signposts were removed correctly while static signposts with high stability were retained, which improved the stability of ORB-SLAM2.

5. Conclusions

The overall idea of this project is to combine semantic information with ORB-SLAM2 and improve the traditional visual SLAM scheme through acquisition of object level semantic information. YOLO aids the process of selecting keyframes and selecting map points through object detection algorithms for visual SLAM object-level semantic information, enabling ORB-SLAM2 to obtain more stable static map points and more stable keyframes for loopback, and to preword wrap objects through keyframes and object models, providing a good database for higher closed-loop testing efficiency.

Key frames received by the local build thread of ORB-SLAM2 are filtered through object level information. By detecting the RGB image corresponding to each key frame through YOLO, the category, position, and size information of the object corresponding to each frame can be obtained. After the dynamic probability of the object is increased according to the category of the object, the dynamic influencing factors are calculated for each key frame based on the above information, and the key frames whose dynamic influencing factors exceed the threshold are eliminated. On the one hand, this improves the reliability of the common view and prevents the co-vision relationship generated under the condition that there are many landmarks related to the dynamic objects in the adjacent key and the degree of co-vision is high. This co-vision relationship is unstable and may no longer exist with the movement of the dynamic objects. On the other hand, it increases the stability of loop detection and prevents the judgment of loop detection from being affected by the repeated detection of large objects with more relevant dynamic landmarks. Through experiments, it was found that the RGB images corresponding to the deleted key frames had larger dynamic objects and more dynamic landmarks, which was in line with the original intention of the design and achieved a better effect.

On the basis of key frame filtering, the dynamic feature points corresponding to the reserved key frames were removed to screen out more reliable signpost points for ORB-SLAM2, which is conducive to the accuracy of positioning and mapping. Through the experiment, it is found that the dynamic landmark points retained by the key frame filter are removed accurately and the desired effect is achieved.

This paper provides reasonable recommendation and removal of inappropriate information for SLAM technology based on feature method. The methodology is to propose a semantic recognition of environment objects in the natural scene through object detection. The method can retain the information of high reliability and good stability for visual SLAM and process the key frames with poor reliability or low stability and the information related to key frames.

Data Availability

This publication was supported by TUM RGB-D datasets, which are openly available at location cited in [25].

Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This research was funded by the Fundamental Research Foundation for Universities of Heilongjiang Province (LGYC2018JQ003) and University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (No. UNPYSCT-2018208).

References

[1] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6243–6252, Seattle, WA, USA, June 2017.

[2] F. Zhong, S. Wang, Z. Zhang, and Y. Wang, "Detect-slam: making object detection and slam mutually beneficial," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1001–1010, IEEE, Lake Tahoe, NV, USA, March 2018.

[3] C. Yu, Z. Liu, X.-J. Liu et al., "Ds-slam: a semantic visual slam towards dynamic environments," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168–1174, IEEE, Madrid, Spain, October 2018.

[4] P. Li and Q. Tong, "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 646–661, Glasgow, UK, August 2018.

[5] R. Wang, M. Schworer, and D. Cremers, "Stereo dso: large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3903–3911, Venice, Italy, October 2017.

[6] J. Pan, C. F. Cristian, and K. McGuinness, "Salgan: visual saliency prediction with generative adversarial networks," 2017, http://arxiv.org/abs/1701.01081.

[7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Seattle, WA, USA, June 2017.

[8] H.-J. Liang, J. Nitin, C. Fermüller, and Y. Aloimonos, "Salientdso: bringing attention to direct sparse odometry," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1619–1626, 2019.

[9] P. Ganti and S. L. Waslander, "Visual slam with network uncertainty informed feature selection," 2018, http://arxiv.org/abs/1811.11946.

[10] V. Murali, H.-P. Chiu, S. Samarasekera, and R. Teddy Kumar, "Utilizing semantic visual landmarks for precise vehicle navigation," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, IEEE, Yokohama, Japan, October 2017.

[11] B. Bescos, J. M. Facil, J. J. Civera, and J. Neira, "Dynaslam: tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[12] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[13] N. Brasch, Aljaz Bozic, L. Joe, and F. Tombari, "Semantic monocular slam for highly dynamic environments," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 393–400, IEEE, Madrid, Spain, October 2018.

[14] E Stenborg, C. Toft, and L. Hammarstrand, "Longterm visual localization using semantically segmented images," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6484–6490, IEEE, Brisbane, Australia, May 2018.

[15] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, John Wiley & Sons, Hoboken, NJ, USA, 2007.

[16] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1722–1729, IEEE, Marina Bay Sands, Singapore, May 2017.

[17] H. Suwoyo, W. Y. Tian, A. L. AdriansyahLi, and G. Yuan, "Adaptive development of svsf for a feature-based slam algorithm using maximum likelihood estimation and expectation maximization," *IIUM Engineering Journal*, vol. 22, no. 1, pp. 269–286, 2021.

[18] Ji-H. Xi, K.-M. L. Dong-Hyun Lee, and C.Ho Lin, "An improved yolov3-based neural network for de-identification technology," in *Proceedings of the 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Korea, March 2019.

[19] F. Zeng and C. Wang, "Visual navigation with asynchronous proximal policy optimization in artificial agents," *Journal of Robotics*, vol. 2020, Article ID 8702962, 7 pages, 2020.

[20] J. Ni, T. Gong, Y. Gu, J. Zhu, and X. Fan, "An improved deep residual network-based semantic simultaneous localization and mapping method for monocular vision robot," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 7490840, 14 pages, 2020.

[21] S. Lloyd, “Least squares quantization in pcm,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] J. Song and L. B. Kish, “On the theory and design of cold resistors,” *Fluctuation and Noise Letters*, vol. 20, no. 1, Article ID 2150001, 2020.

[23] D. Arthur and V. Sergei, “k-means++: the advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, USA, January 2007.

[24] Z. A. Ali, Z. Han, and Bo H. Wang, “Cooperative path planning of multiple UAVs by using max–min ant colony optimization along with cauchy mutant operator,” *Fluctuation and Noise Letters*, vol. 20, no. 1, Article ID 2150002, 2021.

[25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *Proceedings of the International Conference on Intelligent Robot Systems (IROS)*, Vancouver, Canada, October 2012.

Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6634773, 12 pages
https://doi.org/10.1155/2021/6634773

Review Article

Recent Advancements in Autonomous Robots and Their Technical Analysis

Boni Liu

School of Electronic Engineering, Xi'an Aeronautical University, Xi'an 710077, China

Correspondence should be addressed to Boni Liu; 200607004@xaau.edu.cn

Received 8 December 2020; Revised 7 January 2021; Accepted 19 January 2021; Published 3 February 2021

Academic Editor: Bhawani Shankar Chowdhry

The purpose of this paper is to discuss and present a technical analysis of the recent advancements in autonomous robots equipped with a manipulator. The autonomous robots include unmanned aerial vehicle (UAV), unmanned underwater vehicle (UUV), and unmanned ground vehicle (UGV). A manipulator can make an autonomous robot more adaptable and robust but it can also affect its performance as well. Several issues can arise because of the installation of a manipulator like the robot becoming unstable due to the extra weight, slow convergence, and errors in the path planning. Therefore, this study presents the numerous recent techniques that are in use to counter the aforementioned problems. The methodology and approach used in this paper are to first present the dynamic model of the autonomous robot. Then, the study offers a performance analysis of the specific robot in question. Finally, the paper formulates the limitations of the recently proposed techniques in the form of a table for each vehicle. The key findings of this study are a comprehensive review of the aforesaid techniques and their technical analysis. The unique contribution of this study is to present some of the limitations that these methods have so the researcher can better select the method according to the mission requirement.

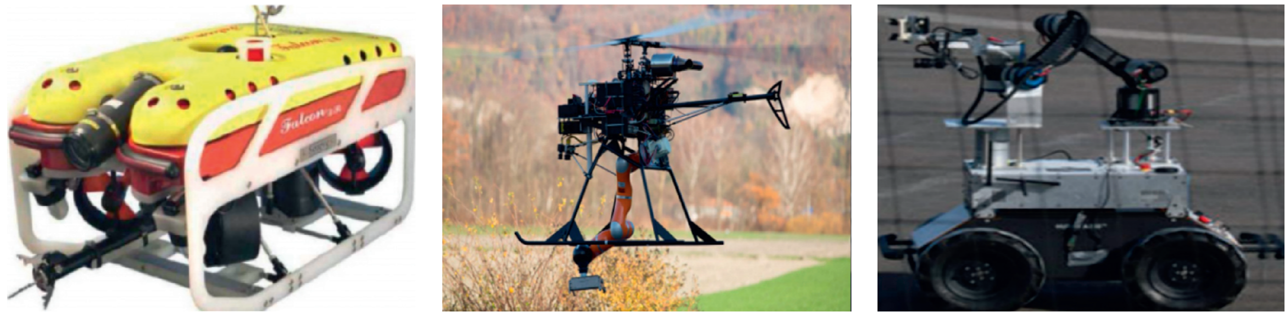
1. Introduction

For the past few years, the researchers are engaged in evaluating the performance of autonomous vehicles with the addition of manipulator design due to the emerging demand in executing the number of flexible tasks in any dull, dirty, difficult, or dangerous environment [1–3]. These manipulators provide easy access to perform several jobs with merely small inertia, high load to weight ratio, and smart flexible structure [4]. For complex dynamic models with time delays in output variables and unmodeled dynamic factors, high-performance tracking has been observed as still one of the challenging tasks.

Acquiring the real and precise dynamics of the system during control design is among the complicated and strenuous activities of the procedure. The researchers in this regard are opting for some hybrid-type control algorithms to improve the tracking performance [5–7]. One may design such control designs, but they require the tuning of several parameters. In short, one researcher has two tough approaches either to acquire an exact mathematical dynamic model of autonomous vehicles or to estimate the numerous parameters for control design to produce refined input logic for the proposed system. Before going through the literature review, one should understand the types of autonomous unmanned vehicles. These vehicles are autonomous because of their ability to perform any sort of task without any intervention of human beings. Figure 1(a) shows a UUV with a manipulator [8], Figure 1(b) shows a UAV with a manipulator [9], and Figure 1(c) presents a UGV with a manipulator [10].

This entire review paper discusses the four types of unmanned vehicles embedded with gripper, that is, unmanned underwater vehicle (UUV), underactuated quadrotor unmanned aerial vehicle (QUAV), unmanned ground vehicle (UGV), and last but not least unmanned air-cushion vehicle (UACV). In addition to this, the paper addresses the constraints such as the occurrence of time delays and exogenous disturbances in a system.

The idea for embedding the UUV with a manipulator is introduced many times, that is, [11, 12]. This is because of



(a) (b) (c)

Figure 1: Types of unmanned vehicles embedded with a manipulator.

enabling an ability to grasp the target in water. Most of the UUVs have six degrees of freedom but only four actuators which make them underactuated system. These underactuated systems are very hard to control. Thus, for the stabilization, various hybrid control designs are introduced by researchers. In the catalog of such controllers, one may see model reference adaptive control (MRAC), sliding mode control (SMC), and many other robust control strategies [13–16].

Since the last decade, the extensive use of UAVs has been observed in various fields, either for commercial purposes, that is, surveillance [17, 18], or for military-oriented tasks. This type of unmanned vehicle got a great boom because of its aggressive maneuverability [19–22] over a long field of distance. Researchers have also tried to embed smart manipulator/gripper mechanism [23, 24], with UAV in order to increase the utility of drones in multiple fields. Researchers previously proposed commonly 01 and 02 DOF-based manipulators with unmanned aerial vehicles, that is, quadrotor [25]. Researchers were engaged initially in optimizing the control performance for the control law associated with the above manipulators [26]. The researchers also proposed some advanced mechanical designs and typical construction of quadrotor embedded with grippers of lightweight but with great capability to grasp the object within the working envelop [27].

Researchers also embedded some smart manipulators on such UGVs, that is, [28]. These smart manipulators have increased the manipulating ability to move up to 250 kg mass from one point to another. It is believed that a UGV must have good speed and navigation systems to monitor and manipulate the objects within harsh terrain [29]. Thus, in literature, one may find several types of manipulators as discussed by [30–32]. In most cases, it is recommended to use servo motors for ideal torque and mass ratio. In addition to this, a servo motor can be controlled easily. In today's era, researchers proposed different microcontrollers for experimental design, that is, Raspberry Pi [30], Arduino [33], or any modular programmable logical controller (PLC).

The motivation behind this paper was to collate the research studies about autonomous robots in one place so that new authors and researchers can easily compare the benefits and limitations of each study and pick the one most optimal for their mission requirement.

The main contributions of the paper are to provide one comprehensive review and technical analysis of the old and new studies about the UUVs, UAVs, and UGVs, to shed light on the limitations of the aforementioned studies in the form of an easily accessible table.

The paper is arranged as follows: Section 2 presents some cutting-edge research into autonomous robots. Section 3 discusses the UUV with the manipulator, its dynamic model, and its performance analysis and finally sums up the limitations of the previous techniques in a table. Similarly, Section 4 deals with the UAV, and Section 5 handles the UGV. Then, Section 6 provides technical analysis, and lastly, Section 7 concludes the whole study.

2. State of the Art

The state-of-the-art approach for UUV is discussed in [34] where a nonlinear observer-based model is amalgamated with dual proportional integral derivative (Dual-PID) design. This research provides comparatively effective results for 06 degrees of freedom (DOF) UUV with 02 DOF manipulator.

Researchers in [35] present a state-of-the-art technique for UAVs using 5G networks in a smart city. The researchers use blockchain-based solutions to secure these 5G networks for industrial and defense purposes.

Academics in [36] offer a novel idea of integrating UGV and UAV for construction site data collection. The UGV is autonomous and travels using the help of its sensors and the UAV which alerts it of any danger not visible to UGV on the ground.

3. Unmanned Underwater Vehicle Equipped with Manipulator Design

For repairing the structures, mostly in the offshore oil industry, these UUVs are highly recommended. This is because of their capability to reach in the depth of the sea unlike humans [37]. These UUVs have been blessed with two main abilities, that is, position stalking and dynamic stalking. This means that a UUV can maintain all positions throughout time with respect to the body.

One should not forget about the underwater dynamics that can lead to huge turbulences. These underwater dynamic factors are hydrodynamic coefficients and the mass flow rate through water inlets [11].

Discussing the previous works related to UUV, in [14], the author addressed the behavior by using SMC. An extended version, that is, higher-order SMC (HOSMC) can also be seen in [16] where the chattering phenomena (high number of oscillations) were reduced using higher order of SMC [13]. In some of the research works, one may see the use of a dual control scheme, that is, using proportional derivative (PD) and proportional integral control (PID) controller to stabilize the underactuated dynamics of UUV. Since this PID and PD, dual scheme produces fine results but in the presence of nonlinearities, this will never hold up the response for so long and shall lead it towards instability.

Researchers [13] proposed multivariable sliding mode control for the stabilization of attitude and position of a UUV equipped with a manipulator. In this case, the proposed UUV is a fully actuated system (number of control inputs are equal to degrees of freedom) that is why it can grasp any object underwater easily but simultaneously the power consumption by the actuators is huge as compared to underactuated UUVs.

3.1. Dynamic Model of UUV.

The dynamic model for UUV is achieved after going through the study of both frames of references, that is, Earth frame of reference (inertial frame) and fixed body frame (noninertial frame). The common design is comprised of six actuators that lead to six DOF easily [38]. In this subsection, the main idea related to the state-of-the-art approach is discussed.

Here, the Dual-PID control techniques are fused with the nonlinear model-based observer to stabilize the fully actuated underwater vehicle. This strategy is applied on six DOF-based UUVs embedded with a gripper/manipulator of two DOF; this makes in total eight DOF to control. Figure 2 shows a six DOF UUV physical model embedded with a manipulator [39]. Table 1 presents the orientation, translational, and angular velocities, forces, and moments along with the degrees of freedom for the UUV.

The position and orientation with respect to the inertial frame are given as

$$\mu_1(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \mu_2(t) = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}. \tag{1}$$

In equation (1), $\mu_1(t)$ and $\mu_2(t)$ are the vectors that describe the position and angular velocities of UUVs. The column matrix $\mu_2(t)$ is also known as the attitude of the vehicle.

$$\begin{aligned} T_1(t) &= \begin{bmatrix} u & v & w \end{bmatrix}^T, \\ T_2(t) &= \begin{bmatrix} p & q & r \end{bmatrix}^T, \end{aligned} \tag{2}$$

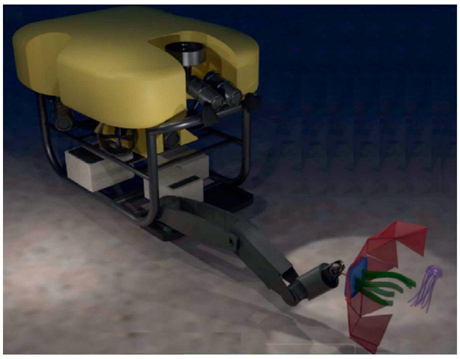


FIGURE 2: UUV with a manipulator.

where $T_1(t)$ represents the translational velocities whereas $T_2(t)$ represents the angular velocities. By combining the translational and angular velocities as in [38], we get

$$v(t) = [T_1(t), T_2(t)]^T \varepsilon \mathfrak{R}^6. \tag{3}$$

The rotation matrix can be derived using Newton–Euler methods as

$$R = \begin{bmatrix} c_\theta c_\psi & -c_\theta c_\psi + c_\psi s_\phi s_\theta & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + c_\phi s_\theta s_\psi \\ -s_\theta & c_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{4}$$

where c_x means $\cos(x)$ and s_x means $\sin(x)$. The dynamic model for UUV is shown as follows:

$$\begin{aligned} M(t)\dot{v}(t) &= g(v(t)), \mu(t), U(t) + \tau_{e(g)}, \\ \dot{\mu}(t) &= h(v(t), \mu(t)). \end{aligned} \tag{5}$$

In the above set of equations, $M(t)$ is the inertial matrix that comprised hydrodynamic mass change and functions (e.g., one can see $(.)$ and $h(.)\ \varepsilon\mathscr{R}^{6x1}$). $M(t)$ is the sum of centripetal mass and Coriolis body mass mentioned as follows [40]:

$$M(t) = M_{CR}(t) + M_{CP}(t). \tag{6}$$

3.2. Kinematics of UUV.

As per the conventional study by [41], the kinematic set of equations are given as in equations (7) and (8):

$$\begin{cases} \dot{\varphi} = p + q\sin\varphi.\tan\theta + r\cos\ \tan\theta, \\ \dot{\theta} = q\cos\varphi. - r\sin\varphi, \\ \overset{\vdots}{\psi} = \dfrac{q(\sin\varphi + r\cos\varphi)}{\cos\theta}, \end{cases} \tag{7}$$

$$\begin{cases} \dot{x} = u(c_\theta c\psi) + (s\theta s\varphi c\psi - s\psi c\varphi)v + (s\theta c\varphi s + s\psi s\varphi), \\ \dot{y} = u(c_\theta s\psi) + (s\theta s\varphi s\psi + c\psi c\varphi)v + (s\theta c\varphi s\psi - c\psi s\varphi)w, \\ \dot{z} = -us_\theta + (c\theta s\varphi)v + (c\theta c\varphi)w. \end{cases} \tag{8}$$

Table 1: Symbols for the 6 DOF UUV.

Position	Linear and angular velocities	Forces and moments	Degree of freedom
x	u	X	Surge
Y	v	Y	Sway
Z	w	Z	Heave
ϕ	p	K	Roll
θ	q	M	Pitch
ψ	r	N	Yaw

3.3. Modeling of Manipulator Design. For modeling the manipulator design, one should consider the moments of the arm as an external torque. Since the attached manipulators connected with UUV are based on two links and one joint mostly using a simple servo motor. The kinematics for this gripper/manipulator is stated by [42] via opting for direct kinematics. This method helps to compute the orientation by finding the nth number of joints and compute the position of the end effector to grasp the object correctly.

Thus, the nth number of joints can be expressed as $q \in R^n$, whereas the position of the end effector is expressed as $\left\{n_p \in R^3, n_0 \in R^3\right\}$. One can now develop a relationship between position and orientation easily as provided as follows:

$$n_{p,o} = \left[n_p, n_o\right]^T \in R^6. \tag{9}$$

Researchers have used the Denavit–Hartenberg (DH) formulation to find the configuration of an end effector of the gripper. Moreover, the dynamics of UUV stated that the total forces and torque that are acting on the body of UUV in the deep sea can be expressed in generic as

$$\begin{cases} f_i = m_i\left[a_i + \dot{T}_2(t) \times r_i + T_2(t) \times \left(T_2(t) \times r_i\right)\right], \\ T_m = I \times \dot{T}_2(t) + T_2(t) \times \left(I \times \dot{T}_2(t)\right]. \end{cases} \tag{10}$$

In equation (10), m_i is an additional mass due to the manipulator and the weight of the object that must be grasped by the manipulator. Moreover, r_i is the vectoral distance from the origin frame I towards the center of gravity of the link. The variable a_i is the translational acceleration from the origin of the frame, whereas $\dot{T}_2(t)$ is the vector denoting the change in angular velocities where $T_2(t)$ is the vector consisting of rotational velocities.

3.4. Performance Analysis of UUV Equipped with Manipulator. There is an effective need for an autonomous unmanned underwater vehicle due to several issues. The important thing at this moment is to save the lives of our divers and get efficient results by using UUVs beneath the sea more than the depth covered by divers. Figure 3 shows a UUV equipped with a manipulator that has multiple links and joints to grasp the object [38].

Researchers also used the bioinspired dolphin algorithm for controlling the locomotion of UUVs like a real dolphin. Figure 4 presents a UUV hardware design based on a bioinspired dolphin algorithm [11].

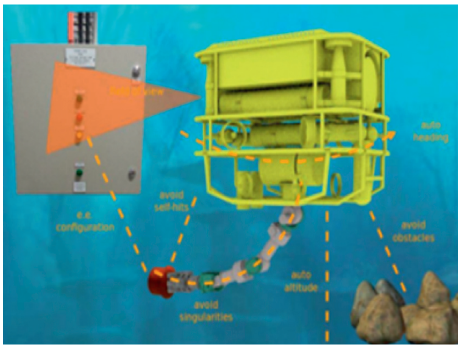


Figure 3: UUV equipped with a manipulator.

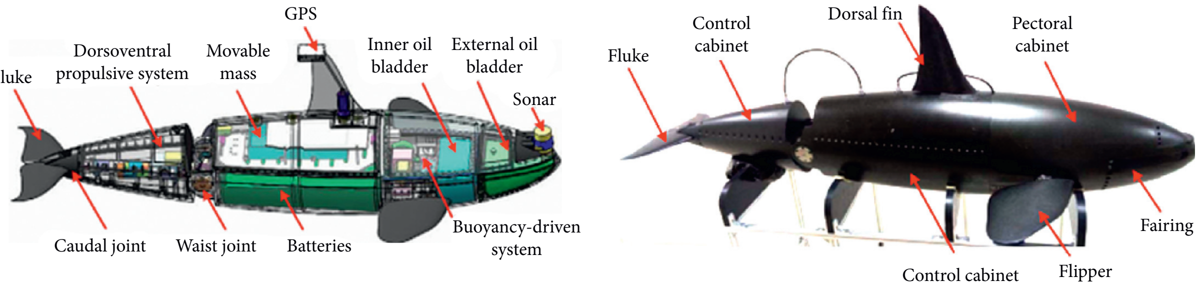
Using the Newton–Euler dynamic method shown in Figure 5, we can derive the equation of motions [40]. This method is easy but has some limitations such as gimbal lock due to singularity issues.

After going through the latest papers and current state-of-the-art approaches, Table 2 presents the previously proposed techniques for UUVs, the hardware they are applied on, and their limitations.

4. Underactuated Quadrotor UAV Equipped with Manipulator Design

Like underwater vehicles, unmanned aerial vehicle and its dynamic model are also derived from the Newton–Euler method. This method is frequently opted for by various researchers because of less complexity. The only limitation of this approach is the gimbal lock due to the singularity issue which can be reduced, not eliminated completely through hyperbolic tangent function. These equations involve the trigonometric functions; hence, the computation time for these terms is usually huge, and therefore expensive programmable controller is selected which leads to an expensive hardware design [44].

The focus for the UAVs in this paper is set on quadrotor type of UAVs. This is because of a fewer number of actuators that result in less power consumption and long battery time for flight [44–46]. Researchers modified the quadrotor with multirotors as well as the manipulator designs too. There are also some hybrid control schemes too, previously proposed by [47, 48], for the stabilization of the entire behavior of UAV with a gripper mechanism. The same control laws that were proposed before for UAV are proposed here too such as model reference adaptive control, a hybridized version with sliding mode control for quadrotor UAV (QUAV) equipped with 2 DOF manipulator [49].



(a)

(b)

Figure 4: UUV hardware design.

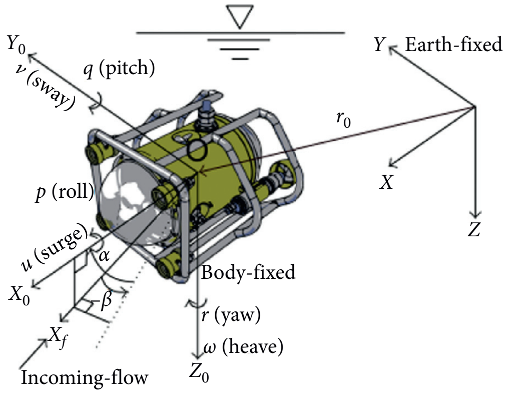


Figure 5: Deriving the equations of motion using Newton–Euler method.

4.1. Performance Analysis of UAV Equipped with Manipulator. The paper discusses two categories of unmanned aerial vehicles. One is a cable-driven UAV and the second one is equipped with a manipulator design. The current approaches for the trajectory tracking of UAVs are discussed by [50, 51], in which an adaptive robust control law is proposed. The lumped dynamics are estimated using estimator design and in addition to this, the chattering phenomenon is also eliminated [52]. For unknown modeled factors, one can see several such hybrid techniques. Researchers have also proposed subcontrol blocks, that is, time delay estimator, supertwisting law, and fractional-order SMC technique to minimize the dynamic error in the system. Researchers in this way obtained 20% efficiency for quadrotor equipped with 2 DOF manipulator on a referred path [53]. The hybridized version of the regulation, pole-placement, and tracking (RST) control design with MRAC and the stability for tracking and grabbing the objects both are proved using MIT rules. The work proposed by [53], where quadrotor varies its mass from 0.5 kg to 5.0 kg, and the manipulation related parameters are achieved using the Denavit–Hartenberg principle. Table 3 presents the previously proposed techniques for quadrotor UAVs, the hardware they are applied on, and their limitations.

4.2. Dynamic Model of QUAV with 2 DOF Gripper. Researchers avoided the different parameters, that is, aerodynamic effect, ground effects, and flapping of blades, and proposed an overall dynamic model. Figure 6 shows the overall model of a dynamic quadrotor with a 2 DOF manipulator [53, 54]. The Newton–Euler method is the most frequent method used and stated by the majority of the people. One can find the separate models as well like in [54] but this will increase the complexity.

The overall mathematical model including the gripper dynamics is derived and stated as

$$\begin{bmatrix} f_q(U) \\ T_q\left(U,q_A^j\right) \end{bmatrix} + \begin{bmatrix} f_A\left(\partial q_A^j\right) \\ T_A\left(\partial q_A^j\right) \end{bmatrix} = \begin{bmatrix} m\bar{I} & 0 \\ 0 & I\left(q_A^j\right) \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \times I\omega \end{bmatrix}. \tag{11}$$

In equation (11), the left-hand side is the dynamics of quadrotor and manipulator, respectively, whereas the right-hand side is the vector form of rigid quadrotor body. The term like $f_q(U)$ is the control input force and torque given as $T_q(U, q_A^j)$ for quadrotor whereas the force exerted by manipulator is given as $f_A(\partial q_A^j)$ and torque as $T_A(\partial q_A^j)$. m is the total mass and I is an identity matrix of (3×3) order. $I(q_A^j)$ is the total inertia. Moreover, $\dot{v}$ and $\dot{\omega}$ are angular velocities.

5. Unmanned Ground Vehicle Equipped with Manipulator Design

5.1. Dynamic Model of UGV with Manipulator Designs. As discussed, these unmanned vehicles are deployed in such tasks that are far away from human management. UGVs are among the prominent vehicles that are used for surveillance purposes and can tackle high-risk crises [28]. The part of deploying sensors is the core part that guides the UGV in indoor/outdoor space.

The main motive of proposing the UGV equipped with manipulator design is to navigate it in a space with a specific trajectory tracking and manipulating the objects to mentioned coordinates. Generally, two approaches are commonly adapted with control designs such as mentioned as follows: understanding of failure modes [55].

5.2. Failure Analysis of Acquired Data and Its Usage. The researchers opted for the Newton–Euler method most frequently for UGVs like UUVs and UAVs. Here, the robotic manipulator and the payload duly manipulated are driven using a free diagram as shown in Figure 7 [56]

Table 2: Limitations of previously proposed techniques for UUVs.

Techniques	Applied on	Limitations	References
Self-tuning fuzzy proportional integral derivative (PID) nonsingular fast terminal sliding mode control (STF-PID-NFTSM)	PUMA560 robotic manipulator	(1) The average convergence time is 3.5 seconds in the presence of uncertainties (2) The response experiences an undershoot in the presence of unmodeled uncertainties	[7]
Nonlinear disturbance observer-based sliding mode control law	UUV with 2 DOF manipulator	(1) The proposed UUV is a fully actuated system and thus consumes huge power and process time (2) It has a small working envelope and does not consider unmodeled dynamic factors (3) For estimating the sudden hydrodynamic coefficients, the strategy is complex	[39]
Bioinspired dolphin algorithm embedded with disturbance rejection scheme	UUV with fins like a dolphin	(1) The disturbance taken in the simulation is limited up to $0.01 \sin(t)$ (2) Undersea there are so many other factors, that is, hydrodynamic coefficients which were assumed through ANSYS and computational fluid dynamics	[11]
Adaptive iterative approach with boundary layer and hyperbolic tangent function	2nd-order nonlinear system	(1) Bounded conditions have been defined already (2) Control input somehow experiences the chattering-like noise	[6]
Underwater long-arm manipulator (ULAM) with an improved hydraulic driving system (SHDS) with fuzzy-based PID control	UUV with long-arm gripper/manipulator	(1) Either in low or in high pressure, there is an average tracking error of 5.12 percent (2) Each joint experiences an average overshoot of 1.5% and a steady-state error of 0.015 (3) Fuzzy-based PID slows down the maneuverability and increases hardware cost	[43]
The nonlinear model-based observer design using the linearization of the model to estimate the current state	UUV Visor3	(1) Chattering effect is available along with some transient and steady-state issues (2) Hardware implementation will be costly as compared to other previously proposed UUVs	[40]
DexROV, an EC Horizon 2020-funded project	UUV with TITAN4 from shilling robotics	(1) The output response, that is, angular and translational velocities, experiences a chattering effect too with reasonable tracking error because of a long length of gripper and time delays	[38]
Hybrid high-order terminal sliding mode (HHOTSM) control approach for (MIMO) uncertain nonlinear systems	Underwater robotic manipulator	(1) First convergence time but output has a response which has some overshoots too (2) Chattering is still there due to switching mode b/w terminal SMC and higher-order SMC	[16]
Sliding mode control-based dynamic positioning system	Applied on ship model in the deep sea	(1) parameters were assumed for fully loaded and ballasted conditions and did not consider unmodeled dynamic factors (2) Responses, that is, surge, sway, and yaw, experience 12%–20% overshoots in simulations and experimental work, respectively	[14]
Supervisory feedforward artificial neural network- (ANN-) based fuzzy control law to address friction and elasticity issues of a manipulator	Flexible joint-based manipulator design	(1) There is a reasonable error in load and the motor trajectories (2) The convergence rate on the time axis is not suitable for any sensitive pick and drop tasks (3) The technique is based on ANN and a fuzzy set of rules; it will only be implemented on field-programmable gate array (FPGA) or digital signal processing (DSP) kits that lead us to an expensive hardware design	[5]
PID guidance and control laws to perform basic control tasks such as autoheading, autospeed, and straight line	Unmanned surface vehicle	(1) Time delays due to Global Positioning System- (GPS-) based communication. (2) Due to these time delays, the responses have tracking errors	[13]

Table 3: Limitations of previously proposed techniques for Quadrotor Unmanned Aerial Vehicles.

Technique	Applied on	Limitations	References
Adaptive control-based regulation, pole-placement, and tracking (RST) control law	QUAV with 2 DOF gripper	(1) Sudden fluctuations are experienced due to variation in altitude (2) Sluggish in grasping the objects (need an improvement for fast grasp)	[53]
Adaptive time delay control (ATDC) scheme using a fractional-order nonsingular terminal sliding mode (FONTSM)	Cable-driven manipulators	(1) The tuning function for ATDC is a complex procedure (2) The root means square error for joint 01/02 is 0.3 and 0.32. 5 chattering appears at torque responses in the presences of payload	[50]
Continuous nonsingular fast terminal sliding mode (CNFTSM) control scheme using a modified supertwisting algorithm (STA)	QUAV with a cable-driven manipulator	(1) The response experiences chattering in experimental work (2) The software simulations also show some overshoots too (3) There is also a steady-state error available for manipulator	[52]
A composite controller scheme using 02 subcontrol blocks (one for arm/gripper and second for QUAV), gain tuning method	QUAV with a cable-driven manipulator	(1) Gimbal lock due to Newton–Euler method (2) Due to cosine and sine terms, the computation time increases; therefore, an expensive DSP kit is proposed for the algorithm (3) The chattering phenomenon appears while real flight takes off and path planning	[48]
An adaptive terminal sliding mode controller for the trajectory tracking of robotic manipulators using radial basis function neural networks (RBFNNs)	Robotic manipulators	(1) In the presence of any disturbance factor, that is, wind disturbance, the robotic manipulator will be diverted from trajectory for some time and refollow the trajectory again	[49]
Combination of gain scheduling and Lyapunov-based model reference adaptive control (MRAC)	UAV for manipulation	(1) There is undershoot and overshoot in yaw angular velocity (2) While performing pitch, the rotors experience oscillations for 10 seconds	[47]
Backstepping control design using an admittance subcontrol block for the manipulator design	Octa-copter UAV with 07 DOF manipulator	(1) Due to the change of the reference values, too fast several joints and their servos are not able to follow, and the response lags the reference for some time	[26]
Proportional integral derivative- (PID-) based wireless control of quadrotor at hovering state	Quadrotor with a payload of 250 grams	(1) The major shortcoming is the control using the Zigbee module which is sluggish and hence QUAV will never be able to perform aggressive maneuvers (2) At the payload of 250 grams, the QUAV cannot balance itself in a hovering state	[46]
Nested controller scheme for attitude stabilization, vision-based navigation, and guidance, with the aerial gripping	QUAV with aerial gripping task	(1) There is an error in both hovering and path tracking states (i.e., the deviation in between ±13 cm). At outdoor operations, this error increases up to ±20 cm. (2) During wind disturbance, the quadrotor deviates from the path for a few seconds and refollows the path again.	[27]
Direct approximate-adaptive control using cerebellar model arithmetic computer (CMAC) nonlinear approximators	Quadrotor with multiple payload variation	(1) There must not be extreme variation in payload mass, and this will lead to bursting or instability because of the previously trained weights	[45]
Discrete proportional integral derivative control design	QUAV with 4 g weight and 1 kg payload mass	(1) After visualizing the pitch and roll response, there are drastic oscillations in the output response due to the ground effect	[44]

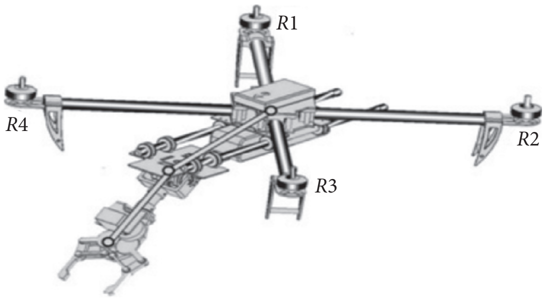


Figure 6: UAV with a manipulator.

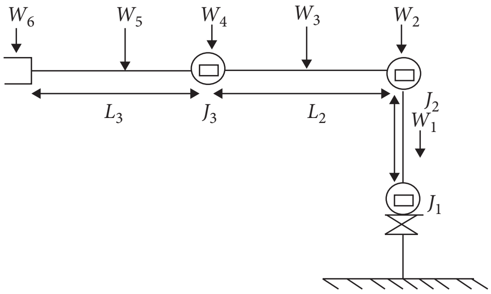


Figure 7: Manipulator-free diagram for torque.

$$T = FL, \tag{12}$$

where “T” is the torque, “F” is the force in Newton, and “L” is the perpendicular distance between the point of rotation and applied force.

$$F = W = mg. \tag{13}$$

The term “m” is the mass “g” which is the gravitation acceleration; hence, with equation (13), equation (12) can be transformed as

$$T = mgL. \tag{14}$$

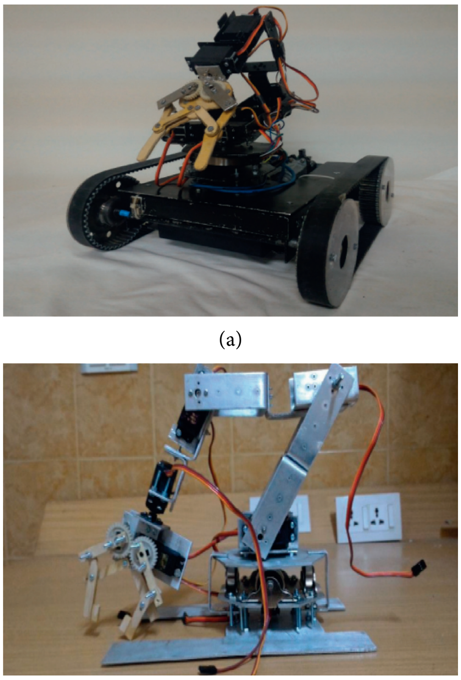
Researchers have proposed various dynamic models for UGV and its manipulators such as in [57]. Figure 8 presents an example of a UGV [31] and a manipulator design [56].

Table 4 presents the previously proposed techniques for UGVs, the hardware they are applied on, and their limitations. Moreover, Table 5 summarizes Tables 2–4 into one easily comprehensible table to further elucidate the shortcomings of all the techniques used for an unmanned vehicle.

6. Technical Analysis and Discussion

If someone is working on UUV, then one should work on the constraints, that is, chattering effect, producing cost-effective hardware design, minimizing the power consumption, and process time. This manuscript provides an opportunity to evaluate either robust or adaptive control laws with nonlinear observer designs.

Table 2 presents the previously proposed techniques for UUVs, the hardware they are applied on, and their limitations. It is a helpful guide for any future researchers to choose the best strategy according to their mission requirements.


(a)

(b)

Figure 8: UGV with a manipulator. (a) UGV model. (b) Manipulator design.

For quadrotor UAV, one must concern the issues such as the elimination of gimbal lock, chattering noise, and some serious undershoots/overshoots due to unmodeled dynamic factors. The paper suggests a serious need for reviewing the adaptive control law and their amalgamation with state observer design. The emerging bioinspired algorithms such as the pigeon algorithm are recommended while designing the observer design.

Table 3 presents the previously proposed techniques for quadrotor UAVs, the hardware they are applied on, and their limitations. Any future academic researching this field would find this table useful for deciding the best technique for their study.

For UGV, the processing time and hardware designs are emerging issues, and hence paper reviewed some of the fuzzy logic-oriented designs which produce fine response outcomes but are slower due to the fuzzy inference system. Therefore, it is suggested to use a single dimension-based fuzzy logic controller as they minimize the processing time. Once the processing time will be reduced, then a hardware designer may opt for a cheap microcontroller for programming.

Table 4 presents the previously proposed techniques for UGVs, the hardware they are applied on, and their limitations. It delineates the drawbacks of the mentioned strategies and would help in selecting the best method for the UGV.

After going through several research papers, Table 5 has been stated in this paper. This table shows the limitations of

Table 4: Limitations of previously proposed techniques for Unmanned Ground Vehicles.

Technique	Applied on	Limitations	References
Computer vision and control approach	UGV with variable payload up to 4 kg and 6 DOF manipulator	(1) NRF module is used to communicate with UGV and hence with the displacement in the antenna's direction may lead to loss of connection. (2) With 06 DOF, still there are certain easy moves that it cannot do and with the UGV prototype, it consumes relatively high power. (3) Without load, its RPMs are between 50 and 70. The approach may not be affordable for heavy payloads. The image processing approach is restricted to only color. It is studied that various approaches recommend tracking object with respect to color, shape, and histogram	[56]
Computer vision and Internet of things (IoT) control approach	UGV with sharp rudder to clear sewerage pipelines	(1) The response and work are fine but the time delay is experienced in the response due to the internet of things and the sensor's resolution factor	[58]
Novel hybrid additive-subtractive manufacturing (HASM)	06 DOF robotic manipulator	(1) Due to fast response, there is a time delay in system response (2) Sudden overshoots are also experienced	[29]
DTMF-based control law for robotic arm	Robotic arm for a coconut tree climber	(1) The simple dual tone multiple frequency-based control is proposed which may face serious distortions in the presence of any external noise. (2) The approach is involved in cutting and climbing on coconut trees to pick and drop the coconuts. Thus, the minor error may destabilize the system and the cutter part may harm any human standing beneath (hazardous in nature).	[59]
European ICARUS project towards the development of unmanned search and rescue (SAR) robots	UGV with large gripper	(1) Focused on only rescue missions but there are some huge tracking errors while performing manipulation of the objects	[28]
An alternative and comprehensive map-generating algorithm	UGV with 2 DOF gripper	(1) The trajectories are fine with no error but in the absence of unmodeled factors 2 The output response for gripper experiences the steady-state error of 0.55%	[31]
The design and control of robotic search and rescue system based on an immunocontrol framework	UGV with navigation systems	(1) The communication is performed using a Zigbee module, which is sluggish and thus brings time delay in responses	[30]
CoMoRAT (configurable mobile robot for all-terrain applications) for the installation of variable payloads	Modular UGV	(1) The modular approach is presented that leads to the installation of additional components on UGV to make it affordable for variable load conditions (2) Expensive hardware approach	[33]

Table 5: Summary of all previously proposed control laws for all unmanned vehicles.

References	Limitations/shortcomings									Vehicle type
	TD	OS/US	SC	CE/EN	GL	SSE	EHD	PC	T_proc	
[7]		✓	✓		✓					UUVs with manipulators
[39]								✓	✓	
[11]		✓		✓	✓					
[6]										
[43]						✓	✓	✓	✓	
[40]				✓			✓	✓	✓	
[38]		✓		✓						
[16]			✓				✓	✓	✓	
[14]				✓	✓					
[5]			✓				✓			
[13]	✓				✓	✓				

Table 5: Continued.

References	Limitations/shortcomings									Vehicle type
	TD	OS/US	SC	CE/EN	GL	SSE	EHD	PC	T_proc	
[53]		✓		✓						QUAVs with manipulator design
[50]						✓			✓	
[52]		✓								
[48]				✓	✓		✓	✓	✓	
[49]			✓							
[47]		✓								
[26]	✓	✓				✓			✓	
[46]	✓	✓		✓	✓					
[27]			✓		✓	✓				
[45]									✓	
[44]			✓	✓						
[56]	✓						✓	✓	✓	UGV with manipulator design
[58]			✓		✓	✓				
[29]		✓								
[59]	✓							✓		
[28]		✓		✓		✓			✓	
[31]					✓					
[30]	✓		✓						✓	
[33]							✓	✓	✓	

TD: time delay. SC: slow convergence rate. GL: gimbal lock. EHD: expensive hardware. T_proc: process time. OS/US: over/undershoots. CE/EN: chattering effect/external noise effect. SSE: steady-state error. PC: power consumption.

all previously proposed control laws over the unmanned vehicles embedded with several manipulator types.

7. Conclusion

This review paper presents a detailed review of the current state-of-the-art approaches and control laws proposed already for three types of unmanned vehicles, that is, UUVs, UAVs (more specifically quadrotors), and UGVs. The manuscript comes up with the limitations in Table 5. By reading Tables 2 to 5, one may see the most frequent problems in such unmanned vehicles, especially when embedded with manipulator design. The control laws so far proposed are fine until the degree of freedom for the manipulator is 02.

If the DOF value increases, the tracking performance also degrades and one may experience the chattering noise and deviation from tracking for some time and refollow the path. In addition to this, there is also degradation in transient and steady-state performances, that is, steady-state error and slow convergence. These vehicles are designed for fast maneuvers and aggressive operations with greater reliability but with these constraints, these unmanned vehicles compromise on their overall performance.

For future research ideas, one could revisit the robust and adaptive control laws with an amalgamation of bioinspired algorithms and smart observer designs to manage these problems. Also, the current team is planning to evaluate the bioinspired algorithms for individual unmanned vehicle types and may come up with another review.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This paper was supported by the Science Research Fund of Xi'an Aeronautics University (Grant no. 2019KY0208).

References

[1] N. U. Rehman, K. Kumar, and G. E. M. Abro, "Implementation of an autonomous path planning & obstacle avoidance UGV using SLAM," in *Proceedings of the 2018 International Conference on Engineering and Emerging Technologies, ICEET 2018*, pp. 1–5, Lahore, Pakistan, January 2018.

[2] J. Sánchez-García, "A survey on unmanned aerial and aquatic vehicle multi-hop networks: wireless communications, evaluation tools and applications," *Computer Communications*, vol. 119, pp. 43–65, 2018.

[3] G. E. M. Abro, B. Jabeen, and A. Manan, "Stabilization of non-holonomic 03 dof hovercraft using robust RST control design," in *Proceedings of the Sukkur IBA Journal of Emerging Technologies*, vol. 2, no. 1, pp. 45–50, Sukkur, Pakistan, August 2019.

[4] Y. Wang, S. Jiang, B. Chen, and H. Wu, "A new continuous fractional-order nonsingular terminal sliding mode control for cable-driven manipulators," *Advances in Engineering Software*, vol. 119, no. January, pp. 21–29, 2018.

[5] H. Chaoui, P. Sicard, and W. Gueaieb, "ANN-based adaptive control of robotic manipulators with friction and joint elasticity," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 3174–3187, 2009.

[6] J. Wei, "Adaptive iterative learning control of a class of nonlinear time-delay systems with unknown backlash-like

hysteresis input and control direction," *ISA Transactions*, vol. 70, pp. 79–92, 2017.

[7] M. Van, X. P. Do, and M. Mavrovouniotis, "Self-tuning fuzzy PID-nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators," *ISA Transactions*, vol. 96, pp. 60–68, 2020.

[8] C. Shen, B. Buckham, and Y. Shi, "Modified C/GMRES algorithm for fast nonlinear model predictive tracking control of AUVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1896–1904, 2016.

[9] M. J. Kim, K. Kondak, and C. Ott, "A stabilizing controller for regulation of UAV with manipulator," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1719–1726, 2018.

[10] J. Carius, M. Wermelinger, B. Rajasekaran, K. Holtmann, and M. Hutter, "Deployment of an autonomous mobile manipulator at MBZIRC," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1342–1357, 2018.

[11] Z. Wu, J. Yu, J. Yuan, and M. Tan, "Towards a gliding robotic dolphin: design, modeling, and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 260–270, 2019.

[12] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, 2016.

[13] M. Caccia, M. Bibuli, R. Bono, and G. Bruzzone, "Basic navigation, guidance and control of an unmanned surface vehicle," *Autonomous Robots*, vol. 25, no. 4, pp. 349–365, 2008.

[14] E. A. Tannuri, "Dynamic positioning systems: an experimental analysis of sliding mode control," *Control Engineering Practice*, vol. 18, no. 10, pp. 1121–1132, 2010.

[15] F. A. Azis, M. S. M. Aras, M. Z. A. Rashid, M. N. Othman, and S. S. Abdullah, "Problem identification for underwater remotely operated vehicle (rov): a case study," *Procedia Engineering*, vol. 41, pp. 554–560, 2012.

[16] X.-T. Tran and H.-J. Kang, "Adaptive hybrid High-Order terminal sliding mode control of MIMO uncertain nonlinear systems and its application to robot manipulators," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 2, pp. 255–266, 2015.

[17] J. Lee, D. H. Shim, S. Cho et al., "A mission management system for complex aerial logistics by multiple unmanned aerial vehicles in MBZIRC 2017," *Journal of Field Robotics*, vol. 36, no. 5, pp. 919–939, 2019.

[18] L. Li, Y. Zhang, M. Ripperger, J. Nicho, M. Veeraraghavan, and A. Fumagalli, "Autonomous object pick-and-sort procedure for industrial robotics application," *International Journal of Semantic Computing*, vol. 13, no. 2, pp. 161–183, 2019.

[19] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sensing*, vol. 9, no. 4, p. 312, 2017.

[20] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: a review," *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, 2017.

[21] M. Silvagni, "Multipurpose UAV for search and rescue operations in mountain avalanche events," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.

[22] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, "An amateur drone surveillance system based on the cognitive internet of things," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, 2018.

[23] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—a literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, 2018.

[24] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, vol. 32, no. 1, pp. 200–214, 2019.

[25] B. W. Davis, J. T. Diep, and S. Jose, *(12) United States Patent*, Moderna, vol. 2, no. 12, Cambridge, MA, USA, 2017.

[26] G. Heredia, "Control of a multirotor outdoor aerial manipulator," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 3417–3422, Chicago, IL, USA, September 2014.

[27] V. Ghadiok, J. Goldin, and W. Ren, "On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor," *Autonomous Robots*, vol. 33, no. 1-2, pp. 41–68, 2012.

[28] D. Doroftei, A. Matos, and G. de Cubber, "Designing search and rescue robots towards realistic user requirements," *Applied Mechanics and Materials*, vol. 658, pp. 612–617, 2014.

[29] L. Li, A. Haghighi, and Y. Yang, "A novel 6-axis hybrid additive-subtractive manufacturing process: design and case studies," *Journal of Manufacturing Processes*, vol. 33, no. March, pp. 150–160, 2018.

[30] A. W. Y. Ko and H. Y. K. Lau, "Intelligent robot-assisted humanitarian search and rescue system," *International Journal of Advanced Robotic Systems*, vol. 6, no. 2, pp. 12–128, 2009.

[31] J. Suthakorn, "On the design and development of a rough terrain robot for rescue missions," in *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008*, pp. 1830–1835, Chicago, IL, USA, September 2001.

[32] J. M. Ed and D. Hutchison, "Autonomous vehicles," *Auto Tech Review*, vol. 64, 2012.

[33] G. Bayar, A. B. Koku, and E. I. Konukseven, "CoMoRAT: a configurable all terrain mobile robot," in *Proceedings of the 11th Wseas International Conference on Automatic Control, Modelling and Simulation*, pp. 507–512, Istanbul, Turkey, October 2009.

[34] Z. A. Ali, X. Li, and M. A. Tanveer, "Controlling and stabilizing the position of remotely operated underwater vehicle equipped with a gripper," *Wireless Personal Communications*, vol. 116, 2019.

[35] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: challenges, solutions, and comparisons," *Computer Communications*, vol. 151, pp. 518–538, 2020.

[36] K. Asadi, A. Kalkunte Suresh, A. Ender et al., "An integrated UGV-UAV system for construction site data collection," *Automation in Construction*, vol. 112, p. 103068, 2020.

[37] P. I. Macreadie, D. L. McLean, P. G. Thomson et al., "Eyes in the sea: unlocking the mysteries of the ocean using industrial, remotely operated vehicles (ROVs)," *Science of the Total Environment*, vol. 634, pp. 1077–1091, 2018.

[38] P. A. Di Lillo, E. Simetti, D. De Palma et al., "Advanced ROV autonomy for efficient remote control in the DexROV project," *Marine Technology Society Journal*, vol. 50, no. 4, pp. 67–80, 2016.

[39] N. R. Sinatra, B. Clark, D. M. Vogt, K. Kit Parker, D. F. Gruber, and R. J. Wood, "Ultragentle manipulation of delicate structures using a soft robotic gripper," *Science Robotics*, vol. 4, p. 33, 2019.

[40] S. Rúa and R. E. Vásquez, "Development of a low-level control system for the ROV Visor3," *International Journal of Navigation and Observation*, vol. 2016, Article ID 8029124, 12 pages, 2016.

[41] A. J. Healey, "Model-based maneuvering controls for autonomous underwater vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, no. 4, pp. 614–622, 1992.

[42] G. Antonelli, “Underwater robots fourth edition,” 2018, http://www.springer.com/series/5208.

[43] C. Yang, Y. Wang, and F. Yao, “Driving performance of underwater long-arm hydraulic manipulator system for small autonomous underwater vehicle and its positioning accuracy,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, pp. 172988141774710–172988141774718, 2017.

[44] P. Pounds, R. Mahony, and P. Corke, “Modelling and control of a large quadrotor robot,” *Control Engineering Practice*, Elsevier, vol. 18, no. 7, pp. 691–699, 2010.

[45] C. Nicol, C. J. B. MacNab, and A. Ramirez-Serrano, “Robust adaptive control of a quadrotor helicopter,” *Mechatronics*, Elsevier, vol. 21, no. 6, pp. 927–938, 2011.

[46] D. Hanafi, M. Qetkeaw, R. Ghazali, M. N. M. Than, W. M. Utomo, and R. Omar, “Simple GUI wireless controller of quadcopter,” *International Journal of Communications, Network and System Sciences*, vol. 6, no. 1, pp. 52–59, 2013.

[47] M. Orsag, C. M. Korpela, S. Bogdan, and P. Y. Oh, “Hybrid adaptive control for aerial manipulation,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1–4, pp. 693–707, 2014.

[48] Y. Yu, X. Ding, and J. J. Zhu, “Dynamic modeling and control for aerial arm-operating of a multi-propeller multifunction aerial robot,” *Advanced Robotics*, vol. 31, no. 13, pp. 665–679, 2017.

[49] M.-D. Tran and H.-J. Kang, “A novel adaptive finite-time tracking control for robotic manipulators using nonsingular terminal sliding mode and RBF neural networks,” *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 7, pp. 863–870, 2016.

[50] Y. Wang, F. Yan, J. Chen, F. Ju, and B. Chen, “A new adaptive time-delay control scheme for cable-driven manipulators,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3469–3481, 2019.

[51] Y. Wang, L. Liu, D. Wang, F. Ju, and B. Chen, “Time-delay control using a novel nonlinear adaptive law for accurate trajectory tracking of cable-driven robots,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5234–5243, 2020.

[52] Y. Wang, F. Yan, J. Chen, and B. Chen, “Continuous nonsingular fast terminal sliding mode control of cable-driven manipulators with super-twisting algorithm,” *IEEE Access*, vol. 6, pp. 49626–49636, 2018.

[53] Z. A. Ali and X. Li, “Controlling of an under-actuated quadrotor UAV equipped with a manipulator,” *IEEE Access*, vol. 8, pp. 34664–34674, 2020.

[54] M. Orsag, C. Michael Korpela, S. Bogdan, and P. Y. Oh, “Hybrid adaptive control for aerial manipulation,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1–4, pp. 693–707, 2014.

[55] M. Ahsan, “Modification of a toy helicopter into a highly cost effective, semi-autonomous, reconnaissance unmanned aerial vehicle,” in *Proceedings of the 2012 International Conference on Robotics and Artificial Intelligence, ICRAI 2012*, pp. 49–54, IEEE, Rawalpindi, Pakistan, October 2012.

[56] U. F. Ahmed, S ur-Rehman, Z. Hameed, F. Ahmed, T. A. U’Chong, and M. M. A. Asif, “Design and development of control system for unmanned ground vehicle and its manipulator,” in *Proceedings of the 2020 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–8, Lahore, Pakistan, February 2020.

[57] M. Mustafa, *Modeling and Dynamic Control of Autonomous Ground Mobile Manipulators*, University of Calgary, Calgary, Canada, 2016.

[58] G. M. Abro, B. Jabeen, Ajodhia et al., “Designing smart sewerbot for the identification of sewer defects and blockages,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 2, pp. 615–619, 2019.

[59] R. Jabeen, T. Pathmakumar, T. Venugopal, G. Maruthiyodan, and A. Philip, “DTMF based robotic arm design and control for robotic coconut tree climber,” in *Proceedings of the 2015 International Conference on Computer, Communication and Control (IC4)*, Indore, India, September 2015.