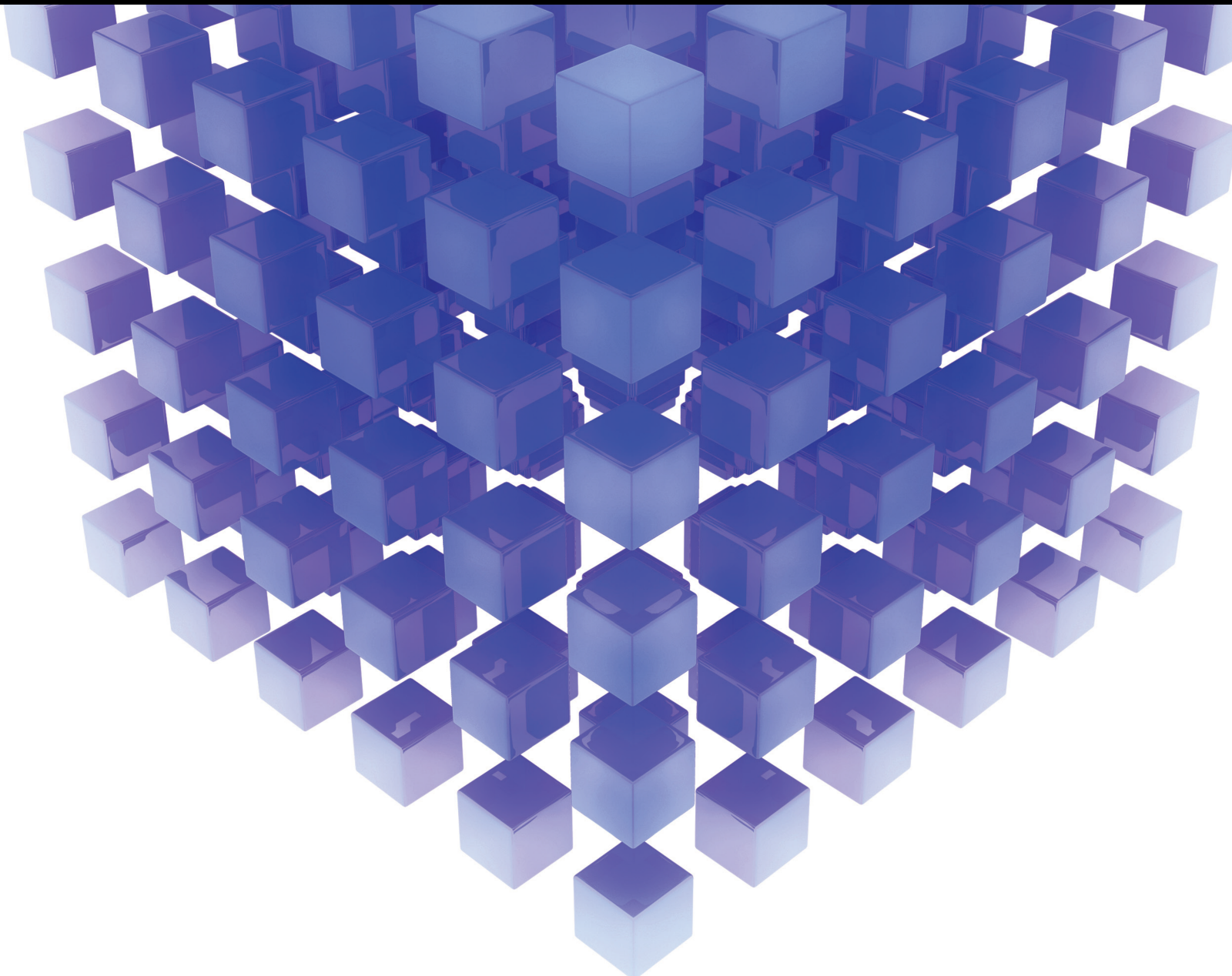


Mathematical Problems in Engineering

# Mathematical Problems of Applied System Innovations for IoT Applications

Lead Guest Editor: Teen-Hang Meen

Guest Editors: Wenbing Zhao and Cheng-Fu Yang





---

**Mathematical Problems of Applied System  
Innovations for IoT Applications**



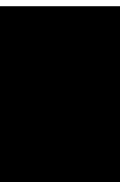
Mathematical Problems in Engineering

---

**Mathematical Problems of Applied  
System Innovations for IoT Applications**

Lead Guest Editor: Teen-Hang Meen

Guest Editors: Wenbing Zhao and Cheng-Fu Yang




---

Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in “Mathematical Problems in Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

Guangming Xie , China

## Academic Editors

Kumaravel A , India  
Waqas Abbasi, Pakistan  
Mohamed Abd El Aziz , Egypt  
Mahmoud Abdel-Aty , Egypt  
Mohammed S. Abdo, Yemen  
Mohammad Yaghoub Abdollahzadeh  
Jamalabadi , Republic of Korea  
Rahib Abiyev , Turkey  
Leonardo Acho , Spain  
Daniela Addessi , Italy  
Arooj Adeel , Pakistan  
Waleed Adel , Egypt  
Ramesh Agarwal , USA  
Francesco Aggogeri , Italy  
Ricardo Aguilar-Lopez , Mexico  
Afaq Ahmad , Pakistan  
Naveed Ahmed , Pakistan  
Elias Aifantis , USA  
Akif Akgul , Turkey  
Tareq Al-shami , Yemen  
Guido Ala, Italy  
Andrea Alaimo , Italy  
Reza Alam, USA  
Osamah Albahri , Malaysia  
Nicholas Alexander , United Kingdom  
Salvatore Alfonzetti, Italy  
Ghous Ali , Pakistan  
Nouman Ali , Pakistan  
Mohammad D. Aliyu , Canada  
Juan A. Almendral , Spain  
A.K. Alomari, Jordan  
José Domingo Álvarez , Spain  
Cláudio Alves , Portugal  
Juan P. Amezcua-Sanchez, Mexico  
Mukherjee Amitava, India  
Lionel Amodeo, France  
Sebastian Anita, Romania  
Costanza Arico , Italy  
Sabri Arik, Turkey  
Fausto Arpino , Italy  
Rashad Asharabi , Saudi Arabia  
Farhad Aslani , Australia  
Mohsen Asle Zaem , USA

Andrea Avanzini , Italy  
Richard I. Avery , USA  
Viktor Avrutin , Germany  
Mohammed A. Awadallah , Malaysia  
Francesco Aymerich , Italy  
Sajad Azizi , Belgium  
Michele Bacciocchi , Italy  
Seungik Baek , USA  
Khaled Bahlali, France  
M.V.A Raju Bahubalendruni, India  
Pedro Balaguer , Spain  
P. Balasubramaniam, India  
Stefan Balint , Romania  
Ines Tejado Balsera , Spain  
Alfonso Banos , Spain  
Jerzy Baranowski , Poland  
Tudor Barbu , Romania  
Andrzej Bartoszewicz , Poland  
Sergio Baselga , Spain  
S. Caglar Baslamisli , Turkey  
David Bassir , France  
Chiara Bedon , Italy  
Azeddine Beghdadi, France  
Andriette Bekker , South Africa  
Francisco Beltran-Carbajal , Mexico  
Abdellatif Ben Makhlof , Saudi Arabia  
Denis Benasciutti , Italy  
Ivano Benedetti , Italy  
Rosa M. Benito , Spain  
Elena Benvenuti , Italy  
Giovanni Berselli, Italy  
Michele Betti , Italy  
Pietro Bia , Italy  
Carlo Bianca , France  
Simone Bianco , Italy  
Vincenzo Bianco, Italy  
Vittorio Bianco, Italy  
David Bigaud , France  
Sardar Muhammad Bilal , Pakistan  
Antonio Bilotta , Italy  
Sylvio R. Bistafa, Brazil  
Chiara Boccaletti , Italy  
Rodolfo Bontempo , Italy  
Alberto Borboni , Italy  
Marco Bortolini, Italy

Paolo Boscariol, Italy  
Daniela Boso , Italy  
Guillermo Botella-Juan, Spain  
Abdesselem Boulkroune , Algeria  
Boulaïd Boulkroune, Belgium  
Fabio Bovenga , Italy  
Francesco Braghin , Italy  
Ricardo Branco, Portugal  
Julien Bruchon , France  
Matteo Bruggi , Italy  
Michele Brun , Italy  
Maria Elena Bruni, Italy  
Maria Angela Butturi , Italy  
Bartłomiej Błachowski , Poland  
Dhanamjayulu C , India  
Raquel Caballero-Águila , Spain  
Filippo Cacace , Italy  
Salvatore Caddemi , Italy  
Zuowei Cai , China  
Roberto Caldelli , Italy  
Francesco Cannizzaro , Italy  
Maosen Cao , China  
Ana Carpio, Spain  
Rodrigo Carvajal , Chile  
Caterina Casavola, Italy  
Sara Casciati, Italy  
Federica Caselli , Italy  
Carmen Castillo , Spain  
Inmaculada T. Castro , Spain  
Miguel Castro , Portugal  
Giuseppe Catalanotti , United Kingdom  
Alberto Cavallo , Italy  
Gabriele Cazzulani , Italy  
Fatih Vehbi Celebi, Turkey  
Miguel Cerrolaza , Venezuela  
Gregory Chagnon , France  
Ching-Ter Chang , Taiwan  
Kuei-Lun Chang , Taiwan  
Qing Chang , USA  
Xiaoheng Chang , China  
Prasenjit Chatterjee , Lithuania  
Kacem Chehdi, France  
Peter N. Cheimets, USA  
Chih-Chiang Chen , Taiwan  
He Chen , China

Kebing Chen , China  
Mengxin Chen , China  
Shyi-Ming Chen , Taiwan  
Xizhong Chen , Ireland  
Xue-Bo Chen , China  
Zhiwen Chen , China  
Qiang Cheng, USA  
Zeyang Cheng, China  
Luca Chiapponi , Italy  
Francisco Chicano , Spain  
Tirivanhu Chinyoka , South Africa  
Adrian Chmielewski , Poland  
Seongim Choi , USA  
Gautam Choubey , India  
Hung-Yuan Chung , Taiwan  
Yusheng Ci, China  
Simone Cinquemani , Italy  
Roberto G. Citarella , Italy  
Joaquim Ciurana , Spain  
John D. Clayton , USA  
Piero Colajanni , Italy  
Giuseppina Colicchio, Italy  
Vassilios Constantoudis , Greece  
Enrico Conte, Italy  
Alessandro Contento , USA  
Mario Cools , Belgium  
Gino Cortellessa, Italy  
Carlo Cosentino , Italy  
Paolo Crippa , Italy  
Erik Cuevas , Mexico  
Guozeng Cui , China  
Mehmet Cunkas , Turkey  
Giuseppe D'Aniello , Italy  
Peter Dabnichki, Australia  
Weizhong Dai , USA  
Zhifeng Dai , China  
Purushothaman Damodaran , USA  
Sergey Dashkovskiy, Germany  
Adiel T. De Almeida-Filho , Brazil  
Fabio De Angelis , Italy  
Samuele De Bartolo , Italy  
Stefano De Miranda , Italy  
Filippo De Monte , Italy



































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 Eftekhari Azam, USA  
Said El Kafhali , Morocco  
Antonio Elipse , 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 Faruqi, 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  
Stylianios Georgantzinou, 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-Díaz , 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  
Sarfranz 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

Francisco J. Martos , Spain  
Elio Masciari , Italy  
Paolo Massioni , France  
Alessandro Mauro , Italy  
Jonathan Mayo-Maldonado , Mexico  
Pier Luigi Mazzeo , Italy  
Laura Mazzola, Italy  
Driss Mehdi , France  
Zahid Mehmood , Pakistan  
Roderick Melnik , Canada  
Xiangyu Meng , USA  
Jose Merodio , Spain  
Alessio Merola , Italy  
Mahmoud Mesbah , Iran  
Luciano Mescia , Italy  
Laurent Mevel , France  
Constantine Michailides , Cyprus  
Mariusz Michta , Poland  
Prankul Middha, Norway  
Aki Mikkola , Finland  
Giovanni Minafò , Italy  
Edmondo Minisci , United Kingdom  
Hiroyuki Mino , Japan  
Dimitrios Mitsotakis , New Zealand  
Ardashir Mohammadzadeh , Iran  
Francisco J. Montáns , Spain  
Francesco Montefusco , Italy  
Gisele Mophou , France  
Rafael Morales , Spain  
Marco Morandini , Italy  
Javier Moreno-Valenzuela , Mexico  
Simone Morganti , Italy  
Caroline Mota , Brazil  
Aziz Moukrim , France  
Shen Mouquan , China  
Dimitris Mourtzis , Greece  
Emiliano Mucchi , Italy  
Taseer Muhammad, Saudi Arabia  
Ghulam Muhiuddin, Saudi Arabia  
Amitava Mukherjee , India  
Josefa Mula , Spain  
Jose J. Muñoz , Spain  
Giuseppe Muscolino, Italy  
Marco Mussetta , Italy

Hariharan Muthusamy, India  
Alessandro Naddeo , Italy  
Raj Nandkeolyar, India  
Keivan Navaie , United Kingdom  
Soumya Nayak, India  
Adrian Neagu , USA  
Erivelton Geraldo Nepomuceno , Brazil  
AMA Neves, Portugal  
Ha Quang Thinh Ngo , Vietnam  
Nhon Nguyen-Thanh, Singapore  
Papakostas Nikolaos , Ireland  
Jelena Nikolic , Serbia  
Tatsushi Nishi, Japan  
Shanzhou Niu , China  
Ben T. Nohara , Japan  
Mohammed Nouari , France  
Mustapha Nourelfath, Canada  
Kazem Nouri , Iran  
Ciro Núñez-Gutiérrez , Mexico  
Włodzimierz Ogryczak, Poland  
Roger Ohayon, France  
Krzysztof Okarma , Poland  
Mitsuhiro Okayasu, Japan  
Murat Olgun , Turkey  
Diego Oliva, Mexico  
Alberto Olivares , Spain  
Enrique Onieva , Spain  
Calogero Orlando , Italy  
Susana Ortega-Cisneros , Mexico  
Sergio Ortobelli, Italy  
Naohisa Otsuka , Japan  
Sid Ahmed Ould Ahmed Mahmoud , Saudi Arabia  
Taoreed Owolabi , Nigeria  
EUGENIA PETROPOULOU , Greece  
Arturo Pagano, Italy  
Madhumangal Pal, India  
Pasquale Palumbo , Italy  
Dragan Pamučar, Serbia  
Weifeng Pan , China  
Chandan Pandey, India  
Rui Pang, United Kingdom  
Jürgen Pannek , Germany  
Elena Panteley, France  
Achille Paolone, Italy

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 Rianza , 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 Vitek , 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  
Qihong 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

## **Automated Classification System for Tick-Bite Defect on Leather**

Y. S. Gan, Wei-Chuen Yau, Sze-Teng Liong , and Chih-Cheng Chen 




Research Article (12 pages), Article ID 5549879, Volume 2022 (2022)

## **A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO**

Cheng-Jian Lin , Shiou-Yun Jeng, and Hong-Wei Lioa




Research Article (10 pages), Article ID 1577614, Volume 2021 (2021)

## **Hilbert–Schmidt Independence Criterion Regularization Kernel Framework on Symmetric Positive Definite Manifolds**

Xi Liu , Zengrong Zhan , and Guo Niu 



Research Article (11 pages), Article ID 2402292, Volume 2021 (2021)

## **Simple and Ingenious Mobile Botnet Covert Network Based on Adjustable Unit (SIMBAIDU)**

Min-Hao Wu , Chia-Hao Lee , Fu-Hau Hsu, Kai-Wei Chang , Tsung-Huang Huang, Ting-Cheng Chang, and Li-Min Yi

Research Article (6 pages), Article ID 9920883, Volume 2021 (2021)

## **Influence Maximization Algorithm Based on Reverse Reachable Set**

Gengxin Sun  and Chih-Cheng Chen 





Research Article (12 pages), Article ID 5535843, Volume 2021 (2021)

## **Efficient Visualization Method and Implementation of Reservoir Model Based on WPF**

Shanshan Liu , Xiaoqi Wang , Yueli Feng , Xianlu Cai, Pengyin Yan , and Binwang Li 

Research Article (17 pages), Article ID 5581282, Volume 2021 (2021)

## **Utilizing Technology Acceptance Model for Influences of Smartphone Addiction on Behavioural Intention**

Chih-Wei Lin , Yu-Sheng Lin , Chia-Chi Liao , and Chih-Cheng Chen 




Research Article (7 pages), Article ID 5592187, Volume 2021 (2021)

## **Design of a Cryptographic System for Communication Security using Chaotic Signals**

Jai-Houng Leu , Jung-Kang Sun, Ho-Sheng Chen , Chong-Lin Huang, Dong-Kai Qiao, Tian-Syung Lan , Yu-Chih Chen, and Ay Su

Research Article (7 pages), Article ID 5585079, Volume 2021 (2021)

## **An Optimized and Efficient Routing Protocol Application for IoV**

Kiran Afzal, Rehan Tariq , Farhan Aadil, Zeshan Iqbal , Nouman Ali , and Muhammad Sajid





Research Article (32 pages), Article ID 9977252, Volume 2021 (2021)

## **Nonpreferential Attachment Leads to Scale-Free or Not**

Chuankui Yan , Nan Meng , and Yu Yang 






Research Article (9 pages), Article ID 5530048, Volume 2021 (2021)

### **Cross-Platform Drilling 3D Visualization System Based on WebGL**

Shanshan Liu , Yueli Feng , Xiaoqiu Wang , and Pengyin Yan 



Research Article (18 pages), Article ID 5516278, Volume 2021 (2021)

### **A Validated Study of a Modified Shallow Water Model for Strong Cyclonic Motions and Their Structures in a Rotating Tank**

Hung-Cheng Chen , Jai-Houng Leu , Yong Liu , He-Sheng Xie , and Qiang Chen 


Research Article (15 pages), Article ID 5529601, Volume 2021 (2021)

### **Relationship between Bitcoin Exchange Rate and Other Financial Indexes in Time Series**

Chien-Yun Chang, Chien-Chien Lo, Jui-Chang Cheng, Tzer-Long Chen , Liang-Yun Chi, and Chih-Cheng Chen 



Research Article (9 pages), Article ID 8842877, Volume 2021 (2021)

### **Integrated Image Sensor and Light Convolutional Neural Network for Image Classification**

Cheng-Jian Lin , Chun-Hui Lin, and Shyh-Hau Wang

Research Article (7 pages), Article ID 5573031, Volume 2021 (2021)

### **Multistep Prediction of Bus Arrival Time with the Recurrent Neural Network**

Zhi-Ying Xie, Yuan-Rong He, Chih-Cheng Chen , Qing-Quan Li, and Chia-Chun Wu 

Research Article (14 pages), Article ID 6636367, Volume 2021 (2021)

## Research Article

# Automated Classification System for Tick-Bite Defect on Leather

Y. S. Gan,<sup>1</sup> Wei-Chuen Yau,<sup>2</sup> Sze-Teng Liong ,<sup>3</sup> and Chih-Cheng Chen <sup>4,5</sup>

<sup>1</sup>School of Architecture, Feng Chia University, Taichung 407, Taiwan

<sup>2</sup>School of Computing and Data Science, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang, Selangor Darul Ehsan, Malaysia

<sup>3</sup>Department of Electronic Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>4</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>5</sup>Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

Correspondence should be addressed to Sze-Teng Liong; [stliong@fcu.edu.tw](mailto:stliong@fcu.edu.tw) and Chih-Cheng Chen; [ccc@gm.cyut.edu.tw](mailto:ccc@gm.cyut.edu.tw)

Received 4 February 2021; Revised 5 April 2021; Accepted 25 August 2021; Published 16 February 2022

Academic Editor: Teen-Hang Meen

Copyright © 2022 Y. S. Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Natural leather is a durable, breathable, stretchable, and pliable material that comes in various styles, colors, finishes, and prices. It is an ideal raw material to manufacture luxury products such as shoes, dresses, and luggage. The leather will be categorized into different grades that are determined by visual appearance, softness, and natural defects. This grading process requires a manual visual inspection from experienced experts to ensure proper quality assurance and quality control. To facilitate the inspection process, this paper introduces an efficient automated defect classification framework that is capable to evaluate if the sample patches contain defective segments. A six-step preprocessing procedure is introduced to enhance the quality of the leather image in terms of visibility and to preserve important features representation. Then, multiple classifiers are utilized to differentiate between defective and nondefective leather patches. The proposed framework is capable to generate a classification accuracy rate of 94% from a collection of samples of 1600 pieces of calf leather patches.

## 1. Introduction

Leather is the most popular raw material made from animal skins (i.e., cow, lamb, deer, elk, pig, etc.) Amongst them, most leather is made from cowhide as it is relatively easier to acquire in large quantities and its thick characteristics make it desirable for various types of products. Nevertheless, each leather piece comes with some imperfections that may result in the grain surface or structure of a hide. Common unsightly appearance existing on natural leather surfaces include scars, flay cuts, vainness, and irregular coloring. The surface appearance of a leather piece is an important indicator to determine its grading and hence affecting the selling price. To date, the conventional method to manually inspect the quality of the leather pieces is still adopted in the industrial manufacturing process.

In brief, the basic procedures to convert the raw animal hides into leather are as follows: (1) Soaking: to remove the dirt and curing salts by immersing the leather in water for several hours to several days; (2) liming: to remove the

epidermis, hair, and subcutaneous materials; (3) tanning: to create the protein cross-links in the collagen by penetrating the chemicals into the hides; (4) drying: to get rid of excess water; and (5) dyeing: to produce the desired custom color. Some of the natural defects are inconspicuous before the tanning process and they gradually appear to be apparent during the leather finishing process. On the other hand, those defective areas with minor damage will be repaired and roughed up with fillers to create a smooth and even surface. Finally, the finishing leather pieces will be graded before shipping to the customer.

The grading process is one of the most critical and exhausting procedures because it involves a manual assessment to visually inspect the defective parts of the leather. Particularly, the type of defect (i.e., cuts, wrinkles, and scabies), the defective size, and the severity level (i.e., critical, major, minor, or trivial) are the major aspects of quality control. The inspectors require to conduct a thorough manual evaluation on the same piece of leather multiple



times, viewing from multiple angles, distances, and lighting conditions to ensure the correctness and completeness. However, it should be noted that each judgment is subjective as it highly relies on the individual. Thus, human inspection is costly, time-consuming, inefficient, and inconsistent. It can be prone to human mistakes or errors due to this boring and repetitive task, or when the labor is feeling stressed and rushing to complete the task. Therefore, there is vital to design an automatic leather defect inspection system in order to improve the grading and inspection processes, in the meantime cutting off unnecessary costs.

The final goal of this paper is to classify the leather sample into either defective and nondefective classes. In brief, the four primary contributions of this research work are summarized as follows:

- (1) Proposal of six preprocessing steps and XBoost ANN classifier to categorize the defective leather patch
- (2) Verification the robustness of the proposed algorithm by validating them on several distinct machine learning classifiers
- (3) Comprehensive experimental evaluation and comparative analysis are carried out on over 1600 leather images
- (4) Demonstration of the promising classification results by reporting both the qualitative and quantitative findings

The subsequent sections of the paper are arranged as follows. A review of related literature is presented in Section 2. Then, Section 3 describes the proposed framework in detail which includes the intuition and explanation of the principal image processing techniques exploited. The experimental design such as the details of the database used, performance metrics, and the configuration of the parameters in the experiment are presented in Section 4. The classification performance is presented and discussed in Section 5 with further analysis. Finally, the conclusion is drawn in Section 6, accompanied by methodological recommendations for future research.

## 2. Literature Review

To date, the literature that carried out the automatic classification or segmentation tasks on the leather pieces is yet limited [1–3]. Besides, the experimental data are varied and hence it is difficult to make a fair test of performance to verify the effectiveness of the proposed methods. For instance, reference [4] collects the leather patch dataset using a robot arm such that each image is captured under consistent lighting source, same viewing angle, and distance. In total, the dataset contains 584 images. Then, a series of procedures are introduced to localize the tick-bite defects on leather patches. Succinctly, a segmentation algorithm, namely, Mask Region-based Convolutional Neural Network (Mask R-CNN) is adopted to learn the local features from 84 defective images. As a result, a classification accuracy of  $\sim 70\%$  is obtained when evaluated on 500 testing images.

Later, reference [5] employs the same data elicitation process to collect a different piece of calf leather. In brief, 27

images are collected and each piece is partitioned into 24 small patches. Thus, in total, 648 images are used in the experiment. Different from reference [4, 5] conducts both the classification and segmentation processes to predict two types of defects, namely, black lines and wrinkle. A transfer learning technique is adopted to fine-tune the parameters in AlexNet architecture for the classification task, whereas UNet architecture is employed for the segmentation task. As a result, the classification performance attained is 95% and the segmentation task obtained an Intersection over Union rate of close to 100%. However, it should be noted that the black line and wrinkle defects are relatively obvious and occupy a larger region. Thus, a reasonably higher classification result can be achieved.

Reference [6] designs a statistical approach based on the image intensity to tackle the classification task for both the datasets released by references [4, 5]. Briefly, this work adopts simple statistical features operations such as mean, variance, variance, skewness, kurtosis, lower, and upper quartile values. Then a feature selection method of the 2-sample Kolmogorov–Smirnov test is exploited to determine meaningful features. Then, three methods are applied to eliminate redundant features: percentile thresholding, Gaussian mixture model (GMM), and K-means clustering. Finally, seven types of classifiers are adopted to differentiate between the defective and nondefective leather patches. The best classification accuracy generated are 99% and 77% on two different datasets (i.e., [4, 5]), respectively. In short, this paper successfully outperforms reference [4] by 7% while obtaining a comparable performance with reference [5].

On the other hand, conventional methods such as feature extraction and reduction are adopted for leather defect detection task in which deep learning methods are not applied. For example, reference [7] utilizes the FisherFace feature reduction technique to project the local features of the leather images from high-dimensional image space to a lower-dimensional feature space in order to effectively distinguish the targeted classes. Concisely, the feature size of each image sample has been reduced from 4202 to 160. The extracted features include the attributes of color details, histograms of the color, co-occurrence matrix, Gabor filters, and the original pixels. To validate the effectiveness of the proposed method, the experiment was tested on 2000 samples that are composed of seven defective classes. Then, three types of classifiers are employed to predict the defective type. The best classification accuracies obtained are  $\sim 88\%$  for wet blue and  $\sim 92\%$  for rawhide images.

On the other hand, a leather type classification task was performed by reference [8] that evaluated 1000 leather sample images to differentiate among monitor lizard, crocodile, sheep, goat, and cow. Despite each leather type may contain samples with different colors, the proposed method is capable to distinguish the texture and characteristics of each leather type. Thus, a 99.9% classification accuracy was achieved by adopting the pretrained AlexNet architecture. However, no defect inspection or defect classification task is involved in the experiment.

Based on the aforementioned discussion, the research works conducted thus far are manageably finite. Inspired by

reference [4, 6], this paper aims to enhance the classification performance by introducing a simple yet effective solution. Particularly, the type of defect class in this classification task is strictly limited to only the tick bite. In brief, six preprocessing steps are applied to improve the images and to extract the local information of the leather patches. Next, the feature sets are fed into several two-class classifier models independently by exposing the relationships between the encoded features, in order to generate corresponding predicted labels. The classifiers involved in the experiment herein include decision tree, SVM,  $k$ -NN, Artificial Neural Network (ANN), XBoost ANN, and others which are employed to categorize the testing data.

### 3. Proposed Method

There are two major steps proposed in the algorithm, namely, the preprocessing and classification. The flowchart of the process is portrayed in Figure 1. Concisely, the images are first passed to a series of preprocessing steps, such as histogram matching, resizing, grayscale, Canny edge detection, Gaussian blurring, and histogram of the gradient. On the other hand, the classification task employs state-of-the-art supervised classifiers such as decision tree [9], discriminant analysis [10], SVM [11],  $k$ -Nearest Neighbor (NN) [12], Artificial Neural Network (ANN), XBoost Artificial Neural Network, and others.

The details of mathematical derivations of the aforementioned preprocessing methods and classifiers are elaborated in Section 3.1 and Section 3.2, respectively.

**3.1. Preprocessing Procedure.** The six preprocessing techniques employed in the experiment are shown in Figure 2 and each step is described as follows. In addition, sample images are shown in Figure 3 to illustrate the effect in each preprocessing step.

#### Step 1. Histogram Matching

The images have performed a histogram matching with the ground truth template image to standardize the new image so that to eliminate the difference of brightness or contrast due to the environmental situation. The idea is to map the probability density function  $P_r(r)$  of the original image into the desired output  $P_z(z)$ , where  $r$  and  $z$  are intensity values of color spaces such as HSV/HLS, YUV, and YCbCr. The mapping is built by finding the best matches  $P_z(z)$  for each input in  $P_r(r)$  such that satisfied the following equation:

$$|P_r(r) - P_z(z)| = \min_k |P_r(r) - P_z(k)|. \quad (1)$$

#### Step 2. Resizing

The image is then resized from  $400 \times 400$  to  $100 \times 100$ . This downsampling step is to minimize the computational complexity; meanwhile, the execution speed is increased. Besides, it reduces the background noise.

#### Step 3. Gray Scale

The image is converted from color into a grayscale image. It can minimize redundancy and dimensionality; thus the computational requirements are also reduced.

#### Step 4. Gaussian Filter

A Gaussian blur is applied to smooth the background area and the defective area. Gaussian blur transforms each pixel in the image to produce normally distributed pixel values by its local neighbor through a mathematical function defined as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-x^2+y^2/2\sigma^2}, \quad (2)$$

where  $x$  is the distance from the center to the horizontal axis,  $y$  is the distance from the center to the vertical axis, and  $\sigma$  is the standard deviation of the Gaussian distribution. The values from this distribution are used to build a convolution matrix that is applied to the original image. Furthermore, by using a suitable filter size, it will produce more vulnerability intensity of new pictures. If the areas in an image are the same, it will generate a similar intensity. Hence, it increases the discriminant effect of the defective and nondefective areas.

#### Step 5. Canny Edge Detection:

Up to this step, the defective and nondefective areas should be easier to differentiate. The image is then enhanced by focusing on the gradient difference in the intensity of images. Succinctly, the process of the Canny edge detection algorithm can be implemented using these five steps:

- (i) A Gaussian filter is adopted to remove the image noise and suppress the meaningless information.
- (ii) The intensity gradients of the image are obtained by applying the edge detector operators like Sobel, Prewitt, and Robert.
- (iii) A nonmaximum suppression is employed to eliminate the spurious response such as spikes or noises.
- (iv) The lower and upper threshold values are specified to identify potential edges.
- (v) The edges are tracked by hysteresis whereby the *weak* edges that are not connected to *strong* edges are minimized.

The experiment conducted in this paper considers the Sobel operator. It applies convolution on the image with a separable, integer, and small-valued filter in the horizontal/vertical directions. Particularly, the Sobel operator approximates the gradient of the image by applying convolution on the image with a separable kernel in either horizontal or vertical directions. In general,  $3 \times 3$  kernels are used on both the horizontal and vertical derivative approximation that are denoted as  $G_x$  and  $G_y$ , respectively:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (3)$$

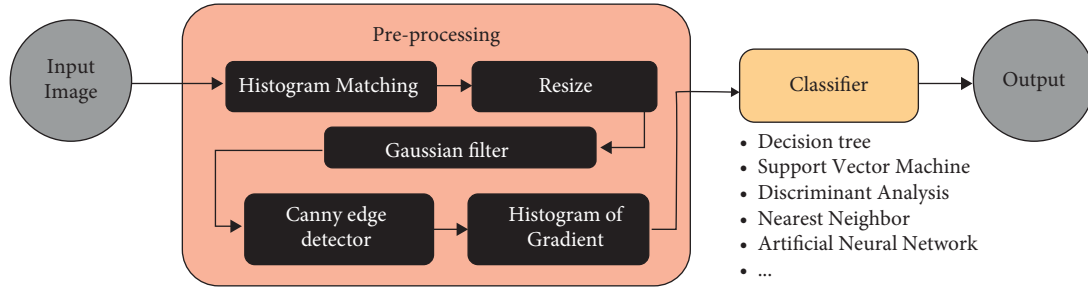


FIGURE 1: The proposed tick-bite defect classification system.

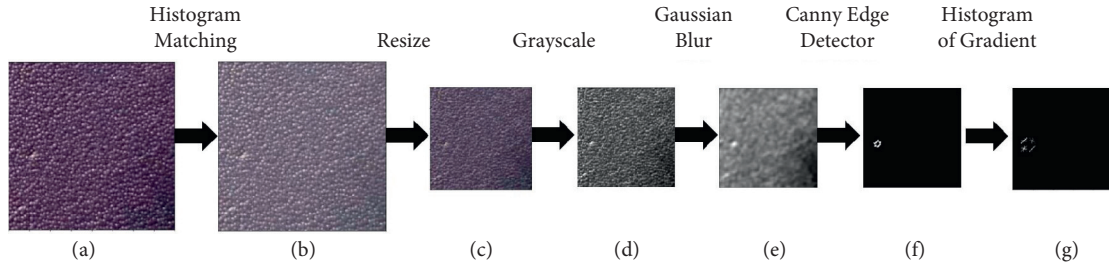


FIGURE 2: The example of after applying the steps of preprocessing methods: (a) original image; (b) histogram matching; (c) resizing; (d) grayscale; (e) Gaussian blur; (f) Canny edge detection; and (g) histogram of gradient.

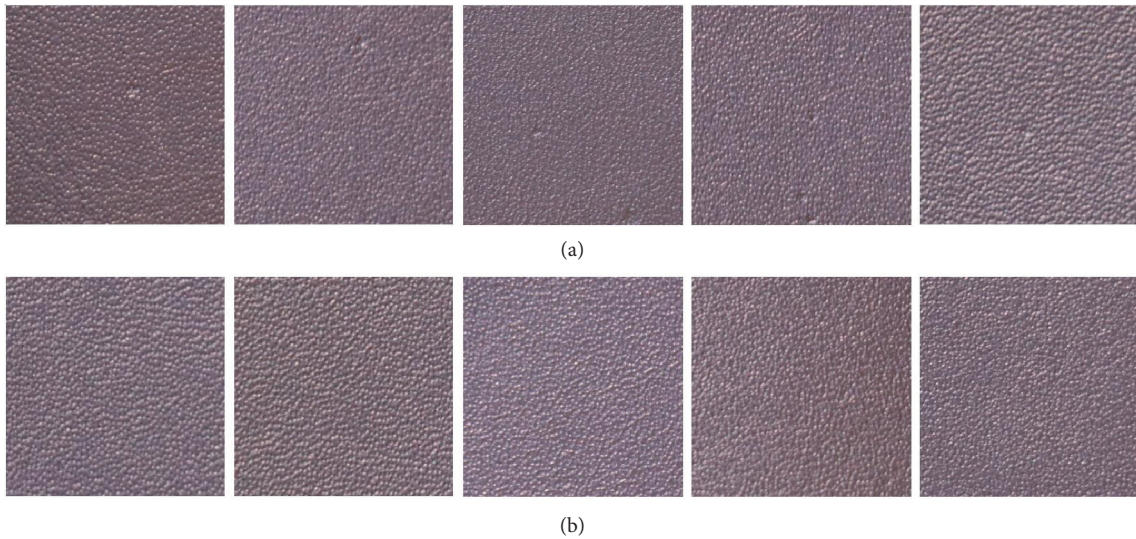


FIGURE 3: Sample leather images that contain (a) defect and (b) no defect.

At each regularly spaced sample points, the gradient approximations can be derived and represented to the gradient's magnitude ( $\rho$ ) and gradient's direction ( $\theta$ ):

$$\rho = \sqrt{G_x^2 + G_y^2}. \quad (4)$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right). \quad (5)$$

Step 6. Histogram of Gradient

The Canny edge detector in Step 5 returns a binary image. It is then split into  $8 \times 8$  pixels per cell to calculate the histogram of gradient of the binary images. This reveals the frequency histogram gradient of the orientation of the edges at each local patch, especially for the defective areas. On a uniform grid of cells, HOG summarizes the intensity gradients based on their respective directions to derive the local appearance features that describe the focus information of the corresponding image. Then, the histogram of gradient directions within the connected cells is concatenated such that an enriched resultant feature vector is constructed. Owing to the advantages of the HOG descriptor such as fast



computation speed and effectiveness in encoding the local shape information, it is one of the feature extractors that had been widely adopted in the research community. Specifically, the steps to realize the HOG algorithm are as follows.

**3.1.1. Gradient Image Generation.** The image filtering method can be applied by utilizing the kernels that contain both the horizontal and vertical kernels, namely,  $[-1\ 0\ 1]$  and  $[-1\ 0\ 1]^T$  sliding window. Concisely, the kernel convolutes with the original image from left-right, to top-bottom. Then, (4) and (5) is applied to acquire the pixel-wise magnitude and orientation maps. As a result, the regions with constant and similar color intensity are eliminated from the image; in the meantime, the important outlines or edges are kept without change.

**3.1.2. HOG Computation in  $m \times n$  Cells.** Each image is partitioned into  $m \times n$  cells such that a more compact feature representation can be constructed. Thus, a 9-bin histogram that falls in the angles range of  $0, 20, 40, 60, \dots, 180$  is computed.

**3.1.3. Cell's Blocks Normalization.** The magnitude computed may be vulnerable and sensitive to changes in illumination. Therefore, a simple normalization operation is implemented locally for each block. Finally, the resultant feature vector is enriched by concatenating all the histograms.

**3.2. Classifier.** This subsection briefly elaborates on the characteristics of classifiers. Concretely, both the functions of the conventional classifiers (i.e., decision tree, SVM, NN, discriminant analysis, etc.) and the ANN are stated in Subsections 3.2.1 and 3.2.2.

**3.2.1. Conventional Classifier.** After attaining the feature vectors from the feature descriptors discussed in the previous section, they are then processed by the classifiers to distinguish the defective status. Some widely known classifiers that available in Sklearn Package are utilized, namely, decision tree, SVM, NN, and ensemble classifier. Note that the classifiers adopted herein are supervised machine learning approach:

- (1) *k*-Nearest Neighbor (k-NN) [12]: this is one of the simplest classifiers as it is easy to implement and no training time is required. The predicted outcome is identified based on the simple majority vote system and determination of the number of nearest neighbors.
- (2) Support Vector Classification (SVC) [13]: it can be used for either the classification or regression analysis. It involves at least a quadratically fitting scale with the number of samples.
- (3) Linear Support Vector Machine (SVM) [11]: a linear kernel is utilized to project the input data to a higher

dimensional space. This data transformation process finds an optimal boundary between the possible outcomes.

- (4) Decision tree [9]: it builds a classification model by adopting simple decision rules. A tree-structured model is created by outlining all the possible consequences. In brief, the decision tree consists of the root, nodes, branches, and leaves. The predicted response is generated by following the decision from the root node down to the leaf node.
- (5) Random forest [14]: it is a collection of simple tree estimator that process various subsamples of the dataset and obtain the average values to boost the classification accuracy and prevent over-fitting.
- (6) Multilayer perceptron (MLP) [15]: it composes three basic layers, namely, the input, hidden, and output layers. Each layer may contain a different number of the neuron. Specifically, the neurons in the input layer depend on the dimension of the input data. The number of neurons in the hidden layer is subjective as it relies on the function's complexity and the attribute properties of the targeted classes. Finally, the number of neurons in the output layer is the number of output classes.
- (7) Adaptive Boosting (AdaBoost) [16]: it is a meta-estimator that learns a single "strong classifier" from several "weak" classifiers. It produces a set of optimal features that consider the weights factor before the combination of the classifiers.
- (8) Discriminant analysis [10]: it has a quadratic decision boundary to develop discriminant functions to examine the difference between the predictor variables.
- (9) Extreme gradient boosting [17]: it trains many weak prediction models sequentially and ensembles them. The typical models are decision trees and the learning procedure generalizes the new model to provide a more accurate and optimized predictor.

**3.2.2. Artificial Neural Network (ANN) Learning Features.** ANN is a significant part of artificial intelligence as it mimics the computational principles of neural networks of an animal. Owing to its remarkable generalization capability and promising correlation-based feature selector, it has been extensively used in the research field such as handwritten text recognition [18], weather forecasting [19], financial economics [20], and agricultural land assessment [21].

Basically, ANN incorporates three layers, namely, the input, hidden, and output layers. Concisely, the neurons in both the hidden and output layers adopt the sigmoid activation functions in performing the backpropagation operation. The output of the ANN can be acquired by the following equation:

$$y_{\text{output}} = \frac{1}{1 + \exp^{-(b_2 + W_2 (\max(0, b_1 + W_1 * X_n)))}}, \quad (6)$$

where  $W_1, W_2, b_1, b_2$  are weights and biases parameters and  $X_n$  refers to the data input. The Adam optimization algorithm [22] is adopted to adaptively update the learning rates during the model training. In addition, we propose the “XBoosting ANN” by implementing an extreme gradient boosting method onto the ANN outputs.

## 4. Experiment

**4.1. Database.** The experimental data adopt the database released by reference [4]. Concretely, the database consists of 1605 leather patches that have the size of  $90 \times 60 \text{ mm}^2$  (width  $\times$  length). Amongst them, 503 images contain one or more tick-bite defects, whereas the remaining 1102 images are nondefective images. In brief, all the images are collected using a 6-axis articulated robot arm DRV70L from Delta, which load-bearing capacity is 5 kg. The robot arm is equipped with a Canon 77D camera fitted with a 135 mm focal length lens. Each captured data has an image resolution of  $2400 \times 1600 \text{ pixels}^2$ . A lightning source is utilized to guarantee a consistent brightness distribution on the leather pieces. A screenshot of the experimental setup is illustrated in Figures 3 and 4 which shows the samples for the defective and nondefective images.

An illustrative example that describes the bounding box with the estimated size is shown in Figure 5. Besides, the largest and the smallest defect samples are depicted in Figure 6.

**4.2. Experiment Configuration.** In the classification stage, 5-fold cross-validation is applied to test the unseen data. The dataset is by the first split to a ratio of 7 : 3 for the train: test subsets. Then, the training subset is further partitioned into 7 : 3 into train and validation subsets. Therefore, the final division of the dataset is about a ratio of 5 : 3:2 for the train: test:validation subsets, which consists of 785 : 483 : 337 images, respectively. Concretely, the train features will be fed into the classification model; in the meantime, the validation images are utilized in order to determine the optimal experimental configuration and parameter settings (i.e., filter size of the Gaussian filter and threshold value for the Canny edge detector). Finally, the refined model is used to validate the test images, and the performance metrics are described in the following subsection.

**4.3. Performance Metrics.** This is a binary classification problem where the output should produce the label of “defect” or “no defect.” Thus, the following four metrics can be derived from the  $2 \times 2$  confusion matrix:

$$\text{Accuracy} := \frac{TP + TN}{TP + FP + TN + FN}, \quad (7)$$

where TP is the predicted pixel that has correctly identified the defective pixel; TN is the nondefective pixel that has been correctly predicted; FP is the pixel that is incorrectly predicted as defective pixel; and FN is the undetected defective pixels.



FIGURE 4: Illustration of the experimental setup.

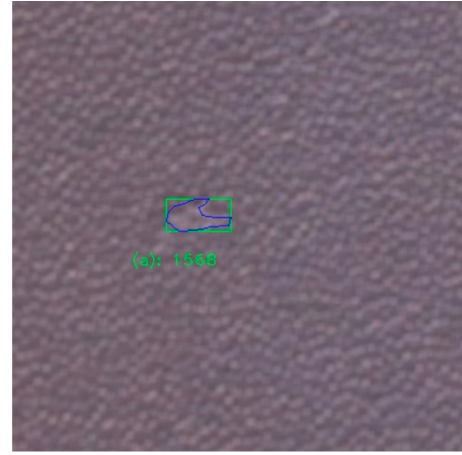


FIGURE 5: The surface area of the defect with a bounding box.

On the other hand, F1-score performance metric is computed:

$$F1 - \text{score} := 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (8)$$

for

$$\text{recall} := \frac{TP}{TP + FN}, \quad (9)$$

and

$$\text{precision} := \frac{TP}{TP + FP}. \quad (10)$$

There are two types of F1-score, namely, macro-averaged and weighted-average. The former is simply the mean of per class F1-score, which is similar to the macro-averaged precision and macro-averaged recall, which are calculated by the mean of precision and recall, respectively. For the weighted-average F1-score, the weighted-precision and weighted-recall are calculated by considering the weights to each class:



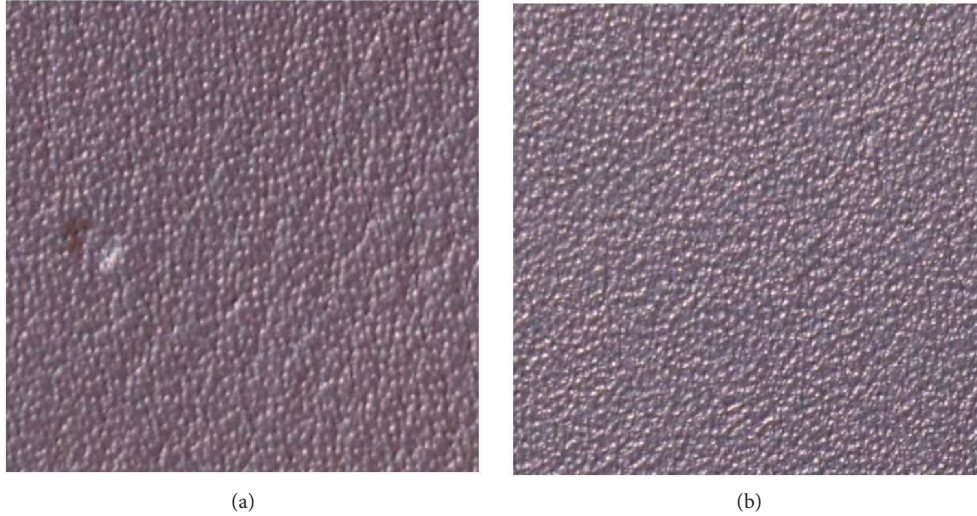


FIGURE 6: The defective samples that has the (a) largest and (b) smallest tick-bite defects.

$$\text{weighted - average F1 - score} = \frac{(w_1 S_1 + w_2 S_2 + \dots + w_n S_n)}{(w_1 + w_2 + \dots + w_n)}, \quad (11)$$

where  $w_i$  is the number of samples for class  $i$  and  $S_i$  belongs to F1-score, precision, and recall for each class  $i$ .

## 5. Result and Discussion

It should be noted that for the Canny edge detector, different size of Gaussian filters will detect differently sized features in the input image and producing distinct-sized feature maps. To seek an optimal configuration of the edge detector, its kernel size is first fixed to the common sizes, namely, (5,5), (7,7), and (9,9), respectively. Then, the range of the threshold values is set to [80, 230]. The validation accuracy of the classification accuracy against the threshold parameter in the Canny edge detector is portrayed in Figure 7. There is a similar trend for the kernel sizes of 7 and 9, whereby the best results are obtained when the threshold is minimal (i.e., 84 to 104). In contrast, when kernel size = 5, a low threshold did not outperform, compared to that of the kernel of 7 and 9. In addition, it can also be observed that when kernel size = 5, the highest accuracy (when threshold = 160) obtained is relatively lower.

From the preliminary result performed in Figure 7, we opt to select the optimal threshold of 92 for all the kernel sizes throughout the remaining of the experiments. The results of the accuracy when adopting ANN and XBoost ANN are shown tabulated in Table 1, with the detailed TP, FP, FN, TN, and F1-score. It can be seen that when the preprocessing steps do not involve HoG (the first three rows), the accuracy in the ANN classifier is 69%, while higher accuracy is attained when utilizing XBoost ANN (up to 82%). On the contrary, when HoG is added as one of the preprocessing steps, all the accuracies in both the ANN and XBoost ANN improved. Specifically, a promising classification result of 94% is exhibited when kernel size = 7.

To further analyze the impact of the HoG in the preprocessing step, a receiver operating characteristic (ROC) can illustrate the effectiveness of the statistical model of the classifier. Particularly, when HoG is not applied as one of the preprocessing steps, the ROC curve is shown in Figure 8. It is observed that the accuracy of the micro-average ROC is 69%, whereas the macro-average ROC is 50%. On the other hand, when HoG is included in the proposed method, the accuracies of the ROC improved up to 95%, as demonstrated in Figure 9.

In addition, we opt for ANN and Xboost ANN as the classifiers in our experiment. The reason being is because other classifiers are not outperforming based on the features extracted, especially in this binary classification task. The classification results are summarized in Figure 10. It can be seen that both the ANN and Xboost ANN achieve an accuracy of more than 90%. For SVM with linear kernel and random forest classifiers, their results are promising as well ( $\sim 90\%$ ). Other classifiers like k-NN, SVC, decision tree, AdaBoost, gradient boosting, and discriminant analysis seem not suitable to be adopted in this experiment.

The proposed framework is compared to three other works that performed the binary classification on the same leather dataset. Concretely, the three other methods utilized the ANN [4], AlexNet [4], and statistical analysis [6] as the key feature descriptors. The results comparison is summarized in Table 2 whereby the metrics presented are accuracy and F1-score. It is interesting to highlight that the deep learning network such as AlexNet does not perform well in this classification task. This may in part due to the overfitting phenomena. With a relatively small dataset and highly imbalanced data, where the number of the nondefective images is doubled of the defective ones, the network is not able to generalize well and thus leads to a poor classification result. Notably, the results generated in this paper outperform the state-of-the-art, in which the accuracy and F1-score reported are both 94%. However, it should be noted that the training images considered in the experiment herein are almost half of

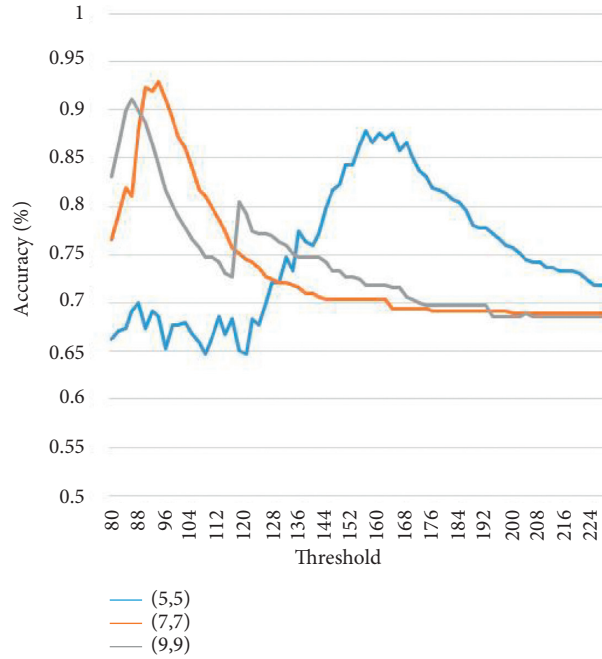


FIGURE 7: The validation accuracy of the classification accuracy against the threshold parameter in the Canny edge detector.

TABLE 1: Comparison of results when employing ANN (A) and XBoost ANN (X) as the classifiers.

Methods	TP		FP		FN		TN		Accuracy		Macro-avg		Weighted-avg	
	A	X	A	X	A	X	A	X	A	X	F1-score		F1-score	
											A	X	A	X
<i>w/o</i> HoG (5,5)	331	266	0	65	151	114	0	37	0.69	0.63	0.41	0.52	0.56	0.61
<i>w/o</i> HoG (7,7)	331	274	0	57	151	48	0	103	0.69	0.78	0.41	0.75	0.56	0.78
<i>w/o</i> HoG (9,9)	331	281	0	50	151	35	0	116	0.69	0.82	0.41	0.8	0.56	0.83
With HoG (5,5)	314	313	17	18	50	49	101	102	0.86	0.86	0.83	0.83	0.86	0.86
With HoG (7,7)	325	325	6	6	21	25	130	126	<b>0.94</b>	<b>0.94</b>	0.93	0.92	<b>0.94</b>	0.93
With HoG (9,9)	323	323	8	8	41	41	110	110	0.90	0.90	0.87	0.87	0.89	0.89

The bold numbers represent the highest values within the experimental results presented.

the dataset. Nonetheless, with the utilization of neural network architecture and gradient features, it achieves unprecedented improvements in the classification results.

In a nutshell, this paper proposes a new feature enhancement pipeline in classifying the defective leather image. Specifically, a large portion of the contribution is attributed to the preprocessing stage, in which the processes include histogram matching, resizing, grayscale normalization, Gaussian blurring, and Canny edge detection. Thereafter, the defective region becomes clearer and

noticeable. The HOG descriptor is utilized to convert the image into a 1D feature vector. Finally, multiple classifiers are employed to evaluate the robustness of the proposed mechanism. As a whole, the proposed method requires relatively lower computational resources whilst achieving promising classification accuracy of up to 94%. A brief comparison of the state-of-the-arts is provided in Table 3 to show the primary difference in tackling this leather defective classification problem. Note that the total number of leather samples for the experiments is slightly different.

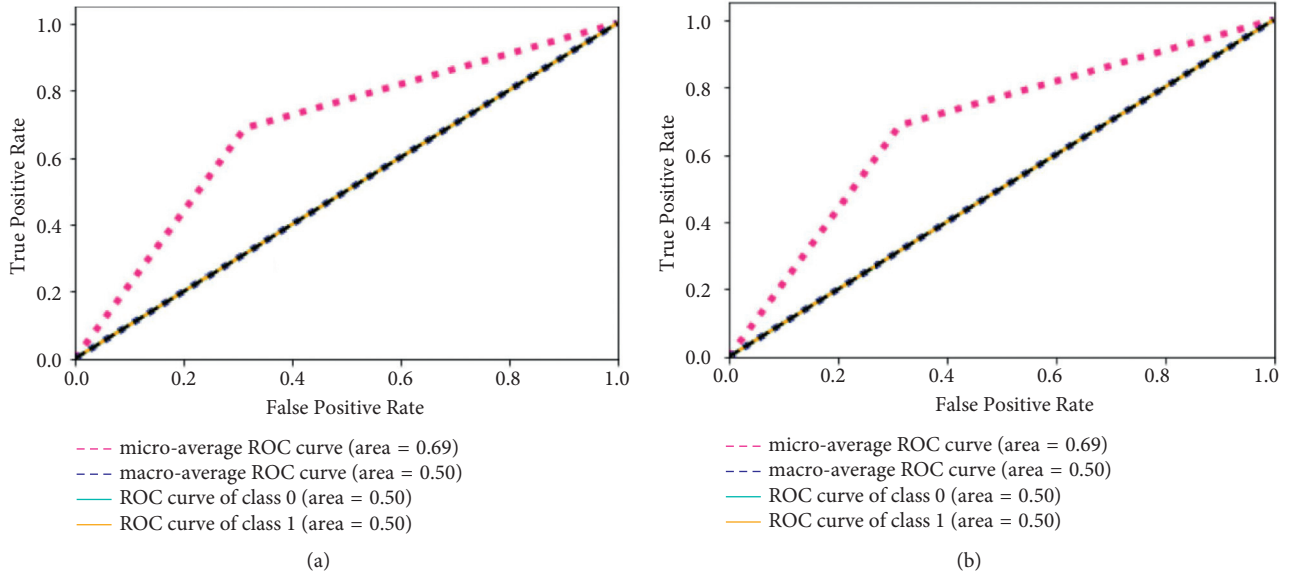


FIGURE 8: ROC curve when histogram of gradient is not included as one of the preprocessing steps, when evaluated on (a) validation set and (b) testing set.

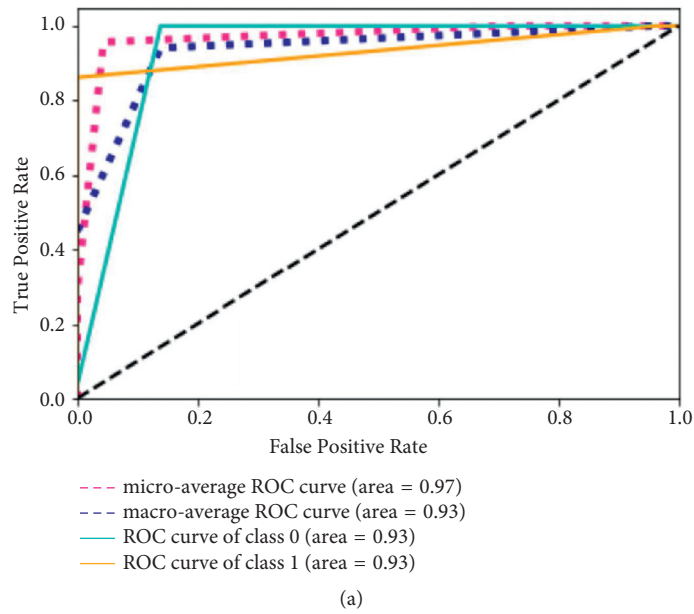


FIGURE 9: Continued.

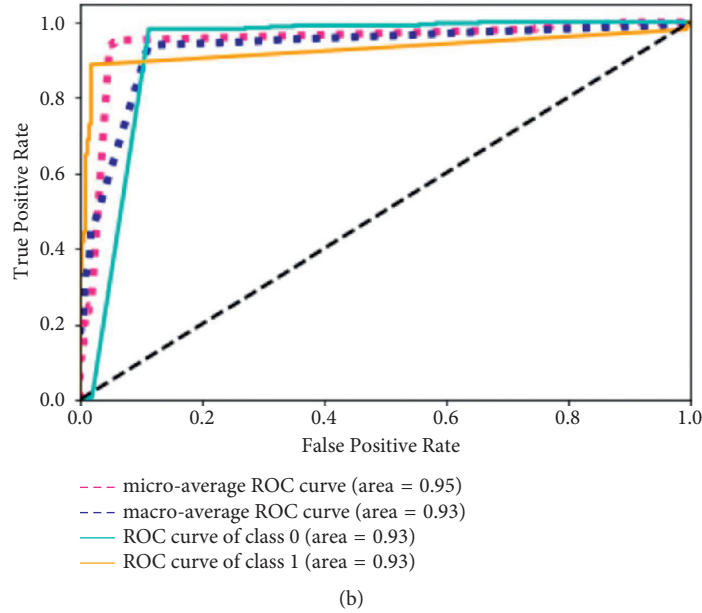


FIGURE 9: ROC curve when applying histogram of gradient as one of the preprocessing steps, when evaluated on (a) validation set and (b) testing set.

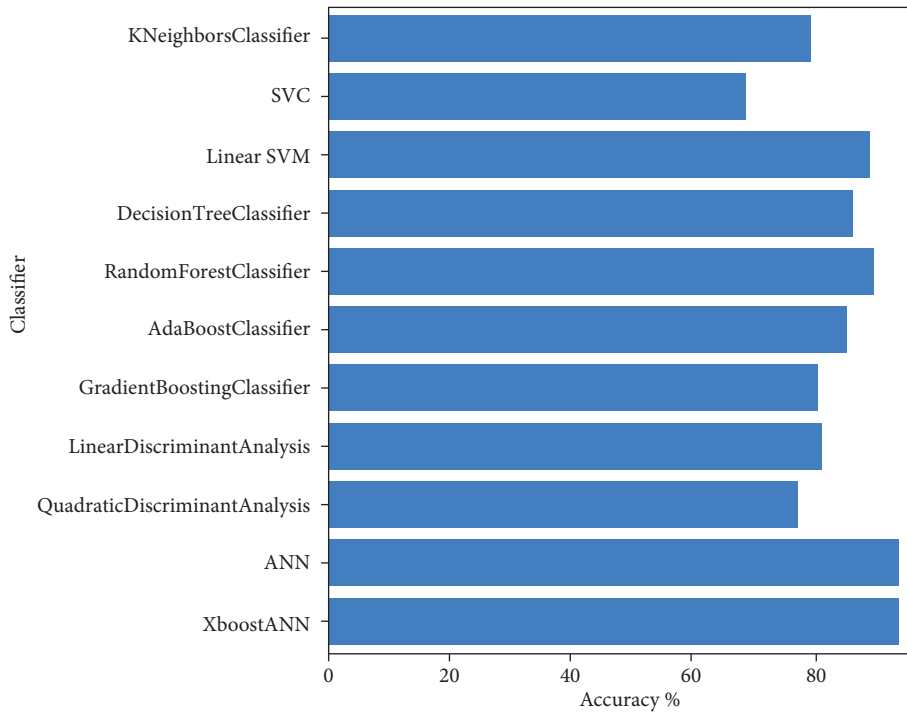


FIGURE 10: Classification performance of the proposed method when employing different classifiers.

TABLE 2: Classification performance comparison to the state-of-the-arts on the same dataset.

Method	Defective: nondefective	Train: validation: testing split	Method	Accuracy (%)	F1-score (%)
[4]	370 : 1527	60 : 5:35	ANN	80	89
[4]	233 : 699	90 : 0:10	AlexNet	74	84
[6]	503 : 1102	70 : 0: 30	Statistical analysis + SVM	74	97
<b>Ours</b>	503 : 1102	49 : 21: 30	<b>XBoost ANN + HOG</b>	<b>94</b>	<b>94</b>

The bold numbers represent the highest values within the experimental results presented.

TABLE 3: Classification performance comparison to the state-of-the-arts on the same dataset.

Method	Preprocessing	Feature Extractor	Classifier	Accuracy (%)
[6]	Grayscale normalization, Gaussian filter	Statistical features (mean, variance, skewness, kurtosis, lower, and upper quartile value)	SVM	74
[23]	Grayscale normalization, resizing, Canny edge detection, block partition		ANN	80
[23]	Grayscale normalization, resizing, Canny edge detection, block partition		AlexNet	76
[4]	—		Mask R-CNN	80
Ours	Histogram matching, resizing, grayscale normalization, Gaussian blurring, and Canny edge detection	Histogram of gradient	k-NN, SVC, SVM, MLP, AdaBoost, decision tree, random forest, discriminant analysis, extreme gradient boosting	<b>94</b>

The bold numbers represent the highest values within the experimental results presented.

## 6. Conclusion

This study introduced a binary classification system to distinguish if a leather image contain tick-bite defect. Thorough experiments and analyses have been conducted to verify the robustness of the proposed algorithm. Overall, promising results are exhibited when exploiting a series of preprocessing methods and two neural network classifiers. As a result, the best classification accuracy obtained is 94% when employing ANN and XBoost ANN as the classifiers. As this experiment strictly limited to the defect type of tick-bite defect, potential direction for future research in this area includes the development of classification or segmentation system to determine the defect types such as open cuts, closed cuts, wrinkles, holes, and scabies. Apart from investigating the defect type, the experiment can be extended to evaluate on other leather type such as the hides of other animals lamb, crocodile, and snakes. Ultimately, a fully automated hardware setup that consists of the functions of capturing leather image patches, identifying the defective areas, and of laser cutting of the leather can be developed in the future.

## Data Availability

The nature of the data in an Excel File, and the data and the code can be accessed on the following website: <https://github.com/christy1206/XBoost-leather>. There are no restrictions on data access. The [Excel File] data used to support the findings of this study are included in the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was funded by Ministry of Science and Technology (MOST) (Grant Number: MOST 109-2221-E-035-065-MY2, MOST 110-2221-E-035-052).

## References

- [1] M. Aslam, T. M. Khan, S. S. Naqvi, G. Holmes, and R. Naffa, "On the application of automated machine vision for leather defect inspection and grading: a survey," *IEEE Access*, vol. 7, Article ID 176065, 2019.
- [2] M. Jawahar, N. K. Babu, K. L. J. A. Vani, L. J. Anbarasi, and S. Geetha, "Vision Based Inspection System for Leather Surface Defect Detection Using Fast Convergence Particle Swarm Optimization Ensemble Classifier Approach," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4203–4235, 2020.
- [3] M. K. Kasi, J. B. Rao, and V. K. Sahu, "Identification of leather defects using an autoadaptive edge detection image processing algorithm," in *Proceedings of the 2014 International Conference on High Performance Computing and Applications (ICHPCA)*, pp. 1–4, IEEE, Bhubaneswar, India, December 2014.
- [4] S. T. Liong, Y. S. Gan, Y. C. Huang, C. A. Yuan, and H. C. Chang, "Automatic Defect Segmentation on Leather with Deep Learning," March 2019, <https://arxiv.org/abs/1903.12139>.
- [5] S.-T. Liong, D. Zheng, Y.-C. Huang, and Y. Gan, "Leather defect classification and segmentation using deep learning architecture," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 10-11, pp. 1105–1117, 2020.
- [6] Y. Gan, S.-S. Chee, Y.-C. Huang, S.-T. Liong, and W.-C. Yau, "Automated leather defect inspection using statistical approach on image intensity," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–17, 2020.
- [7] W. P. Amorim, H. Pistori, M. C. Pereira, and M. A. C. Jacinto, "Attributes reduction applied to leather defects classification," in *Proceedings of the 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 353–359, IEEE, Gramado, Brazil, August 2010.
- [8] S. Winiarti, A. Prahara, and D. P. I. Murinto, "Pre-trained convolutional neural network for classification of tanning leather image," *Network (CNN)*, vol. 9, no. 1, 2018.
- [9] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [10] W. R. Klecka, G. R. Iversen, and W. R. Klecka, *Discriminant analysis*, Sage, California, CA, USA, 1980.



- [11] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [12] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [13] S. R. Gunn, "Support vector machines for classification and regression," *ISIS technical report*, vol. 14, no. 1, pp. 5–16, 1998.
- [14] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
- [15] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14–15, pp. 2627–2636, 1998.
- [16] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.
- [17] T. Chen, T. He, M. Benesty et al., *Xgboost: Extreme Gradient Boosting, R Package Version*, vol. 1, no. 4, pp. 1–4, 2015.
- [18] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011.
- [19] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 626–632, 2002.
- [20] Y. Li and W. Ma, "Applications of artificial neural networks in financial economics: a survey," vol. 1, pp. 211–214, in *Proceedings of the 2010 International symposium on computational intelligence and design*, vol. 1, IEEE, Hangzhou, China, October 2010.
- [21] F. Wang, "The use of artificial neural networks in a geographical information system for agricultural land-suitability assessment," *Environment and Planning A: Economy and Space*, vol. 26, no. 2, pp. 265–284, 1994.
- [22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," December 2014, <https://arxiv.org/abs/1412.6980>.
- [23] S.-T. Liong, Y. S. Gan, K.-H. Liu et al., "Efficient neural network approaches for leather defect classification," June 2019, <https://arxiv.org/abs/1906.06446>.

## Research Article

# A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and YOLO

Cheng-Jian Lin <sup>1,2</sup>, Shiou-Yun Jeng,<sup>3</sup> and Hong-Wei Lioa<sup>1</sup>

<sup>1</sup>Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan

<sup>2</sup>College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan

<sup>3</sup>Department of Business Administration, Asia University, Taichung 413, Taiwan

Correspondence should be addressed to Cheng-Jian Lin; [cjlin@ncut.edu.tw](mailto:cjlin@ncut.edu.tw)

Received 7 May 2021; Revised 30 July 2021; Accepted 7 October 2021; Published 2 November 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Cheng-Jian Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, vehicle detection and classification have become essential tasks of intelligent transportation systems, and real-time, accurate vehicle detection from image and video data for traffic monitoring remains challenging. The most noteworthy challenges are real-time system operation to accurately locate and classify vehicles in traffic flows and working around total occlusions that hinder vehicle tracking. For real-time traffic monitoring, we present a traffic monitoring approach that overcomes the abovementioned challenges by employing convolutional neural networks that utilize You Only Look Once (YOLO). A real-time traffic monitoring system has been developed, and it has attracted significant attention from traffic management departments. Digitally processing and analyzing these videos in real time is crucial for extracting reliable data on traffic flow. Therefore, this study presents a real-time traffic monitoring system based on a virtual detection zone, Gaussian mixture model (GMM), and YOLO to increase the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. In this study, the Montevideo Audio and Video Dataset (MAVD), the GARM Road-Traffic Monitoring data set (GRAM-RTM), and our collection data sets are used to verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively. In addition, the average absolute percentage error of vehicle speed estimation with the proposed method is about 7.6%.

## 1. Introduction

Traffic monitoring with an intelligent transportation system provides solutions to various challenges, such as vehicle counting, speed estimation, accident detection, and assisted traffic surveillance [1–5]. A traffic monitoring system essentially serves as a framework to detect the vehicles that appear on a video image and estimate their position while they remain in the scene. In the case of complex scenes with various vehicle models and high vehicle density, accurately locating and classifying vehicles in traffic flows is difficult [6, 7]. Moreover, limitations occur in vehicle detection due to environmental changes, different vehicle features, and

relatively low detection speeds [8]. Therefore, an algorithm must be developed for a real-time traffic monitoring system with the capabilities of real-time computation and accurate vehicle detection. Therefore, the accurate and quick detection of vehicles from traffic images or videos has theoretical and practical significance.

With the rapid development of computer vision and artificial intelligence technologies, object detection algorithms based on deep learning have been widely investigated. Such algorithms can extract features automatically through machine learning; thus, they possess a powerful image abstraction ability and an automatic high-level feature representation capability. A few excellent object detection

networks, such as single-shot detection (SSD) [9], Fast R-CNN [10], YOLOv3 [11], and YOLOv4 [12], have been implemented for traffic detection using deep learning object detectors [13]. For example, Biswas et al. [14] implemented SSD to estimate traffic density. Yang et al. [15] proposed a multitasking-capable Faster R-CNN method that uses a single image to generate three-dimensional (3D) space coordinate information for an object with monocular vision to facilitate autonomous driving. Huang et al. [8] proposed a single-stage deep neural network called YOLOv3 and applied it to data sets generated in different environments to improve its real-time detection accuracy. Hu et al. [16] proposed an improved YOLOv4-based video stream vehicle target detection algorithm to solve the problem in the detection speed. In addition, the most noteworthy challenges associated with traffic monitoring systems are real-time operation for accurately locating and classifying vehicles in traffic flows and total occlusions that hinder vehicle tracking. Therefore, YOLO was developed as a regression-based, high-performance algorithm for the real-time detection of and statistics collection from vehicle flows.

The robustness of YOLOv3 and YOLOv4 to road marking detection improves its accuracy in small target detection. The model that is based on the TensorFlow framework, to enhance the real-time monitoring of traffic-flow problems by an intelligent transportation system [17]. The YOLOv3 network comprises 53 layers. It uses the Feature Pyramid Network for pedestrian detection to handle general multiscale object detection problems and the deep residual network (ResNet) ideas to extract image features for achieving a trade-off between detection speed and detection accuracy [18]. In addition to leveraging anchor boxes with predesigned scales and aspect ratios to predict vehicles of different sizes, YOLOv3 and YOLOv4 can realize real-time vehicle detection with a top-down architecture [19]. Moreover, a real-time vehicle detection and classification system can perform foreground extraction, vehicle detection, vehicle feature extraction, and vehicle classification [20]. To test the proposed method for vehicle classification, a vehicle-feature-based virtual detection zone and virtual detection line, which are predefined for each frame in a video, are used for vehicle feature computation [21]. Grents et al. [22] proposed a video-based system that uses a convolutional neural network to count vehicles, classify vehicles, and determine the vehicle speed. Tabassum et al. [23, 24] applied YOLO and a transfer learning approach to recognize native vehicles and vehicle classification on Bangladeshi Roads. Therefore, YOLO can be used to obtain a better matching map.

To increase vehicle counting and classification problems in real-time traffic monitoring, this study presents a real-time traffic monitoring system based on a virtual detection zone, Gaussian mixture model (GMM), and YOLO to increase the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. The major contributions of this study are described as follows: (1) A real-time traffic monitoring system is developed

to perform real-time vehicle counting, vehicle speed estimation, and vehicle classification; (2) the virtual detection zone, GMM, and YOLO are used to increase vehicle counting and classification efficiency; (3) the distance and time traveled by a vehicle is proposed to estimate the vehicle speed; and (4) the MAVD, GRAM-RTM, and our collection data sets are used to verify various methods and the proposed method with YOLOv4 achieving the highest classification accuracy in the three data sets.

The remainder of this study is organized as follows. Section 2 describes the materials and methods, including data set preparation, vehicle counting method, and vehicle classification method. Section 3 presents the results of and a discussion on the proposed real-time vehicle counting, speed estimation, and classification system based on a virtual detection zone and YOLO. Finally, Section 4 presents a few concluding remarks and an outline for future research on real-time traffic monitoring.

## 2. Materials and Methods

To count vehicles from traffic videos, this study proposes a real-time vehicle counting, speed estimation, and classification system based on the virtual detection zone and YOLO. We combined a vehicle detection method with a classification system on the basis of two conditions between the virtual detection zone and the virtual detection lane line. To detect vehicles, a Gaussian mixture model (GMM) is applied to detect moving objects in each frame of a traffic video. Figure 1 shows a flowchart of the vehicle counting and classification process used in the proposed real-time vehicle counting, speed estimation, and classification system. In this study, first, traffic videos are collected to train the image data and used to perform vehicle classification verification. Next, GMM and virtual detection zone are used for vehicle counting. Finally, YOLO is used to perform vehicle classification in real time. In this study, the three steps are described as follows:

Part 1: Collect traffic videos from online cameras.

In this study, traffic videos were collected from online cameras and used for image data training and vehicle classification verification, as described in Section 2.1.

Part 2: Perform vehicle counting using GMM and virtual detection zone.

To realize real-time vehicle counting, object detection and recognition are performed. A virtual detection lane line and virtual detection zone are used to perform vehicle counting and speed estimation, respectively, as described in Section 2.2 and Section 2.4, respectively.

Part 3: Perform vehicle classification and speed estimation using the YOLOv3 and YOLOv4 algorithms.

*2.1. Data Set Preparation.* The data set used in this study was prepared by collecting traffic videos recorded with online cameras installed along various roads in Taiwan. Image data



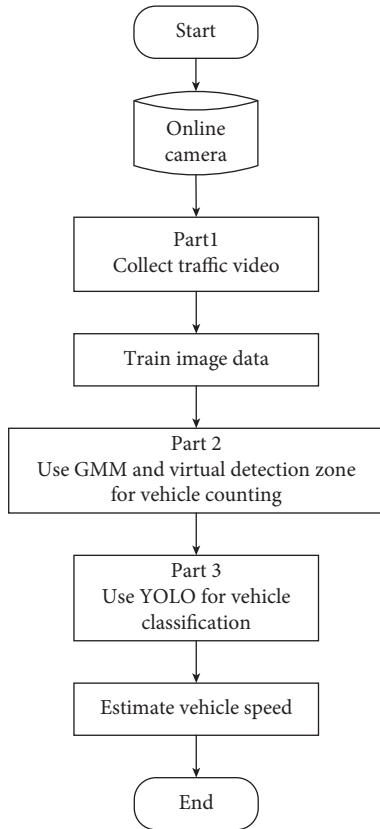




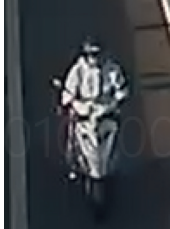



FIGURE 1: Flowchart of the vehicle counting and classification process.

were extracted from the traffic videos using a script, and labeling was performed using an open-source software application called “labeling” [25]. According to the common types of vehicles on the road are announced by the Directorate General of Highways, Ministry of Transportation and Communications (MOTC) in Taiwan, this study divides six different sizes, such as sedans, trucks, scooters, buses, hlinkcars, and flinkcars, in the training process, and the vehicle lengths of these six vehicle classes are listed in Table 1. In this study, we used YOLO to perform vehicle classification without using the length of the vehicle.

**2.2. Vehicle Counting.** To count vehicles, a GMM is used for the background subtraction in the complex environment to identify the regions of moving objects. The GMM is quite reliable in the background extraction and foreground segmentation process, so the characteristics of a moving object in video surveillance are easier to detect [26, 27]. The virtual detection zone is predefined in each video and used for vehicle feature computation. When the vehicle enters a virtual detection zone and virtual detection lane line, the GMM is used for vehicle counting. The vehicle counting window is depicted in Figure 2.

**2.3. Vehicle Detection and Classification.** This study uses the YOLO algorithm to classify vehicles into six classes. The validation method is used for verifying the vehicle

TABLE 1: Vehicle classification.

Class	Vehicle	Length (m)	Image
1	Sedan	3.6–5.5	
2	Truck	>5.5–11	
3	Scooter	1–2.5	
4	Bus	7–12.2	
5	Hlinkcar	15–18	
6	Flinkcar	18–20	

classification in the collected videos. A visual classifier based on the YOLO algorithm is used to verify the vehicle classification capability. Figure 3 depicts the architecture of the visual classifier based on the YOLO algorithm that is used for classifying each vehicle into one of six classes. In the training process, when a vehicle belonging to one of the six classes is detected, all bounding boxes are extracted, their classes are manually labeled, and the labeled data are passed to the YOLO model for classifying the vehicle.

The YOLOv3 model architecture displayed in Figure 4 was used in this study. Images of size  $416 \times 416$  px were input into the Darknet-53 network. This feature extraction network comprises 53 convolutional layers, and thus, it is called

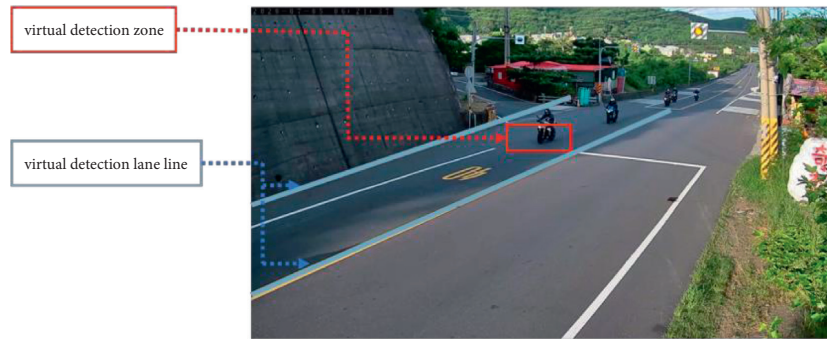


FIGURE 2: Object detection window.

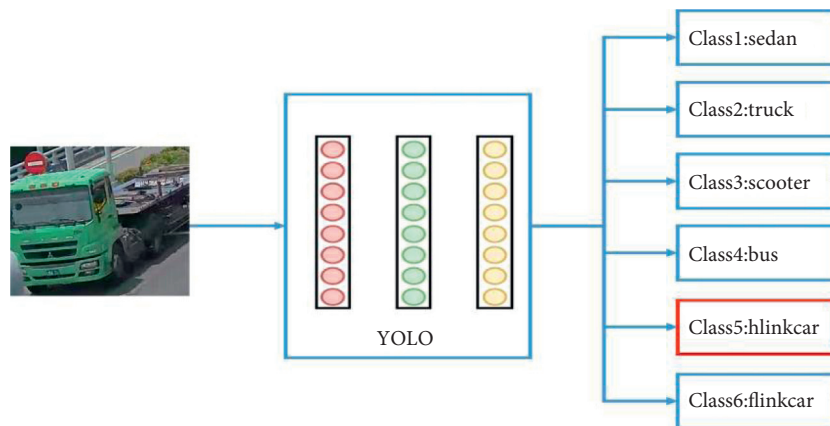


FIGURE 3: Architecture of visual classifier based on the YOLO algorithm for verifying the vehicle classification.

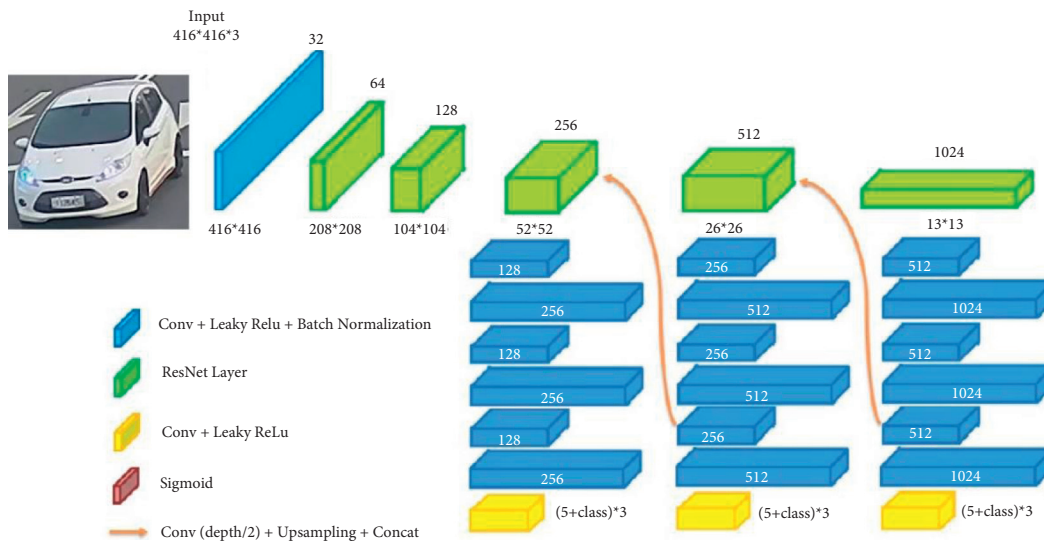


FIGURE 4: YOLOv3 model architecture.

Darknet-53 [11]. In Darknet-53, alternating convolution kernels are used, and after each convolution layer, a batch normalization layer is used for normalization. The leaky rectified linear unit function is used as the activation function, the pooling layer is discarded, and the step size of the convolution kernel is increased to reduce the size of the feature map. The YOLOv3 model uses ResNet for feature

extraction and subsequently uses the feature pyramid top-down and lateral connections to generate three features with sizes of  $13 \times 13 \times 1024$ ,  $26 \times 26 \times 512$ , and  $52 \times 52 \times 256$  px. The final output depth is  $(5 + \text{class}) \times 3$ , which indicates that the following parameters are predicted: four basic parameters and the credibility of a box across three regression bounding boxes as well as the possibility of each class being

contained in the bounding box. YOLOv3 uses the sigmoid function to score each class. When the class score is higher than the threshold, the object is considered to belong to a given category, and any object can simultaneously have multiple class identities without conflict.

The loss function of YOLOv3 is mainly divided into four parts. *A* denotes the loss of the identified center coordinates that is used to predict  $(x, y)$  in the bounding box to ensure that it is only valid for the highest predicted target. *B* is the loss of  $(w, h)$  width and height in the predicted bounding box, and the error value reflects the bounding box of different sizes in the object to predict the square root of the width and height instead of directly predicting the width and height of the bounding box. *C* is the loss of the predicted object category, assuming that each box is a cell; if the center of the object detection is in this cell, then mark the cell with bounding box  $(x, y, w, h)$ , and there is also category information to meet which object in the image to predict in the cell. *D* denotes the loss of the credibility of the predicted object to calculate the credibility in each bounding box to know that when the bounding box predicts the object. When the object is not predicted, there will be a credibility prediction penalty  $\lambda_{\text{noobj}} = 0.5$ , and it is defined as follows:

$$\begin{aligned} & \overbrace{\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \ell_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]}^A \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \ell_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \underbrace{\sum_{i=0}^{S^2} \ell_i^{\text{obj}} \sum_{j=0}^B [(p_i(c) - \hat{p}_i(c))^2]}_C \\ & + \underbrace{\sum_{i=0}^{S^2} \sum_{j=0}^B \ell_{ij}^{\text{obj}} [(C_i - \hat{C}_i)^2] + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \ell_{ij}^{\text{noobj}} [(C_i - \hat{C}_i)^2]}_D, \end{aligned} \quad (1)$$

where  $x_i, y_i$  is the location of the centroid of the anchor box and  $w_i, h_i$  is the width and height of the anchor box.  $C_i$  is the *Objectness*, i.e., confidence score of whether there is an object or not, and  $p_i(c)$  is the classification loss.

YOLOv4 is the latest algorithm of YOLO series, which is the basis of YOLOv3, scales both up and down and is applicable to small and large networks while maintaining optimal speed and accuracy, and the network architecture is shown in Figure 5. Compared with YOLOv3, YOLOv4-tiny is an extended version of YOLOv3. The original Darknet53 network is added with a CSP network. Backbone is CSPO-SANet proposed by Cross Stage Partial Network (CSPNet) + One-Shot Aggregation Network (OSANet), plus Partial in Computational Blocks (PCB) technology. CSPNet can be applied to different CNN architectures to reduce the amount of parameters and calculations while improving accuracy. OSANet is derived from the OSA model in VoVNet. Its central idea is improved by the DenseNet module. At the end, all layers are connected to allow input consistent with the

number of output channels; PCB technology can make the model more flexible because it can be adjusted according to the structure to achieve the best accuracy-speed balance.

The loss function remains the same as the YOLOv4 model, which consists of three parts: classification loss, regression loss, and confidence loss [28]. Classification loss and confidence loss remain the same as the YOLOv3 model, but complete intersection over union (CIoU) is used to replace mean-squared error (MSE) to optimize the regression loss [29]. The CIoU loss function is shown as follows:

$$\begin{aligned} \text{LOSS} &= 1 - \text{IoU} + \frac{\rho^2(b, b^{\text{gt}})}{c^2} + \alpha v \\ & - \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] \\ & - \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] \\ & - \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in \text{classes}} [\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))], \end{aligned} \quad (2)$$

where  $S^2$  represents  $S \times S$  grids; each grid generates  $B$  candidate boxes, and each candidate box gets corresponding bounding boxes through the network; finally,  $S \times S \times B$  bounding boxes are formed. If there is no object (noobj) in the box, only the confidence loss of the box is calculated. The confidence loss function uses cross entropy error and is divided into two parts: there is the object (obj) and noobj. The loss of noobj increases the weight coefficient  $\lambda$ , which is to reduce the contribution weight of the noobj calculation part. The classification loss function also uses cross entropy error. When the  $j$ -th anchor box of the  $i$ -th grid is responsible for certain ground truth, then the bounding box generated by this anchor box will calculate the classification loss function.

**2.4. Speed Estimation.** The real-time vehicle speed is also calculated in this study. Figure 6 shows the video images taken along the direction parallel to the length of the car (defined as the  $y$ -axis) and parallel to the width of the car (defined as the  $x$ -axis). First, as per the scale in the video, the yellow line (referred to as  $L$ ) in the red circle has a length of 4 m in accordance with traffic laws. A GMM is used to draw a virtual detection zone (blue box) on the road to be tested (referred to as  $Q$ ). The green box is the car frame (referred to as  $C$ ), and the midpoint of the car is  $Ct$ .

$$u_0 = \frac{\overline{L_{AB}}(px)}{\overline{L_{AB}}(m)}, \quad (3)$$

$$\alpha = \frac{u_0}{T},$$

$$\text{If } \alpha > 1, \quad u_i = u_0 \alpha^i, \quad u_j = \frac{u_0}{\alpha^j}, \quad (4)$$

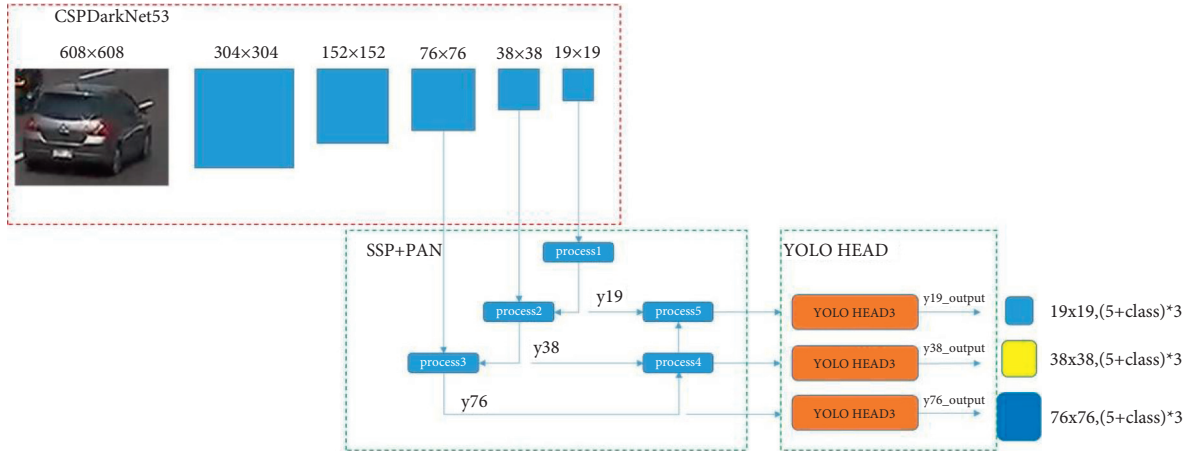
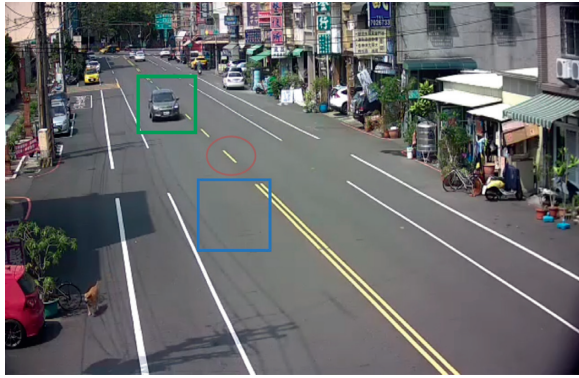
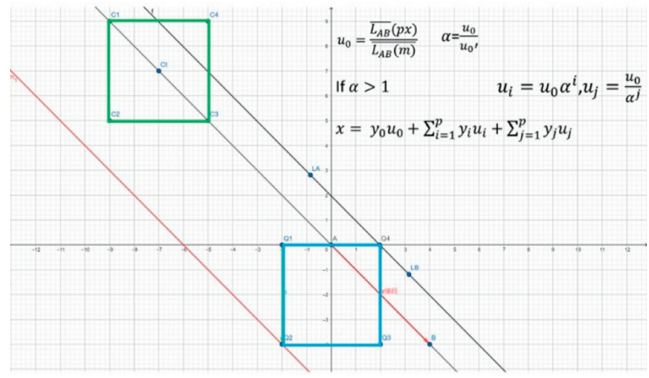


FIGURE 5: YOLOv4 model architecture.



(a)



(b)

FIGURE 6: Diagram of speed estimation ((a): real picture on the video; (b): control scale).

where  $u_0$  is the scale,  $i$  is the scale of the blue box,  $j$  is the scale of the green box,  $px$  is the length of the video, and  $m$  is the actual length. The parameter  $\alpha$  denotes the increase or decrease in relationship of the scale per unit length on the  $y$ -axis. If  $\alpha > 1$ , the speed calculation is performed using equation (4).

To calculate the parallel  $L$  line segment of  $Ct$  (referred to as  $L^*$ ), the algorithm computes the  $L^*$  distance  $y$  between  $A$  and  $B$ . Then, it restores  $y$  from its scale relationship with the actual line segment length  $x$ , where  $x$  denotes the distance traveled by the vehicle in  $Q$ .

$$x = y_0 u_0 + \sum_{i=1}^p y_i u_i + \sum_{j=1}^p y_j u_j. \quad (5)$$

In the calculation process, the program is used to determine the frame rate of the video and calculate the number of frames for which the vehicle travels in  $Q$  (referred to as  $p$ ). Equation (6) is used to find the travel time of the vehicle from  $A$  to  $B$  in  $Q$ .

$$t = \frac{p}{\text{fps}}, \quad (6)$$

$$v = \frac{x}{t}, \quad (7)$$

$$v' = x \times \frac{3.6}{t}. \quad (8)$$

Equation (7) is used for calculating vehicle speed. After unit conversion (m/s to km/h), Equation (8) provides the vehicle speed.

### 3. Results and Discussion

All experiments in this study were performed using the YOLO algorithm under the Darknet framework, and the program was written in *Python 2.7*. To validate the real-time traffic monitoring system, we used a real-world data set to perform vehicle detection, vehicle counting, speed estimation, and classification. In this study, three test data sets were used to evaluate the proposed method. One of these data sets was mainly derived from traffic video images of online cameras on various roads in Taiwan, and it contains 12,761 training images and 3,190 testing images. Second, the Montevideo Audio and Video Dataset (MAVD), which contains data on different levels of traffic activity and social use characteristics in Montevideo city, Uruguay, was used as the other traffic data set [30]. Finally, GARM Road-Traffic Monitoring (GRAM-RTM) data set [21] has four categories (i.e., cars, trucks, vans,

and big-trucks). The total number of different objects in each sequence is 256 for M-30, 235 for M-30-HD, and 237 for Urban 1. In this study, the definition of accuracy is based on the classification of vehicles in the database. In the video verification, if the results of manual and proposed system classification of the vehicles are the same, it means that the count is correct; otherwise, it is the wrong vehicle counting.

**3.1. Vehicle Counting.** Seven input videos of the road, each ranging in length between 3 and 5 minutes, were recorded at 10 am and 8 pm. In addition, eleven input videos of the road in the rain were also recorded for testing. Each frame in these traffic videos was captured at 30 fps. The first experimental results of real-time vehicle counting using the proposed method during the day are summarized in Table 2. The symbols *S* and *L* denote small and large vehicles, respectively. The vehicle counting accuracy of the proposed method at 10 am was 95.5%. The second experimental results of real-time vehicle counting using the proposed method during the night are summarized in Table 3. The vehicle counting accuracy of the proposed method at 8 pm was 98.5%. In addition, the third experimental results of real-time vehicle counting using the proposed method in the rain are summarized in Table 4. The vehicle counting accuracy of the proposed method was 94%. Screenshots of vehicle detection with the proposed real-time vehicle counting and classification system are depicted in Figure 7, where the detected vehicles are represented as green rectangles.

Vehicle counting in online videos is delayed due to network stoppages or because the target vehicle may be blocked by other vehicles on the screen, which causes the count to be missed. In addition, poor lighting in the rain and night affects the vehicle recognition capabilities of YOLOv3 and YOLOv4. These challenges can be overcome using a stable network connection and adjusting the camera brightness, respectively. Therefore, the novelty of this study is to solve the problem of unclear recognition in the rain.

**3.2. Speed Estimation.** In this subsection, the vehicle speed can be estimated using the proposed method. Table 5 lists the actual and the estimated speeds of the vehicles. The results indicate that the average absolute percentage error of vehicle speed estimation was about 7.6%. The use of online video for vehicle speed estimation will cause large speed errors due to network delays. Therefore, network stability is essential to reduce the percentage error in the speed estimation.

**3.3. Comparison Results Using the MAVD and GRAM-RTM Data Sets.** MAVD traffic data set [30] and GARM Road-Traffic Monitoring (GRAM-RTM) data set [21] were used for evaluating the vehicle counting performance of the proposed method. The videos were recorded with a GoPro Hero 3 camera at a frame rate of 30 fps and a resolution of 1920 × 1080 px. We analyzed 10 videos, and the vehicle counting accuracy of the proposed method at 10 am for the

TABLE 2: The real-time vehicle counting using the proposed method in the daytime.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	<i>S</i>	<i>L</i>	Total	<i>S</i>	<i>L</i>	Total
1	31	7	38	29	6	35
2	18	0	18	18	0	18
3	16	5	21	16	5	21
4	28	0	28	25	0	25
5	22	3	25	20	3	23
6	11	0	11	11	0	11
7	11	1	12	9	0	9

TABLE 3: The real-time vehicle counting using the proposed method in the night time.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	<i>S</i>	<i>L</i>	Total	<i>S</i>	<i>L</i>	Total
1	23	8	31	23	7	30
2	31	5	36	29	5	34
3	15	2	17	14	1	15
4	10	3	13	9	3	12
5	19	4	23	19	4	23
6	11	2	13	11	2	13
7	35	5	38	34	3	37

TABLE 4: The real-time vehicle counting using the proposed method in the raining day.

Video no.	Actual number of vehicles			Estimated number of vehicles		
	<i>S</i>	<i>L</i>	Total	<i>S</i>	<i>L</i>	Total
1	9	0	9	9	0	9
2	12	0	12	10	1	11
3	13	0	13	13	0	13
4	7	0	7	8	0	8
5	11	1	12	14	1	15
6	15	0	15	17	0	17
7	10	0	10	13	0	13
8	7	0	7	10	0	10
9	12	0	12	15	0	15
10	12	0	12	14	0	14
11	17	0	17	19	0	19

MAVD traffic data set was 93.84%. Vehicle classification results of the proposed method using MAVD traffic data set are listed in Table 6.

In summary, three data sets, namely, MAVD, GRAM-RTM, and our collection data sets, were used to verify the proposed method and Fast RCNN method [10]. The MAVD training and testing samples contains vehicles belonging to four categories (i.e., cars, buses, motorcycles, and trucks). The GRAM-RTM data set has four categories (i.e., cars, trucks, vans, and big-trucks). The total number of different objects in each sequence is as follows: 256 for M-30, 235 for M-30-HD, and 237 for Urban 1. Table 7 shows the classification accuracy results of three data sets using various methods. In Table 7, the



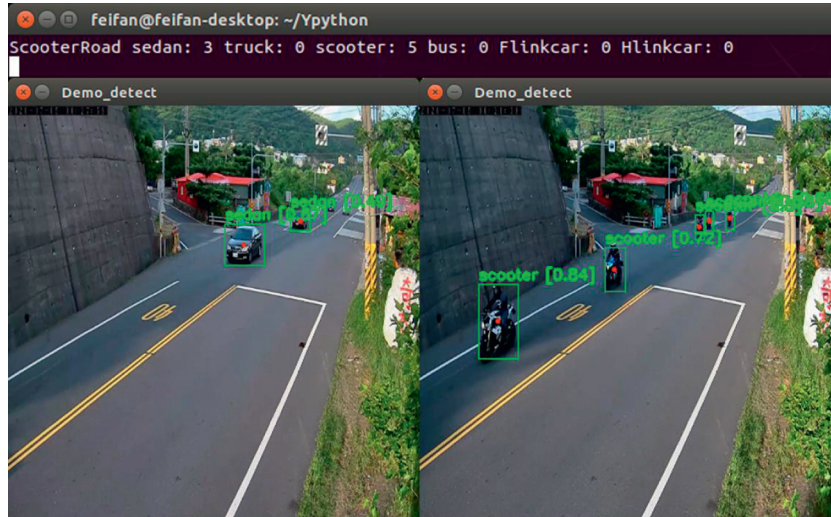


FIGURE 7: Screenshots from the proposed real-time vehicle counting, speed estimation, and classification system.

TABLE 5: The actual and the estimated vehicle speeds using the proposed method.

Vehicle ID	Actual speed	Estimated speed	Difference	Error (%)
1	60	63	3	5
2	70	75	5	7
3	72	63	-9	12.5
4	99	100	1	1
5	84	85	1	1
6	67	60	-7	10
7	73	71	-2	2.7
8	67	64	-3	4.4
9	37	43	6	16
10	73	77	4	5
11	55	50	-5	9
12	48	54	6	12.5
13	111	127	16	14.4
14	79	75	-4	5
15	69	71	2	2.8
16	82	75	-7	8.5
17	83	73	-10	12
<b>Average error</b>				<b>7.6</b>

TABLE 6: Vehicle classification results of the proposed method using MAVD traffic data set.

Video no.	Total number of vehicles			Number of counted vehicles		
	S	L	Total	S	L	Total
1	8	0	8	8	0	8
2	5	0	5	4	0	4
3	4	2	6	3	2	5
4	4	0	4	3	0	3
5	3	0	3	3	1	4
6	1	0	1	1	0	1
7	7	2	9	7	2	9
8	11	0	11	10	0	10
9	9	0	9	9	0	9
10	9	0	9	8	0	8

proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, three different environments (i.e., daytime, night time, and rainy day) are used to verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively.

Recently, some researchers have adopted various methods for vehicle classification using GRAM-RTM data set, such as Faster RCNN [10], CNN [31], and DNN [32]. Therefore, we use the same GRAM-RTM data set to compare the proposed method with other methods. Table 8 shows the comparison results. In Table 8, the results show that the proposed method with YOLOv4 can perform better than the other methods.

TABLE 7: Classification accuracy results of three data sets using various methods.

Data sets		Methods	Accuracy (%)	FPS
MAVD		Faster RCNN [10]	97.21	5
		Proposed method with YOLOv3	97.66	15
		Proposed method with YOLOv4	98.91	15
GRAM-RTM		Faster RCNN [10]	91.54	5
		Proposed method with YOLOv3	98.02	15
		Proposed method with YOLOv4	99.5	15
Our data set	Daytime	Faster RCNN [10]	97.7	5
		Proposed method with YOLOv3	98	15
		Proposed method with YOLOv4	99.1	15
	Night time	Faster RCNN [10]	93.59	5
		Proposed method with YOLOv3	98	15
		Proposed method with YOLOv4	98.6	15
	Rainy day	Faster RCNN [10]	87.5	5
		Proposed method with YOLOv3	90	15
		Proposed method with YOLOv4	98	15

TABLE 8: Classification accuracy results of various methods using GARM-RTM data set.

Methods	Faster RCNN [10]	Gomaa et al. [31]	Abdelwahab [32]	Our proposed method	
				YOLOv3	YOLOv4
Accuracy (%)	91.54	96.8	93.51	98.02	99.5

## 4. Conclusions

In this study, a real-time traffic monitoring system based on a virtual detection zone, GMM, and YOLO is proposed for increasing the vehicle counting and classification efficiency. GMM and a virtual detection zone are used for vehicle counting, and YOLO is used to classify vehicles. Moreover, the distance and time traveled by a vehicle are used to estimate the speed of the vehicle. In this study, MAVD, GRAM-RTM, and our collection data sets are used to verify the proposed method. Experimental results indicate that the proposed method with YOLOv4 achieved the highest classification accuracy of 98.91% and 99.5% in MAVD and GRAM-RTM data sets, respectively. Moreover, the proposed method with YOLOv4 also achieves the highest classification accuracy of 99.1%, 98.6%, and 98% in daytime, night time, and rainy day, respectively. In addition, the average absolute percentage error of vehicle speed estimation with the proposed method is about 7.6%. Therefore, the proposed method can be applied to vehicle counting, speed estimation, and classification in real time.

However, the proposed method has a few limitations. The vehicles appearing in the video are assumed to be inside the virtual detection zone; thus, the width of the virtual detection zone should be sufficiently large for counting the vehicles. In the future work, we will focus on algorithm acceleration and model simplification.

## Data Availability

The MAVD and GRAM-RTM traffic data sets are available at [https://zenodo.org/record/3338727#\\_YBD8B-gzY2w](https://zenodo.org/record/3338727#_YBD8B-gzY2w) and <https://gram.web.uah.es/data/datasets/rtm/index.html>.

## Conflicts of Interest

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

## Acknowledgments

This research was funded by the Ministry of Science and Technology of the Republic of China, grant number MOST 110-2221-E-167-031-MY2.

## References

- [1] Y. Mo, G. Han, H. Zhang, X. Xu, and W. Qu, "Highlight-assisted nighttime vehicle detection using a multi-level fusion network and label hierarchy," *Neurocomputing*, vol. 355, pp. 13–23, 2019.
- [2] D. Feng, C. Haase-Schuetz, L. Rosenbaum et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, p. 3, 2019.
- [3] Z. Liu, Y. Cai, H. Wang et al., "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [4] Y. Qian, J. M. Dolan, and M. Yang, "DLT-NET: joint detection of drivable areas, lane lines, and traffic objects," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4670–4679, 2020.
- [5] Y. Cai, L. Dai, H. Wang et al., "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

- [6] Á. Llamazares, E. J. Molinos, and M. Ocaña, "Detection and tracking of moving obstacles (DATMO): a review," *Robotica*, vol. 38, no. 5, pp. 761–774, 2020.
- [7] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds, and S. Atkinson, "A vision-based pipeline for vehicle counting, speed estimation, and classification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.
- [8] Y.-Q. Huang, J.-C. Zheng, S.-D. Sun, C.-F. Yang, and J. Liu, "Optimized YOLOv3 algorithm and its application in traffic flow detections," *Applied Sciences*, vol. 10, Article ID 3079, 2020.
- [9] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the Computer Vision—ECCV 2016*, pp. 21–37, Amsterdam, Netherlands, October 2016.
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [11] X. Zhang and X. Zhu, "Vehicle detection in the aerial infrared images via an improved Yolov3 network," in *Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, pp. 372–376, Wuxi, China, July 2019.
- [12] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, <https://arxiv.org/abs/2004.10934v1>.
- [13] J. Redmon and A. Farhadi, "YOLO V3: an incremental improvement," pp. 1–22, 2018, <http://arxiv.org/abs/1804.02767>.
- [14] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using single shot detection (SSD) and mobilenet-SSD," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 110, pp. 176–184, 2019.
- [15] W. Yang, Z. Li, C. Wang, and J. Li, "A multi-task Faster R-CNN method for 3D vehicle detection based on a single image," *Applied Soft Computing*, vol. 95, Article ID 106533, 2020.
- [16] X. Hu, Z. Wei, and W. Zhou, "A video streaming vehicle detection algorithm based on YOLOv4," in *Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2081–2086, Chongqing, China, March 2021.
- [17] C. Y. Cao, J. C. Zheng, Y. Q. Huang, J. Liu, and C. F. Yang, "Investigation of a promoted You Only Look once algorithm and its application in traffic flow monitoring," *Applied Sciences*, vol. 9, Article ID 3619, 2019.
- [18] H. Zhou, L. Wei, C. P. Lim, D. Creighton, and S. Nahavandi, "Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7074–7085, 2018.
- [19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Honolulu, HI, USA, July 2017.
- [20] C.-Y. Chen, Y.-M. Liang, and S.-W. Chen, "Vehicle classification and counting system," in *Proceedings of the 2014 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 485–490, Shanghai, China, July 2014.
- [21] N. Seenoung, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "Vehicle detection and classification system based on virtual detection zone," in *Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, July 2016.
- [22] A. Gretns, V. Varkentin, and N. Goryaev, "Determining vehicle speed based on video using convolutional neural network," *Transportation Research Procedia*, vol. 50, pp. 192–200, 2020.
- [23] S. Tabassum, M. S. Ullah, N. H. Al-Nur, and S. Shatabda, "Native vehicles classification on Bangladeshi roads using CNN with transfer learning," in *Proceedings of the 2020 IEEE Region 10 Symposium (TENSYP)*, Dhaka, Bangladesh, June 2020.
- [24] S. Tabassum, S. Ullah, N. H. Al-nur, and S. Shatabda, "Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification," *Data in Brief*, vol. 33, Article ID 106465, 2020.
- [25] LabelImg (accessed on 5 March 2018), <https://github.com/tzutalin/labelImg>.
- [26] A. Nurhadiyatna, B. Hardjono, A. Wibisono et al., "Improved vehicle speed estimation using Gaussian mixture model and hole filling algorithm," in *Proceedings of the 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Sanur Bali, Indonesia, September 2013.
- [27] A. Ghosh, M. S. Sabuj, H. H. Sonet, S. Shatabda, and D. M. Farid, "An adaptive video-based vehicle detection, classification, counting, and speed-measurement system for real-time traffic data collection," in *Proceedings of the 2019 IEEE Region 10 Symposium (TENSYP)*, Kolkata, India, June 2019.
- [28] L. Wu, J. Ma, Y. Zhao, and H. Liu, "Apple detection in complex scene using the improved YOLOv4 model," *Agronomy*, vol. 11, no. 3, p. 476, 2021.
- [29] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," in *Proceedings of the 2020 AAAI Conference on Artificial Intelligence*, pp. 12993–13000, New York, NY, USA, February 2020.
- [30] P. Zinemanas, P. Cancela, and M. Rocamora, "MAVD: a dataset for sound event detection in urban environments," in *Proceedings of the 4th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2019)*, New York, NY, USA, October 2019.
- [31] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R.-I. Taniguchi, "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow," *Sensors*, vol. 19, no. 20, Article ID 4588, 2019.
- [32] M. A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *Proceedings of the 2019 International Conference on Innovative Trends in Computer Engineering (ITCE'2019)*, Aswan, Egypt, February 2019.



## Research Article

# Hilbert–Schmidt Independence Criterion Regularization Kernel Framework on Symmetric Positive Definite Manifolds

Xi Liu <sup>1</sup>, Zengrong Zhan <sup>1</sup> and Guo Niu <sup>2</sup>

<sup>1</sup>School of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou 510483, China

<sup>2</sup>School of Electronics and Information Technology, Foshan University, Foshan 510275, China

Correspondence should be addressed to Xi Liu; liux239@mail2.sysu.edu.cn

Received 12 May 2021; Revised 16 August 2021; Accepted 22 September 2021; Published 11 October 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Xi Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image recognition tasks involve an increasingly high amount of symmetric positive definite (SPD) matrices data. SPD manifolds exhibit nonlinear geometry, and Euclidean machine learning methods cannot be directly applied to SPD manifolds. The kernel trick of SPD manifolds is based on the concept of projecting data onto a reproducing kernel Hilbert space. Unfortunately, existing kernel methods do not consider the connection of SPD matrices and linear projections. Thus, a framework that uses the correlation between SPD matrices and projections to model the kernel map is proposed herein. To realize this, this paper formulates a Hilbert–Schmidt independence criterion (HSIC) regularization framework based on the kernel trick, where HSIC is usually used to express the interconnectedness of two datasets. The proposed framework allows us to extend the existing kernel methods to new HSIC regularization kernel methods. Additionally, this paper proposes an algorithm called HSIC regularized graph discriminant analysis (HRGDA) for SPD manifolds based on the HSIC regularization framework. The proposed HSIC regularization framework and HRGDA are highly accurate and valid based on experimental results on several classification tasks.

## 1. Introduction

Vision recognition tasks are often encountered in real-life application [1–3]. Most traditional image recognition algorithms are constructed in the Euclidean space [4, 5]. Recently, symmetric positive definite (SPD) matrices [6] have received more and more attention in terms of region covariance descriptor [7–9], Gaussian mixture model (GMM) [10], diffusion tensors [11, 12], and structure tensors [13, 14]. These descriptors utilize second-order statistical information to capture the correlation between different features and are effective in various applications [15–19]. The SPD matrices lie on an SPD manifold when endowed with an appropriate Riemannian metric. It is not adequate to directly use most of the conventional machine learning methods on SPD manifolds [20, 21]. Therefore, developing methods for classifying the points on SPD manifolds is of significant interest.

Most existing classification methods on SPD manifolds employ Riemannian metrics and matrix divergences as the

dissimilarity measurement [22–26], e.g., the log-Euclidean Riemannian metric (LERM) [21] and the affine-invariance Riemannian metric (AIRM) [20], and Jensen–Bregman LogDet divergence (JBLD) [27, 28] is not a real Riemannian metric, it facilitates a quick and approximate computation of the distance. However, these methods cannot be developed to other manifolds owing to specific metrics. Another common approach maps the points of a manifold to the tangent space of a specific matrix. Under this approach, traditional dimensionality reduction methods have been extended to Riemannian manifolds [29]. Tuzel et al. applied LogitBoost for classifying SPD manifolds [30], and the method employed in this case was generalized to multiclass classification [31]. Tangent space approximations could preserve the manifold geometry to some extent. However, mapping the points to the tangent space may bring inaccurate distance measurement, particularly for points far away from the center. In [32], sparse coding by embedding the SPD matrices to the unified tangent space was proposed.

More recent studies have addressed the nonlinearity by adopting the kernel trick. Kernel methods project the SPD manifold to a reproducing kernel Hilbert space (RKHS) and further project the points to linear spaces. Thus, classification algorithms can be extended to SPD manifolds [33–36]. However, mapping from the RKHS to the Euclidean space is based on a linear assumption; the intrinsic relationship of input data and projections is not considered.

In this study, a novel kernel framework of SPD manifolds by considering the connection of SPD matrices and linear projections is proposed to address the problem. An intrinsic relationship between SPD matrix and low-dimensional representation is introduced herein to make the low-dimensional representations more respectful to the intrinsic feature of the input data. In the proposed framework, the relationship of the SPD matrices and projections can be reflected by the Hilbert–Schmidt independence criterion (HSIC), and an HSIC regularization term is added to the traditional kernel framework. HSIC is usually used to express the statistical correlation between two datasets; it has been extended to supervised sparse learning feature selection [37–39], dictionary learning [40, 41], subspace learning [42], and nonlinear dimensionality reduction [43]. Although HSIC has been found in many applications, it seems not to have been directly applied on SPD manifolds. This study is an extension of our previously published work [44]. In [44], we proposed a method named HSIC subspace learning (HSIC-SL) by using global HSIC maximization. Compared with HSIC-SL, we use HSIC herein in the form of regularization term to build a novel kernel framework on SPD manifolds. The most significant aspect of the proposed framework is that it allows us to extend the traditional kernel methods to new HSIC regularization kernel methods, and the effectiveness of new methods can be improved. Additionally, this paper proposes an algorithm based on the HSIC regularization framework and graph embedding. Our primary contributions are summarized as follows:

- (1) The HSIC is applied to the kernel framework, and an HSIC regularization kernel framework is proposed. The application of the proposed framework to most of the existing kernel methods on SPD manifolds improves the effectiveness of these methods. This work can provide important contributions to the development of kernel methods on SPD manifolds.
- (2) Three kernel functions involved in HSIC regularization are presented. The most appropriate kernel function can be selected based on the target application, which increases the flexibility of the proposed framework.
- (3) A method called HSIC regularized graph discriminant analysis (HRGDA) on SPD manifolds is proposed based on the HSIC regularization kernel framework. HRGDA uses a LogDet divergence kernel for embedding and a variant of kernel linear discriminant analysis (LDA) for learning.

## 2. Related Work

As mentioned previously, the kernel trick is the most common method for addressing the nonlinearity of SPD manifolds. Riemannian locality preserving projections (RLPP) [45] embed Riemannian manifold in vector space via a Riemannian kernel; however, their time cost is high and the kernel is not always SPD. Jayasumana et al. [46] presented a theorem to judge the SPD of Gaussian radial basis function (RBF) kernels. Harandi et al. executed sparse coding by embedding SPD matrices into RKHS through matrix divergences [47]. Zhuang et al. proposed kernel learning and Riemannian metric (KLRM) based on data-dependent kernel learning [48]. Covariance discriminative learning (CDL) [8] maps the SPD matrices to a vector space by using matrix logarithm. Kernel-based subspace learning (KSLR) [19] defines an improved log-Euclidean RBF kernel and seeks the optimal subspace by using linear discriminant analysis (LDA).

*2.1. Kernelized Schemes.* The framework of kernel methods on SPD manifolds is presented in Figure 1. Here, let  $X = \{X_1, X_2, \dots, X_N\} \subseteq M$  be  $N$  samples on an SPD manifold  $M$ . First, the input SPD matrices on the SPD manifold  $M$  are embedded onto a high-dimensional RKHS  $H$  with a pre-defined kernel function  $k(X_i, X_j)$ . Second, the data are further projected on an  $m$ -dimensional linear subspace  $H'$ , which is isomorphic to a vector space  $R^m$ . Under the isomorphism assumption of kernel mapping, the projection  $y_i \in R^m$  of  $X_i \in M$  is obtained using  $y_i = AK_i^T$ , where  $A$  is the transformational matrix and  $K_i = [k_{i1} \ \dots \ k_{iN}]$ ,  $k_{ij} = k(X_i, X_j)$ . Furthermore, we have  $Y = AX$ . Third, the transformation matrix  $A$  is learned using the training data. To this end, manifold learning methods are performed on the low-dimensional Euclidean space. Therefore, the learning methods transform into the following optimization problem:

$$A = \arg \max_A f(A), \quad (1)$$

where  $f(A)$  is a cost function. Occasionally, the optimization problem is a minimum optimization problem [45, 46]. Finally, cluster and classification tasks can be realized in the vector space.

The optimization problem of equation (1) is customary to be designed as the cost function in many methods such that the optimization problem can be solved by eigenvalue decomposition [8, 19, 45]. Then, the objective functions of these methods can be constructed as

$$A = \arg \max_A (A^T F A). \quad (2)$$

The proposed HSIC regularization kernel framework is designed to improve the performance of these methods, e.g., RLPP, CDL, and KSLR, in the form of equation (2).

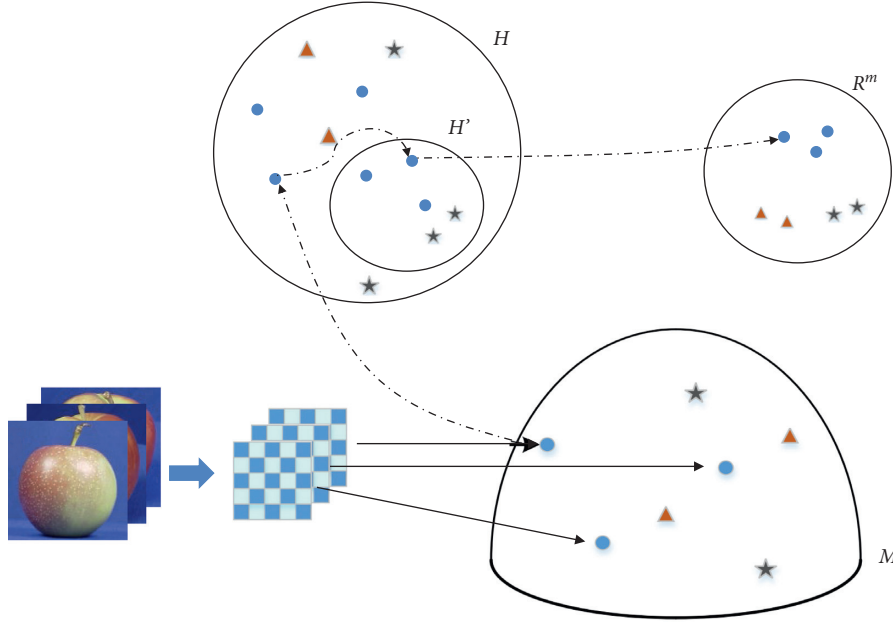


FIGURE 1: Framework of kernel methods on SPD manifolds.

2.2. *RLPP*. RLPP exploits locality preserving projections for discriminative learning; it tries to seek the optimal  $A$  by preserving the local geometry of Riemannian manifolds, which is reflected by a similarity matrix. Here, the binary matrix provides a penalty if adjacent points from the same class are mapped far away. The binary matrix  $W$  is defined as follows:

$$W_{ij} = \begin{cases} 1, & \text{if } l_i = l_j, \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where  $l_i$  and  $l_j$  are the labels.

The objective function is obtained using the following minimum optimization problem:

$$\begin{aligned} \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 W_{ij} &= \sum_i A^T K_i K_i^T A W_{ii} - \sum_{i,j} A^T K_i K_i^T A W_{ij}, \\ &= A^T K D K^T A - A^T K W K^T A = A^T K L K^T A, \end{aligned} \quad (4)$$

where  $K$  is the kernel matrix,  $D_{ii} = \sum_j W_{ij}$ , and  $L = D - W$ .

Therefore, the minimization problem is reduced to solve the following:

$$A = \arg \min_A A^T K L K^T A. \quad (5)$$

2.3. *CDL*. CDL uses the matrix logarithm operator to define the kernel function. Moreover, the objective function of CDL is provided as follows:

$$A = \arg \max_A A^T \frac{K W K}{K K^T} A, \quad (6)$$

where  $W$  is the connection matrix defined as

$$W_{ij} = \begin{cases} \frac{1}{q_c} & \text{if } l_i = l_j, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Both RLPP and CDL are constructed based on the traditional kernel framework, which means that the map from the input space to the projective space is a linear assumption. Thus, this paper proposes the HSIC regularization kernel framework by introducing statistical correlation between SPD matrices and low-dimensional representations. In the HSIC regularization kernel framework, the intrinsic connection of SPD matrices and low-dimensional projections is measured by HSIC. Under this framework, some existing algorithms based on kernel trick can be developed into new HSIC regularization kernel algorithms. As for the subspace learning methods in these algorithms, both RLPP and CDL only consider local geometric features and ignore the discriminative information. This paper proposes HRGDA based on the HSIC regularization framework. HRGDA combines local geometry and label information to learn the transformational matrix.

### 3. Preliminaries

3.1. *Geometry of SPD Manifolds*. Let  $S^d$  be a  $d \times d$  real symmetric matrix space. A symmetric matrix  $A$  is considered to be positive semidefinite if  $x^T A x \geq 0$  holds for all nonzero vectors, which is denoted as  $A \geq 0$ . Let  $\lambda(X)$  be the eigenvalues of  $A$ ; thus,  $\lambda(X)$  has nonnegative values. This property is derived from the implicit structure of matrix  $A$ . If the eigenvalues of  $A$  are positive, then  $A$  is an SPD matrix. Correspondingly, the inequality  $x^T A x > 0$  holds for any nonzero  $x$ , which is also denoted as  $A > 0$ . The  $d \times d$  real SPD matrix space is denoted as  $S_{++}^d$ , and the space forms an SPD

manifold when endowed with a Riemannian metric. Manifolds are Hausdorff topological spaces with countable basis. For every point, there is an open set neighborhood local homeomorphism to the  $n$ -dimensional vector space. For differentiable manifolds, all tangent vectors at a specific point are included in the tangent space of that point. The inner product is called the Riemannian metric. The norm of a tangent vector  $v$  can be derived from the inner product, i.e.,  $\|v\|_X^2 = \langle v, v \rangle_X$ .

Let  $X_i \in M$  be the point of manifold and  $\vec{X}_i \vec{X}_j$  be a tangent vector of  $X_i$ . Here, there exists exactly one geodesic corresponding to the tangent vector  $\vec{X}_i \vec{X}_j$ . The geodesic connecting  $X_i$  and  $X_j$  is transformed into a straight line, and the distance of the geodesic is equal to the length of the line. The Riemannian distance between  $X_i$  and  $X_j$  on the manifold is obtained using the geodesic from  $X_i$  to  $X_j$ , and this relation is illustrated in Figure 2.

Furthermore, the exponential map maps  $\vec{X}_i \vec{X}_j$  to  $X_j$ , i.e.,  $X_j = \exp_{X_i}(\vec{X}_i \vec{X}_j)$ . The inverse operation of  $\exp_X$  is the logarithmic map, i.e.,  $\vec{X}_i \vec{X}_j = \log_{X_i}(X_j)$ . The definition of exponential and logarithmic maps is given as follows:

$$\begin{aligned} \exp_{X_i}(v) &= X_i^{(1/2)} \exp(X_i^{-(1/2)} v X_i^{-(1/2)}) X_i^{(1/2)}, \\ \log_{X_i}(X_j) &= X_i^{(1/2)} \log(X_i^{-(1/2)} X_j X_i^{-(1/2)}) X_i^{(1/2)}. \end{aligned} \quad (8)$$

In the symmetric matrix case, the exponential and logarithmic maps can be computed using the eigenvalue decomposition. The symmetric matrix  $\Sigma$  can be decomposed as  $\Sigma = U \Lambda U^T$ . Thus,  $\exp_X$  and  $\log_X$  can be, respectively, computed as follows:

$$\begin{aligned} \exp(\Sigma) &= U \text{DIAG}(\exp(d_i)) U^T, \\ \log(\Sigma) &= U \text{DIAG}(\log(d_i)) U^T. \end{aligned} \quad (9)$$

**3.2. Riemannian Metrics.** The geodesic distance is the most common distance measure for two SPD matrices. The affine-invariant distance defined by AIRM [20] can be obtained as follows:

$$D_{AI}(X_i, X_j) := \left\| \log(X_i^{-(1/2)} X_j X_i^{-(1/2)}) \right\|_F, \quad (10)$$

where  $\|\cdot\|_F$  is the Frobenius norm. This metric inherits the characters of invariant distance; however, it exhibits high time complexity when practically implemented.

Another approach for measuring the geodesic distance is LERM [21]. The log-Euclidean distance can be defined as

$$D_{LE}(X_i, X_j) := \left\| \log(X_i) - \log(X_j) \right\|_F. \quad (11)$$

The log-Euclidean distance is close to the actual geodesic distance and is easier to be computed than AIRM. However, the computational cost can be high for applications with numerous input matrices owing to matrix logarithms.

Driven by concerns regarding simple calculations, the matrix divergence is a fast and approximate candidate of the geodesic distance. Maher et al. introduced Jeffreys Kullback–Leibler divergence (JKLD) as

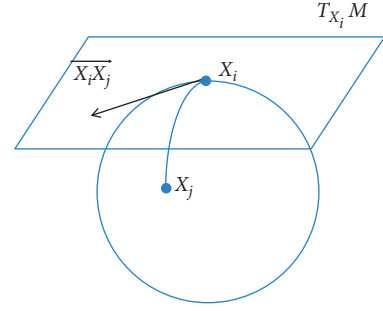


FIGURE 2: Illustration of tangent space  $T_{X_i} M$  at point  $X_i$ .

$$D_{Jkl}(X_i, X_j) := \left( \frac{1}{2} \text{Tr}(X^{-1}Y + Y^{-1}X - 2I) \right)^{(1/2)}, \quad (12)$$

where  $\text{Tr}(\cdot)$  is the matrix trace. This measure is fast; however, it can overestimate the geodesic distance.

Recently, the JBLD has been discussed as a proxy for the Riemannian distance. The JBLD is defined as follows:

$$D_{l,d}(X_i, X_j) = \left( \log \left| \frac{X_i + X_j}{2} \right| - \frac{1}{2} \log |X_i X_j| \right)^{1/2}, \quad (13)$$

where  $|\cdot|$  is the matrix determinant. This divergence is efficient because it does not require matrix logarithms.

## 4. HSIC Regularization

**4.1. HSIC Regularization Kernel Framework.** HSIC [49] is used to characterize the internal connection between two random vectors. The theoretical basis of HSIC is a bit complex. The derivation of HSIC relies on complex mathematical theories such as HS operators [50], cross-covariance operator, mean function [51], and functional analysis. However, HSIC can be calculated by an empiric HSIC. The empiric HSIC can be calculated as

$$\text{HSIC}(X, Y) \approx \frac{1}{N^2} \text{tr}(K_Y C_N K_X C_N), \quad (14)$$

where  $C_N = I_N - (1/N) \Gamma_N \Gamma_N^T$  is the centralizing matrix and  $K_X$  and  $K_Y$  are the kernel matrices of  $X$  and  $Y$ , respectively. We have summarized the relevant theories of HSIC in [44]. For detailed derivation, please refer to our previously published work.

The HSIC of the SPD matrices  $X$  and the low-dimensional representations  $Y$  is denoted by  $\text{HSIC}(X, Y)$ . To facilitate easy calculation, we select the linear kernel as the kernel function of  $Y$ , i.e.,  $k_Y(y', y'') = y'^T y''$ . Then, the kernel matrix  $K_Y$  of  $Y$  can be computed as

$$K_Y = \begin{bmatrix} y_1^T y_1 & \cdots & y_1^T y_N \\ \vdots & \ddots & \vdots \\ y_N^T y_1 & \cdots & y_N^T y_N \end{bmatrix} = Y^T Y. \quad (15)$$

Thus,  $\text{HSIC}(X, Y)$  can be computed as

$$\begin{aligned} \text{HSIC}(X, Y) &= \frac{1}{N^2} \text{tr}(Y^T Y C_N K_X C_N) \\ &= \frac{1}{N^2} \text{tr}(Y C_N K_X C_N Y^T). \end{aligned} \quad (16)$$

Since  $N$  has no relation with  $Y$ , we ignore the coefficient  $(1/N^2)$  in equation (16). We obtain

$$\text{HSIC}(X, Y) = \text{tr}(A K_X C_N K_X C_N K_X^T A^T) = \text{tr}(A L_H A^T), \quad (17)$$

where  $L_H = K_X C_N K_X C_N K_X^T$ .

As described in Figure 1, to address non-Euclidean geometry of SPD manifolds, the traditional kernel-based framework first embeds the input SPD matrices onto a high-dimensional RKHS  $H$  with a predefined kernel and then projects them into a low-dimensional linear subspace. The relationship between low-dimensional projection  $y_i \in R^m$  and SPD matrix  $X_i \in M$  is actually a linear assumption. Under this assumption of the traditional kernel framework, the intrinsic connection of  $y_i$  and  $X_i$  is ignored. The statistical correlation of input data and projective data should be taken into account during transformation. Thus, HSIC of SPD matrices and low-dimensional representations is introduced herein to align the low-dimensional representation with the intrinsic features of input data, where HSIC is typically used to measure the statistical dependence between two datasets. In order to make the proposed kernel framework applicable to the traditional kernel-based methods, this paper proposes to add the HSIC regularization term to equation (1). Then, we have

$$A = \arg \max_A (f(A) + \lambda \text{HSIC}(X, Y)), \quad (18)$$

where  $\lambda$  is a regular term coefficient.

In summary, the objective function is formulated as

$$A = \arg \max_A (f(A) + \lambda \text{HSIC}(X, Y)) = \arg \max_A (A(F + \lambda L_H)A^T), \quad (19)$$

where  $F$  depends on the traditional method.

**4.2. Kernel Pool.** In the calculation of HSIC, the reproducing kernel  $k_Y$  of  $H_Y$  is fixed; however, the kernel function  $k_X$  is alternative. It can be selected from the kernel pool based on specific practical applications. To generate a valid Hilbert space, the kernel function must be SPD [45]. Here, we discuss three alternative kernels.

**4.2.1. Log-Linear Kernel.** The log-linear kernel is generalized using the polynomial kernel in a Euclidean space. It is defined as follows:

$$k_{LL}(X_i, X_j) = \text{tr}(\log(X_i)\log(X_j)). \quad (20)$$

**4.2.2. Log-Gaussian Kernel.** The log-Gaussian kernel is expressed as

$$k_{LE}(X_i, X_j) = \exp\left\{-\frac{D_{LE}^2(X_i, X_j)}{2\sigma^2}\right\}. \quad (21)$$

It replaces the Euclidean distance between  $x_i$  and  $x_j$  with the LERM in the popular Gaussian RBF kernel.

**4.2.3. LogDet Divergence Kernel.** The LogDet divergence kernel is defined as

$$k_{LD}(X_i, X_j) = \exp\{-\beta D_{LD}^2(X_i, X_j)\}. \quad (22)$$

The kernel is a conditionally positive kernel [47], and it is guaranteed to be an SPD kernel for the following:

$$\beta \in \left\{\frac{1}{2}, \frac{2}{2}, \dots, \frac{n-1}{2}\right\} \cup \left\{\tau \in R: \tau > \frac{1}{2}(n-1)\right\}. \quad (23)$$

**4.3. HSIC Regularization Graph Discriminant Analysis.** The HRGDA is proposed based on the HSIC regularization framework. This method uses the LogDet divergence kernel in equation (22) for embedding and a variant of kernel LDA for learning. In kernel-based methods on SPD manifolds, the log-linear and log-Gaussian kernels are commonly used for Hilbert space embedding. Specifically, the log-linear kernel is used in CDL and the log-Gaussian kernel is adopted in KPCA and KSLR. The LogDet divergence kernel used in HRGDA is an effective kernel, since the computation complex of JBLD is lower than LERM and AIRM. The HRGDA seeks for a transformation matrix which maximized the between-class graph matrix and minimized the within-class graph matrix. The graph discriminant analysis is defined as follows:

$$A = \arg \max_A \frac{A S_B A^T}{A S_W A^T}, \quad (24)$$

where  $A$  is the transformational matrix and  $S_B$  and  $S_W$  are the between-class and within-class graph matrices, respectively.

The within-class graph matrix  $S_W$  is defined from the following function:

$$\min \sum_{i,j} (y_i - y_j)^2 W_{ij}. \quad (25)$$

The adjacency graph  $W_{ij}$  contains local geometry and is defined as

$$W_{ij} = \begin{cases} \exp\left(-\frac{d_{i,d}^2(X_i, X_j)}{2\sigma^2}\right), & \text{if class label } l_i = l_j, \\ 0, & \text{else.} \end{cases} \quad (26)$$

By substituting equation (26) into equation (25), we obtain the following:



$$\begin{aligned}
\sum_{i,j} (y_i - y_j)^2 W_{ij} &= \sum_{i,j} (y_i^2 + y_j^2 - 2y_i y_j) W_{ij} \\
&= \left( \sum_i y_i^2 D_{ii}^W + \sum_j y_j^2 D_{jj}^W - 2 \sum_{i,j} y_i y_j W_{ij} \right) \\
&= 2Y(D^W - W)Y^T = 2AK(D^W - W)K^T A^T,
\end{aligned} \tag{27}$$

where the element on the diagonal of  $D^W$  is row sums of  $A$ ,  $D_{ii}^W = \sum_j W_{ij}$ . Thus,  $S_W$  can be defined as

$$S_W = K(D^W - W)K^T. \tag{28}$$

The between-class scatter  $S_B$  is defined from

$$\max \sum_{p=1, q=1}^c (m_p - m_q)^2 B_{pq}, \tag{29}$$

where  $c$  is the number of classes,  $m_p$  is the center of the  $p$ -th class, and  $B_{pq} = \exp(-d_{i_a}^2(m_p, m_q)/2\sigma^2)$ .

Similarly,  $S_B$  can be defined as

$$S_B = K(D^B - B)K^T, \tag{30}$$

where  $D^B$  is a diagonal matrix and  $D_{pp}^B = \sum_q B_{pq}$ .

To sum up, the HRGDA can be formulated as

$$\begin{aligned}
A &= \arg \max_A \left( \frac{AS_B A^T}{AS_W A^T} + \lambda AL_H A^T \right) \\
&= \arg \max_A A \left( \frac{S_B}{S_W} + \lambda L_H \right) A^T.
\end{aligned} \tag{31}$$

The optimal problem can be solved through eigenvalue decomposition.

**4.4. Computational Complexity.** The time complexity of HSIC regularization kernel framework contains three main parts: (1) calculating the learning function, i.e.,  $f(A)$ ; (2) calculating the HSIC, i.e.,  $\text{HSIC}(X, Y)$ ; and (3) conducting eigenvalue decomposition of optimization problem.

The computational complexity for calculating the learning function  $f(A)$  is the same as the traditional methods. Besides, the cost of eigenvalue decomposition is  $O(N^3)$ . Thus, the proposed framework brings no additional calculations of parts (1) and (3). Part (2) is an additional calculation. It can be seen from equation (17) that the computational complexity of  $\text{HSIC}(X, Y)$  is mainly in the calculation of kernel matrix  $K_X$ . The kernel pool provides three useful kernel functions. The computational complexity of log-linear, log-Gaussian, and LogDet divergence kernel is  $O(N(d^3 + 2d^2))$ ,  $O((N(N+1)/2)(d^3 + 2d^2))$ , and  $O(Nd(N+1)(d+1))$ , respectively. Among the three kernel functions, the LogDet divergence kernel has the lowest computational complexity. The reason is that the computational complexity of matrix determinant is lower than matrix logarithm.

## 5. Experiments

The effectiveness of HSIC regularization framework is testified in the experiments. Here, we consider four widely used datasets for performing visual recognition tasks, i.e., the QMUL [52], FERET [53], COIL-20 [54], and ETH80 [55] datasets.

**5.1. Datasets and Settings.** The QMUL dataset comprises a set of 20,005 images of human heads captured by using cameras at an airport terminal. This dataset is split into different sets according to the direction of the head, namely, “back,” “front,” “left,” “right,” and “background.” Sample images are presented in Figure 3. The images are divided into training and testing sets beforehand. To compute the region covariance descriptor of each image, the feature vector is expressed as follows:

$$F(x, y) = \left[ I_L, I_a, I_b, \sqrt{I_x^2 + I_y^2}, \arctan\left(\frac{|I_x|}{|I_y|}\right), G_1, \dots, G_8 \right], \tag{32}$$

where  $I_L, I_a$ , and  $I_b$  are the values of the CIELAB color space,  $I_x$  and  $I_y$  are the first-order gradients of  $I_L$ , respectively, and  $G_i$  is the response of eight difference-of-Gaussians filters. We randomly select 200 and 100 images of each class as training and testing data, respectively.

We choose the “b” subset of the FERET dataset for face recognition experiments. The FERET contains 2000 face images of 200 people. The size of pictures is  $64 \times 64$  pixels. The training set comprises images of “ba,” “bc,” “bh,” and “bk” classes. The remaining constitutes the test data. The feature vector is expressed as  $F(x, y) = [x, y, I, |G_{00}|, \dots, |G_{47}|]$ , where  $I$  is the gray scale and  $G_{uv}$  are the response values of Gabor filter. The values of  $u$  and  $v$  are 0–4 and 0–7, respectively.

The COIL-20 dataset consists of 20 objects, each comprises 72 pictures with a size of  $128 \times 128$  pixels. The sample images are presented in Figure 4. Gray scale, first- and second-order gradients are used to compute the region covariance descriptor. Hence, the region covariance descriptor of an image is a  $5 \times 5$  matrix. 10 images are randomly selected to the training data, and the remaining images are the testing data.

ETH80 is an object dataset, including images of apples, pears, cars, and dogs. The dataset contains a total of 410 images from 10 instances. The size of images in ETH80 is  $128 \times 128$  pixels (Figure 5). For the region covariance descriptor, we extract the following features:

$$F(x, y) = [x, y, R, G, B, I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|], \tag{33}$$

where  $R, G$ , and  $B$  represent red, green, and blue color scale, respectively,  $I$  is the gray scale, and  $|I_x|, |I_y|, |I_{xx}|$ , and  $|I_{yy}|$  are the first- and second-order gradients of gray scale. The dimensionality of region covariance descriptor is  $10 \times 10$ . In

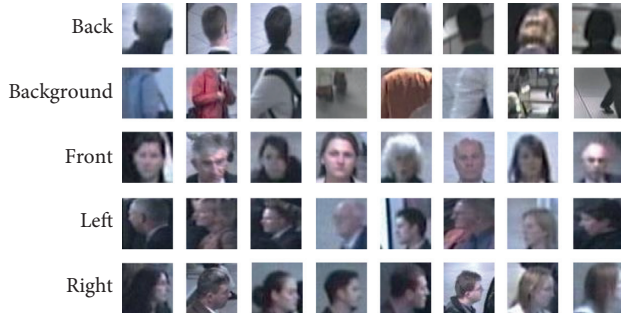


FIGURE 3: Example images from the QMUL dataset.



FIGURE 4: Example images from the COIL-20 dataset.

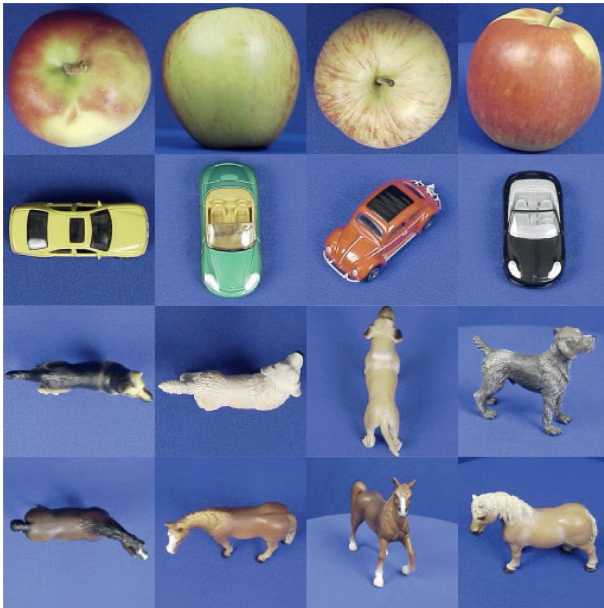


FIGURE 5: Example images from the ETH80 dataset.

each object, half of the instances are randomly selected for training and the remaining are used for testing. 100 images are randomly selected for each instance.

5.2. *Compared Methods.* To verify the performance of the HRGDA and HSIC regularization framework, RLPP [45], KSLR [19], and CDL [8] were combined with the proposed HSIC regularization, denoted as RLPP-HR, KSLR-HR, and CDL-HR, respectively. The kernel used in HSIC ( $X, Y$ ) is the log-Gaussian kernel. The methods for HSIC regularization are compared with several recognition methods on SPD manifolds, namely, HSIC-SL [44], KPCA [46], RSR [47], TSC [2], Riem-DLSC [22], logEuc-SC [32], and KLRM-DL [48]. All the parameters of the compared algorithms are set based on the recommendation provided in the corresponding literature. The kernel function in HSIC-SL is log-Gaussian kernel. The recognition accuracies of the QMUL and FERET datasets are given in Tables 1 and 2, respectively.

5.3. *Comparison of Kernels in HSIC Regularization.* In this experiment, three kernel functions are used to compute HSIC regularization, namely, the log-linear, log-Gaussian, and LogDet divergence kernel; here, the HSIC regularization is presented as HR (log-linear), HR (log-Gaussian), and HR (LogDet divergence), respectively. Table 3 lists the recognition results of different kernels on the COIL-20 and ETH80 datasets.

5.4. *Discussion.* The detailed classification results of all methods on four image datasets (i.e., QMUL, FERET, COIL-20, and ETH80 datasets) are presented in Tables 1–3. The discussion was presented as follows:

- (1) To show that the proposed HSIC regularization kernel framework improves the effectiveness of traditional kernel framework, we compare the classification accuracy of the HSIC regularization kernel methods with that of the traditional kernel methods. RLPP-HR, CDL-HR, and KSLR-HR are the HSIC regularization kernel methods corresponding to RLPP, CDL, and KSLR, respectively. As shown in Tables 1 and 2, the classification accuracy of RLPP-HR is higher than that of RLPP. Similarly, CDL-HR and KSLR-HR achieve better classification accuracy than CDL and KSLR, respectively. This conclusion is more obvious in Table 3. Irrespective of the type of kernel function employed in HSIC, the classification accuracy of the HSIC regularization kernel methods is better than that of the traditional methods. These results indicate that HSIC regularization considerably improves the performance of traditional algorithms. Moreover, the proposed HSIC regularization kernel framework is superior to the traditional kernel framework because the former considers the intrinsic connection of SPD matrices and low-dimensional projections.
- (2) Based on the proposed HSIC regularization kernel framework, this study presents a new method called HRGDA. As shown in Table 1, the recognition accuracy of HRGDA is the best on the QMUL dataset. Table 2 presents the performances of all methods on FERET. The recognition accuracy of HRGDA on

TABLE 1: Classification accuracy on the QMUL dataset.

Method	Accuracy (%)
KPCA	42.5
logEuc-SC	66.3
RSR	73.2
TSC	61.7
Riem-DLSC	36.6
KLRM-DL	70.74
HSIC-SL	76.22
CDL	76
CDL-HR	77.4
RLPP	58.4
RLPP-HR	60.4
KSLR	77.4
KSLR-HR	80
HRGDA	<b>80.8</b>

TABLE 2: Classification accuracy on the FERET dataset.

Methods	bd	be	bf	bg	Average
logEuc-SC	74.00	94.00	97.50	80.50	86.50
RSR-S	82.50	94.50	<b>98.00</b>	83.50	89.63
RSR-J	79.50	<b>96.50</b>	97.50	86.00	89.88
TSC	36.00	73.00	73.50	44.50	56.75
Riem-DLSC	88.25	93.50	96.50	91.75	92.50
KLRM-DL	<b>89.50</b>	96.00	97.00	94.00	<b>94.13</b>
HSIC-SL	88.00	88.50	95.00	93.00	91.13
CDL	76.50	75.00	88.50	84.50	81.13
CDL-HR	81.5	83	95	90.5	87.5
RLPP	58.40	60.00	67.00	60.50	61.48
RLPP-HR	63.50	62.00	74.50	71.50	67.88
KSLR	83.00	90.00	96.00	91.00	90.00
KSLR-HR	86.00	90.00	97.50	92.00	91.38
HRGDA	<b>89.50</b>	93.00	96.50	<b>94.50</b>	93.38

TABLE 3: Classification accuracy (%) of different kernels.

Method	COIL-20	ETH80
KPCA	81.05	72.61
logEuc-SC	89.76	73.75
RSR	93.95	77.74
TSC	80.16	65.88
Riem-DLSC	87.74	75.63
KLRM-DL	97.33	80.13
HSIC-SL	96.87	82.40
RLPP	85.89	74.08
RLPP-HR (log-linear)	87.58	77.88
RLPP-HR (log-Gaussian)	88.55	77.25
RLPP-HR (LogDet divergence)	87.58	74.38
CDL	94.54	79.92
CDL-HR (log-linear)	97.58	82.63
CDL-HR (log-Gaussian)	97.26	81.38
CDL-HR (LogDet divergence)	97.18	82.5
KSLR	96.24	81.66
KSLR-HR (log-linear)	97.98	<b>85</b>
KSLR-HR (log-Gaussian)	<b>98.87</b>	84.5
KSLR-HR (LogDet divergence)	97.82	84.88
HRGDA	<b>98.87</b>	84.88

“bd” and “bg” is higher than those of the other methods; however, in the case of average accuracy, the proposed HRGDA is the second. The reason

might be that FERET is a face recognition dataset which contains subtle features. The HRGDA performs slightly worse when classifying datasets comprising subtle features. Table 3 gives the classification accuracy of all methods for the COIL-20 and ETH80 datasets. The HRGDA method is superior to the compared methods for the COIL-20 dataset. In the case of ETH80, the performance of HRGDA is slightly worse than that of KSLR-HR (log-linear). It is worth noting that the KSLR-HR (log-linear) is a new method generated from the proposed HSIC regularization kernel framework. In other words, the HRGDA is still better than the traditional methods for the ETH80 dataset. Among the four classification experiments, the classification accuracy of HRGDA is the best in the case of QMUL, COIL-20, and ETH80 datasets and the second in the case of FERET. In general, the HRGDA is indeed an excellent algorithm on SPD manifolds.

- (3) We compare the effectiveness of the kernel functions for computing HSIC in Table 3. It is shown that the selection of the kernel function affects the performance of HSIC regularization. The performance of different kernels is diverse across different datasets. However, irrespective of the type of kernel functions employed in HSIC, the classification accuracy increases by 2–8% on the basis of the traditional methods. User can compare the result of different kernels and select the most appropriate one.

Overall, we consider that the experimental results verify the proposed HSIC regularization framework as an effective framework for kernel methods on SPD manifolds. Also, the proposed HRGDA is an accurate and valid learning method of SPD manifolds.

## 6. Conclusions

Herein, we propose an HSIC regularization kernel learning framework to improve traditional kernel framework on SPD manifolds by introducing the HSIC of SPD matrices and low-dimensional projections. Traditional kernel framework neglects the connection of SPD matrices and linear projections. To solve this problem, the proposed framework uses HSIC to measure the statistical correlation between SPD matrices and projections. The proposed framework can be applied to some specific forms of kernel methods on SPD manifolds, such as RLPP, CDL, and KSLR. Moreover, HSIC regularization consistently improves the classification accuracy of the traditional methods. To improve the applicable scenarios of this framework, we investigate different kernel functions to calculate the HSIC, i.e., log-linear, log-Gaussian, and LogDet divergence kernels. Additionally, we propose a method on the basis of the HSIC regularization kernel framework. Experiments demonstrate that the HSIC regularization consistently improves the classification accuracy of the traditional algorithms. We believe that the finding of this work can provide important contributions to the development of kernel methods on SPD manifolds.



Furthermore, the proposed HRGDA is also found to be an effective method on SPD manifolds.

However, there are still several deficiencies in applications. The performance of HRGDA is slightly worse than that of other methods in the classification of subtle texture. We will develop additional kernel functions on SPD manifolds for this framework because having access to diverse kernel functions will increase the flexibility and applicability of HSIC regularization.

## 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 they have no conflicts of interest.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China through the project “Research on Nonlinear Alignment Algorithm of Local Coordinates in Manifold Learning” under grant no. 61773022, the Character and Innovation Project of Education Department of Guangdong Province under grant no. 2018GKTSCX081, the Young Innovative Talents Project of Education Department of Guangdong Province under grant no. 2020KQNCX191, the Guangzhou Science and Technology Plan Project of Bureau of Science and Technology of Guangzhou Municipality under grant no. 202102020700, and the Educational Big Data Enterprise Lab of Guangzhou Panyu Polytechnic under grant no. 2021XQS05.

## References

- [1] J. Chen, Z. Ma, and Y. Liu, “Local coordinates alignment with global preservation for dimensionality reduction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 106–117, 2013.
- [2] J. Yu, Z. Qin, T. Wan, and X. Zhang, “Feature integration analysis of bag-of-features model for image retrieval,” *Neurocomputing*, vol. 120, pp. 355–364, 2013.
- [3] J. Yu, X. Yang, F. Gao, and D. Tao, “Deep multimodal distance metric learning using click constraints for image ranking,” *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4014–4024, 2017.
- [4] B. Li, J. Li, and X.-P. Zhang, “Nonparametric discriminant multi-manifold learning for dimensionality reduction,” *Neurocomputing*, vol. 152, no. 25, pp. 121–126, 2015.
- [5] X. Liu and Z. Ma, “Discriminant analysis with local Gaussian similarity preserving for feature extraction,” *Neural Processing Letters*, vol. 47, no. 1, pp. 39–55, 2018.
- [6] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Geometric means in a novel vector space structure on symmetric positive definite matrices,” *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 1, pp. 328–347, 2006.
- [7] F. Porikli, O. Tuzel, and P. Meer, “Covariance tracking using model update based on Lie algebra,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2006.
- [8] R. Wang, H. Guo, L. S. Davis, and Q. Dai, “Covariance discriminative learning: a natural and efficient approach to image set classification,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2496–2503, Providence, RI, USA, June 2012.
- [9] O. Tuzel, F. Porikli, and P. Meer, “Region covariance: a fast descriptor for detection and classification,” *Computer Vision—ECCV 2006*, Springer, vol. 3952, pp. 589–600, Berlin, Germany, 2006.
- [10] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, “Discriminant analysis on riemannian manifold of Gaussian distributions for face recognition with image sets,” *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 27, no. 1, pp. 151–163, 2018.
- [11] I. Dryden, A. Koloydenko, and D. Zhou, “Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging,” *Annals of Applied Statistics*, vol. 3, no. 3, pp. 1102–1123, 2009.
- [12] D. Le Bihan, J.-F. O. Mangin, C. Poupon et al., “Diffusion tensor imaging: concepts and applications,” *Journal of Magnetic Resonance Imaging*, vol. 13, no. 4, pp. 534–546, 2001.
- [13] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista, “A nonparametric Riemannian framework on tensor field with application to foreground segmentation,” in *Proceedings of the 2011 International Conference on Computer Vision*, vol. 45, no. 11, pp. 1–8, Barcelona, Spain, November 2011.
- [14] A. Thomason and J. Gregor, “Higher order singular value decomposition of tensors for fusion of registered images,” *Journal of Electronic Imaging*, vol. 20, no. 1, pp. 13–23, 2011.
- [15] T. Hazan, S. Polak, and A. Shashua, “Sparse image coding using a 3D non-negative tensor factorization,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05)*, vol. 1, pp. 50–57, Beijing, China, October 2005.
- [16] H. Tan and Y. Gao, “Patch-based principal covariance discriminative learning for image set classification,” *IEEE Access*, vol. 5, pp. 15001–15012, 2017.
- [17] A. Ognjen, S. Gregory, F. John, C. Roberto, and D. Trevor, “Face recognition with image sets using manifold density divergence,” in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA, June 2005.
- [18] M. Lovrić, M. Min-Oo, and E. A. Ruh, “Multivariate normal distributions parametrized as a Riemannian symmetric space,” *Journal of Multivariate Analysis*, vol. 74, no. 1, pp. 36–48, 2000.
- [19] X. Liu and Z. Ma, “Kernel-based subspace learning on Riemannian manifolds for visual recognition,” *Neural Processing Letters*, vol. 51, pp. 147–165, 2019.
- [20] X. Pennec, P. Fillard, and N. Ayache, “A Riemannian framework for tensor computing,” *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.
- [21] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Log-euclidean metrics for fast and simple calculus on diffusion tensors,” *Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, 2006.
- [22] A. Cherian and S. Sra, “Riemannian dictionary learning and sparse coding for positive definite matrices,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 12, pp. 2859–2871, 2017.
- [23] D. Tosato, M. Farenzena, M. Cristani, M. Spera, and V. Murino, “Multi-class classification on Riemannian manifolds for video surveillance,” *Computer Vision—ECCV 2010*, Springer, Berlin, Switzerland, 2010.

- [24] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," *Computer Vision—ECCV 2012*, Springer, Berlin, Switzerland, 2012.
- [25] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Tensor sparse coding for positive definite matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 592–605, 2014.
- [26] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric means," in *Proceedings of the 25th International Conference on Neural Information Processing Systems—NIPS'12*, vol. 1, pp. 144–152, Lake Tahoe, NV, USA, December 2012.
- [27] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, "Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2161–2174, 2013.
- [28] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 505–512, Pittsburgh, PA USA, June 2006.
- [29] A. Goh and R. Vidal, "Clustering and dimensionality reduction on Riemannian manifolds," in *Proceedings of the 2008 IEEE Conference on Computer Vision & Pattern Recognition*, pp. 626–632, Anchorage, AK, USA, June 2008.
- [30] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.
- [31] C. Rui, P. Martins, J. F. Henriques, F. S. Leite, and J. Batista, "Rolling Riemannian manifolds to solve the multi-class classification problem," in *Proceedings of the 2013 IEEE Conference on Computer Vision & Pattern Recognition*, vol. 9, no. 4, pp. 41–48, Portland, OR, USA, June 2013.
- [32] K. Kai Guo, P. Ishwar, and J. Konrad, "Action recognition from video using feature covariance matrices," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2479–2494, 2013.
- [33] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: a kernel approach," *Computer Vision—ECCV 2012*, Springer, vol. 7573, no. 2, pp. 216–229, Berlin, Germany, 2012.
- [34] R. Vemulapalli, J. K. Pillai, and R. Chellappa, "Kernel learning for extrinsic classification of manifold features," in *Proceedings of the 2013 IEEE Conference on Computer Vision & Pattern Recognition*, vol. 9, no. 4, pp. 1782–1789, Portland, OR, USA, June 2013.
- [35] Z. Huang, R. Wang, S. Shan, and X. Chen, "Learning euclidean-to-Riemannian metric for point-to-set classification," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1677–1684, Columbus, OH, USA, June 2014.
- [36] Z. Huang, R. Wang, S. Shan, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 720–729, Lille, France, July 2015.
- [37] M. J. Gangeh, H. Zarkoob, and A. Ghodsi, "Fast and scalable feature selection for gene expression data using hilbert-schmidt independence criterion," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 167–181, 2017.
- [38] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature selection via dependence maximization," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1393–1434, 2012.
- [39] G. Camps-Va, J. Mooij, and B. Schölkopf, "Remote sensing feature selection by kernel dependence measures," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 3, pp. 587–591, 2010.
- [40] M. J. Gangeh, A. Ghodsi, and M. S. Kamel, "Kernelized supervised dictionary learning," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4753–4767, 2013.
- [41] M. J. Gangeh, P. Fewzee, A. Ghodsi, M. S. Kamel, and F. Karray, "Multiview supervised dictionary learning in speech emotion recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1056–1068, 2014.
- [42] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Zolghadri Jahromi, "Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds," *Pattern Recognition*, vol. 44, no. 7, pp. 1357–1371, 2011.
- [43] Z. Ma, Z. Zhan, X. Ouyang, and X. Su, "Nonlinear dimensionality reduction based on HSIC maximization," *IEEE Access*, vol. 6, pp. 55537–55555, 2018.
- [44] X. Liu, P. Yang, Z. Zhan, and Z. Ma, "Hilbert-Schmidt independence criterion subspace learning on hybrid region covariance descriptor for image classification," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6663710, 15 pages, 2021.
- [45] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell, "Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures," in *Proceedings of the IEEE Workshop on the Applications of Computer Vision*, pp. 433–439, Breckenridge, CO, USA, January 2012.
- [46] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on Riemannian manifolds with Gaussian RBF kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2464–2477, 2015.
- [47] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson, "Sparse coding on symmetric positive definite manifolds using Bregman divergences," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 6, pp. 1294–1306, 2016.
- [48] R. Zhuang, Z. Ma, W. Feng, and Y. Lin, "SPD data dictionary learning based on kernel learning and Riemannian metric," *IEEE Access*, vol. 8, pp. 61956–61972, 2020.
- [49] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," *Lecture Notes in Computer Science*, Springer, Berlin, Switzerland, pp. 63–77, 2005.
- [50] I. Gohberg, S. Goldberg, and M. A. Kaashoek, "Hilbert-Schmidt operators," *Classes of Linear Operators*, Springer, Berlin, Switzerland, pp. 138–147, 1990.
- [51] E. Kreyszig, *Introductory Functional Analysis with Applications*, Vol. 1, Wiley, New York, NY, USA, 1978.
- [52] D. Tosato, M. Spera, M. Cristani, and V. Murino, "Characterizing humans on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1972–1984, 2013.
- [53] P. J. Phillips, H. Hyeonjoon Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

- [54] S. Nene, S. Nayar, and H. Murase, "Columbia object image library (COIL-20)," An Academic Publisher, Cambridge, MA, USA, CUCS-005-96, 1996.
- [55] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, no. 2, pp. 409–415, Madison, WI, USA, June 2003.

## Research Article

# Simple and Ingenious Mobile Botnet Covert Network Based on Adjustable Unit (SIMBAIDU)

Min-Hao Wu <sup>1</sup>, Chia-Hao Lee <sup>2</sup>, Fu-Hau Hsu,<sup>2</sup> Kai-Wei Chang <sup>2</sup>,  
Tsung-Huang Huang,<sup>3</sup> Ting-Cheng Chang,<sup>1</sup> and Li-Min Yi<sup>1</sup>

<sup>1</sup>College of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou, Guangdong 511483, China

<sup>2</sup>Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan

<sup>3</sup>Green Energy and Environment Research Laboratories (GEL), Industrial Technology Research Institute (ITRI), Chutung, Taiwan

Correspondence should be addressed to Kai-Wei Chang; [popdata520@gmail.com](mailto:popdata520@gmail.com)

Received 19 March 2021; Revised 9 July 2021; Accepted 26 July 2021; Published 4 August 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Min-Hao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Various services through smartphones or personal computers have become common nowadays. Accordingly, embedded malware is rapidly increasing. The malware is infiltrated by using short message service (SMS), wireless networks, and random calling and makes smartphones bots in botnets. Therefore, in a system without an appropriate deterrent, smartphones are infiltrated easily. In the security threats by malware, random calling has become serious nowadays. To develop the defensive system against random calling and prevent the infiltration of the malware through random calling, it is required to understand the exact process of how to make bots in the botnet. Thus, this research develops a simple and ingenious mobile botnet covert network based on adjustable ID units (SIMBAIDU) to investigate how a botnet network is established by using phone numbers. Perfect octave coding (P8 coding) turns out to be effective in infiltrating smartphones and executing commands, which is used for botnets. The results provide the basic process of P8 coding which is useful for developing defensive systems of smartphones.

## 1. Introduction

The information era has brought many conveniences into people's lives. The internet allows fast information transmission that makes the way of communication significantly different from the past. However, the increasing use of smartphones or personal computers for such purposes also causes malware embedding increasingly. As the smartphone combines the properties of computers and telephones and is always connected to the network, personal and financial information in them is vulnerable. Thus, a botnet easily threatens security as involving substantial economic loss [1]. It remotely controls the operating system (OS) by back-dooring the system and embedding malware, which allows cyberattackers to steal any information from the system. The botnet is different from traditional computer viruses as it is always hidden. Therefore, users do not recognize that their systems are included in the botnet. Malware such as Trojan horse backdoors on the device and configures it to autostart

without the user's acknowledgment. The botnet is easily infiltrated into smartphones as making them the bots for control. The obscurity and complexity of the botnet are critical issues as it bypasses the phone's commands in their OSs.

Many researchers have tried to find a way to protect from botnet infiltration and its cyberattacks with the aims of reinforcing security and preventing malicious attacks [2]. Earlier, the research focused on how to detect the botnet [3] using a system such as a honeypot. Later on, an additional passive system was added for web traffic diagnostics [4]. These technologies were based on several different types such as signature-, anomaly-, DNS-, and mining-based approaches.

In the honeypot, the higher the ability to attract malicious attacks, the more efficient to collect information and find new episodes. Whether low or high interactivity, the honeypot increases the probability of being attacked as much as possible and then collects the expected information.



Several honeypots comprise a whole honeynet that spreads the honeynet spots and integrates the honeypot data for malicious behavior analysis. The honeypot help understands the techniques and features of the botnet. However, it usually takes a long time. The web traffic surveillance pinpoints the botnet's existence in the passive monitoring and the analysis of network traffic. However, the mining-based diagnostic [4] only spots well-known botnets; signature-, anomaly-, and DNS-based diagnostics detect the bots or botnets that have not been discovered before. When an attacker modifies the network architecture or protocol into a botnet, the users can use DNS and mining-based diagnostic detection technology to detect botnets. These diagnostics make it easy to identify botnets, regardless of whether the attacker is adjusting its process or designing [4] the potential for information security in both methods.

The botnet diagnostics are based on the peer-to-peer (P2P) or hybrid system [5]. As it processes Internet relay chat (IRC) and hypertext transfer protocol (HTTP), P2P is regarded to be relatively good for the detection of botnet infiltration [6]. However, due to the limitation of the power and memory of the smartphone, antivirus software is not appropriate for botnet diagnostics of the smartphone. As smartphones are using various applications, the botnet causes severer security problems for the smartphone than any other operating system [7]. Therefore, there has been much research on the mobile botnet. How to detect and prevent the related cyberattacks, the new covert channel, and the botnet control mode are critical issues nowadays. However, as the mobile botnet is originated from the traditional botnet of personal computers, its details have not been discussed considerably [8–11].

Zeng et al. [12] showed that the botnet is more vigorous on personal computers than the mobile devices. For the mobile botnet, technologies related to Bluetooth, short message service (SMS), and commands and control (C&C) are required to consider [8, 10, 13–15]. Recent research has been focused on the Android botnet such as DroidDream [16] as the use of SMS as a covert channel in the mobile botnet was found. To prevent such incidents, modifying caller ID numbers is recommended, which solves the encountered problems on many mobile OSs. As the caller ID numbers keep changing, the attacker cannot decide which data or command to transmit. They also need to start networking services whenever trying to manipulate the callee. As long as the victim's smartphone is turned on and infected by malware, the attacker uses caller ID numbers for including them as bots without the limits of cost and location. However, when the caller ID numbers keep changing, the smartphone becomes self-protecting and concealment. Despite the easiness and effectiveness, there are not many research results on the use of the caller ID number for preventing mobile botnet infiltration. Thus, this paper focuses on hidden communication and the control model on the smartphone with the following research objectives.

We aim to propose a hidden communication model by using caller IDs to transmit the binary file, a simple and ingenious mobile botnet covert network based on

adjustable ID units (SIMBAIDU). This applies to any platform regardless of its operating system. Through the test of the proposed model on Windows Mobile 6 (WM6), we also intend to use caller IDs to send binary executable files, which proves that using caller IDs is used for preventing covert channel controls. As the maximum number of digits of caller IDs is 15 according to the E.164 standard, the experiment results provide a compression method in encoding transferring files and the permutations and combinations of the caller IDs. The model enables sending commands at different times by using changing caller IDs. The proposed model prevents the infiltration of the mobile botnet, which is applied to the networks such as general packet radio service (GPRS), 4 G, 5 G, and Wi-Fi for the concealed communication by modifying the caller IDs.

## 2. Proposed System and Simulation

*2.1. System Structure.* The structure of the proposed system is shown in Figure 1. The botmaster or botherder is the master and manipulator of a botnet. The victim is the person with an infiltrated mobile device by malware. Generally, the attacker makes the bot program that runs automatically after each reboot and is hidden by the bot program. The proposed system uses P8 coding for changing the original caller IDs. The caller mechanism here assumes the attacker's infiltration and spoofing by using the caller IDs. Any command or binary execution file is sent through different phone numbers (caller IDs) and caller ID modification. In the proposed model, we cooperated with a telecommunication operator to change the caller IDs.

*2.2. Operation of Malware.* In hidden communication, the attacker downloads and sets up malware through social engineering as unnoticed by the user. The attacker uses malicious code for calling which sends binary files and then issues a command to execute calling. This is a typical process of making a botnet. If the malware gives an instruction, it executes the corresponding operation. With a part of a binary file, it is hidden in the device. By executing the files, the malware operates the mobile device as the attacker wants. Without knowing, the victim may pay extra charges and start specific programs that delete the data remotely. The attacker turns the victim's device to be a springboard to call others and tries to infiltrate other devices that are connected to the victim's device. This approach has three essential features, so-called, three zeros: zero cost to spend, zero packets to use, and zero chance to be revealed.

In sending binary executable files, Win32 API and the registry keys are used to capture caller IDs. Caller IDs, in general, have phone numbers with up to nine digits, ten different names, and decimal separators. We used phone numbers as caller IDs in this study. As computers commonly use hexadecimal from zero to F, using binary files requires changing caller IDs based on P8 coding. Sending commands and files demands changing phone numbers as the malware uses different arrangements and frequencies of the numbers

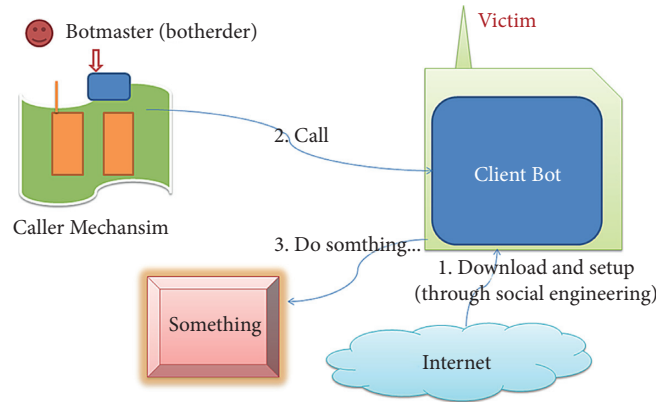


FIGURE 1: The structure of the proposed system.

based on P8 coding. The implementation of the arrangement is a permutation with or without repetition as follows:

$$\text{permutations with repetition : } n^r \text{ commands,} \tag{1}$$

$$\text{permutations without repetition : } P(n, r) = \frac{n!}{(n - r)!} \text{ commands,} \tag{2}$$

where  $n$  is the total number of phone numbers and  $r$  is the number of the phone numbers for finding victims.  $r$  is usually determined when the attacker programs the bot malware. Equations (1) and (2) present the total number of commands that the attacker issues. To reduce the complexity of communications when controlling and avoid annoyance, permutations without repetition are usually used. With permutations without repetition, the attacker uses one command for one device or one command for several devices according to the designed order. For instance, six IDs create 30 available commands ( $6!/4! = 30$ ). Therefore, it is enough to use 2 or 3 numbers to issue commands to control simple malicious behavior.

The history of calls is recorded by the telecommunication carrier as long as the callee answers in Taiwan. Thus, the carrier does not have any record of when the attacker infiltrates without the callee’s answer. Therefore, the attacker then hides the identity, and the infiltrated device as a bot executes any given commands.

### 2.3. Compression and Encoding

**2.3.1. Compression and Decompression.** There are a huge number of applications that implement on the victim’s smartphone through the attacker’s call. The binary executable files are sent in general. Thus, octal is used to code the related program and confirm the loss-less data compression.

In the Perfect Octave Coding (P8 Coding), octal couples with clefs (eight and nine) and a run-length approach. This method is based on encoding numbers into musical notes. In sending, the sender needs to convert the hexadecimal data

into a phone number. As the first step, three bytes of hexadecimal data are read as shown in Figure 2. At the second step, hexadecimal codes are converted into binary codes (Figure 3). Then, the binary codes are packaged into octal codes (Figure 4). Finally, the codes are compressed to obtain the second sequence. To packet the second sequence and to group 15 digits of a phone number, phone numbers are used as caller IDs for calling the victim’s device.

**2.3.2. P8 Coding.** When the malware of the victim’s device receives the decompressed command, its binary executable file does not appear on the device. Its conversion is carried out in the reverse order of the sender’s process. The data compression uses a run-length encoding method that combines octal numbers and the number of conversions to decompress them without error.

A sample of the syntax of P8 coding is similar to music scores. A clef is used as a flag, and key signatures present the number of repetitions. Time signatures represent the name of exact repeating times for coding patterns. Notes are the numbers from zero to seven. This system uses the clef, time signature, notes, and key signatures depending on actual compression requirements. The brackets in Figure 5 showing the syntax of the compression represent the option fields that are optional in compression.

According to the syntax, Figures 6 and 7 show examples of P8 codings. When a clef is eight, the number of repetitions is defined from 5 (one digit) to 99 (two digits). When a clef is 9, the number of repetitions is greater than 99. It continuously squeezes adjacent repeating fingers into six or more numeric characters. As previously described, the number of



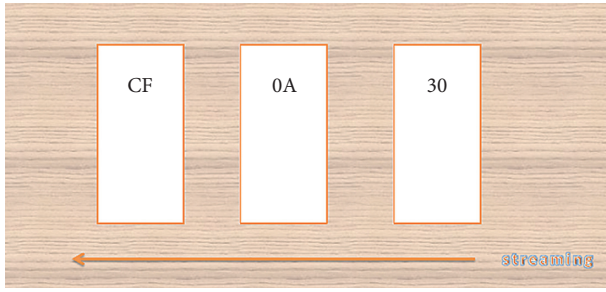


FIGURE 2: The original hexadecimal code.

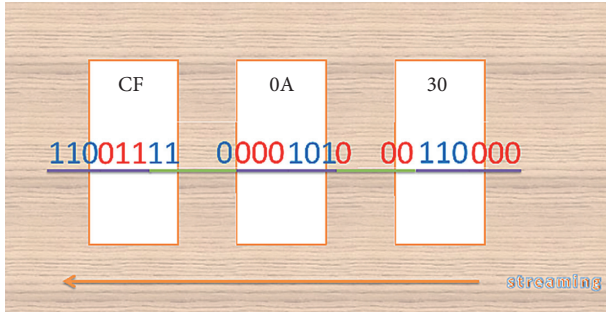


FIGURE 3: Converted binary codes from the hexadecimal codes.

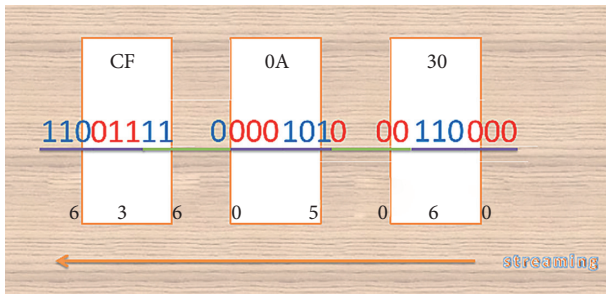


FIGURE 4: The octal codes are converted from the binary codes.

repetitions is greater than 100, a clef digit is 9. When a key signature digit indicates the number of repetitions to be 3, then repeating times are in three digits. For example, in case of “931342,” “9” is for the number of repetitions greater than 100, “3” indicates the number in three digits, “134” is the number, and “2” is for repeating the process twice. In the case of “9410003,” “4” indicates the number in four digits and the repetition occurs 1000 times. As we use octal codes, the used numbers 8 and 9 are the signs for information. If a time signature has only one digit, we need “8” as an end code.

If the location of  $b_{i+1}$  is 8, then  $b_i$  must be 0 to 7, indicating that the number of exact times ( $b_{i-1}$ , time signature) is five to nine. If  $b_{i+1}$  is zero to seven, then  $b_i$  must be zero to nine.  $b_{i-1}$  and  $b_i$  represent ten and nine digits of the exact times.  $b_{i+1}$  represents the number of digits (in Figure 8).

The role of clefs is shown in Figure 9 that represents a mutually exclusive relation. When the exact time is great than 100, it resolves the digit of the number of actual times. Then, that will use the clef of 9 as shown in Figure 7. “8” on

the left is a sign to reveal the meaning of the following digits. The purpose of the design is to reduce the amount of data and for loss-less transmission. “9” as the first digit signals another encoding rule for “key signature” as follows: “9,” “the digits of duplicate times,” “duplicate times,” “octal number.” This is generalized as the following syntax of P8 coding: “clef 9,” “key signature,” “time signature,” and “note.”

Therefore, the syntax needs one more field to indicate the number of digits of exact times (key signature). The exact times from five to nine have a single digit. Then, the file is compressed with a clef of eight or nine. When the adjacent number is seven, and the number of exact duplicate times is nine, as shown in Figure 10. As discussed before, the compressed length by a duplicate time of 8 or 9 is the same. We use a clef of 9 for a duplicate time of over 100 and 8 for that of below 99.

**2.4. Simulation.** The experiment was simulating the syntax on the desktop computer in the Windows 7 operating system. The computer was equipped with a 3.00 GHz AMD Athlon(TM) II X2 250 processor and 2 GB of random access memory (RAM). The binary files corresponding to P8 coding were compressed and analyzed by using several caller IDs. We created a program called “hello-world binary executable file” whose size was 3500 bytes. The file displays the word “hi” in a message box when it modifies or overrides the function of the smartphone or uses another Windows API. With the file, we created the other binary executable file for calling other smartphones. The size of the original file was 7680 bytes. The files were extracted and converted to a binary file by using a mobile device of HP iPAQ. They used specific phone numbers to execute conversion and data-saving commands. Windows Mobile 6 Professional Emulator displayed the particular phone number which they want to release. The files were programmed to appear in a specific dictionary by the attacker and operate.

The experiment included the following steps: (1) mastering several groups of available phone numbers, (2) calling the victim’s smartphone to make it a bot, (3) infiltrating into other smartphones by calling from the smartphone, and (4) running a simple executable file. As calling and infiltrating into random smartphones is illegal, we simulated the process by using Windows Mobile 6 Professional Emulator.

### 3. Results and Discussion

In the simulation, sending the file to the victim’s smartphone numbers from calling to executing the file took about 7.8 s on average. Sending the file to 212 smartphones took 27.5 m to complete the operation. When the file was octal coded, the size of the file was 9559 bytes which was 2.7 times larger than the original file. The compressed file with P8 coding had a reduced size of 3135 bytes in the second sequence. The file compression ratio of the octal coding was 32.8%, while that of the P8 coding was 87.5%. This result reveals the compression ability of P8 coding.

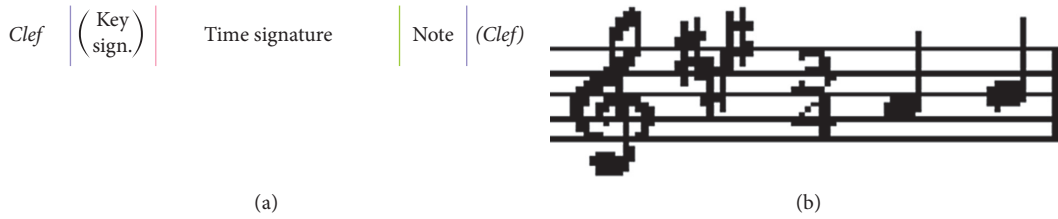


FIGURE 5: The (a) syntax of P8 coding and (b) music notes.

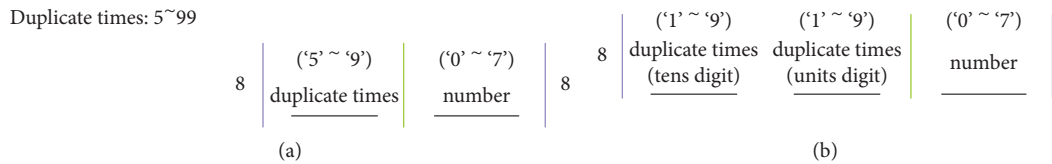


FIGURE 6: Examples of P8 coding with the clef of 8. (a) Duplicate times of 5 to 9. (b) Duplicate times of 10 to 99.

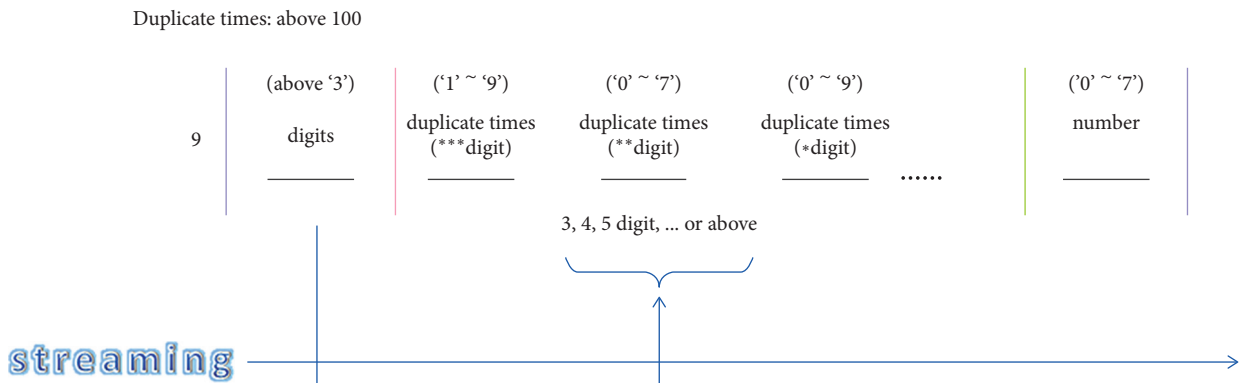


FIGURE 7: Examples of P8 coding with the clef of 9.

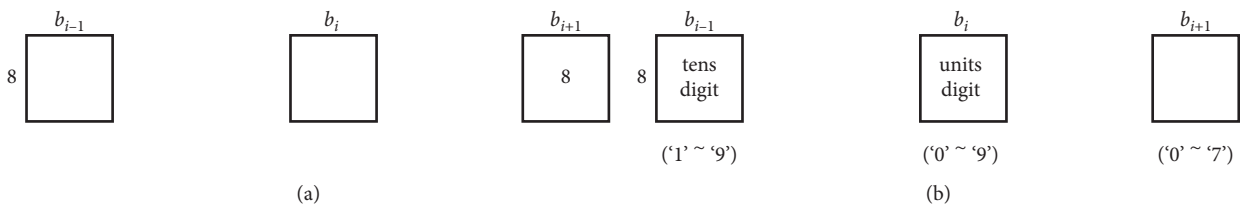


FIGURE 8: Compression format of P8 coding.

(if  $b_{i+1} == '8'$ )  $\Rightarrow$  (Done  $\Rightarrow b_i == '0' \sim '7'$ )  
 (if  $b_{i+1} == '0' \sim '7'$ )  $\Rightarrow$  (Done  $\Rightarrow b_i == '0' \sim '9' \ \&\& \ (b_{i-1}, b_i) = (\text{tens digit}, \text{units digit})$ )

FIGURE 9: Using clefs in P8 coding that is coded by using C++.

In the process, the data need to be expanded to octal numbers first. Since P8 coding compresses the file, it is important to consider the compression ratio in the following process: another customized binary executable of the file function, for example, is calling with some of the related operations. This original file is 7680 bytes. Octal coding increased the size to 20482 bytes. After compression by P8

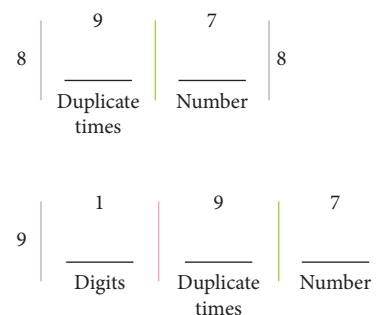


FIGURE 10: The syntax with a duplicate time of a single digit (5 to 9).

coding, it decreased to 13446 bytes. The compression ratio of the P8 coding was 65.6%. When compared to the size of the original file, no compression effect was observed. The file size was increased by 2.7 times by the octal coding when multiplying by eight and dividing by three. The compression process opened the sequence of octal codes rather than the original data in the coding method of this study. Thus, there is an indirect compression of the original file. If an improved compression ratio of the original file is needed, the compression ratio needs to be increased in octal coding. Then, the compression effect is enhanced. Antivirus software such as Aircanner Antivirus for Windows Mobile and Trend Micro Mobile Security Enterprise 5.5 scanning did not detect the files in the mobile device for the experiment.

#### 4. Conclusions

This paper proposes a botnet that regulates caller IDs, especially phone numbers, as a simple and ingenious mobile botnet covert network based on adjustable ID units (SIMBAIDU). SIMBAIDU is the first systematic way to establish malicious attacks. By using perfect octave coding (P8 coding), hexadecimal codes are converted into binary codes and then into octal codes to call and execute a command to control the infiltrated smartphone. After calling the phone numbers stored in the victim's smartphones, a bot program decompresses the caller IDs of the smartphone and sends binary executable files. This process allows manipulating the victims' smartphones remotely and establishing covert channels.

For preventing this approach of establishing the covert channel, the VoIP carriers need to offer the users the right to change their caller IDs frequently. Potential threats by mobile botnets are regarded to be through SMS and wireless networks yet, but this study proves that caller IDs are also a covert channel. The fraudsters from overseas keep trying to call with modified phone numbers and some websites provide a tool for changing them. In this situation, the results of this study show how caller IDs are used as a channel for cyberattacks using the botnet. As P8 coding is used for making botnets, the proposed process provides the basis for suggesting possible defenses to prevent such threats.

#### 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 conflicts of interest.

#### Acknowledgments

The authors thank Guangzhou Panyu Polytechnic Innovation and Entrepreneurship Education Center and Panyu Polytechnic Innovation Team support. The Consortium was funded by the Guangzhou Panyu Polytechnic Innovation and Entrepreneurship Education Center under grant no.

210113263 and Panyu Polytechnic Innovation Team under grant no. 2020CXTD003.

#### References

- [1] M. Schipka, "Dollars for downloading," *Network Security*, vol. 2009, no. 1, pp. 7–11, 2009.
- [2] S. Zander, G. Armitage, and P. Branch, "Covert channels and countermeasures in computer network protocols," *IEEE Communications Magazine*, vol. 45, no. 12, pp. 136–142, 2007.
- [3] W. Lee, C. Wang, and D. Dagon, *Botnet Detection: Countering The Largest Security Threat*, Springer Science & Business Media, Berlin, Germany, 2007.
- [4] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Proceedings of the 2009 Third International Conference On Emerging Security Information, Systems And Technologies*, pp. 268–273, IEEE, Athens, Greece, June 2009.
- [5] S. Chang, L. Zhang, Y. Guan, and T. E. Daniels, "A Framework for p2p botnets," vol. 3, pp. 594–599, in *Proceedings of the 2009 WRI International Conference On Communications And Mobile Computing*, vol. 3, pp. 594–599, IEEE, Kunming, China, January 2009.
- [6] D. Dagon, G. Gu, C. P. Lee, and W. Lee, "A taxonomy of botnet structures," in *Proceedings of the Thirtieth Annual Computer Security Applications Conference (ACSAC 2007)*, pp. 325–339, IEEE, San Juan, PR, USA, December 2007.
- [7] A. Karim, S. A. A. Shah, R. B. Salleh, M. Arif, R. M. Noor, and S. Shamshirband, "Mobile botnet attacks – an emerging threat: classification, review and open issues," *KSII transactions on internet and information systems*, vol. 9, pp. 1471–1492, 2015.
- [8] A. Flo and A. Josang, "Consequences of botnets spreading to mobile devices," in *Short-Paper Proceedings of the 14th Nordic Conference on Secure IT Systems (NordSec 2009)*, pp. 37–43, Oslo, Norway, October 2009.
- [9] M. Ahamad, *Emerging Cyber Threats Report For 2009*, Georgia Tech Information Security Center, Atlanta, GA, USA, 2008.
- [10] C. Mulliner, "Smartphone botnets," in *Proceedings of the 5. GI FG SIDAR Graduierten-Workshop über Reaktive Sicherheit*, p. 10, Nancy, France, October 2010.
- [11] A. Aprville, "Symbian worm Yxes: towards mobile botnets?" *Journal in Computer Virology*, vol. 8, no. 4, pp. 117–131, 2012.
- [12] Y. Zeng, K. G. Shin, and X. Hu, "Design of SMS commanded-and-controlled and P2P-structured mobile botnets," in *Proceedings of the Fifth ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 137–148, Tucson, AZ, USA, April 2012.
- [13] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating bluetooth as a medium for botnet command and control," *Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 6201, pp. 61–80, 2010.
- [14] C. Mulliner and C. Miller, "Injecting SMS messages into smart phones for security analysis," in *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, vol. 29, Montreal, Canada, August 2009.
- [15] C. Mulliner and J.-P. Seifert, "Rise of the iBots: owning a telco network," in *Proceedings of the 2010 5th International Conference On Malicious And Unwanted Software*, pp. 71–80, IEEE, Nancy, France, October 2010.
- [16] S. Perez, "More droidDream details emerge: it was building a mobile botnet," 2011, <http://www.readwriteweb.com/archives/droiddream-malware-was-going-to-install-more-apps-on-you-%20r-phone.php>.

## Research Article

# Influence Maximization Algorithm Based on Reverse Reachable Set

Gengxin Sun <sup>1</sup> and Chih-Cheng Chen <sup>2,3</sup>

<sup>1</sup>School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China

<sup>2</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>3</sup>Department of Aeronautical Engineering, Chaoyang University of Technology, Taiwan 413310, Taiwan

Correspondence should be addressed to Chih-Cheng Chen; [ccc@gm.cyut.edu.tw](mailto:ccc@gm.cyut.edu.tw)

Received 11 February 2021; Revised 5 April 2021; Accepted 19 July 2021; Published 28 July 2021

Academic Editor: Isabella Torricollo

Copyright © 2021 Gengxin Sun and Chih-Cheng Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most of the existing influence maximization algorithms are not suitable for large-scale social networks due to their high time complexity or limited influence propagation range. Therefore, a D-RIS (dynamic-reverse reachable set) influence maximization algorithm is proposed based on the independent cascade model and combined with the reverse reachable set sampling. Under the premise that the influence propagation function satisfies monotonicity and submodularity, the D-RIS algorithm uses an automatic debugging method to determine the critical value of the number of reverse reachable sets, which not only obtains a better influence propagation range but also greatly reduces the time complexity. The experimental results on the two real datasets of Slashdot and Epinions show that D-RIS algorithm is close to the CELF (cost-effective lazy-forward) algorithm and higher than RIS algorithm, HighDegree algorithm, LIR algorithm, and pBmH (population-based metaheuristics) algorithm in influence propagation range. At the same time, it is significantly better than the CELF algorithm and RIS algorithm in running time, which indicates that D-RIS algorithm is more suitable for large-scale social network.

## 1. Introduction

Because the rapid development of social networks, the number of users, and the scale of information dissemination continue to expand, the problem of maximizing influence has received more and more attention. It is widely used in “viral marketing” [1, 2]. “Viral marketing” is a way to maximize brand awareness through word-of-mouth effects among users. Therefore, with limited resources, the key to maximizing influence is to select the appropriate initial communication users to maximize the final communication effect.

Richardson et al. [1] regard the problem of maximizing influence as an algorithmic problem; that is, under a specific information dissemination model, select  $k$  initial seed node sets from a social network to maximize the final influence dissemination range. Kempe et al. [3] proved for the first time that impact maximization is an NP-hard subject based

on the Independent Cascade model (IC model) [4] and the Linear Threshold model (LT model) [5]. At the same time, a greedy algorithm (GA) is proposed. The algorithm selects the node with the most considerable marginal effect by iteration to ensure that it is close to the optimal solution within the range of  $(1 - (1/e) - \epsilon)$ . Due to the high time complexity, this algorithm is not suitable for large-scale social networks. Therefore, many researchers have proposed some optimization algorithms for the low efficiency of greedy algorithms. In 2007, Leskovec et al. [6] proposed the Cost-Effective Lazy Forwards (CELF) algorithm. It uses the characteristics of the inter-node influence propagation function to satisfy the submodularity, which increases the running speed of the greedy algorithm by 700 times. In 2011, Goyal et al. [7] proposed the CELF++ algorithm, which further reduced the time complexity of the CELF algorithm. These algorithms have achieved a certain degree of speed improvement. However, each time a node is selected to join



the node set, the increase in the influence of the node is calculated, so the operating efficiency is still very low, and it is difficult to apply to large-scale social networks. At present, most scholars use heuristic algorithms to improve the running speed. Literature [8, 9] proposed different influence maximization algorithms on the basis of degree centrality. In 2010, Chen et al. [10] proposed the PMIA algorithm based on the maximum influence propagation path between nodes. In 2012, Jung et al. [11] proposed the IRIE heuristic algorithm for the IC model. In addition, heuristic influence maximization algorithms based on network topology have been proposed successively [12–14]. In 2016, Xie et al. [15] proposed a new heuristic algorithm to improve operational efficiency. Cao et al. [16] proposed a CCA algorithm based on K core. Still, these algorithms only focus on the topological structure of the network and lack a specific theoretical guarantee, which may cause the algorithm to fail to obtain the optimal solution. Based on the above problems, Sun et al. [13] proposed a RIS algorithm that combines theory and actual efficiency. The algorithm selects nodes by generating a certain number of reverse reachable sets and then calculates node influence many times so that the time complexity is close to linear, and there is a specific theoretical guarantee. Although the RIS algorithm has many advantages, it still has disadvantages such as insufficient accuracy and stability in selecting the number of the reverse reachable sets. Therefore, a lot of calculation costs are required in practice.

Any social animal has mutual influence between groups and individuals. As an advanced social animal with complex means of communication, interpersonal and social influence are everywhere in our social life. In-depth understanding of the generation and transmission mode of influence helps us understand the behavior of human groups and individuals, so as to predict people's behavior and provide reliable basis and suggestions for the decision-making of government, institutions, enterprises, and other departments.

In this paper, we propose a dynamic-reverse reachable set (D-RIS) algorithm based on reverse reachable set. The algorithm does not need to preset the theoretical threshold of the number of reverse reachable sets in advance but based on the monotonicity and submodularity of the influence propagation function, set the judgment conditions for generating the critical value of the random reverse reachable set, and automatically debugs the generation A certain number of reverse reachable sets can avoid time wastage while obtaining a better influence spread range.

## 2. Influence Maximization Algorithm Based on Reverse Reachable Set

The social network is abstracted as a network graph  $G$  with a node-set  $V$  (user) and a directed edge set  $E$  (the relationship between users), with  $G = (V, P, E)$ ,  $|V| = n$ ,  $|E| = m$ , and  $p \in (0, 1)$ . Assume that each edge  $e$  in  $G$  has a propagation probability  $p(E) \in (0, 1)$ ; then,  $p(u, v) \in p(u, v \in V)$  represents the probability that node  $u$  activates node  $v$ . For the convenience of presentation, Table 1 lists the symbols commonly used in this article.

**2.1. Communication Model and Question Description.** When looking for a specific set of seed nodes with the most significant influence in social networks, it is necessary to use a particular spread model to simulate the rules of spreading information on the network. The current classic information dissemination models include the IC model and the LT model.

The experiment in this article uses the IC model to simulate the maximum spread of user influence. In this model, a directed weighted graph  $G$  with  $n$  nodes and  $m$  edges is given to represent the underlying network. The weight of edge  $e = (v, u)$  represents the probability  $P$  that node  $v$  propagates to node  $u$  along edge  $e$ . Nodes in the IC model are divided into three states: activated state, newly activated state, and inactive state. Each newly activated node has one and only one chance to try to start adjacent nodes that are not activated with probability  $P$ . The higher the value of  $P$ , the greater the possibility of activation. When there are no influential active nodes in  $G$ , the propagation process ends. The influence of propagation simulation on the IC model is started by random propagation from a set of seed nodes. Let  $I(S)$  be the number of random nodes eventually infected by the propagation simulation process and  $E[I(S)]$  be the ultimate propagation impact of the node sets. This model simulates the propagation process of the infectious disease model [15, 16]. The seed set  $S$  is similar to a group of infected individuals, and the propagation simulation process of activating its adjacent nodes is identical to the spread of disease from one individual to another.

The following example describes how influence spreads in the IC model.

Figure 1 is an initial graph of a social network composed of four nodes, and the weight on each edge represents the propagation probability from the outside node to the in-side node. The activation probability of all nodes in the social network is defined as 0.5. The information propagation process of the social network is simulated as in Figure 1.  $S = \{a\}$  is the initial seed set. Node  $a$  is activated at time  $T_1$ . Then, at time  $T_2$ , node  $a$  has a probability of 0.2 to activate node  $c$  and probability of 0.8 to activate node  $d$ , because of  $p_{ac} = 0.5 > p = 0.2$ . At time  $T_2$ , node  $d$  is activated,  $S = \{a, d\}$ . At time  $T_3$ , node  $c$  and node  $b$  have a probability of 1 being activated by node  $d$ . Suppose that node  $d$  activates node  $c$  but not node  $b$ , which affects the end of the propagation process. Because no new nodes on the network can be activated, the total number of nodes activated in the propagation process is 3; that is,  $I(S) = 3, S = \{a, b, c\}$ . If node  $b$  is activated at time node  $b$ , then  $I(S) = 4, S = \{a, b, c, d\}$ . Since the IC model is a probability model [17], the propagation process and the final propagation result are not necessarily. The Monte Carlo method [14] is often used in experiments to take the average of multiple runs to ensure the accuracy of the results.

Given the social network  $G$  and a constant  $k$ , the problem of maximizing influence is to find a set of seed nodes  $S$  in  $G$  so that it has the widest range of influence under the IC propagation model, that is, finding the node set  $S \in V$  and  $|S| = k$  such that  $E[I(S)]$  is maximum.

TABLE 1: Frequently used notations.

Notation	Description
$G$	A social network
$n$	The number of nodes in $G$
$m$	The number of edges in $G$
$k$	The size of the seed set of influence maximization
$P(E)$	The propagation probability of an edge $e$
$S$	A node set
$I(S)$	The spread of a node set $S$ in an influence propagation process on $G$
$E[I(S)]$	The maximum propagation expectation for a node set
$R$	The set of all RR sets generated
$R_j$	A random RR set
$\theta$	The critical value of the number of RR sets
$\alpha$	The ratio of the RR sets

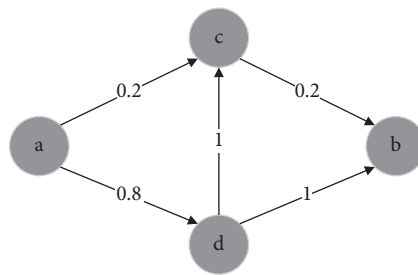


FIGURE 1: Initial diagram of the influence propagation social network based on the independent cascade model (IC model) seed node set.

2.2. *RIS Algorithm.* Borgs et al. [18] proposed the Reverse Influence Sampling (RIS) algorithm based on the IC model, which is a completely different influence maximization algorithm from other classic algorithms. The algorithm introduces a novel reverse reachable set (reverse reachable set, referred to as RR set) sampling method to replace the Monte Carlo method to calculate the influence of the expected propagation of nodes. The main idea is to generate as few reverse reachable set samples as possible and, finally, obtain a near-optimal solution in the range of  $(1 - (1/e) - \epsilon)$ . This algorithm proves that for any  $\epsilon > 0$ , it can run in the time of  $o(\beta(m + n)k \log_n)$ , and the time complexity is approximately linear time ( $\beta$  is the number of steps to select the reverse reachable centralized operation).

The RIS algorithm avoids the limitation of the high time complexity of the greedy algorithm and also solves the problem that the heuristic algorithm lacks theoretical guarantee and cannot obtain the optimal solution. But this algorithm cannot effectively control the number of random RR sets. They proposed a threshold-based method to generate random RR sets: when the total number of generated nodes and edges reaches a predetermined theoretical threshold, they stop generating random RR sets. Although this method has approximately linear time complexity, there is a great correlation between the generation of reverse reachable sets of fixed theoretical thresholds, and the hidden constants in practice are large, resulting in two shortcomings in the RIS algorithm. (1) The actual RR set sample size generated is greater than the theoretical threshold. (2) There is no guarantee that the theoretical threshold is the minimum number of samples generated in the RR set. Therefore,

the sample size of the RR set selected by this algorithm is not accurate, and it is not well suited for solving large-scale social networks.

2.3. *Based on Reverse Reachable Set: D-RIS Algorithm.* For most of the classic influence maximization algorithms, the time complexity is too high or the optimal solution cannot be obtained. Based on the IC model and combined with the reverse reachable set sampling method, we propose a D-RIS (Dynamic-Reverse Influence Sampling) algorithm for maximizing influence.

The D-RIS algorithm is divided into two steps:

- (1) Generate a reverse reachable set (RR set): randomly select  $n$  nodes with replacement and generate a set  $R$  of  $\theta$  node RR sets by performing propagation simulation on a random graph  $g$ . The value of  $\theta$  is determined by the method in Section 2.3.1.
- (2) Node selection: use the maximum coverage method to find  $k$  nodes that cover the most RR sets and return the node set  $S$ .

Analyzing the theory of the RIS algorithm, it can be known that if the sampling number of the random RR sets is too small, the algorithm will not get the optimal solution due to insufficient selection of nodes. If the sampling number of the random RR set is too large, although the error is reduced, the time complexity will be too high. Therefore, the accuracy of selecting the seed node set determines the final influence spread range and time efficiency. Therefore, the research focus of the algorithm in this paper is how to select the



smallest possible RR set sample size, so that the algorithm can achieve a better balance between the spread of influence and operating efficiency.

This paper firstly refers to the sampling method in [17] to define a unified reverse reachable set sampling framework. On this basis, Section 2.3.1 puts forward a new critical value judgment method, which can dynamically select as few RR set samples as possible. Finally, Section 2.3.2 uses the maximum coverage method to select the seed node set.

Given a network  $G = V, E, P$ , the algorithm captures the influence propagation process of nodes in  $G$  by generating a set  $R$  of random RR sets. Let  $R_z$  be a subset of the RR set of node  $v$ , that is, the random RR set of the node. Graph  $g$  is a random graph obtained by removing edge  $e$  in  $G$  with a probability of  $1 - P(E)$ . The specific definition and sampling process are as follows.

*Definition 1.* Reverse reachable set (RR set)

The set of reachable nodes in the random graph  $g$  (for each node  $u$  in the RR set, there is a directed path from  $u$  to  $v$  in  $g$ ).

Sampling process is as follows. (1) Randomly select a node  $v \in V$ . (2) Generate a sample random graph  $g$  on the network  $G$ . (3) Return the reverse reachable set  $R_z$  of node  $v$  in the random graph  $g$ .

The node  $v$  in the above sampling process is called the source in  $R_z$ , and all nodes in  $R_z$  have a certain probability to activate the source node  $v$ . Therefore, the presence of a certain node in more RR sets means that more nodes can be activated, and, at the same time, this node can produce a larger influence spread range. Based on the same inference, if the node set  $S$  with  $k$  nodes covers a large number of RR sets, the  $k$  nodes in the network  $G$  have strong propagation ability to spread to the maximum range; that is,  $IS = nPr[S \text{ Covers } R_z]$ . Therefore, the influence of the node set  $S$  is proportional to the probability that  $S$  and the RR set intersect. So, to solve the problem of maximizing influence, we determine the lower bound of the  $R$  set. Section 2.3.1, based on this reverse reachable set sampling framework, sets up a dynamic debugging method to determine the minimum number of  $R$  sets.

Use an example to illustrate the process of generating a reverse reachable set for the social network  $G$  in Figure 1 under the IC model, and set  $k = 1$ . Figure 2 shows three random RR sets  $g_1, g_2, g_3$  generated on  $G$ . Three random RR sets,  $R_1 = \{c, a\}$ ,  $R_2 = \{d, a\}$ , and  $R_3 = \{b, c, a\}$ , generated for three randomly selected source nodes  $c, b$ , and  $d$ , respectively. Because node  $a$  appears in three random RR sets, node  $a$  is the most influential node. Therefore, the final return result is  $S = \{a\}$ .

*2.3.1. Determination of the Number of Reverse Reachable Sets.*

Analysis of the selection of the number of random RR sets in the RIS algorithm shows that the more the number (the larger the  $R$  set), the more accurate the selected seed node set, but it will cause a waste of time. Therefore, this section proposes a method to control the number of generated  $R$  sets as small as possible without affecting the final influence spread.

In the experiment of Section 3.2.1, we found that as the number of random RR sets increases, the increase in the spread of influence is not linear but diminishing in utility. Therefore, the relationship between the number of RR sets and the influence propagation range function satisfies both monotonicity and submodularity (diminishing marginal utility), which is defined as follows.

- (1) Monotonicity: set the influence propagation range function  $f$ ; for any number of reverse reachable sets  $q_1 < q_2$ , there are  $f(q_1) \leq f(q_2)$
- (2) Submodularity: for the total number of nodes in the graph  $t$ , set the influence propagation range function  $f$ ; for the number of reverse reachable sets  $q_1 < q_2$ , and all  $a > 0$ , if  $q_1 < q_2 < t$ , there are  $f(q_1 + a) \leq f(q_2 + a)$

Based on the above theory, for a given  $G$ , the algorithm sets a critical value  $\theta$  for the number of random RR sets, where  $\theta = n \times \alpha$  ( $\alpha$  is the random RR set selection ratio). When the number of random RR sets is less than  $\theta$ , the maximum influence spread range cannot be achieved because the number of random RR sets selected is not enough. When the number of random RR sets is greater than  $\theta$ , due to diminishing marginal benefits, the range of influence increases too slowly or no longer increases, resulting in a waste of time. Therefore, based on the current propagation situation of the nodes in the network, the algorithm automatically doubles the generation of reverse reachable sets in each round until the critical value judgment condition set in Algorithm 1 (line 7) is met three times, and the number of RR sets generated by the algorithm is considered to be infinitely close to critical value. The specific description is as follows.

Set the influence spread range of this round to  $f_c$  and the influence spread range of the last round to  $f_p$ . Algorithm 1 gives the pseudocode in the process of generating the reverse reachable set of the D-RIS algorithm. The specific process is as follows:

- (1) Set the initial reverse reachable set number ratio to a very small value  $\alpha$  (e.g., in Algorithm 1, the value of  $\alpha$  is 0.001; then  $\theta = 0.001n$ ), randomly select nodes with a ratio of  $\alpha$  from the node set  $S$  in the graph  $G$  to generate an RR set, and calculate the impact Power transmission range  $f_c$  (Algorithm 1: Lines 4–6).
- (2) Each round doubles the value of  $\alpha$  and calculates the increase in the spread of influence in this round  $I_C$ , which is  $I_C = f_c - f_p$ . The following will make an effective judgment on the increase in the scope of influence in this round (Algorithm 1: Line 7); if the conditions are met, it is determined that this round of  $\alpha$  doubling has no effect on the growth of influence and may have been close to the critical value:

Judgment condition is as follows: if  $I_C \leq 0$  or  $I_C < \text{math} \cdot \log(I_p, 2)$ . That is, the increase in the range of influence of this round is less than or equal to 0 or less than the result of the root sign of the increase in the range of influence of the previous round.

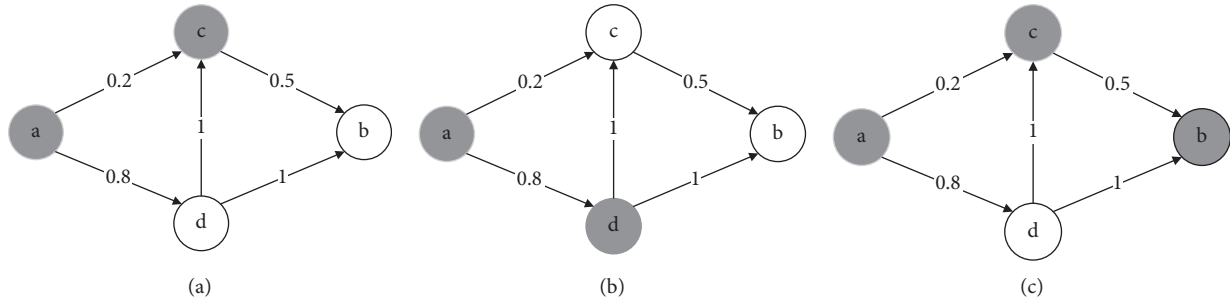


FIGURE 2: Three random graphs generated based on the social network in Figure 1. (a) Random graph  $g_1$ , (b) Random graph  $g_2$ , (c) Random graph  $g_3$ .

- (3) Repeat the above steps until three consecutive  $\alpha$  doublings are invalid or when the value of  $\alpha$  is greater than or equal to 1 and stop generating reverse reachable sets. At this time, the number of random RR sets generated by the algorithm approaches the critical value.

Suppose the final inverse reachable set ratio is  $\alpha_1$ , at this time, a relatively stable and effective critical value of inverse reachable set  $\theta$  is obtained, and at this time  $\theta = n \times \alpha_1$ .

In the process of dynamic debugging to determine the value of  $\alpha$ , the value of  $\alpha$  rises gradually until it approaches the critical value. Except for the first round, each cycle does not generate  $\alpha$  proportional reverse reachable sets but generates  $\alpha/2$  proportional reverse reachable sets. We will scale the previous round  $\alpha/2$  to reverse reachable sets. The reached set is stored to combine the reverse reachable set of the  $\alpha$  ratio of the cost round. That is, the same number of reverse reachable sets are generated based on the original reverse reachable sets to double the effect. Therefore, the time efficiency of the algorithm is greatly improved.

In short, this section proposes a method to determine the critical value  $\theta$  of a random RR set based on the monotonicity and submodularity of the influence propagation function, according to the real-time propagation of nodes in the network, and follows the reverse reachable set sampling framework to generate  $\theta$  reverse reachable sets. Next, the D-RIS algorithm calls Algorithm 2 in Section 2.3.2 to find the set of seed nodes  $S$ .

**2.3.2. Seed Node Selection.** The D-RIS algorithm uses the maximum coverage method for seed node selection. Algorithm 2 gives the pseudocode at this stage. Given  $G, k$  and the number of reverse reachable sets  $\theta$ , first, insert the  $\theta$  random RR sets generated in Algorithm 1 into the set  $R$ . If  $S \cap R_j \neq \emptyset$ , the seed set  $S$  covers a random RR set  $R_j$  and define  $\text{Cover}_R(S) = \sum_{R_j \in R} \min\{|S \cap R_j|, 1\}$ . Then, define the approximate value of  $IS$  as  $I(S) = I_R S = \text{Cover}_R(S)/|R|$ . So, the specific  $k$  iteration process is as follows:

- (1) Each time, the algorithm greedily selects a node  $v$  that covers the greatest number of nodes in the  $R$  set
- (2) Delete all the nodes  $v$  in the  $R$  set in reverse reachable set (i.e., the node  $v$  in the deleted reverse reachable set has a path that can be reached through the node)

- (3) Add the node  $v$  to the set  $S$ , update the  $R$  set, and proceed to the next iteration
- (4) Selected node set  $S = k$  iteration ends

In the process of using the maximum coverage method to select  $k$  node sets, the greedy algorithm is used to repeatedly select the nodes that cover the largest marginal revenue to join the node set  $S$ , so the approximate solution of  $(1 - (1/e) - \epsilon)$  can be returned, and the nearly linear time complexity can be obtained.

The D-RIS algorithm mainly includes two stages. In the first stage,  $n$  nodes are randomly selected to generate  $\theta$  reverse reachable sets, among which  $\theta = n \times \alpha$  ( $\alpha < 1$ ) and the time complexity is  $o(\theta)$ . For any randomly selected node  $v_j$ , suppose the time complexity of the reverse reachable set generated by propagation simulation based on a certain propagation model is  $o(\theta)$ , where EVP is the width of the random RR set (i.e., the number of directed edges pointing to the node  $v_j$  in the random graph  $g$ ), and the time complexity of the first stage of the D-RIS algorithm is  $o(\theta \times \text{EVP})$ . The maximum coverage method used in the second stage selects  $k$  nodes using greedy thinking, which can get linear time complexity. So, the time complexity of the D-RIS algorithm is  $o(\theta \times \text{EVP})$ . We have the time complexity of the greedy algorithm  $o(kmnr)$ , with  $r$  representing the number of times Monte Carlo sampling is used, and  $n$  and  $m$  represent the total number of nodes and edges in the network  $G$ , respectively. The values of  $n, m, r$  are commonly very large. In contrast, D-RIS algorithm has better time complexity. Besides, compared with the RIS algorithm that can also achieve linear time complexity, the D-RIS algorithm is more accurate and reasonable in the selection of the number of reverse reachable sets. The experiment also shows that the operating efficiency of the D-RIS algorithm has a better advantage. According to the above analysis, it can be concluded that the D-RIS algorithm is more suitable for large-scale social networks.

### 3. Experiments and Results

**3.1. Datasets.** In order to verify the timeliness of the D-RIS influence maximization algorithm, we use two real datasets for experiments. As shown in Table 2, the first Slashdot dataset [19] is a dataset of friends sharing technology information websites. The site allows users to mark each other

```

Input:  $G = (V, P, E), k$ 
Output:  $S, \alpha$ 
 $R = \emptyset, \alpha = 0.001, I_p = 0, f_p = 0$ 
While ( $\alpha < 1$  &&  $\text{flag} < 3$ )
  Generate a set of seed nodes with  $\alpha$  ratio
   $z \leftarrow \text{Simulate\_influence\_spread}()$ 
  generate  $\theta$  random RR sets and add all  $R_z$  to  $R$ 
   $f_c = \text{Cnt\_}f_c(), I_c = f_c - f_p$ 
  if  $I_c \leq 0$  or ( $I_p > 0$  and  $I_c < \text{math} \cdot \log(I_p, 2)$ )
     $\text{flag} = \text{flag} + 1$ 
  else  $\text{flag} = 0, f_p = f_c$ 
    if  $\text{flag} = 2$ 
      break
     $\text{proportion} = \text{proportion} * 2$ 
  return  $S, \alpha$ 

```

ALGORITHM 1: D-RIS algorithm (generate reverse reachable set).

```

Input:  $G = (V, P, E), k$ , number of RR sets
Output:  $S$ 
 $S \leftarrow \emptyset$ 
for  $i = 1$  to  $k$  do
   $v = \text{max\_coverage}()$ 
  add  $v$  to  $S$ 
  for RR sets contain  $v$ 
    remove all RR Sets from  $R$ 
return  $S$ 

```

ALGORITHM 2: D-RIS algorithm (node selection).

TABLE 2: Dataset information.

No.	Dataset	Nodes	Edges
1	Slashdot	77357	516575
2	Epinions	75879	508837

as “friends” or “enemies.” Of these, 76.7% of the nodes are in “friend” relationships. Some nodes with few or isolated social relationships are meaningless for the study of influence maximization. Therefore, we need to preprocess the original dataset and only select the nodes with a large number of social relationships.

In order to facilitate the comparison between different algorithms, in this paper, we processed the dataset and kept the friendship between 10,000 nodes. The number of friends after preprocessing is 36,338. The second dataset, Epinions [19], is an online social network based on trust. It is a dataset containing multiple relationships. If there is a directed edge from node to node, the node trusts the node. In this paper, we preserved the trust relationship of 10,000 nodes after preprocessing this dataset, which can be downloaded from the Stanford large network dataset website.

**3.2. Experimental Results and Analysis.** The information dissemination model used in the experiment is the

independent cascade (IC) model, and the dissemination probability is set to 0.08. The experiment was run 10,000 times in Monte Carlo and averaged to obtain the influence propagation range of the simulated propagation process. In order to verify the rationality and timeliness of the D-RIS algorithm, the comparative experiment algorithms we selected are currently five representative algorithms:

CELf algorithm is an improved algorithm of greedy algorithm. The core idea is basically the same, and the efficiency is improved by hundredfold. Therefore, this paper selects the CELf algorithm as a contrast algorithm with greedy thinking.

HighDegree algorithm [20] is the most classic heuristic algorithm based on node centrality;  $K$  nodes with the largest degree value are selected as the seed node set.

LIR algorithm [13] is a heuristic algorithm based on topological structure. This algorithm selects the node with the largest local degree value and sorts it and then selects the seed node set.

pBmH algorithm [14] is a heuristic algorithm, which is based on topological structure; this algorithm takes into account the influence of nodes by multiple neighbor nodes and avoids the phenomenon of rich clubs.

RIS algorithm [17] is an algorithm based on reverse reachable set sampling that generates a certain theoretical threshold number of reverse reachable sets and then selects the seed node set.

We set up the simulation experiment as follows:

D-RIS algorithm rule verification uses the Slashdot dataset to verify and analyze the monotonicity and submodality of the influence propagation function in the RIS algorithm and test this rule on the D-RIS algorithm.

D-RIS algorithm and RIS algorithm comparison experiment verification, the number of reverse reachable sets of different ratios of the RIS algorithm is set on the two datasets of Slashdot and Epinions, which will affect the D-RIS algorithm separately. The influence propagation range and running time are compared and analyzed

Comparison of D-RIS algorithm with other four classic algorithms: Section 3.2.3 of the experiment compares D-RIS algorithm with CELF algorithm, HighDegree algorithm, LIR algorithm, and pBmH algorithm on two different real datasets for influence propagation range, and the comparative analysis of running time verifies that the D-RIS algorithm has better timeliness than that of existing algorithms.

**3.2.1. D-RIS Algorithm Rule Verification.** Set  $k = 5$ ; the RIS algorithm starts to iterate from  $\alpha = 0.001$  and double the ratio of the reverse reachable set in each round until three consecutive doublings are invalid or stop.

It can be seen from Figure 3 that as the reverse reachable set ratio becomes larger, the front part of the curve shows an upward trend. The spread of influence continues to increase, which shows that the spread of influence of the RIS algorithm and the D-RIS algorithm is monotonic. In the RIS algorithm, when the reverse reachable set ratio is more significant than 0.01, the upward curve with the number of reverse reachable sets tends to be flat. This shows that the influence spreading function has the property of diminishing marginal effects due to the submodularity. From the curve in the figure, it can be seen that the expansion of the influence range gradually weakens. When the reverse reachable set ratio is 0.03, the curve's downward trend is slow, which is in line with the actual situation. Theoretically, the influence propagation range of the algorithm is monotonic. Due to the probability model's use as the propagation model, there are inevitable fluctuations in the experiment.

Figure 3 verifies that the basic reverse reachable sets influence propagation function has certain rules based on monotonicity and submodularity. With this rule, the RIS algorithm can be improved, which is also the theoretical basis of the D-RIS algorithm proposed in this paper. In the

figure, the D-RIS algorithm is also verified on the real dataset, and the result shows that the upward trend of the curve increases with the increase of the number of reverse reachable sets and then becomes flat. The D-RIS algorithm only needs to preset a smaller reverse reachable set ratio, and it can automatically double the debugging ratio until the condition is met. It avoids the problem that the unreasonable selection of the reverse reachable set ratio in the RIS algorithm leads to the failure of the optimal propagation range or the wasted time. This experiment shows that the D-RIS algorithm has certain rationality and practical significance.

**3.2.2. D-RIS Algorithm and RIS Algorithm Comparison Experiment Verification.** Set the reverse reachable set ratio of the RIS algorithm to 0.001, 0.2, and 0.5. Compare the influence spread range and running time with the D-RIS algorithm on two different datasets. Figures 4–9 are the comparative experimental results of the two algorithms on two different datasets.

- (1) Set the reverse reachable set ratio of the RIS algorithm to 0.001: when the RIS algorithm's reverse reachable set ratio is 0.001 (Figures 4 and 5), the RIS algorithm runs fast, but the influence spread is smaller than the D-RIS algorithm. Especially when the  $k$  value is low, there is a doubled gap in the spread of influence between the two. This is because the threshold of the number of reverse reachable sets in the RIS algorithm is too small, which results in the insufficient number of seed nodes selected, which affects the final propagation range of the algorithm.
- (2) Set the reverse reachable set ratio of the RIS algorithm to 0.2: as shown in Figures 6 and 7, when the reverse reachable set ratio of the RIS algorithm is 0.2. In the Slashdot dataset, the influence spread of the two algorithms is close, but the time efficiency of the D-RIS algorithm is higher than that of the RIS algorithm. In the Epinions dataset, the D-RIS algorithm greatly improves the running time under the premise of obtaining a larger influence spread range, and the larger the selected seed node set, the more obvious the advantage.
- (3) Set the reverse reachable set ratio of the RIS algorithm to 0.5: as shown in Figures 8 and 9, the RIS algorithm sets the reverse reachable set ratio to 0.5. On the two datasets, the D-RIS algorithm has a better spread range of influence, and the operating efficiency is much higher than that of the RIS algorithm. It can be seen that a too large reverse reachable set ratio will result in a waste of the final time cost of the algorithm. For the Slashdot dataset, the running time of the RIS algorithm is more than twice that of the D-RIS algorithm. For the Epinions dataset, the running time of the RIS algorithm is more than 7 times that of the D-RIS algorithm. Therefore, the D-RIS algorithm in this article is in the running time. The advantages are more obvious.

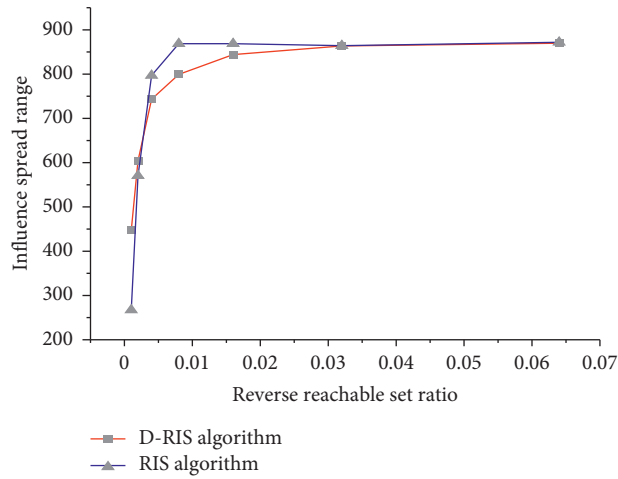


FIGURE 3: Relation between influence spread range and reverse reachable sets of the RIS algorithm and the D-RIS algorithm on Slashdot.

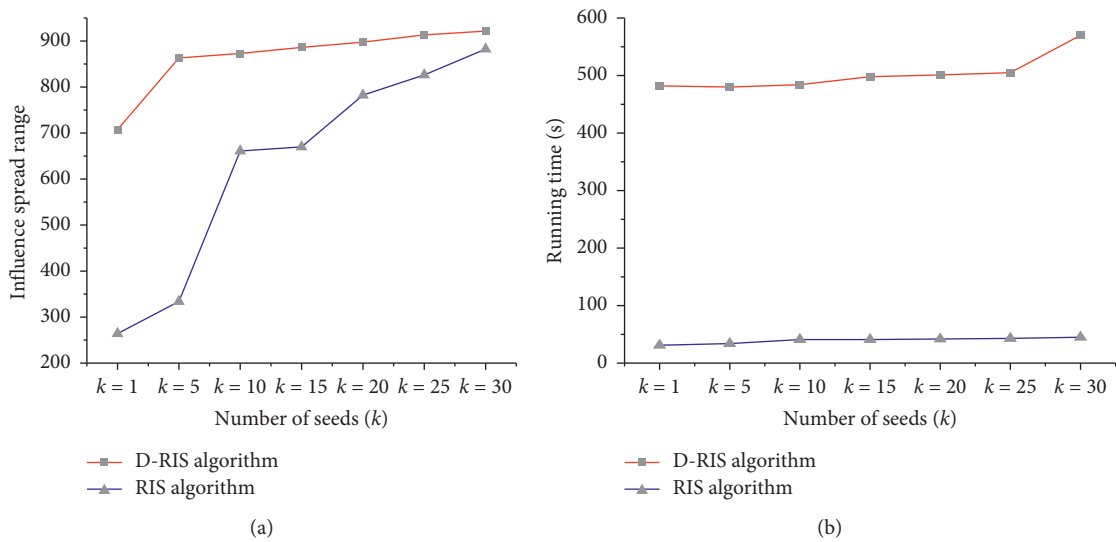


FIGURE 4: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.001) and the D-RIS algorithm on the Slashdot. (a) Comparisons of influence spread range and (b) comparisons of running time.

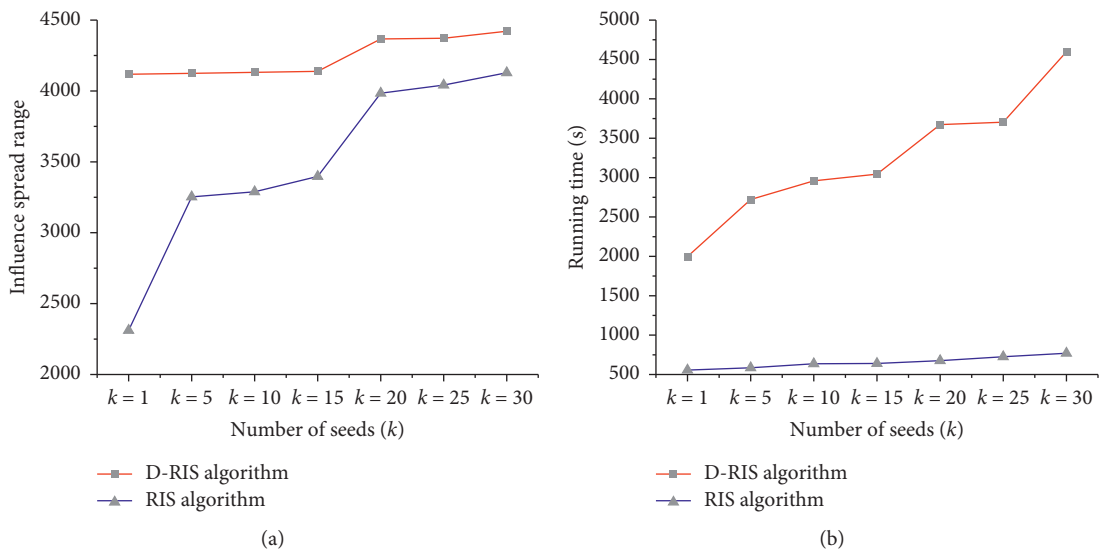


FIGURE 5: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.001) and the D-RIS algorithm on the Epinions. (a) Comparisons of influence spread range and (b) comparisons of running time.



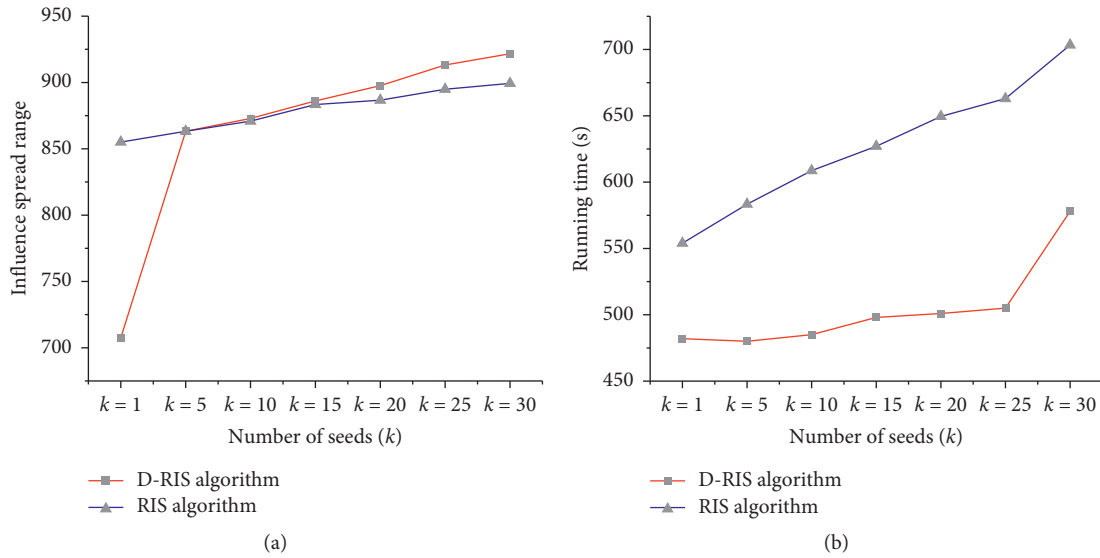


FIGURE 6: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.2) and the D-RIS algorithm on the Slashdot. (a) Comparisons of influence spread range and (b) comparisons of running time.

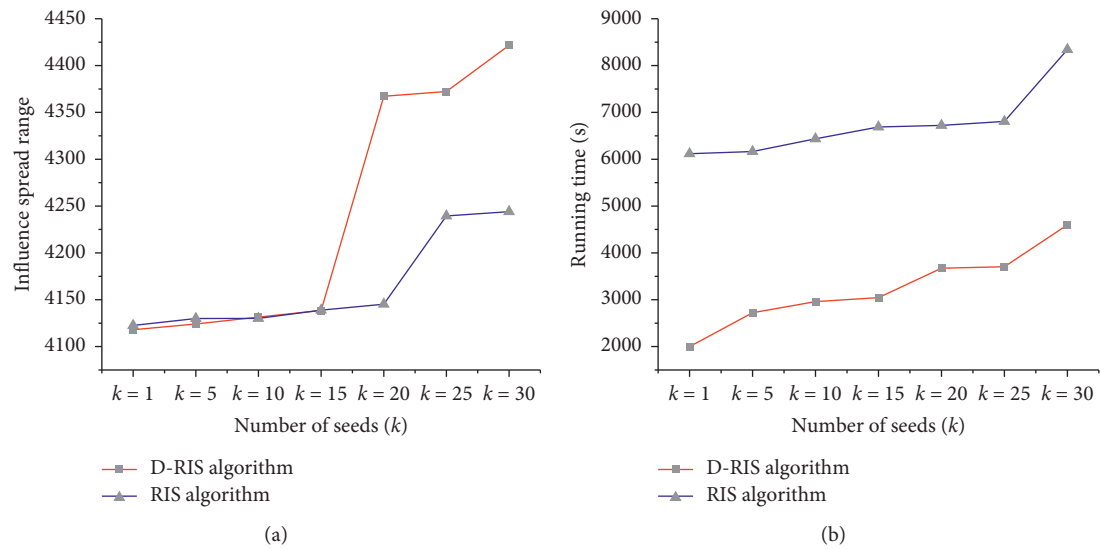


FIGURE 7: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.2) and the D-RIS algorithm on the Epinions. (a) Comparisons of influence spread range and (b) comparisons of running time.

In summary, through experimental verification on two real datasets, it can be seen that when the theoretical threshold of the reverse reachable set of the RIS algorithm is set too small, the influence propagation range is small. When the theoretical threshold of the reverse reachable set is too large, the time efficiency of the RIS algorithm is too poor. The D-RIS algorithm can achieve a better influence spreading range and at the same time run more efficiently.

In addition, compared with the RIS algorithm, the D-RIS algorithm avoids the inaccurate setting of the theoretical threshold of the number of reverse reachable sets, which leads to the problem of not reaching the optimal influence propagation range or causing a large waste of time. For the

current complex social networks, the D-RIS algorithm does not require repeated calculations, and the algorithm automatically debugs to generate a certain ratio of reverse reachable set that is also more suitable for subsequent network structure changes. Therefore, the D-RIS algorithm has certain practical significance.

3.2.3. Comparison of D-RIS Algorithm with Other Four Classic Algorithms. On two different data sets, the D-RIS algorithm is compared with the heuristic HighDegree algorithm, LIR algorithm, and pBmH algorithm and the greedy-based CELF algorithm to compare the influence propagation range and running time.



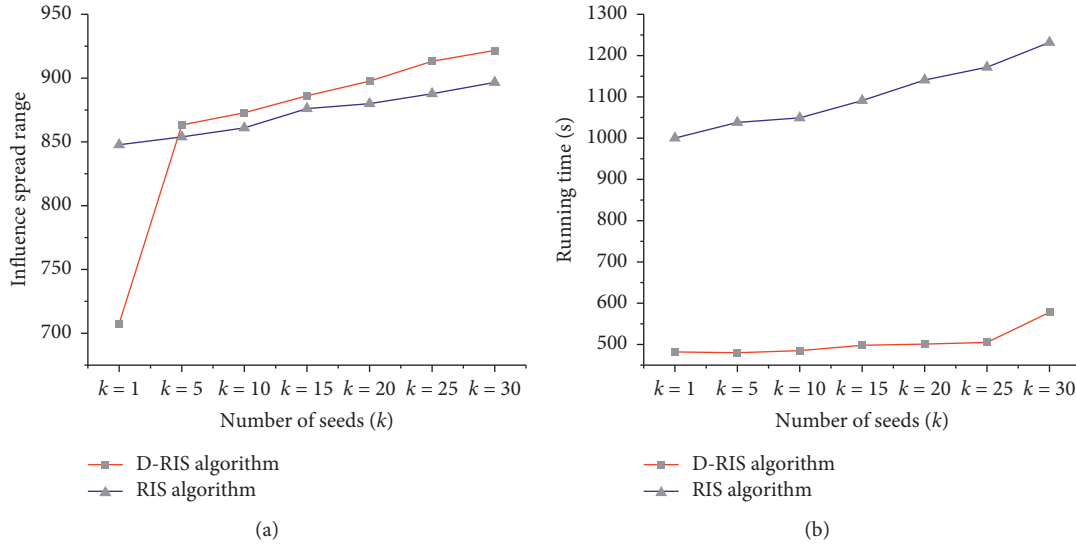


FIGURE 8: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.5) and the D-RIS algorithm on the Slashdot. (a) Comparisons of influence spread range and (b) Comparisons of running time.

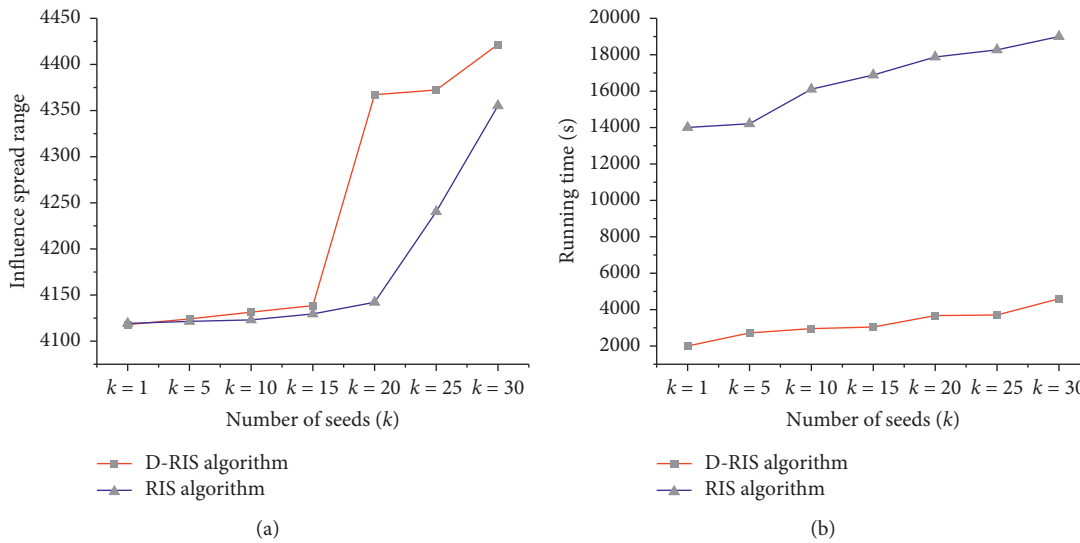


FIGURE 9: Comparisons of the experimental results of the RIS algorithm (reverse reachable set ratio is 0.5) and the D-RIS algorithm on the Epinions. (a) Comparisons of influence spread range and (b) Comparisons of running time.

According to the analysis of the experimental results in Figure 10 (Slashdot dataset) and Figure 11 (Epinions dataset), we have the following:

- (a) The influence propagation range of the D-RIS algorithm is basically similar to the CELF algorithm which is close to the optimal solution within the  $(1 - (1/e) - \epsilon)$  range. But D-RIS runs faster, and the larger the seed node set, the more obvious the advantage; the difference is close to hundreds of times; this is because the CELF algorithm uses the Monte

Carlo method for calculations, resulting in extremely high time complexity, so D-RIS algorithm is more suitable for large-scale social networks.

- (b) Compared with heuristic algorithms (HighDegree algorithm, LIR algorithm, and pBmH algorithm), although the D-RIS algorithm performs poorly in terms of running speed, the spread of the algorithm's influence is much higher than these heuristic algorithms. In the Epinions dataset, the influence spread of the heuristic algorithm is only about 50% of that of

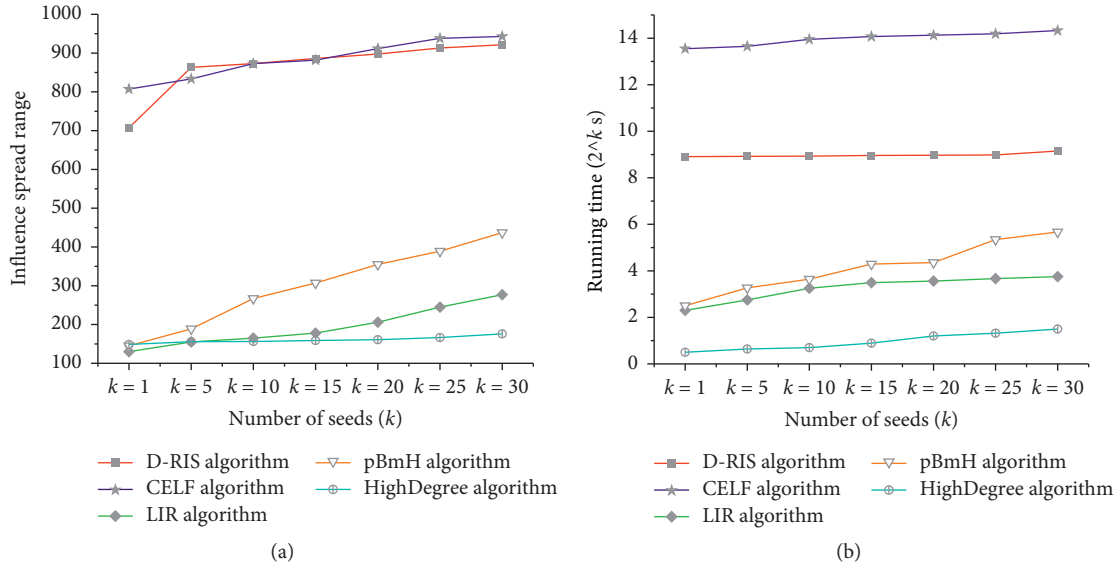


FIGURE 10: Comparisons of the running time of the five algorithms on Slashdot. (a) Comparisons of influence spread range and (b) comparisons of running time; the result of the ordinate is the result of taking the logarithm of 2.

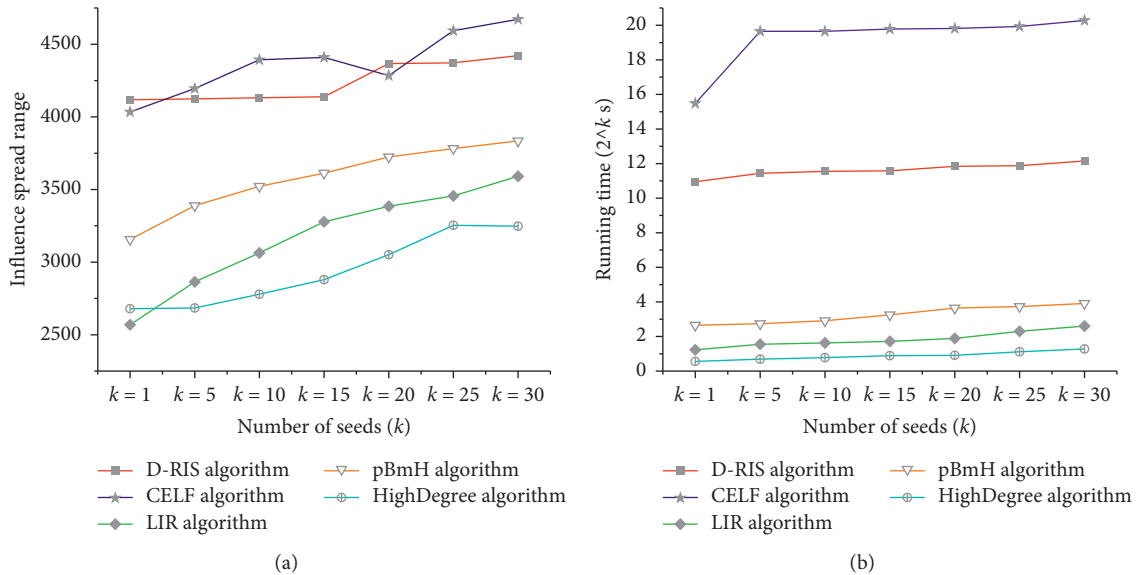


FIGURE 11: Comparisons of the running time of the five algorithms on Epinions. (a) Comparisons of influence spread range and (b) comparisons of running time; the result of the ordinate is the result of taking the logarithm of 2.

the D-RIS algorithm. In the Slashdot dataset, the D-RIS algorithm has more obvious advantages in spreading influence. It can be seen that although the heuristic algorithm has extremely high operating efficiency, it does not take into account that the complex network follow-up structure results in the selection of seed nodes that are not accurate enough, and the spread of influence is small, and the optimal solution is not reached. In addition, the stability of the heuristic algorithm is not good in different datasets.

Based on the comparative experimental analysis of the above algorithms, it can be seen that the D-RIS algorithm

proposed in this paper has achieved a good balance between the influence spread range and time efficiency and has shown good versatility and stability and is more suitable for large-scale social networks.

#### 4. Conclusions

In this paper, we propose a D-RIS influence maximization algorithm based on the independent cascade model combined with the reverse reachable set. Compared with the traditional RIS algorithm, the above algorithm obtains the number of reverse reachable sets by setting the automatic

tuning threshold instead of the fixed threshold. The experimental results show that D-RIS algorithm is close to CELF algorithm and higher than RIS algorithm, HighDegree algorithm, LIR algorithm, and pBmH algorithm in the spread of influence, and it is significantly better than CELF algorithm and RIS algorithm in running time. Therefore, the D-RIS algorithm proposed in this paper has dual advantages in terms of time efficiency and influence spread and can be applied to structural changes and large-scale social networks. In the following research, we will focus on extending the D-RIS algorithm to a more realistic multirelationship influence propagation model and improve the efficiency of the D-RIS algorithm.

## Data Availability

The data used to support the findings of this study are included within the article. The nature of the data is an excel file, and the data can be accessed on <https://github.com/>.

## Conflicts of Interest

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

## Acknowledgments

This research was funded by the Shandong Provincial Natural Science Foundation, China, under Grant no. ZR2017MG011.

## References

- [1] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 61–70, Edmonton, Alberta, Canada, July 2002.
- [2] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 57–66, San Francisco, CA, USA, August 2001.
- [3] D. Kempe and J. Kleinberg, "Maximizing the spread of influence through a social network," in *Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146, ACM, Washington, WA, USA, August 2003.
- [4] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: a complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.
- [5] J. Goldenberg, B. Libai, and E. Muller, "Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata," *Academy of Marketing Science Review*, vol. 9, no. 3, pp. 1–18, 2011.
- [6] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 420–429, ACM, San Jose, CA, USA, August 2007.
- [7] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "CELF++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th International Conference Companion on World Wide Web*, pp. 47–48, ACM, Hyderabad India, March 2011.
- [8] S. Bin and G. Sun, "Matrix factorization recommendation algorithm based on multiple social relationships," *Mathematical Problems in Engineering*, vol. 20218, pages, 2021.
- [9] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 199–208, ACM, Paris, France, June–July 2009.
- [10] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1029–1038, ACM, Washington, WA, USA, July 2010.
- [11] K. Jung, W. Heo, and W. Chen, "IRIE: scalable and robust influence maximization in social networks," in *Proceedings of the 12th IEEE International Conference Data Mining (ICDM)*, pp. 918–923, IEEE, Piscataway, NJ, USA, March 2012.
- [12] Z. Wang, H. Wang, Q. Liu, and E. Chen, "Influence nodes selection: a data reconstruction perspective," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 879–882, ACM, Gold Coast, Queensland, Australia, July 2014.
- [13] G. Sun, C.-C. Chen, and S. Bin, "Study of cascading failure in multisubnet composite complex networks," *Symmetry*, vol. 13, no. 3, p. 523, 2021.
- [14] D.-L. Nguyen, T.-H. Nguyen, T.-H. Do, and M. Yoo, "Probability-based multi-hop diffusion method for influence maximization in social networks," *Wireless Personal Communications*, vol. 93, no. 4, pp. 903–916, 2017.
- [15] S. Xie, Y. Liu, J. Zhu et al., "Research on topic-based local influence maximizing algorithm in social network," *Journal of Frontiers of Computer Science & Technology*, vol. 10, no. 5, pp. 646–656, 2016.
- [16] J. Cao, D. Dong, S. Xu et al., "Self-Interest influence maximization algorithm based on subject preference in competitive environment," *Chinese Journal of Computers*, vol. 2, pp. 238–248, 2015.
- [17] G. L. Tian, S. Zhou, G. X. Sun, and C. C. Chen, "A novel intelligent recommendation algorithm based on mass diffusion," *Discrete Dynamics in Nature and Society*, vol. 2021, Article ID 4568171, 9 pages, 2021.
- [18] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," 2012, <https://arxiv.org/abs/1212.0884>.
- [19] R. M. May and A. Lloyd, "Infection dynamics on scale-free networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 64, no. 2, Article ID 066112, 2001.
- [20] G. Sun and S. Bin, "A new opinion leaders detecting algorithm in multi-relationship online social networks," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4295–4307, 2018.

## Research Article

# Efficient Visualization Method and Implementation of Reservoir Model Based on WPF

Shanshan Liu , Xiaoqiu Wang , Yueli Feng , Xianlu Cai, Pengyin Yan ,  
and Binwang Li 

*College of Petroleum Engineering, China University of Petroleum, 102249 Beijing, China*

Correspondence should be addressed to Xiaoqiu Wang; 13466396559@126.com

Received 21 February 2021; Accepted 31 May 2021; Published 11 June 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Shanshan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In view of the slow speed and poor effect of real-time rendering of large-scale reservoir geological grid model, a new grid model hidden algorithm is proposed by analyzing the Eclipse reservoir grid model storage format, grid model representation, and cell sorting rule, which optimizes the original grid data and improves the rendering speed of reservoir grid model. The algorithm in this paper eliminates hidden points and faces according to the topological relationship of the grid, and finally, only the visible point and face data are extracted as the final visual input data. Through the realization of 3D visualization software of reservoir geological model and well trajectory, the correctness and efficiency of the hidden algorithm are verified. In the software, firstly, the number of display grids is effectively reduced by preprocessing, and the 3D graphics technology of WPF and helix is adopted to realize the high-efficiency display of reservoir grids. The comparison test of different scale reservoir models shows that the method can reduce the point and surface data by more than 85% and shows that the speed optimization effect is significant. The 3D display function realizes the interactive functions such as roaming, zooming, and viewpoint switching of the reservoir model, truly reveals again the geological environment and borehole information of underground drilling, which is helpful for drilling interpretation and decision-making, provides a reasonable drilling tracking geological target drilling scheme for the drilling process, and realizes the seamless connection of geological engineering integration.

## 1. Introduction

Geological engineering integration, as an effective workflow for exploration and development of unconventional oil and gas reservoirs, has become a broad consensus in the industry [1–6] which covers professional fields such as geology, reservoir, geophysical exploration, drilling, mud logging, well testing, production testing, oil and gas production, and downhole operation. The “geology” here does not specifically refer to the geological discipline in the sense of discipline but generally refers to the multidisciplinary comprehensive research work centered on reservoir, including reservoir characterization, geological modeling, and so on [7–12]. “Engineering” refers to a series of engineering technology application and solution optimization from drilling to production in the process of exploration and development. During the drilling process, three-dimensional engineering geological data are loaded into a

unified three-dimensional model, and the three-dimensional scene of the underground space is vividly displayed through software display, which can establish a good communication channel for engineering and geological experts and better realize multidiscipline cooperation. This is one of the important manifestation forms of geological engineering integration. At present, the theoretical research and technical methods of 3D reservoir visualization in China are still in the exploratory stage.

Reservoir geological model is the foundation and core of geological engineering integration of oil and gas exploration and development. Its visualization technology based on 3D geological model of reservoir makes use of the characteristics of virtual reality to truly reproduce the distribution law of reservoir in underground space, and developers can intuitively study complex geological structure and analyze exploration results. Mature large-scale commercial application software mostly contains 3D visualization modeling module.

In order to study and utilize geological research results in mainstream commercial software, it is necessary to analyze grid description methods of different software and write corresponding processing programs to realize efficient visualization of 3D geological model. The organization, loading, and visualization of large-scale reservoir model data require high storage capacity, processing speed, and drawing speed of computer. When rotating and scaling interactive operations are carried out, the amount of calculation data is huge, and the display speed is slow and discontinuous [13–16].

In this paper, the reservoir model is optimized according to the needs of geological engineering integration. By analyzing the commonly used file format of reservoir model data in Eclipse, extracting the required data, and blanking the reservoir grid, the amount of file data is greatly reduced. At the same time, combined with the latest 3D graphical interface, the efficiency of transmission and display of reservoir model is comprehensively improved, which meets the needs of model drawing and visualization platform construction under large-scale reservoir geological data.

To realize the modular development of grid blanking, first read the file, and then extract the read file data. After the data is extracted, it is processed to judge whether the face is completely repeated by judging the coordinate values of 12 points. If the 12 values are exactly the same, the face is repeated; otherwise, it is not repeated. After all the faces of the mesh are judged, the display and hide relations of all the faces are obtained. At this time, as long as the points on the display surface are all the points that need to be displayed, after all the points on the display surface are displayed, all the points that need to be displayed will be displayed, and the relationship between the display and hiding of the points will be clear. At this time, the displayed points will be repeated, because one point may be used by multiple displayed faces. These points that need to be used repeatedly do not need to be displayed repeatedly; they only need to be displayed once. Therefore, it is necessary to arrange the displayed points in the order of natural numbers, and all the displayed points are only arranged once. After sorting, the point set is obtained. At the beginning, the coordinates of points are stored in double type, and the displayed surface is composed of 4 points with 3 values for each point, that is, 12 double type storage values. After the surface set is represented by the point set, a point is composed of four integers, that is, four int type values. It is self-evident that this processing improves the computational efficiency by 5 times and greatly improves the display effect.

## 2. Reservoir Model Data Format

Eclipse is the most commonly used reservoir numerical simulation software, so reservoir departments can generally provide reservoir model data files in Eclipse format. Firstly, this paper interprets the file format of Eclipse model and extracts data content and grid data by keywords according to the attributes to be displayed in visualization, which provides input data for the next step of blanking optimization.

Eclipse reservoir model adopts hexahedron grid, and its basic unit is convex hexahedron. The model file contains multiple data files to describe different contents. The data file types are shown in Table 1.

Model data is stored in sections according to keywords. When the software loads data, it can determine the content of subsequent data according to keywords when corresponding keywords are encountered. Common keywords of Eclipse are shown in Table 2.

The data files are stored in text format and binary format, among which the data files in text format can be viewed directly by text editor, which is easy to read, but it is not conducive to computer processing, with large amount of data and low processing efficiency. Binary files cannot be viewed directly by text, but they can be loaded into computers quickly, with small amount of data and high processing efficiency. The following are introduced separately.

*2.1. Text Formatting.* Take the FGRID grid file in Eclipse as an example, as shown in Figure 1.

Its basic format is segmented according to keywords; for example, dimensions keyword is used to describe dimensions, COORDS keyword is used to describe coordinates of cells, and CORNERS keyword is used to describe coordinates of corners. Because the text format can be read directly, it can be read programmatically according to the meaning of keywords.

*2.2. Binary Format.* There are five data types in Eclipse binary files, as shown in Table 3.

The main point is that the binary type byte coding order of Eclipse file is Big Endian, which is different from the commonly used x86 CPU byte coding order Little Endian. If x86 CPU is adopted, the byte order needs to be reversed when loading data. A perfect computer program can determine the coding order of CPU by dynamically judging byte order at runtime, so as to dynamically determine whether the coding order needs to be processed, which is particularly important for cross-platform development.

## 3. Reservoir Model Pretreatment

The grid data of reservoir model is very huge. In the visualization process, if all grids are displayed directly, it will bring great pressure to the display card, and the general display card can hardly meet the display requirements. On the basis of analyzing the rules of reservoir geological grid data, this paper puts forward a new grid blanking algorithm, optimizes the original grid data, compiles three-dimensional visualization software, and realizes interactive functions such as roaming, zooming, and viewpoint switching of reservoir model in three-dimensional environment. After analysis, there are a large number of occlusion relations among the cells of the grid. On the premise of not affecting the display effect of the reservoir model, the repeated vertices and faces in a large number of 3D cells are hidden and simplified, and finally only the points and faces on the surface can be displayed to achieve the same visualization



TABLE 1: Data files included in the Eclipse model.

Filename	Function
*.GRID or *.FGRID	Grid file in text format
*.EGRID	Grid file in binary format
*.INIT or *.FINIT	Property file
*.PRT	Report output file
*.LOG	Output report during background job
*.DBG	Debugging file
*.SAVE	Restart files quickly
*.RFT	Calculation results of RFT
*.FLUX	Flow boundary

TABLE 2: Common keywords of the Eclipse model.

Keywords	Data
RUNSPEC	Basic information such as the dimension of the model
GRID	Model grid and attribute data
EDIT	Edit pore volume and conductivity
PROPS	PVT properties of fluids and rock data
REGIONS	Partition data
SOLUTION	Balance area data
SUMMARY	Output of calculation result
SCHEDULE	Dynamic data

purpose, and at the same time, the data needed for display can be greatly reduced, the display efficiency can be improved, and the demand for hardware can be reduced.

At the same time, most of the numerical simulation parameters in the model file need not be used in the visualization process, and it is also invalid to transmit this part of data in the network. By preprocessing with software, extracting the data needed for visualization, and storing and transmitting it in an efficient way, the amount of data stored and transmitted can be effectively reduced and the efficiency can be improved.

The pretreatment of reservoir model mainly includes two aspects, one is the blanking calculation of grid model [17], and the other is the efficient storage of blanking results.

According to the characteristics of reservoir geological model grid, this paper puts forward a hidden algorithm based on topological relation of cells, which realizes fast hidden calculation through a set of predefined data of point and plane association relations. This method is fast and stable and has achieved satisfactory results in practical application. The following is the main implementation process of the algorithm.

**3.1. Cell Hexahedron Definition.** The reservoir model is described by hexahedron corner grid. A reservoir grid is composed of  $NX \times NY \times NZ$  hexahedron (cells). These cells are not disordered but arranged in an orderly manner along the  $X$ ,  $Y$ , and  $Z$  directions. Therefore, knowing the number of a cell can determine the position of the cell in the whole grid and its adjacent cells.

First, define a cell, as shown in Figure 2.

As shown in the figure, the numbers of the six faces of each cell are 0, front; 1, back; 2, left; 3, right; 4, up; and 5-

down, with eight vertices: the upper four vertices are numbered from 0 to 3 in counterclockwise order, and the lower four vertices are numbered from 4 to 7.

**3.2. Hidden Data of Reservoir Grid.** In this paper, the hidden algorithm eliminates hidden points and faces according to the topological relations of meshes, in which hidden faces refer to faces where four vertices of two hexahedrons coincide completely. The strategy of the algorithm is to solve the visible point set and face set of the effective grid. Firstly, the algorithm judges the validity of the grid, scans and calculates according to the  $X$ ,  $Y$ , and  $Z$  directions of the cells, judges the coincidence of six faces and adjacent faces of the effective grid, and eliminates the overlapping faces and vertices according to the visibility of the faces. Finally, only the visible point and face data are extracted as the input data of the final visualization, in which the point set also considers the processing problem of coincidence points, thus obtaining an efficient and concise blanking method.

As shown in Figure 3, observing the grid, if the coordinates of four vertices of adjacent faces of adjacent grids coincide completely, then these two faces can be hidden, and if all three faces adjacent to each vertex are hidden faces, the points are changed to hidden points, so the coordinate values of this point do not need to be used when displaying the grid, and only one coincident visible point needs to be kept.

**3.2.1. Hidden Data of Reservoir Grid.** The basic steps of grid model blanking algorithm are shown in Figure 4.

- (i) Initialize according to the validity of cells, and set six faces of all valid grids as visible, and six faces of invalid grids as invisible.
- (ii) Traverse each cell  $Cell(i, j, k)$ , in which  $(1 \leq i \leq NX, 1 \leq j \leq NY, 1 \leq k \leq NZ)$ .
- (iii) Determine whether the current cell is valid. If it is valid, continue the blanking calculation; otherwise, go to step (2).
- (iv) Find adjacent right, back, and lower cells, respectively, that is,  $Cell(i + 1, j, k)$ ,  $Cell(i, j + 1, k)$ , and  $Cell(i, j, k + 1)$ , to determine the visibility of faces 3, 1, and 5. If the adjacent cells are invalid, the faces are visible. Otherwise, if the adjacent faces, i.e., 2, 0, and 4, do not coincide with this face, then both faces are visible; otherwise, they are invisible. Note that, in this step, the visibility of 2, 0, and 4 sides of adjacent cells has been confirmed, so the current cell only needs to judge the visibility of 3, 1, and 5 sides.
- (v) According to the visibility of six faces of the current cell, determine the visibility of eight vertices, that is, if all three faces adjacent to each vertex are invisible, the point is hidden; otherwise, the point is visible.
- (vi) According to the visibility of vertices, determine which vertices need to be output to the final point

```

1  'DIMENS ' 3 'INTE'
2  ' ' 126 75 4
3  'GRIDUNIT' 2 'CHAR'
4  'METRES ' '
5  'MAPAXES ' 6 'REAL'
6  0.00000000E+00 -0.10000000E+04 0.00000000E+00 0.00000000E+00
7  0.10000000E+04 0.00000000E+00
8  'MAPUNITS' 1 'CHAR'
9  'METRES '
10 'RADIAL ' 1 'CHAR'
11 'FALSE '
12 'COORDS ' 7 'INTE'
13 ' ' 1 1 1 1 0 0
14 ' ' 0
15 'CORNERS ' 24 'REAL'
16 0.64477938E+06 -0.51470595E+07 0.92773108E+03 0.64480406E+06
17 -0.51470530E+07 0.92775293E+03 0.64477269E+06 -0.51470350E+07
18 0.92770911E+03 0.64479731E+06 -0.51470280E+07 0.92773096E+03
19 0.64477012E+06 -0.51470420E+07 0.95831512E+03 0.64479350E+06
20 -0.51470355E+07 0.95833014E+03 0.64476350E+06 -0.51470175E+07
21 0.95829987E+03 0.64478669E+06 -0.51470110E+07 0.95831512E+03

```

FIGURE 1: FGRID data file.

TABLE 3: Data types of the Eclipse model.

Type name	Data type	Length	Remarks
INTE	int	4	4-byte integer
REAL	float	4	Single precision floating point
CHAR	char	1	Character
LOGI	bool	4	Logical type
DOUB	double	8	Double precision floating point number

set. However, in this step, it is necessary to consider that the vertices may coincide with the previous cell vertices. To avoid outputting duplicate vertices, it is necessary to compare the left, front, and upper vertices. Only when the adjacent vertices do not coincide, the corresponding vertices will be output, and the index of each visible vertex will be recorded at the same time.

- (vii) Judging whether the traversal is completed.
- (viii) All visible vertices are output to the final point set, and each visible face is output according to the index of the visible vertices. For each face, only four vertex index values of the face need to be output, and the final result is saved to the data file after blanking.

**3.2.2. Efficient Blanking Calculation.** In order to make the program more concise and efficient, the software designed a set of calculation tables according to the topological relations of cells for fast calculation.

The data in Table 4 is used for face blanking, from which it can be seen that only a three-step cycle is needed, and the

adjacent cells in the corresponding direction can be found according to the increment of data  $i, j, k, J,$  and  $K$  in each row. According to the blanking face index and the adjacent face index, the faces to be compared can be determined, and the visibility can be determined only by comparing the vertices in the blanking face vertex index and the adjacent face vertex index.

Whether any vertex in a cell is visible or not can be judged by the visibility of the face where the point is located. It can be seen from the graph observation that each vertex is on three faces, and if any face is visible, the point is visible. The data in Table 5 is used for vertex blanking calculation. For any  $p$  point in the cell,  $0 \leq p \leq 7$ , and its visibility can be determined only by looking at the visibility of three adjacent faces in the corresponding row in the table.

**3.2.3. Optimization of Blanking Results.** Based on the above calculation, the visible and visible points of each cell can be obtained, but the vertices of these faces may overlap. In the process of visualization, the data of coincidence points will occupy a large amount of memory of the graphics card, so it is necessary to further optimize it. The optimization goal is to get a point set and a face set, in which there are no coincident points in the point set, and all visible faces can be represented by these points.

First, process the cells one by one. When the visible points are found, check whether there are overlapping points on the adjacent upper, front, and left sides according to the data shown in Table 6. If there are no overlapping points, output the point to the point set, and record the index value of the point in point set, that is, the order attribute of the cell; otherwise, use the index value of the previously appearing points.

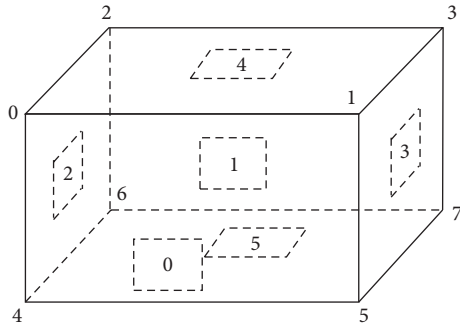


FIGURE 2: Cell basic definition.

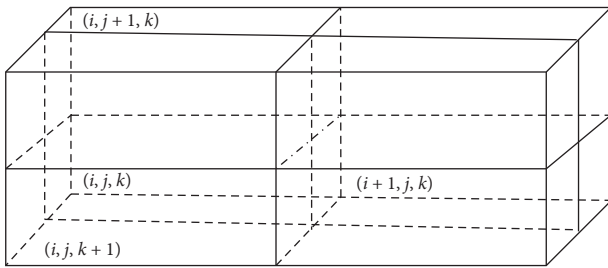


FIGURE 3: Grid diagram.

The number of each group represents the index of the point which may be adjacent to the point among the upper, front, and left cells of the cell. If the point is not adjacent to the corresponding cell, the index of the point is set to  $-1$ . With this set of constants, you can quickly output the point set of noncoincident points in the cell.

After the above processing, the optimized point set data with eliminated coincidence points can be obtained, and only the indexes of the four vertex coordinates in the point set need to be given when the visible faces are expressed.

**3.3. Storage of Blanking Results.** Saving the result of blanking calculation as a data file is beneficial to reduce the amount of data transmitted by the network, is easy to encode and parse, and has good self-description and scalability.

By investigating various text and binary data formats, such as XML, JSON, BSON, MsgPack, etc., the MsgPack format is determined as the basic format, which is a binary format similar to JSON format and has good self-description and extensibility. Generally, the data amount is  $1/20$  of XML and  $1/10$  of JSON. Almost all systems and development languages provide support for MsgPack format.

#### 4. Efficient Visualization of Reservoir Model

**4.1. Visual Mapping of Reservoir Data.** Obtaining a grid data set in a unified format for visualization is the data preprocessing in the prestige of scientific computing visualization process, which is called data manipulation. The next step is to display the data grid, including visual mapping and drawing. Study and analyze the visual mapping scheme, and find the appropriate image display effect and man-machine interaction mode.

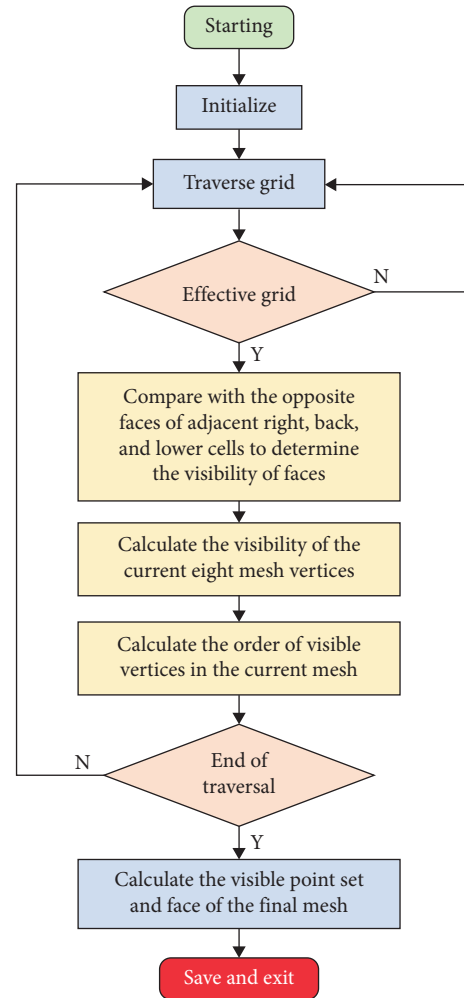


FIGURE 4: Flow chart of grid blanking.

**4.2. Visual Rendering of Reservoir Data.** The drawing process of reservoir grid includes ① drawing geometric shape based on data, including grid and well trajectory, ② determining the relationship between cell parameters and colors, that is, cell coloring, and ③ control of 3D objects, such as rotation, translation, and scaling [18].

**4.2.1. Grid Drawing.** The key to the display of reservoir model is the grid processing and display [19–23]. There is a huge amount of data in reservoir model, which must be preprocessed before displaying. The grid is composed of a series of hexahedrons, in which a large number of cells are continuous and have a mutual shielding relationship; that is, the surfaces of many cells are invisible because of overlapping. In the process of visualization, it is meaningless to display these overlapping surfaces, which will only increase the display burden and affect the performance. In the preprocessing process, all hidden points and faces need to be eliminated according to the topological relations of cells, so as to reduce the amount of displayed data.

Taking a medium-sized model as an example, the size of the model is  $132 \times 92 \times 45 = 564,300$  cells, including 182,104

TABLE 4: Face blanking calculation table.

Blanking surface	$i$ increment	$j$ increment	$k$ increment	Blanking surface index	Adjacent face index	Vertex index of blanking surface	Adjacent face vertex index
Right	1	0	0	3	2	1, 3, 5, 7	0, 2, 4, 6
Down	0	1	0	5	4	2, 3, 6, 7	0, 1, 4, 5
Back	0	0	1	1	0	4, 5, 6, 7	0, 1, 2, 3

TABLE 5: Vertex blanking calculation table.

Pinnacle	Adjacent faces 1	Adjacent faces 2	Adjacent faces 3
0	0	2	4
1	0	3	4
2	1	2	4
3	1	3	4
4	0	2	5
5	0	3	5
6	1	2	5
7	1	3	5

TABLE 6: Overlap point blanking calculation table.

Pinnacle	Adjacent faces 1	Adjacent faces 2	Adjacent faces 3
0	4	2	1
1	5	3	-1
2	6	-1	3
3	7	-1	-1
4	-1	6	5
5	-1	7	-1
6	-1	-1	7
7	-1	-1	-1

effective cells, 1,092,624 points, and 1,092,624 faces. After preprocessing, the visible points are reduced to 171,611 and the visible faces are reduced to 153, as shown in Table 7.

The reservoir model inherits the MeshGeometryModel3D, allowing developers to specify location, common, and texture coordinate information. The key point is to construct the Geometry object through the MeshBuilder class, in which the key attributes are vertex Positions, TriangleIndices, triangle Normals, and TextureCoordinates. The reservoir grid is composed of a series of quadrangles, as shown in Figure 5 for a single quadrangle.

The basic element of 3D graphics is triangle, so a quadrilateral is divided into two triangles  $(p_0, p_1, p_2)$  and  $(p_2, p_3, p_0)$ . The normal vector  $n$  is

$$N = ((p_1 - p_0) \times (p_3 - p_0)). \quad (1)$$

The material is Color Stripe Material, and the material coordinate  $(u, v)$  is

$$(u, v) = \left( \frac{(V - V_{\min})}{(V_{\max} - V_{\min})}, 0 \right), \quad (2)$$

where  $V$  is cell attribute value,  $V_{\min}$  is minimum attribute value, and  $V_{\max}$  is maximum value of attribute value.

**4.2.2. Grid Coloring.** Color code module is the foundation of the whole 3D visualization module, and the design of color

code module directly determines the visualization effect. In the process of visualization, the corresponding attributes of reservoir model can be expressed in different colors, because using colors can better show the trend of the whole formation and the distribution of attributes, and it is easier to judge the current state of the region, so it is very important to design a good color coding system. In this paper, the grid is colored by color code. By choosing different color codes, the grid can be colored with different effects. The basic concept of color code is to use a set of color values to represent numerical values, such as the commonly used rainbow color code. At the same time, this paper also establishes other color codes to meet different needs. The effects are shown in Table 8.

**4.2.3. Well Trajectory Drawing.** Three-dimensional display of well trajectory model plays an important part in the reservoir three-dimensional model [24]. It reflects well trajectory model through three-dimensional visualization technology and combines with reservoir grid model, which not only facilitates reservoir and geological personnel to observe and analyze the well location and well trajectory direction of exploited wells more intuitively, but also understands the drilling depth and the stratum situation of trajectory crossing and also brings convenience for subsequent well trajectory design and control, development scheme formulation, and oilfield development management and decision-making. Trajectory data mainly include measured depth (MD), angle of inclination (Dev), and Azimuth (Azi). As shown in Table 9, the spatial coordinates of the trajectory need to be obtained through calculation: north coordinate  $x$ , east coordinate  $y$ , and vertical depth  $z$ , which, respectively, represent the displacement of a certain point on the trajectory in the north-south direction, east-west direction, and vertical direction relative to the wellhead.

Taking the wellhead as the origin of coordinates, two adjacent measuring points are recursively calculated from the wellhead in turn. By assuming the three-dimensional curve shape of the well section between the two measuring



TABLE 7: The preprocessing results.

Model	Model size	Total number of cells	Effective cell numbers	Contrast	Before blanking	After blanking	Decrement (%)
1	132 × 95 × 45	564,300	182,104	Points	1,456,832	171,611	88.22
				Faces	1,092,624	153,616	85.94
2	224 × 147 × 29	954912	795,035	Points	6,360,280	79,038	99.88
				Faces	4,770,210	76,348	98.40
3	208 × 73 × 99	1,503,216	242,478	Points	1,939,824	178,911	90.78
				Faces	1,454,868	143,742	90.12

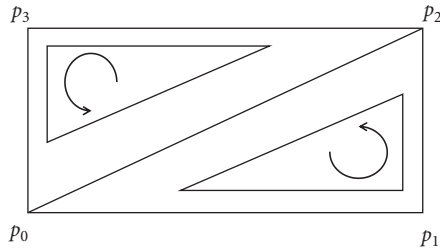


FIGURE 5: Single quadrangle.

points, the vertical increment  $\Delta Z$ , the north-south increment  $\Delta X$ , and the east-west increment  $\Delta Y$  of the measuring well section are calculated, and then they are accumulated to obtain  $x$ ,  $y$ , and  $z$ , and the well trajectory is formed by connecting these data. In some cases, if the well trajectory data is sparse, there will be broken lines in the direct connection line, so the smooth well trajectory can be obtained by spatial arc interpolation or cubic spline interpolation [25].

For a single well, its  $X$ ,  $Y$ , and  $Z$  are relative to its wellhead coordinates. For different wells, the wellhead's north coordinates and east coordinates and wellhead altitude on the map are different. If the multiwell data are placed in the same coordinate system, all well data must be corrected to a unified coordinate system. ( $X_u, Y_u, Z_u$ ):

$$\begin{cases} X_u = W_X + X \\ Y_u = W_Y + Y \\ Z_u = W_b - Z \end{cases}, \quad (3)$$

where  $W_b$  is bushing elevation,  $W_X$  is north coordinate of wellhead map, and  $W_Y$  is east coordinates of wellhead map.

In order to reduce the calculation in actual development, relative coordinates can be used to move the well trajectory to the wellhead position through translation transformation, so that GPU calculation can be fully utilized.

**4.2.4. Three-Dimensional Transformation.** Translation, rotation, and scaling are basic three-dimensional transformations, and formulas can be used to represent each transformation. However, when representing a variety of continuous transformations, the formulas will become very complicated, and it is more convenient to realize various basic transformations and combined transformations by using coordinate transformation matrix.

Perspective transformation is similar to photographic imaging principle, which is to project the three-dimensional geometry of space to the two-dimensional plane, and the perspective view is very close to human vision. The visualization of reservoir model often uses perspective transformation, which can get more realistic display effect.

By setting the parameters of "camera," including position, upward direction, orientation, viewing angle, near plane, and far plane, the user can view the 3D scene from different angles and distances and realize the rotation, scaling, and translation of the whole scene, as shown in Figure 6.




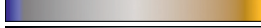

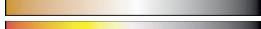


## 5. Efficient Visualization Method and Implementation of the Reservoir Model Based on WPF

In the study of geological engineering integration, engineering and geological data are combined and three-dimensional scenes of underground oil reservoirs are vividly displayed through the visualization technology, which can effectively strengthen multidisciplinary communication, improve the efficiency of technical decision-making, and provide an intuitive technical means for drilling process. In order to realize the visualization of geological reservoir data, the first step is to load the model file, which mainly includes grid and attribute data. The 3D visualization software is developed to display a large amount of data with high performance, which requires the use of efficient graphic interface and organization data. The developed software loads the grid produced by the previous hidden algorithm. Reading Egrid file is a cycle process. A data segment according to the keywords of the data segment is read each time, and the content of the data segment records  $s$  determined and carried out by different processing until all data segments are read.

Most of the previous researches on reservoir data visualization are based on C++ language and OpenGL graphics technology. Commercial software is based on Open Inventor, an advanced commercial software package of OpenGL, and some of them are developed with OSG or OpenGL directly. Windows, as the current mainstream desktop platform, can provide very powerful graphics functions and provides a powerful direct 3D graphics interface and a powerful direct 3D graphics interface for WPF development, which is in the leading position in the field of game development. However, due to historical reasons, there are few researches on reservoir data visualization based on direct 3D, which has not been reported.



TABLE 8: Color maps in software.

Name	Effect
Blue-green-red	
Rainbow	
Green-white	
Blue-brown	
Black-white	
Brown-black	
Orange-black	
Blue-orange	

This paper presents a high-performance 3D visualization solution based on Windows Presentation Foundation (WPF) technology and ModelView-ViewModel (MVVM) design mode and using helix, DirectX graphic interface using C# language. It can not only meet the requirements in display efficiency and effect but also greatly improve the usability of 3D control by applying MVVM mode, which can be easily embedded in WPF applications. This model is of great significance for the realization of software with higher quality and easier maintenance and migration.

This paper adopts the interface design engine WPF under. Net framework which inherits the framework element of WPF in the derived class to realize the specific drawing operation and makes a reasonable design. Helix toolkit is an advanced 3D development technology in .Net. It has a good balance between graphics performance and development efficiency and can meet the needs of most applications. Helix toolkit has two versions of WPF and sharpdx, both of which can perfectly integrate with WPF. The WPF architecture is shown in Figure 7.

MVVM mode divides the user interface into three parts: model, view, and view model according to different responsibilities, which makes the code hierarchical and easy to maintain and transplant. The main function of view layer is to display data, which is designed by XAML. The model layer is mainly used to process data and business logic. View model can process data in model and business logic in view at the same time. The workflow of MVVM logical structure is as follows: user operates view interface; view receives user's operation instructions; the operation instruction is sent to the view model at the same time, as shown in Figure 8. The view model updates the data of the model according to the operation instruction. After the model is updated, the notification is sent to the view model. The view model sends the interface update instruction to the view, and the view adjusts the view interface. MVVM logic structure has the characteristics of low coupling, reusability, and testability.

This software develops in strict accordance with MVVM mode so that the visualization function is completely decoupled from user interface and data model, and the ease of use and maintainability reach a new height.

Based on WPF technology and MVVM design pattern, the visualization function of hidden grid is realized by using

the high-performance 3D visualization solution of Helix and DirectX graphical interface, which can not only meet the requirements in display efficiency and effect, but also greatly improve the usability of 3D controls.

## 6. Application of Reservoir Model in Geological Engineering Integration

The software interface is shown in Figure 9, (a) is model parameter selection interface, (b) is importing file interface, and (c) is color bar selection interface. The software can display more than 20 attributes of reservoir model, such as tops, depth, porosity, permeability, and net to gross ratio. 3D display of well trajectory and reservoir of model 2 in Table 7 is shown in Figure 10. Figure 10 shows the visual effect comparison between the developed software and commercial software CMG. It can be seen that the software developed in this paper is accurate and efficient in showing reservoir model. In order to reduce the number of invalid grids and accelerate the display speed, the hidden algorithm proposed in the previous paper is used to display the reservoir model.

The three-dimensional graphics displayed by the two software pieces are basically the same, but the time of data import and drawing is far different. When the data is not processed, it takes a long time to import the data and draw the graph. According to the example, it takes about half a minute to import the data into the computer and get the image as shown in Figure 10; however, after data simplification, the amount of imported data is greatly reduced, and the import time is also greatly reduced. It takes about 10 seconds to import the same model into the same computer, and the image shown in Figure 10 is obtained. As can be seen from Figure 10(d), due to the great reduction of data to be processed in drawing, the drawing time of the whole process is greatly reduced, almost the data can be quickly imported, and the graphics can be calculated immediately, which greatly facilitates the graphic display, improves the display time, and improves the display efficiency.

It should also be mentioned here that the above calculation is the amount of data to be processed. In fact, there are many calculations inside these data when the actual comparison calculation is made, so the amount of data required for the calculation is much higher than the data obtained above. After processing and simplification, each data is not repeated, only once, so there is no need for internal calculation between data, so the actual amount of calculation and simplification effect will be better. In addition, as the number of grids increases, for example, next time, it will be  $100 \times 100 \times 100$  mesh, the number of simplification will be larger, and the effect of improvement will be more obvious.

Figure 11 shows the 3D reservoir attributes of model 1 and model 3 with different grid color and background color including static-gross ratio (NTG), porosity (PORO), vertical enlargement plus grid display, and grid rotation enlargement display.

TABLE 9: Well trajectory data.

Serial number	Measured depth (m)	Inclination (°)	Azimuth (°)	Vertical depth (m)	North-south displacement (m)	East-west displacement (m)
1	153.32	0.45	3.21	0.00	0.00	0.00
2	180.80	0.39	21.90	44.34	0.11	0.20
3	208.48	0.46	26.04	69.81	-0.21	0.02
...	...	...	...	...	...	...

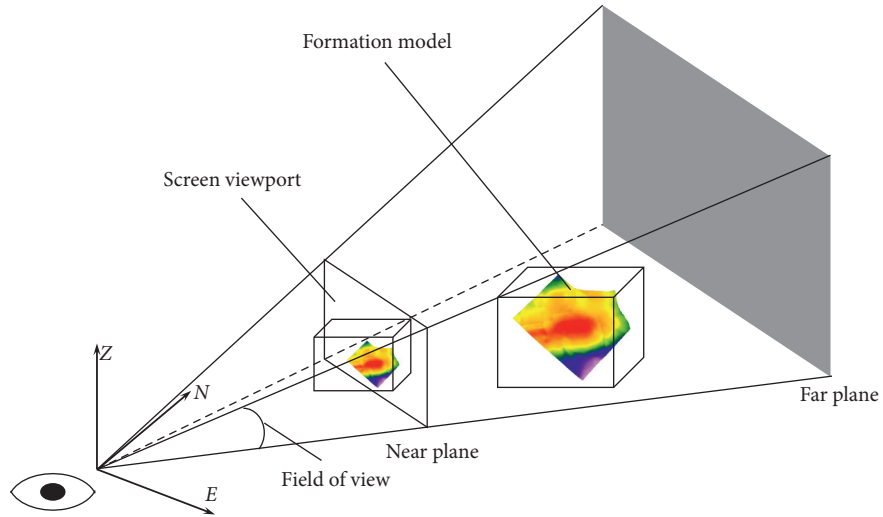


FIGURE 6: Perspective schematic diagram.

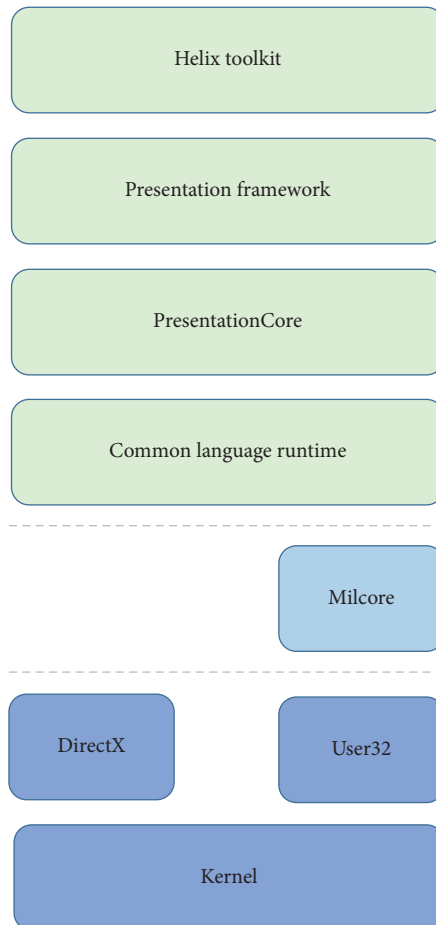


FIGURE 7: MVVM Mode diagram.

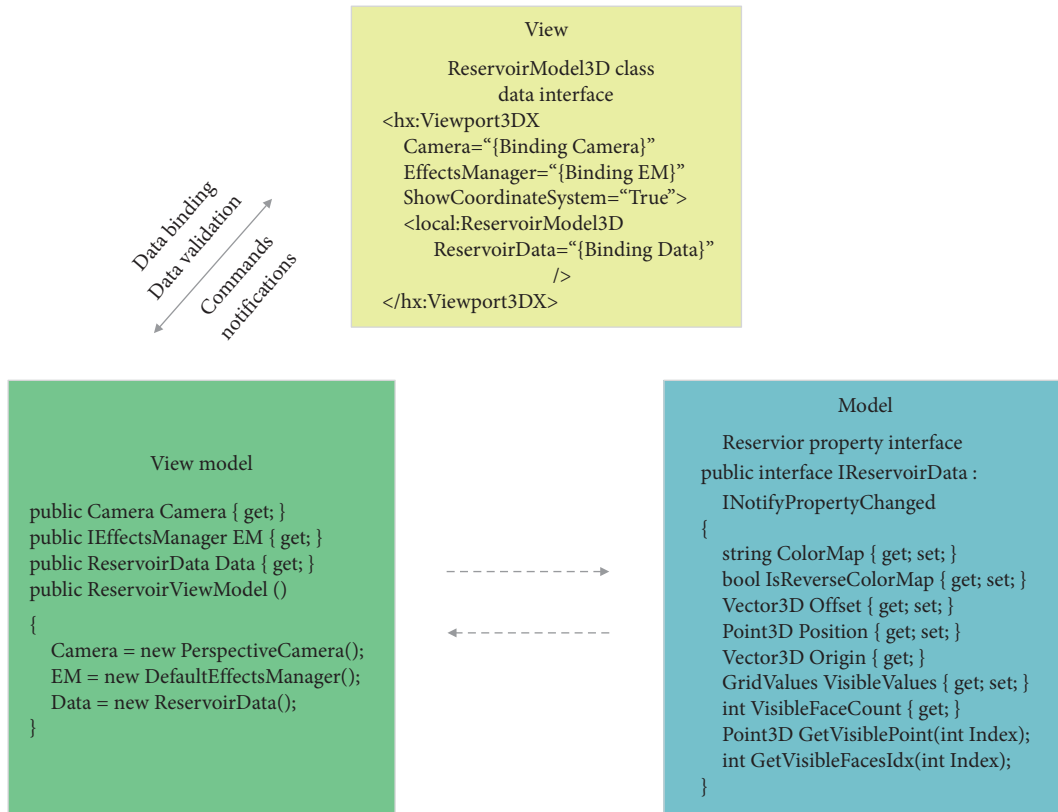
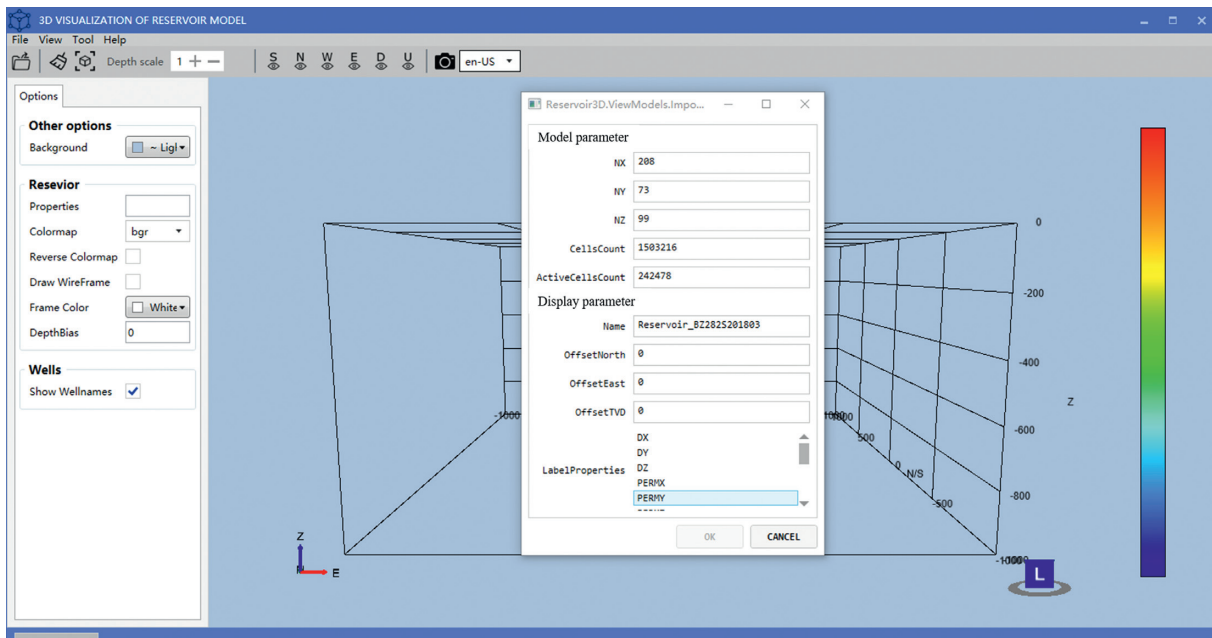
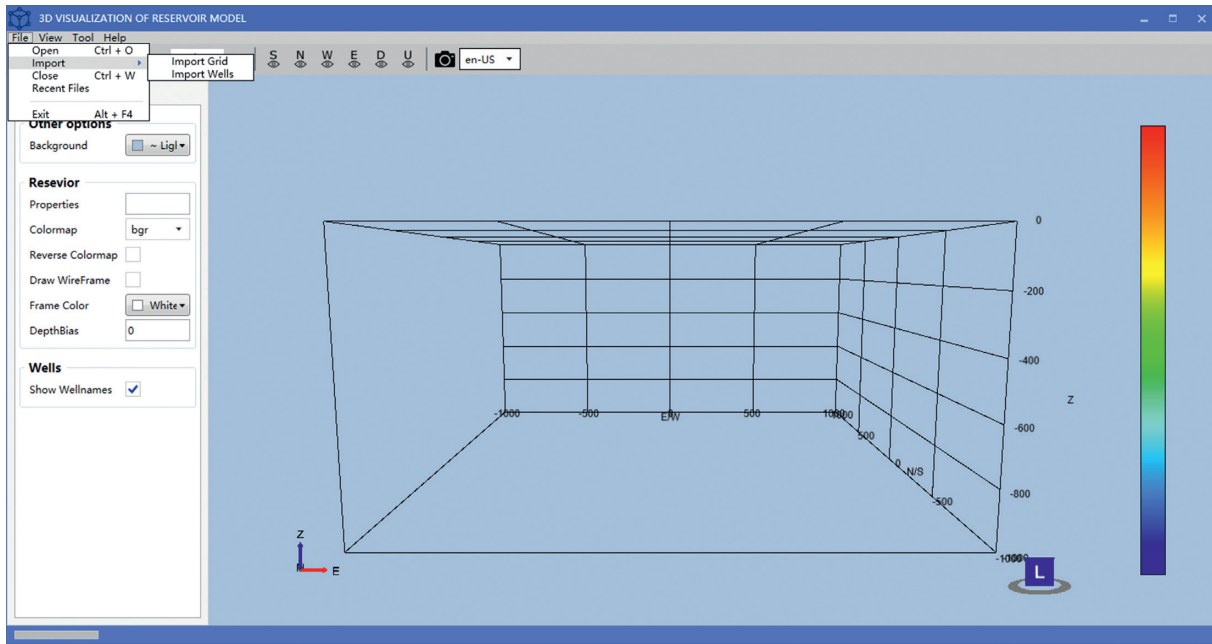


FIGURE 8: WPF architecture diagram.

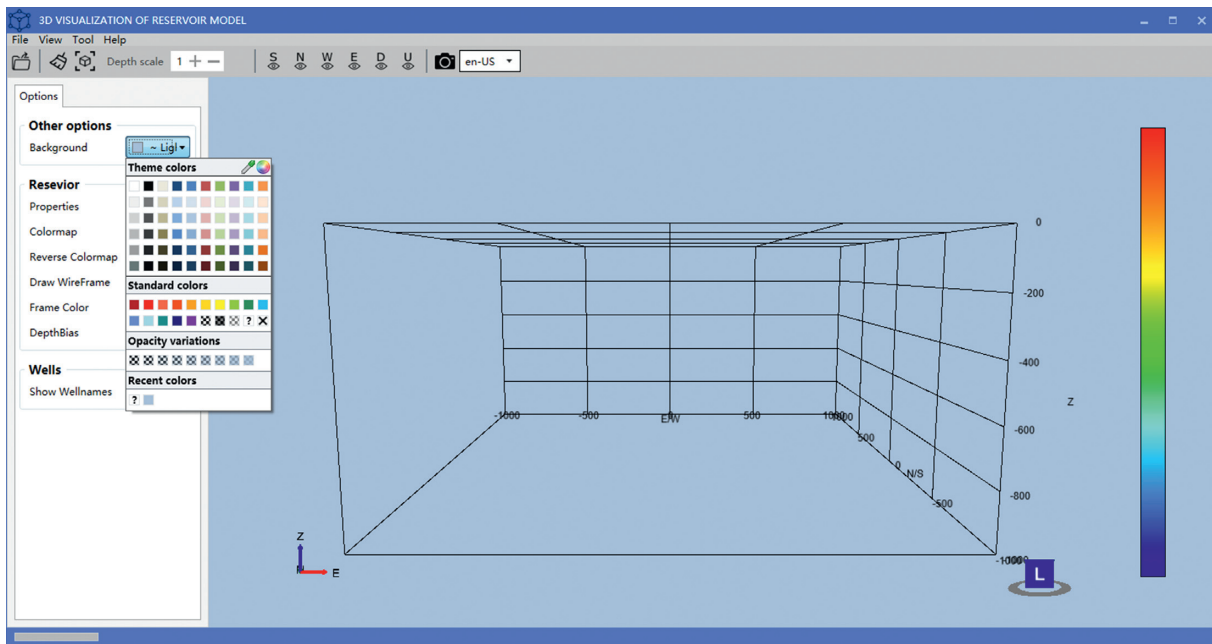


(a)

FIGURE 9: Continued.



(b)



(c)

FIGURE 9: Software interface. (a) Model parameter selection, (b) import file interface, and (c) color bar selection.

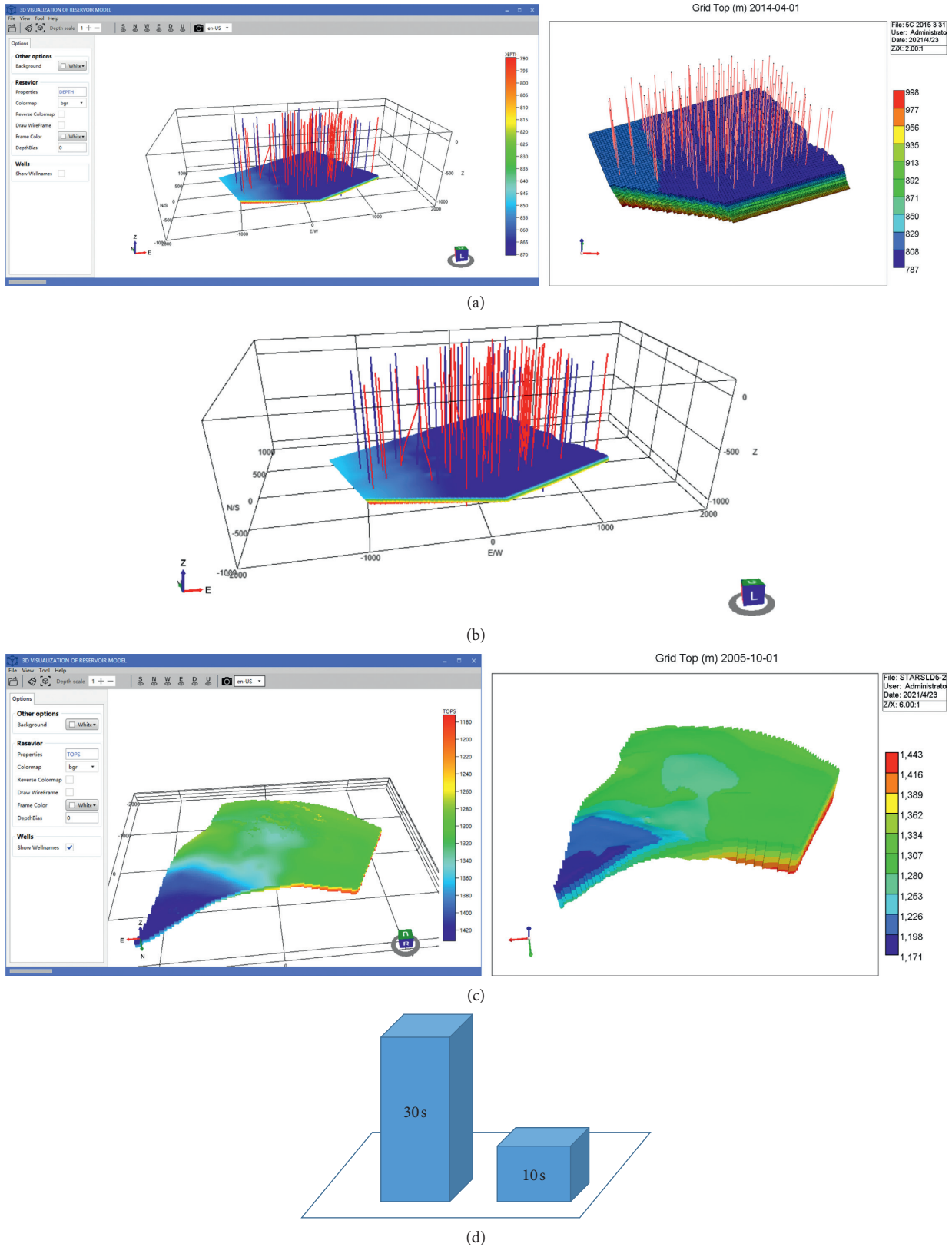
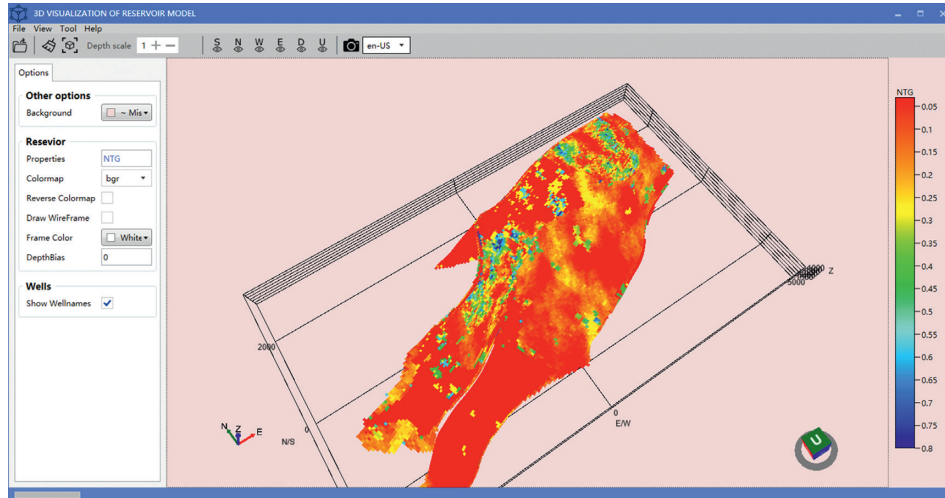
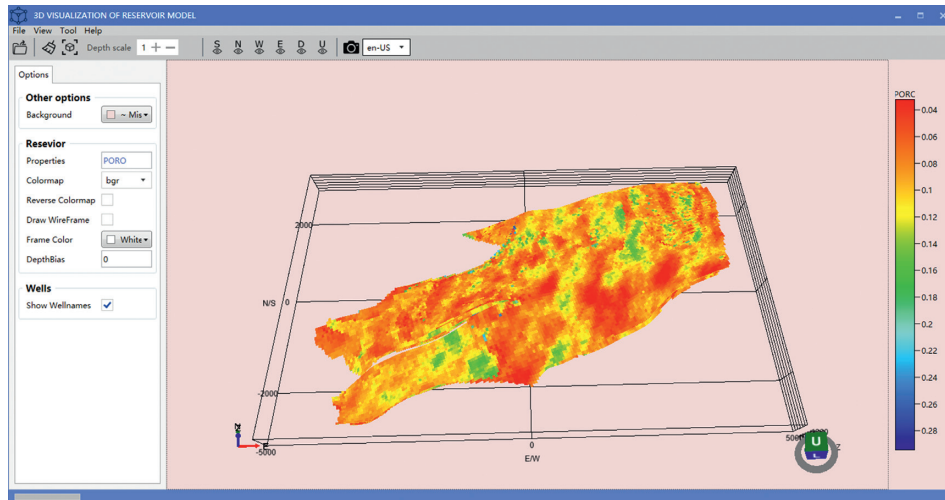


FIGURE 10: 3D display of well trajectory and reservoir of model 2 in Table 7. (a) Demonstrate reservoir grid model and well by CMG software (right) and the software developed in this paper (left), (b) save as image format and output effect, (c) display top attribute by CMG software (right) and the software developed in this paper (left), (d) and compare the time of drawing before (left) and after (right) grid processing.

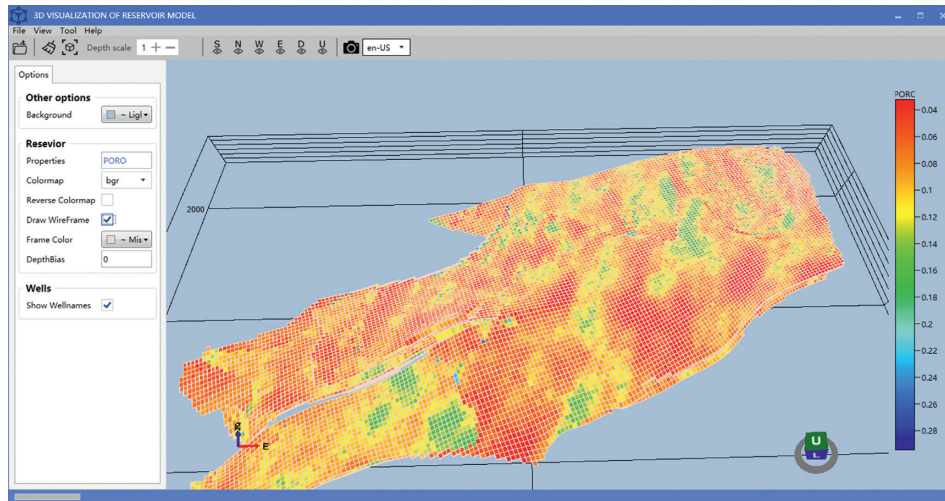




(a)

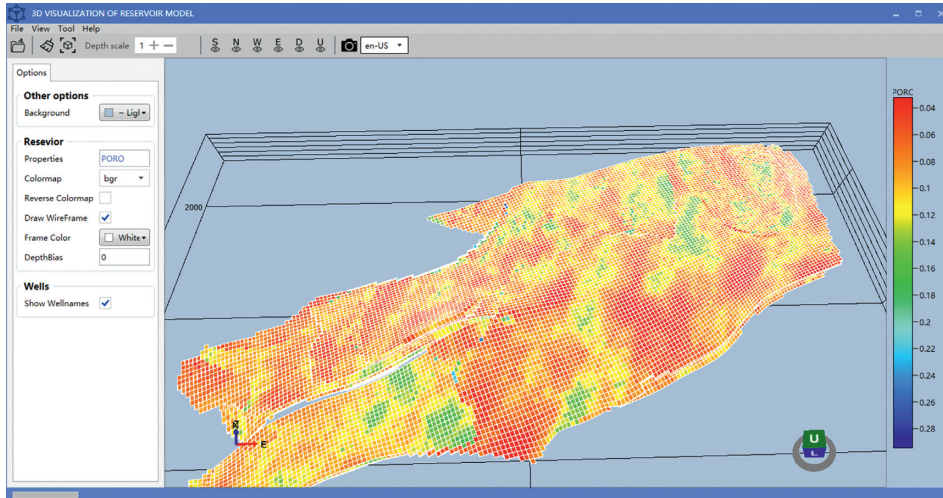


(b)

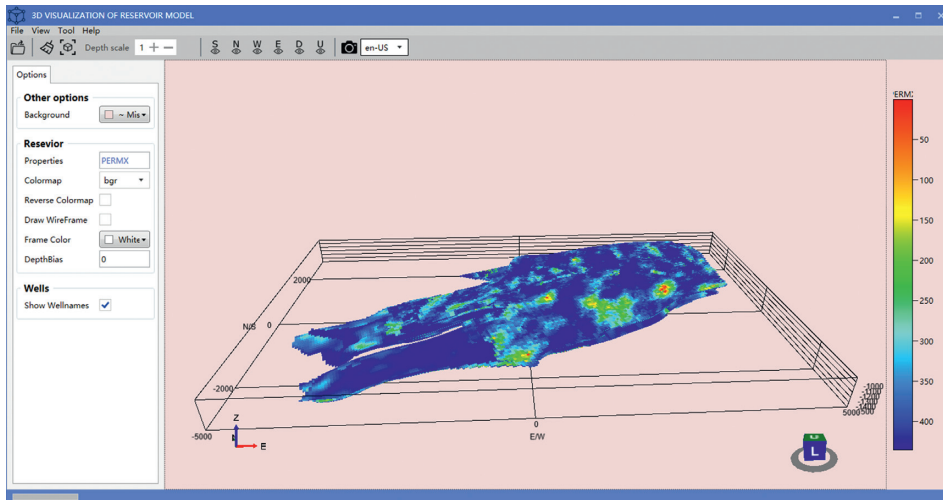


(c)

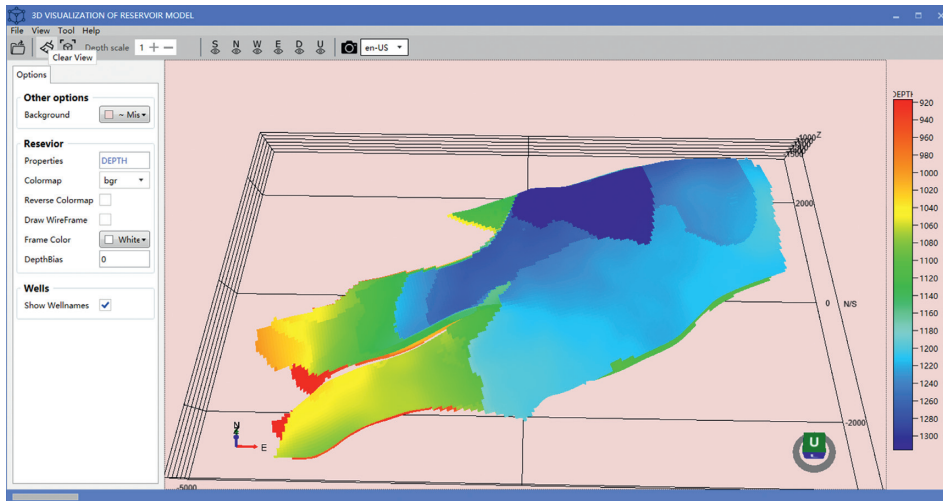
FIGURE 11: Continued.



(d)

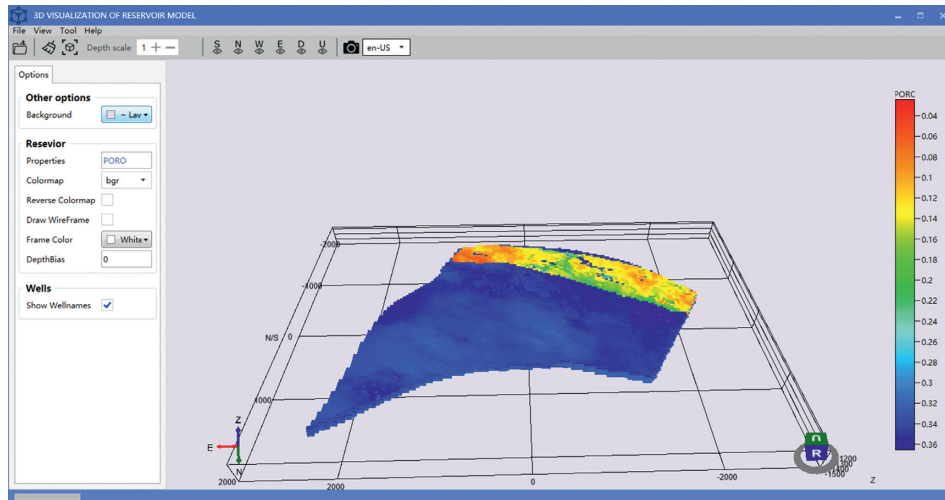


(e)

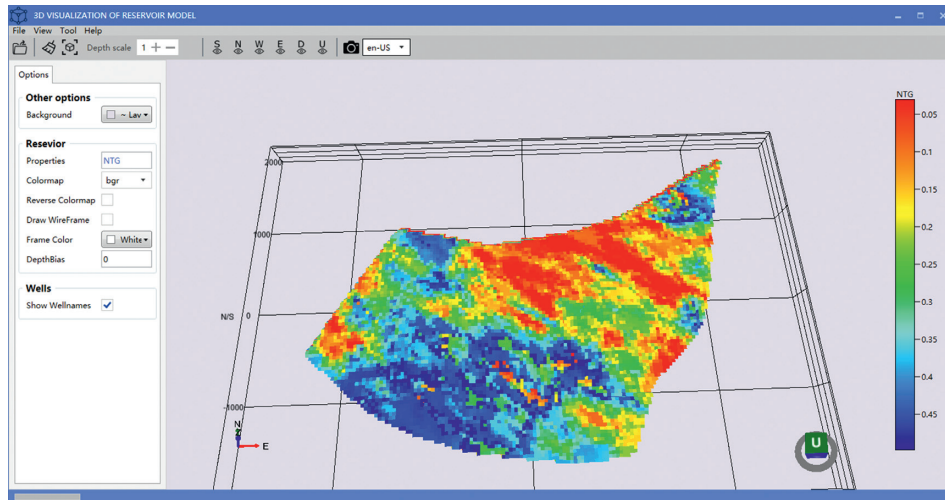


(f)

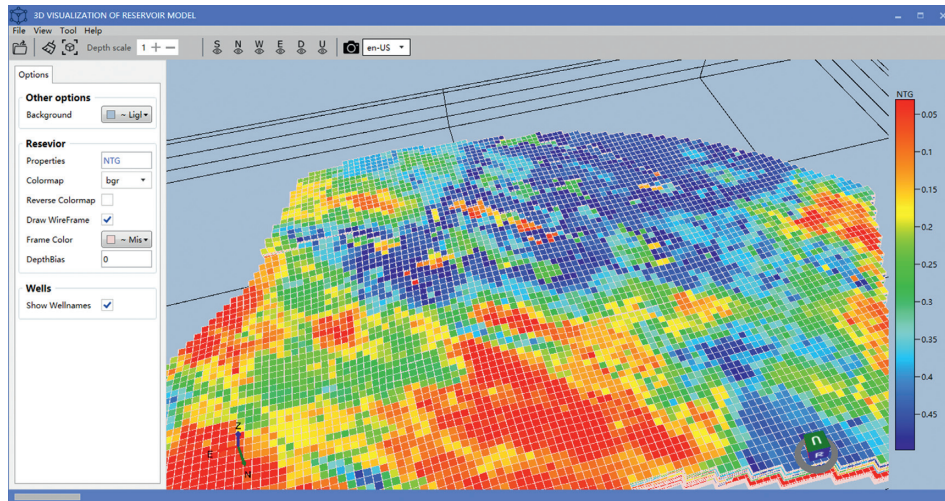
FIGURE 11: Continued.



(g)



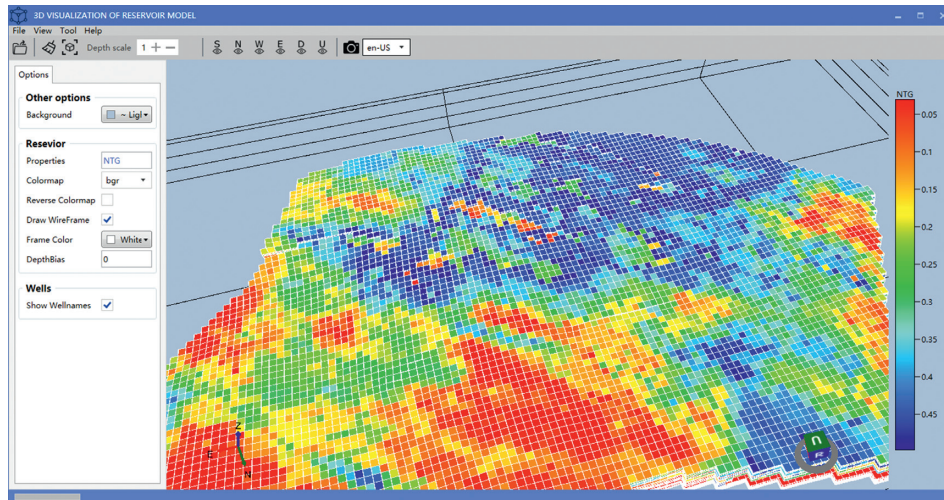
(h)



(i)

FIGURE 11: Continued.





(j)

FIGURE 11: 3D display of reservoir attributes: (a) display net to gross ratio (NTG) of model 3 in Table 7, (b) display porosity (PORO) of model 3 in Table 7, (c) display vertical magnification plus grid (wireframe: misty rose) of model 3 in Table 7, (d) display vertical magnification plus grid (wireframe: white) of model 3 in Table 7, (e) display permeability (PERM) of model 3 in Table 7, (f) display depth of model 3 in Table 7, (g) display porosity (PORO) of model 1 in Table 7, (h) display porosity (NTG) of model 1 in Table 7, (i) vertical magnification plus grid display (wireframe: misty rose) of model 1 in Table 7, and (j) vertical magnification plus grid display (wireframe: white) of model 1 in Table 7.

## 7. Conclusions

- (1) Aiming at the defects of efficient visualization of reservoir models at present, based on the analysis of data format and sorting rules of reservoir geological model grids, this paper puts forward blanking algorithms for reservoir geological model grids. These blanking algorithms based on topological relations greatly reduce the number of cells involved in the final calculation and improve the blanking efficiency. The sorting of unit face numbers directly eliminates the completely invisible grid inside, avoiding a large number of line and face search calculations. By sorting the depth of the cell plane, a large number of intersection and comparison calculations in the blanking algorithm are avoided, and the calculation amount is greatly reduced, thus achieving the goal of fast blanking. Reduce the demand of computer hardware for 3D visualization and improve the display efficiency.
- (2) On the basis of reservoir model, a 3D visualization software of reservoir based on WPF is developed, which realizes interactive operations such as roaming, zooming, moving, and viewpoint switching of reservoir model, and achieves good results. Practical application proves that the algorithm is simple, fast, and stable, and it is a more practical algorithm for hiding geological data grid, which lays a solid foundation for in-depth research and application of geological engineering integration technology.
- (3) Sharpdx and MVVM mode are used to develop 3D components; WPF encapsulates 3D development

related technologies, which makes the development of the system more convenient and can effectively shorten the development cycle. Compared with the traditional development methods, the performance, function, ease of use, and maintainability are greatly improved. The application shows that the system can accurately and in real time visualize the reservoir model data and well trajectory and runs smoothly, with high practical value. At the same time, all the technologies used are open-source technology, without any additional cost, which provides a new economic and efficient solution for the application of data visualization in petroleum industry.

## Data Availability

The data used to support the findings of the study is available from the first author upon request.

## Conflicts of Interest

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

## References

- [1] C. Z. Jia, M. Zheng, and Y. F. Zhang, "Unconventional hydrocarbon resources in China and the prospect of exploration and development," *Petroleum Exploration and Development*, vol. 39, pp. 129–136, 2012.
- [2] Z. X. Guo, P. Chen, and K. J. Zhou, "Application of 3-D visualization technique in petroleum drilling," *Natural Gas Industry*, vol. 24, pp. 60–65, 2004.

- [3] X. M. Ye, "Extraction of geological information and its conversion between different grid systems," *Journal of Xi'an Shiyou University*, vol. 33, pp. 44–48, 2018.
- [4] C. G. Xian, "Shale gas geological engineering integrated modeling and numerical simulation: present conditions, challenges and opportunities," *Oil Forum*, vol. 37, pp. 24–34, 2018.
- [5] Y. J. Zeng, "Integration technology of geology & engineering for shale gas development," *Petroleum Drilling Techniques*, vol. 42, pp. 1–6, 2014.
- [6] X. Z. Zhao, P. Q. Zhao, D. P. Li, X. Wu, W. C. Wang, and S. Z. Tang, "Research and practice of geology engineering integration in the exploration and development of dagang oilfield," *China Petroleum Exploration*, vol. 23, pp. 6–14, 2018.
- [7] J. Xie, "Rapid shale gas development accelerated by the progress in key technologies: a case study of the changning-wei yuan national shale gas demonstration zone," *Natural Gas Industry*, vol. 37, pp. 1–10, 2017.
- [8] J. Xie, "Practices and achievements of the changning-wei yuan shale gas national demonstration project construction," *Natural Gas Industry*, vol. 38, pp. 1–7, 2018.
- [9] G. X. Li, F. Wang, X. J. Pi, and H. Liu, "Optimized application of geology-engineering integration data of unconventional oil and gas reservoirs," *China Petroleum Exploration*, vol. 24, pp. 147–152, 2019.
- [10] W. R. Hu, "Geology-engineering integration A necessary way to realize profitable exploration and development of complex reservoir," *China Petroleum Exploration*, vol. 22, pp. 1–5, 2017.
- [11] X. H. Ma, "Natural gas and energy revolution: a case study of sichuan-chongqing gas province," *Natural Gas Industry*, vol. 37, pp. 1–8, 2017.
- [12] J. Gupta, M. Zielonka, R. A. Albert, A. M. Elrabaa, H. A. Burnham, and N. H. Choi, "Integrated methodology for optimizing development of unconventional gas resources," in *Proceedings of the SPE Hydraulic Fracturing Technology Conference*, Society of Petroleum Engineers, Woodlands, TX, USA, February 2012.
- [13] I. D. Stein, S. Magne, H. Håkon, and B. R. Alf, "Improving visualization of large scale reservoir models," in *Proceedings of the SPE Mathematical Methods in Fluid Dynamics and Simulation of Giant Oil and Gas Reservoirs*, Society of Petroleum Engineers, Istanbul, Turkey, September 2012.
- [14] H. L. Ji and J. X. Gao, "A summary of algorithms for removing the hidden lines and surfaces," *Computer and Digital Engineering*, vol. 9, pp. 27–31, 2006.
- [15] H. Q. Li, B. M. Chen, S. S. Xie, and X. D. Li, "The hidden technology of the continuous slice 3D reconstruction drawing process," *Application of Electronic Technique*, vol. 3, pp. 41–43, 2006.
- [16] X. L. Xia, "Hidden surface removal for 3D object," *Journal of Donghua University*, vol. 2, pp. 137–142, 2002.
- [17] X. C. Ma, X. L. Kong, and J. J. Chen, "A LOD technology for large 3D seismic data volume rendering," *Journal of Northeast Petroleum University*, vol. 32, pp. 23–26, 2008.
- [18] L. B. Shen, *The Research of Key Technologies in 3D Visualization for Geological Objects of Oilfield Exploration and Development*, Ocean University of China, Qingdao, China, 2010.
- [19] Y. B. Wu, Y. T. Zhang, and S. S. Liu, "3D visualized geologic modeling technique based on petrel," *Drilling & Production Technology*, vol. 5, pp. 65–66, 2007.
- [20] G. T. Li, *Research and Application on Key Technology of Reservoir Geological Data 3D Visualization*, China University of Petroleum, Beijing, China, 2017.
- [21] H. Qiao, A. Jia, and Y. S. Wei, "Geological information analysis of horizontal wells and 3D modeling of shale gas reservoir," *Journal of Southwest Petroleum University (Natural Science Edition)*, vol. 40, pp. 78–88, 2018.
- [22] W. Zhang, J. Z. Liu, S. B. Sun, C. Li, and X. Zhang, "Three-dimensional geological modeling of liugou oilfield and study on the visualization of it," *Journal of Xi'an Shiyou University*, vol. 25, pp. 28–31, 2010.
- [23] W. Zhao, *Research and Implementation of Three-Dimensional Visualization Steering System of Horizontal Well Base on Seismic Data*, China University of Geosciences, Beijing, China, 2010.
- [24] Z. Q. Huang, "3D visualization of hole trajectory in directional well drilling," *Journal of Xi'an Shiyou University*, vol. 24, pp. 79–82, 2009.
- [25] Y. X. Duan, Z. Q. Tong, Q. Li, Q. F. Sun, and H. Q. Li, "Wellbore visualization method for logging while drilling," *Journal of China University of Petroleum (Natural Science Edition)*, vol. 40, pp. 63–70, 2016.



## Research Article

# Utilizing Technology Acceptance Model for Influences of Smartphone Addiction on Behavioural Intention

Chih-Wei Lin <sup>1</sup>, Yu-Sheng Lin <sup>2</sup>, Chia-Chi Liao <sup>3</sup>, and Chih-Cheng Chen <sup>4,5</sup>

<sup>1</sup>Department of Leisure Services Management, Chaoyang University of Technology, Taichung 413310, Taiwan

<sup>2</sup>Physical Education Office, Chaoyang University of Technology, Taichung 413310, Taiwan

<sup>3</sup>Department and Graduate Institute of Applied English, Chaoyang University of Technology, Taichung 413310, Taiwan

<sup>4</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>5</sup>Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

Correspondence should be addressed to Chia-Chi Liao; [liao.chiachi@icloud.com](mailto:liao.chiachi@icloud.com) and Chih-Cheng Chen; [ccc@gm.cyut.edu.tw](mailto:ccc@gm.cyut.edu.tw)

Received 31 January 2021; Accepted 16 May 2021; Published 1 June 2021

Academic Editor: Cheng-Fu Yang

Copyright © 2021 Chih-Wei Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of this study was to establish a behavioural model of students' smartphone usage based on the perception of new advanced information technology, perceived playfulness, and smartphone addiction (SA). Undergraduate students were chosen to be the participants from a university in Taiwan. There were 814 valid questionnaires and the valid response rate was 81.4%. Firstly, SA positively and significantly affected perceived playfulness, perceived ease of use, and perceived usefulness. Secondly, perceived usefulness did not significantly influence behavioural intention. However, there was an unforeseen result; the effect of SA on perceived usefulness was barely significant. It can be concluded that the participants in the present study were undergraduate students and they might not intend to use smartphones for their academic performance. The findings indicated that undergraduate students experienced perceived playfulness, perceived ease of use, and perceived usefulness of smartphone from their addiction perception, which further implied that smartphone usage was interesting, easy, and useful. It was inferred that the undergraduate students were already under a high technology addiction (TA) condition. Suggestions indicate that the conjunction of teaching and mobile application should be extensively applied. It should be based on students' dependence on smartphone and smartphone's enjoyment to helpfully improve teaching via smartphones.

## 1. Introduction

The technology acceptance model (TAM) [1] is designed to explore the effects of external variables on perceived usefulness (PU), perceived ease of use (PEOU), attitude towards usage (A), behavioural intentions (BI), and actual behaviour (AB); the purpose is to discuss user's acceptance or rejection of using new information technology with two internal beliefs, which are PU and PEOU, and predict user's A accurately. In TAM, PU is defined as using a specific system which would improve a person's performance at work; PEOU is described as using a particular system which would be effortless [2].

Fishbein and Ajzen [3] pointed out that attitude is a result that comes from past learning experiences. Users

consistently have either likable or dislikeable behaviour on a specific thing; namely, attitude is an overall individual evaluation of specific people or a thing. In light of behavioural intention, Davis, Bagozzi, and Warshaw [1] defined BI as the users' willingness to continuously use a specific system or recommend it to others. An individual's subjective consciousness would decide the possibility of using information technology in the future as his/her AB. Davis [2] and Ahn et al. [4] have shown that PEOU positively affects PU. The connection between PEOU and PU on attitude towards usage shows that the outcomes are worth learning and affective cognition influences user's efficacy to perform [1].

According to Lu, Zhou, and Wang [5], the instant messaging software is a fast and convenient tool for communication and it is easy to use. It has attracted many users,

especially young people. Notably, users have a higher perception of usefulness and ease of use with regard to new technology or new service and his/her attitude towards using would tend to be positive. A is a complement to BI that originates from a positive effect [1]. Lin, Yang, Sia and Tang [6] proposed a smartwatch study, and the results indicated that it positively and significantly affected BI. Users believe that a smartwatch is worth using and his/her willingness is decided by his/her affective consciousness. According to mobile payment research (Lin et al., [7]; August), the researchers illustrated that PU and PEOU positively influenced BI. Users will have a high willingness to use a mobile system if they experience virtual mobile service on a smartphone with ease. Besides, the results of previous studies referring to the Internet, Facebook, instant messaging, and smartphone usage have shown a positive relationship between PU and BI [5, 8–10].

We now turn to the evidence on perceived playfulness. In 2001, Moon and Kim illustrated that the typical TAM does not extremely predict user's motivation; they indicated that much easier IT usage from computer systems will be regarded as a pleasant human-computer interaction [11]. Hence, they proposed a connection between PEOU and perceived playfulness (PP) when using the World Wide Web (www). Based on this, they divided PP into three dimensions: concentration (CON), curiosity (CUR), and enjoyment (ENT). They claimed that the easier the information technology is, the higher intention from the state of playfulness will be considered. According to the research results of user-created content services [12], playfulness is the main factor which determines a user's BI.

Chen, Gillenson, and Sherrell [13] indicated that PP plays a significant role in cyberspace and mobile services; it can stimulate a user's BI. Furthermore, Lin et al. [7] explained that PP, PEOU, and PU of smartwatch usage all positively influence A. Given that PP significantly affects A and BI when developing a new technology system, it further influences subsequent usage behaviour [14–17]. With regard to smartphone addiction (SA), the understanding of SA to date is similar to Internet addiction [18]. Griffiths [11] illustrated that Internet overuse can be regarded as pathological Internet usage or technological addiction. With the diverse features of smartphone, more and more young people have become dependent on or addicted to the smartphone functions. They not only are addicted to sending SMS via smartphones but also rely on other tools of the smartphone [19]. Khang, Kim, and Kim [20] defined SA as digital media addiction; the longer time a user spends on it, the higher the addiction to cyberspace. Leung and Wei [21] found that mobility, immediacy, and functionality were the main motivations that can be used to predict mobile user behaviour. When a user is not satisfied with using his/her smartphone, it can result in the asymmetry of usage time versus usage motive. Simultaneously, an individual's SA may distort his/her internal beliefs, which further increases smartphone usage time.

According to the results of an online auction addiction (eBay users) [22] and social networking websites addiction (Facebook users) [23], the level of addiction strengthens

users' PP perception. In other words, user's recognition of addiction is distorted by the level of how he/she is addicted to technology. Moreover, according to the results of eBay research, the level of online auction addiction significantly deepens the perception of usefulness and playfulness but slightly influences PEOU [22]. Additionally, concentration [15] shows that a user concentrates on a specific activity with playfulness and he/she ignores external interferences and cannot realise how fast time flies. That being said, SA can be described as a user's overly reliance on a smartphone and he/she further exhibits the unable to withdrawal symptom. Addiction is a mental condition that originated from substance abuse and substance dependence which results in overuse. Technology addiction (TA) is a new mental addictive situation which has been incorporated with different technology media. Turel, Serenko, and Giles [22] indicated that TA is a special type of behavioural addiction, which is a psychological dependency on IT usage, which twists the user's perceptions of usefulness, enjoyment, and ease of use towards the system and which makes users become addictive. According to Serenko and Turel [23], a user from a different technology web portal shows different TAs. Additionally, the result showed that Facebook users demonstrate higher TA symptoms than eBay users [22].

*1.1. The Purpose of the Study.* The TAM [1] has been widely used as the theoretical basis in various research fields. However, there are not enough studies using SA as the antecedent variable to strengthen PP, PEOU, and PU. Moreover, there is no direct relationship with attitude from the above-mentioned TA example. Given that, the main purpose of the present study is to bridge the gap and use the expanded TAM to test TA with regard to PP, PEOU, and PU on the smartphone which immerse users in cyberspace. Besides, participants of previous studies mainly focused on high school students. Balakrishnan and Raj [24] indicated that undergraduate students are a veritable group of high-risk Internet addiction. Accordingly, the present study developed a SA model and validated the relationships among all the research variables to understand the effects of university students' perceptions of smartphone usage (PP, PEOU, and PU) on BI through a mediator of positive attitude as shown in Figure 1.

*1.2. The Importance of the Study.* Nowadays, people have become more dependent on their smartphone. This phenomenon may lead to SA. This study has adopted the TAM to test PP, PEOU, and PU of a smartphone which engages undergraduate students with the surroundings. Most importantly, SA in the present study was an antecedent variable that deepens PP, PEOU, and PU. Moreover, the research data of PP, PEOU, and PU which made the participants be addicted to smartphone the most were analysed and the relationships among all variables were tested so as to comprehend the influence.

*1.3. Limitations.* First of all, owing to human resource limitation, the respondents were selected from a university in central Taiwan by purposive sampling. Hence, the

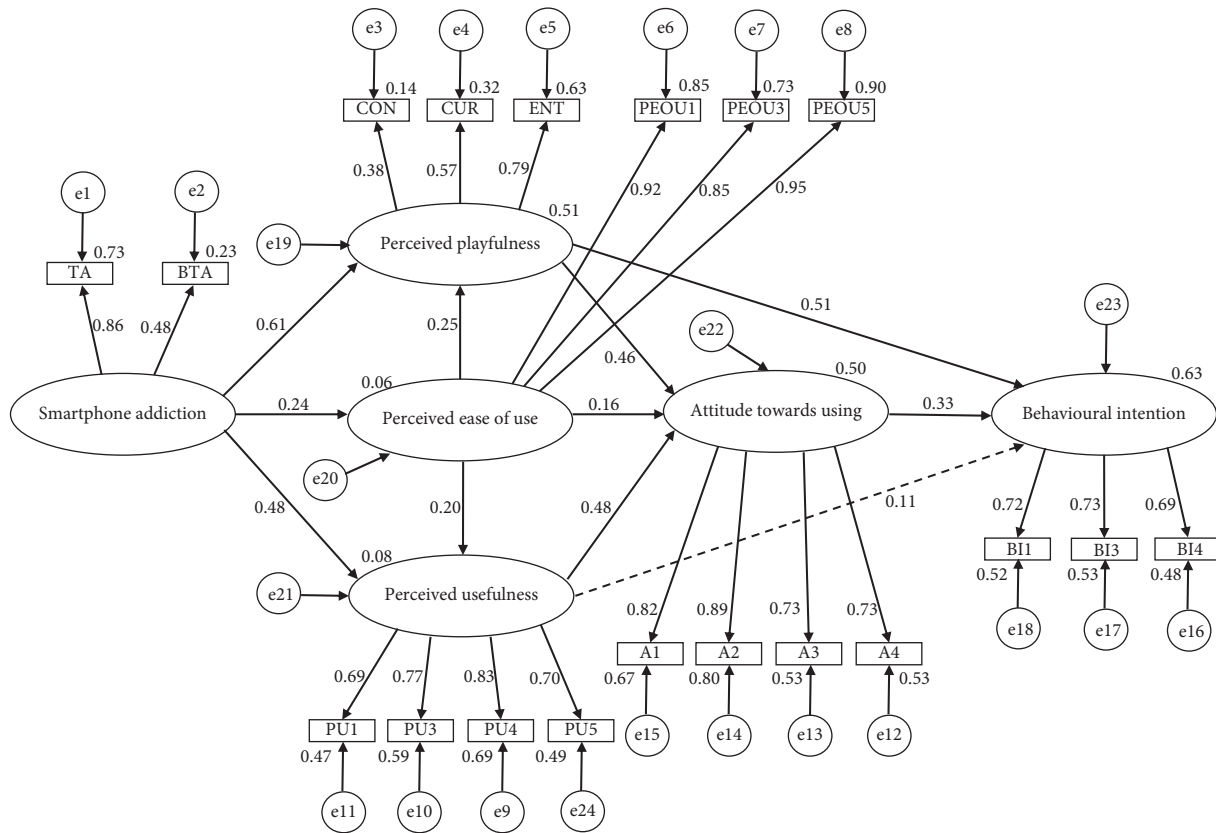


FIGURE 1: Path map of a smartphone addiction (SA) behavioural model. There is no significant difference between perceived usefulness (PU) and behavioural intention (PI) (dashed line).

research data in the present study cannot represent all Taiwanese students. Secondly, the definition of PP, PEOU, and PU does not specifically focus on the same usage motivation. Therefore, the research variable in the present study cannot represent all Taiwanese students. Finally, the function of a smartphone is changed with the advancement in science and technology; hence, the research data in the present study cannot be used to represent all smartphone functions in the future.

## 2. Method

The study was conducted via an online questionnaire survey which was adopted by purposive sampling. The undergraduate students who had smartphones with individual mobile Internet programmes were chosen. One thousand questionnaires were delivered randomly from a university in Taiwan. In total, 186 questionnaires were excluded due to incompleteness or incorrect answers, resulting in a sample of 814 adolescents (81.4% of valid responses). Participants were university students ( $N=814$ ; 201 male and 613 female); the students from 22 different departments were asked to participate. The introduction of the study was explained at first and the research motivation clarified that the questionnaire was designed to measure students' recognition of playfulness, ease of use, and usefulness that were related to their smartphone. There were 38 questions in total in this questionnaire and it took

5–10 minutes to complete. All participants were told that all comments will only be used for academic research, not open to the public.

**2.1. Measures.** Each scale consisted of five items, which were responded to on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). Reliability and validity were tested by confirmatory factor analysis (CFA). Firstly, for SA, the TA scale [23] was adapted to measure students' perceptions of TA and behavioural technology addiction (BTA). The factor loading for SA was from 0.61 to 0.78; composite reliability (CR) was 0.86; and average variance extracted (AVE) was 0.46. Secondly, for PP, the www context scale [15] was adapted for students' perceptions of CON, CUR, and ENT. The factor loading for PP was from 0.41 to 0.93; CR was 0.91; and AVE was 0.54. Thirdly, for PEOU and PU, the innovative use of smartphone scale [9] and mobile services quality scale [25] were used to assess students' perceptions of smartphone's ease of use and usefulness at school for academic performance. The factor loading for PEOU was from 0.85 to 0.98; CR was 0.93; and AVE was 0.83. The factor loading for PU was from 0.67 to 0.84; CR was 0.83; and AVE was 0.56. Finally, for A and BI, the www context scale [15] and innovative use of smartphone scale [9] were used to measure students' perceptions of positive and negative evaluations and future willingness. The factor loading for A was from 0.71 to 0.92; CR was 0.87;

and AVE was 0.63. The factor loading for BI was from 0.69 to 0.77; CR was 0.77; and AVE was 0.52.

**2.2. Data Analysis.** Firstly, descriptive statistics was used to understand the distribution of demographic information. This study used the frequency allocation table, percentage, average, and standard deviation to explore the current situation of TAM, PP, and SA. Furthermore, CFA and structural equation model (SEM) with SPSS AMOS 23 were used. Model adequacy checking was evaluated by using absolute fit indices, relative fit indices and parsimonious fit indices which contain root mean square residual (RMR), goodness of fit index (GFI), error of approximation (RMSEA), comparative fit index (CFI), normed fit index (NFI), parsimonious normed fit index (PNFI), chi-square ( $\chi^2$ )/degree of freedom (df) ratio, and Hoelter's critical N (CN). A cut-off value greater than 0.90 was for GFI, CFI, and NFI; a cut-off value lesser than 0.50 was for the RMR; a cut-off value greater than 0.50 was for the PNFI; cut-off values lesser than 0.08 or greater than 0.05 were for the RMSEA, indicating acceptable model fit [26]; a cut-off value lesser than 5 was for the chi-square ( $\chi^2$ )/degree of freedom (df) ratio; and a cut-off value greater than 200 was for the CN.

### 3. Results

**3.1. Sample Distribution.** There were 814 participants in total, out of which 201 were male (24.7%) and 613 were female participants (75.3%). In light of the division, over 700 respondents (86%) studied in day school. The education background of most of the participants was freshman (195, 24%). As for academy, 327 (40.2%) respondents studied in College of Management. In terms of department, there were 22 departments in this university. The highest response was from the "Department and Graduate Institute of Finance," with 60 people (7.4%). Conversely, the lowest one was the "Department of Landscape and Urban Design," with 7 people (0.9%).

SEM was used to examine a SA behavioural model with direct paths. Firstly, the paths were from the dimensions of SA (TA and BTA) to PP, PEOU, and PU. Secondly, the paths were from PP (CON, CUR, and ENT), PEOU, and PU to A. Thirdly, the paths were from PEOU to PP and PU. Finally, the paths were from PP (CON, CUR, and ENT), PU, and A to BI. Besides, the indirect effects from the dimensions of SA on BI were also examined. After each variable model was examined, the results showed that SA indirectly influenced BI via the mediators of PP, PEOU, and PU. The structural model exhibited a good model of fit. Given that, the final SA behavioural model and standardised coefficients are shown in Figure 1; the fit test indices are shown in Table 1.

Table 1 shows that PEOU and BI were saturated; PNFI values of PU and A were lower than 0.5. Besides,  $\chi^2/df$  value of each variable was lower than 5; GFI value, CFI value, and NFI value of each variable were higher than 0.9; RMR value of each variable was lower than 0.05; RMSEA value of each variable was lower than 0.08; CN value of each variable was higher than 200, which reached the standard. Overall, the

TABLE 1: Fit test indices.

Variable	RMR	GFI	RMSEA	CFI	NFI	PNFI	$\chi^2/df$	CN
PU	0.01	0.99	0.07	0.99	0.99	0.33	5.30	460
A	0.01	0.99	0.06	0.99	0.99	0.33	4.26	572
PP	0.03	0.97	0.06	0.97	0.96	0.64	4.04	306
SA	0.04	0.98	0.06	0.98	0.97	0.60	3.85	363
Overall	0.05	0.98	0.02	0.99	0.98	0.74	1.26	646

Note. PEOU and BI were saturated.

goodness of fit test of the SA behavioural model conformed to the standard.

Figure 1 and Table 2 display that A directly affected BI (with a path coefficient of 0.330). Moreover, PP directly and indirectly influenced BI. However, PU only indirectly affected BI (with a path coefficient of 0.116). Oum and Han [12] indicated a nonstatistically significant relationship between PU and BI; the result supports the notion that there is no significant difference between PU and BI (with a path coefficient of 0.110). Thirdly, PEOU indirectly affected BI via A, PP, and PU. Finally, SA indirectly influenced BI through PP, PEOU, and PU. Besides, comparing three independent variables of the total effect on BI, the most powerful independent variable was PP with an effect of 0.662, followed by PEOU with 0.263; the weakest one was PU with 0.226. However, as for the effect of the antecedent variable (SA) on BI, PP was the most influential with the indirect effect of 0.404, whereas PU was the weakest with 0.038.

### 4. Discussion

Firstly, according to Table 3, PEOU was the highest item, indicating that the undergraduate students could mostly use their smartphone with no difficulty. Hence, ease of use of smartphone could be an influential factor which helps a student improve his/her life or task. Moreover, different perceptions of ease of use were found. For example, PEOU1 represented that most of the participants could use their smartphone effortlessly. On the contrary, PEOU3 represented that most of the undergraduate students could use their smartphone skilfully.

Secondly, the total effects on PU towards the undergraduate students were satisfactory. Notably, the undergraduate students generally felt that their smartphone was useful. Therefore, the usefulness of smartphone could be an influential factor which helped the student enhance his/her life or task. In addition, the highest item PU5 illustrated that smartphones made the participants feel helpful, whereas the lowest item PU1 indicated that improving their academic performance via smartphone was acceptable. Thirdly, the total cognitions on A pointed out that undergraduate students had a positive and pleasant attitude when using their smartphones. Besides, A3 with the highest score showed that smartphone usage made the respondents feel pleasant. On the contrary, A4 was the lowest one, showing that there was a moderate recognition of positive idea for using smartphones. The undergraduate students mostly assented to participate in using smartphones again. Furthermore, BI3 showed that the participants had a mid-to-high cognition of



TABLE 2: Direct effects, indirect effects, and total effects of each variable on behavioural intention.

Antecedent variable	Independent variable		Mediator variable	Direct effect	Indirect effect	Total effect
	PP	—	A	0.330		0.330
				0.510		0.662
	PEOU	PP	A		0.152	0.263
			A		0.053	
	PU	—	PU		0.165	0.226
					0.045	
			A	0.110	0.116	
SA		PP			0.404	0.505
		PEOU			0.063	
		PU			0.038	

TABLE 3: Average and standard deviation of each variable ( $n = 814$ ).

Variable	Item/aspect	Mean	SD
PEOU	PEOU1	4.43	0.66
	PEOU3	4.35	0.72
	PEOU5	4.42	0.67
	Overall	4.40	0.64
PU	PU1	3.26	0.84
	PU3	3.60	0.97
	PU4	3.29	0.93
	PU5	3.74	0.82
	Overall	3.47	0.73
A	A1	3.84	0.76
	A2	3.69	0.80
	A3	3.88	0.80
	A4	3.50	0.81
	Overall	3.73	0.67
BI	BI1	3.78	0.84
	BI3	3.79	0.85
	BI4	3.41	0.94
	Overall	3.73	0.67
PP	CON	3.08	0.78
	CUR	3.44	0.73
	ENT	3.86	0.70
	Overall	3.46	0.55
SA	TA	3.62	0.70
	BTA	2.81	0.87
	Overall	3.16	0.68

frequently using smartphones in the future. BI4 was with the lowest but a medium score.

Fourthly, the overall effects on PP were moderate. Besides, the highest ENT showed that the playfulness of smartphones could make the undergraduate students feel happy, whereas the lowest CON indicated that the undergraduate students were conscious when using smartphones. Finally, the overall average of SA was moderate and the undergraduate students had a mid-to-low-end recognition of TA. In other words, the undergraduate students had awareness of smartphone usage and they could restrict themselves from being addicted. In addition, TA was higher than BTA, which represented that the meaning of TA could make better impressions on the participants. However, the overall average of TA and BTA was not ideal. It was inferred

that the items of TA originated from a compulsive buying tendency. People may have salience, withdrawal, conflict, relapse, and mood modification symptoms when shopping online. Moreover, SA in the present study was used to test the undergraduate students who may have the same symptoms when using smartphones. Therefore, the undergraduate students may have a higher recognition for TA to answer TA. Furthermore, it was assumed that the items of BTA originated from the eBay environment and were used to test for shopping addiction. People may have obsessive-compulsive disorder while online shopping and might be sensitive about the price. Nevertheless, SA in the present study was not used to test the undergraduate students' purchasing power via their smartphones. Therefore, the undergraduate students may have a lower recognition for BTA and the items may not be completely suitable for answering TA. Thus, it was suggested that TA and BTA should be tested separately in the future.

*4.1. Mediator Effects of PP, PEOU, and PU on BI via A.* Attitude was influenced by PP, PEOU, and PU in the present study; it is a mediator variable which affects BI. Besides, the results indicated that PP, PEOU, and PU indirectly influenced BI through A. It can be inferred that undergraduate students are pleasant and satisfied when they are using their smartphones; they are relatively pleased and have active playfulness of their smartphone. A joyful smartphone recreation function may enhance the pleasure in students' life and strengthen the degree of devoting time to their life. Therefore, the most influential direct variable was PP with an effect of 0.510. However, the effect of PP on BI via A was 0.152 as shown in Table 2. It was assumed that the diversified developments of mobile applications combine with smartphone enjoyment so as to enhance the user's dependence on smartphones. Additionally, PEOU and PU directly influenced A with an effect of 0.16 and 0.35, respectively. These results are supported by the notion that was proposed by Lin et al. [7].

Nevertheless, both PEOU and PU adjusted BI through A; the outcomes showed a slightly indirect effect (PEOU with an effect of 0.263 and PU with 0.226). It was concluded that university students spend one-third of their time in the learning environment at school. However, the case of



teaching via smartphone function by teachers is rare. Therefore, university students cannot properly recognise the practical utility of smartphones in their life or learning environment. Moreover, education in Taiwan has recently popularised online learning, distance learning, massive open online courses, flipped classroom, and digital teaching materials compiled on campus in order to integrate with smart learning. It encourages people to learn unlimitedly through the technical function of smartphones or tablets.

*4.2. Effects of SA on PP, PEOU, and PU.* The results verified that a higher SA perception enhances a smartphone's performance, especially playfulness; it leads the undergraduate students to appreciate their smartphone and further continue to use their smartphone. Besides, it was inferred that the undergraduate students were already under a high TA condition before they sensed PP, PEOU, and PU. However, the level of SA did not totally deepen PP, PEOU, and PU. The effects of SA on PU were slightly significant. It was concluded that TA of mobile device users and TA of cyberspace users may exhibit different TA symptoms from different using motivations. Notably, the level or the influence of TA should be determined by the IT feature. With regard to PP, it is always the main factor of using cyberspace or technology product [13, 15]. When a user has a cognitive bias by overusing or abusing smartphone, his/her dependent behaviour will result in addictive symptom. In light of this, PP was shown to be the most powerful variable in this study; the result supported the notion that PP has the strongest effect than PEOU and PU. If smartphone usage cognition has a positive relationship with BI, the students will believe that the smartphone is beneficial and they will use their smartphones carefully. As pointed out in the introduction of this paper, there is a significant connection among PP, PEOU, and PU. The higher PP, PEOU, and PU are, the higher BI is. As expected, the results support this standpoint. The outcome in this study explained that total effect of SA on PP, PEOU, and PU was 0.505 as shown in Table 2.

According to online auction addiction [22] and social networking websites addiction [23], both studies explained that addiction distorts users' perceptions of usefulness and enjoyment. However, the effect of addiction on PEOU was weak [22]. Serenko and Turel [23] indicated that social networking website addiction distorted technology perceptions. Comparing the two results, it can be seen that using social networking website is easier than using online auction website. Basically, Facebook provides its users with a hedonic platform, whereas online auction website provides more complex information (e.g., payment process). Hence, it was concluded that social networking website users have a stronger potential to show TA symptoms. Based on the previous findings, the notions proposed by Turel et al. [22] and Serenko and Turel [23] are supported. Due to the different IT usages, the model of using smartphone properly, to find the solutions or use smartphone improperly to solve the problem, was verified in this study. Finally, in the present study, the results support the notion that addiction positively influences PP, PEOU, and PU.

## 5. Conclusions and Theoretical Implications

The result of this study showed that SA successfully mixed with the TAM model. The purpose of this study was to comprehend what SA stood for when it was combining with the TAM model. Moreover, the results verified that higher SA perception enhanced smartphone's performance, especially playfulness; it led the undergraduate students to have an appreciation on their smartphone and further continue to use their smartphone. However, SA should be efficiently faced; otherwise, it would distort the way people interact with technology systems. The undergraduate students perceived PP, PEOU, and PU of smartphone from their addiction perception which further caused them to feel smartphone using that was interesting, easy, and useful. According to research results, it was inferred that the undergraduate students were already under a high TA condition before they sensed PP, PEOU, and PU perception. Keep using smartphone that results in physical problems.

In the present study, the level of SA did not totally deepen the perceptions of PP, PEOU, and PU. The effects of SA on PU were slightly significant. Comparing with Facebook addiction [23], the addiction results were different. Hence, it was concluded that TA of 3C mobile device user and TA of cyberspace user may exhibit different TA 114 symptoms from different using motivations. Namely, the level or the influence of TA should be determined by the IT feature. Moreover, the influences of PEOU and PU on smartphone in the present study were different from the TAM model. The results showed that PU could not directly affect BI; it had to be decided via attitude towards using. Namely, there was a disagreement between undergraduate students' subjective appraisal of performance on smartphone and smartphone itself. However, the TAM model combining with SA enhanced the effects from antecedent variable. Hence, for multimedia materials teaching class, special attention should be paid to students who may have a symptom of addiction.

## 6. Suggestions

- (1) This study focused on the effects of SA on behavioural intention. It is suggested that future studies should first investigate smartphone usage's purpose from the participants; it may be helpful for future researches to enlarge the research range so as to obtain more accurate research data.
- (2) It was recommended that future studies should collect the questionnaires from different universities or different cities in Taiwan in order to propose different viewpoints of SA on behavioural intention.
- (3) It is suggested that SA caused by playfulness from the university students should be weakened, whereas the perception of dependence on using smartphone in their life and learning environment should be strengthened.

- (4) It is proposed to upgrade the Internet facilities on campus and allow teachers to practice digital teaching materials and other related study courses.

## Data Availability

All data generated or analyzed during this study are included within this manuscript and are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments




The work was partially supported by the National Science Council of Taiwan under Grant MOST 107-2221-E-507-002-MY3.

## References

- [1] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: a comparison of two theoretical models," *Management Science*, vol. 35, no. 8, pp. 982–1003, 1989.
- [2] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.
- [3] M. Fishbein and I. Ajzen, *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*, Addison-Wesley, Reading, MA, USA, 1975, [https://www.researchgate.net/publication/233897090\\_Belief\\_attitude\\_intention\\_and\\_behaviour\\_An\\_introduction\\_to\\_theory\\_and\\_research](https://www.researchgate.net/publication/233897090_Belief_attitude_intention_and_behaviour_An_introduction_to_theory_and_research).
- [4] T. Ahn, S. Ryu, and I. Han, "The impact of web quality and playfulness on user acceptance of online retailing," *Information & Management*, vol. 44, no. 3, pp. 263–275, 2007.
- [5] Y. Lu, T. Zhou, and B. Wang, "Exploring Chinese users' acceptance of instant messaging using the theory of planned behavior, the technology acceptance model, and the flow theory," *Computers in Human Behavior*, vol. 25, no. 1, pp. 29–39, 2009.
- [6] C. W. Lin, C. C. Yang, W. Y. Sia, and K. Y. Tang, "Examining the success factors of smart watch: a behavioral perspective on consumers," *Polish Journal of Management Studies*, vol. 20, 2019.
- [7] C. W. Lin, S. S. Lee, K. Y. Tang, Y. X. Kang, C. C. Lin, and Y. S. Lin, "Exploring the users behavior intention on mobile payment by using TAM and IRT," in *Proceedings of the 2019 3rd International Conference on E-Society, E-Education and E-Technology*, pp. 11–15, Taipei Taiwan, August 2019.
- [8] S. Lee and M. Cho, "Social media use in a mobile broadband environment: examination of determinants of twitter and facebook use," *International Journal of Mobile Marketing*, vol. 6, no. 2, pp. 71–87, 2011.
- [9] Y. Park and J. V. Chen, "Acceptance and adoption of the innovative use of smartphone," *Industrial Management & Data Systems*, vol. 107, no. 9, pp. 1349–1365, 2007.
- [10] T. S. H. Teo, V. K. G. Lim, and R. Y. C. Lai, "Intrinsic and extrinsic motivation in internet usage," *Omega*, vol. 27, no. 1, pp. 25–37, 1999.
- [11] M. Griffiths, "Does internet and computer "addiction" exist? some case study evidence," *CyberPsychology & Behavior*, vol. 3, no. 2, pp. 211–218, 2000.
- [12] S. Oum and D. Han, "An empirical study of the determinants of the intention to participate in user-created contents (UCC) services," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15110–15121, 2011.
- [13] L.-D. Chen, M. L. Gillenson, and D. L. Sherrell, "Enticing online consumers: an extended technology acceptance perspective," *Information & Management*, vol. 39, no. 8, pp. 705–719, 2002.
- [14] R. Agarwal and E. Karahanna, "Time flies when you're having fun: cognitive absorption and beliefs about information technology usage," *MIS Quarterly*, vol. 24, no. 4, pp. 665–694, 2000.
- [15] J.-W. Moon and Y.-G. Kim, "Extending the TAM for a worldwide-web context," *Information & Management*, vol. 38, no. 4, pp. 217–230, 2001.
- [16] H. Van Der Heijden, "User acceptance of hedonic information systems," *MIS Quarterly*, vol. 28, no. 4, pp. 695–704, 2004.
- [17] J. Zhang and E. Mao, "Understanding the acceptance of mobile SMS advertising among young Chinese consumers," *Psychology and Marketing*, vol. 25, no. 8, pp. 787–805, 2008.
- [18] M. Kwon, J. Lee, W. Won et al., "Development and validation of a smartphone addiction scale (SAS)," *PLoS One*, vol. 8, no. 2, pp. 1–7, 2013.
- [19] M. C. Bian, "Linking psychological attributes to smart phone addiction," Hong Kong, China, The Chinese University of Hong Kong, 2012, [http://pg.com.cuhk.edu.hk/pgp\\_nm/projects/2012/BIAN%20Mengwei%20Casey.pdf](http://pg.com.cuhk.edu.hk/pgp_nm/projects/2012/BIAN%20Mengwei%20Casey.pdf).
- [20] H. Khang, J. K. Kim, and Y. Kim, "Self-traits and motivations as antecedents of digital media flow and addiction: the internet, mobile phones, and video games," *Computers in Human Behavior*, vol. 29, no. 6, pp. 2416–2424, 2013.
- [21] L. Leung and R. Wei, "More than just talk on the move: uses and gratifications of the cellular phone," *Journalism & Mass Communication Quarterly*, vol. 77, no. 2, pp. 308–320, 2000.
- [22] O. Turel, A. Serenko, and P. Giles, "Integrating technology addiction and use: an empirical investigation of online auction users," *MIS Quarterly*, vol. 35, no. 4, pp. 1043–1062, 2011.
- [23] A. Serenko, O. Turel, and O. Turel, "Integrating technology addiction and use: an empirical investigation of facebook users," *AIS Transactions on Replication Research*, vol. 1, pp. 1–18, 2015.
- [24] V. Balakrishnan and R. G. Raj, "Exploring the relationship between urbanized Malaysian youth and their mobile phones: a quantitative approach," *Telematics and Informatics*, vol. 29, no. 3, pp. 263–272, 2012.
- [25] F. B. Tan and J. P. C. Chou, "The relationship between mobile service quality, perceived technology compatibility, and users' perceived playfulness in the context of mobile information and entertainment services," *International Journal of Human-Computer Interaction*, vol. 24, no. 7, pp. 649–671, 2008.
- [26] L. T. Hu and P. M. Bentler, "Cutoff criteria for fit indexes in covariance structure analysis: conventional criteria versus new alternatives," *Structural Equation Modeling: A Multidisciplinary Journal*, vol. 6, no. 1, pp. 1–55, 1999.

## Research Article

# Design of a Cryptographic System for Communication Security using Chaotic Signals

Jai-Houng Leu <sup>1</sup>, Jung-Kang Sun,<sup>2</sup> Ho-Sheng Chen <sup>3</sup>, Chong-Lin Huang,<sup>3</sup>  
Dong-Kai Qiao,<sup>3</sup> Tian-Syung Lan <sup>3</sup>, Yu-Chih Chen,<sup>4</sup> and Ay Su<sup>2</sup>

<sup>1</sup>Shandong Polytechnic, No.23000 Jin Ten East Road, Jinan, Shandong Province, China

<sup>2</sup>Department of Mechanical Engineering, Yuan Ze University, Taoyuan 32003, Taiwan

<sup>3</sup>College of Mechatronic Engineering, Guangdong University of Petrochemical Technology, Maoming, Guangdong 525000, China

<sup>4</sup>Aerospace Science and Technology Research Center, National Cheng Kung University, Tainan, Taiwan

Correspondence should be addressed to Ho-Sheng Chen; [hschen98.tw@gmail.com](mailto:hschen98.tw@gmail.com)

Received 12 February 2021; Revised 6 April 2021; Accepted 12 May 2021; Published 27 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Jai-Houng Leu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disturbance or corresponding errors of the transmission of information affect the ability of error detection. The chaotic encryption system prevents errors and secures the transmission system safely. The security assures by updating chaotic signals with the parameters of the chaotic circuits which are frequently changed. The data decipher and the encryption by the chaotic signaling system renews and changes the initial condition of a chaotic electric circuit. When most of the decimal portions are less than the threshold, the transmission is accepted, and all the noninteger numbers are rounded to their nearest integers. The criterion allows the error-detection function in the security system that is proposed in this paper. The chaotic encryption system for information is applied to public channels by the authorized individual. Three pictorial examples transmitted in the proposed system successfully demonstrate the security and performance. The new system provides high efficiency in the satellite communication network.

## 1. Introduction

Communication via satellite has been a common way for information exchange since the twentieth century. New ways of communication such as distance education and video conferencing with mobile devices need satellite networks. Despite the easy and convenient transmission of the information, there is always a problem of protecting private and secret information. Illegal eavesdropping or wiretapping causes considerable loss of the users. As transmitted messages through the satellite are easily interfered with and tampered with, important data of the defense navigation or business messages may not be delivered appropriately. Therefore, communication security in satellite networks has been attracting increasing interest from industry and academia. In general, the encryption system adopts public (secret) key [1–4] or private key cryptography [5–7]. The former was introduced by Diffie and Hellman [1]. They designed a cryptosystem that uses the same private key for

encrypting and decrypting. That is, two terminals share the same identification code for encrypting the cryptography of the private key by designing an encryption algorithm in the system as a data encryption standard (DES) [5]. The private-key cryptosystem provides strong security for public-key cryptosystem whose speed of authentication is slower than that of the private-key cryptosystem. Systems using satellite communication such as mobile devices and communication platforms for video conferencing usually use public (secret) keys.

Thus, how to protect important information in private key in the transmission is critical. Therefore, a new cryptographic technology for network security for satellite communication is required.

This research aims to propose a new system for the security of the satellite communication network by using a chaotic signal as a carrier and the Haar wavelets for multiplexing and demultiplexing. The proposed system is different from the conventional encryption algorithm as the

chaotic encryption system used for this study has a noncycle and complex time behavior. The new nonlinear method that uses the initial condition of a chaotic system as a private key masks the information-bearing signals by chaotic signals in the system (Figure 1). Then, the information is decrypted based on the carrier after it is accepted at the end of the transmission channel. The chaotic system transforms the private key of the system and the Haar wavelet by multiplexing and decrypts the key by the demultiplexing (Figure 2) [8]. This process finds out the transmission error easily and prevents the interception of information from the public channel. Thus, an effective way of satellite communication is obtained. The proposed system proves the effect of the divergence of a chaotic system which is suppressed according to the behavior of a nonlinear system in the new encryption scheme. The system provides a new security system of satellite communication network and protects the data and messages from various cyber attacks.

## 2. Methods

**2.1. System Design.** There are different projects that encode the public key since the public-key-encrypted project arose. Its safety always sets up the most complex mathematics problems. The encrypted key and cracked key in the symmetric encryption system are the same key. The major problem is that how the sender transmits the encrypted key to the receiver in safety after the information was encrypted, and let both share the secret key to decode it. If we use the key list in a trusted Internet, maybe we can solve this problem [9].

Through the encryption algorithm, we can do every kind of replacement to plaintext, and the input to encryption algorithm is the secret key. The key is unrelated data to plaintext; we use the key not only to encrypt the plaintext but also to crack the ciphertext. That is, we use the same secret key to encrypt or crack the text in the symmetric encryption system, so the transceiver must own the same key. Therefore, how to transmit the key to the receiver validly and guard the information against hackers is an important problem. [10, 11].

Everyone has a public key and private key in the asymmetric encryption system. The private key must be kept by an individual carefully. Under the asymmetric encryption system, every participator can get everyone's public key and own his own private key, so the private key does not need be transmitted in the net. If the public key encrypted one message, then it must be cracked by the private key, and vice versa. [12].

The state trajectory of a chaotic system is indeterminable. Thus, the divergence of nearby trajectories causes any small error to be magnified as the equations are integrated with the specified initial conditions. Even a small effect affects the system in a long term. The sensitivity of the system depends on initial conditions in the chaotic behavior of the system. The effect of the divergence of a chaotic system is suppressed in a nonlinear system where a message of plain texts is converted into a Haar wavelet form by the encoder matrix. It gives not only an encrypted message but also a transmitted

error checking [13]. The Haar wavelets signal can be carried by one state of the chaotic signals (Figure 3). Then, it is sent to a public channel, decrypted at the receiving end, and demultiplexed by using the decoder matrix. The method uses the initial conditions of a chaotic system as a private key in addition to the Haar wavelet transform for multiplexing and demultiplexing to form the nonlinear system. The messages are securely encrypted, and its transmission errors are easily detected. No one can decrypt the intercepted messages from a public channel without the private key. The Haar wavelet of information in Chua's circuit is transmitted to a public channel as it is decoded at the end of communication by a demultiplexer. The process is presented in Figure 3. The security of the system is decided by the initial condition of the chaotic signal of the information. The original information is transformed by the Haar wavelet by the encoder matrix.

**2.2. Encryption.** Encrypting the chaotic cryptosystem is carried out according to the following steps:

- (1) First, both the transmitter and the receiver are assigned to have the same private key that contains the chaotic parameters  $(\alpha, \beta, a, b)$ , the initial conditions  $(x_0, y_0, z_0)$ , and the rank of the encoder matrices  $\mathbf{H}_n$
- (2) The transmitter obtains the plaintext data  $[\mathbf{C}]$  and calculates  $[\mathbf{m}] = [\mathbf{C}] * \mathbf{H}_n$
- (3) It generates the signal states of  $(\mathbf{X}$  or  $\mathbf{Y}$  or  $\mathbf{Z})$  on a fixed time interval in  $x, y,$  and  $z$  channels in Chua's circuit using the parameters in step 1
- (4) When  $\hat{\mathbf{Z}} = \mathbf{Z} + [\mathbf{m}]$ , a chaotic signal of chaotic masking denotes and transmits  $\hat{\mathbf{Z}}$  to the receiver
- (5) The transmitter calculates the new chaotic parameters as follows:

$$\left\{ \begin{array}{l} \alpha' = f(\alpha), \\ \beta' = f(\beta), \\ a' = f(a), \\ b' = f(b), \\ x'_0 = f(x_0), \\ y'_0 = f(y_0), \\ z'_0 = f(z_0), \end{array} \right. \quad (1)$$

where  $f()$  is a collision-free one-way function [7, 14] for both ends of the transmitter and receiver

For example,

$$f(x) = (a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0) \bmod p. \quad (2)$$

If the transmitter sends the next frame message, steps 2~6 should be repeated. The receiver obtains the encrypted messages  $\mathbf{Z}$  from the public channel and uses the following procedures for decryption.



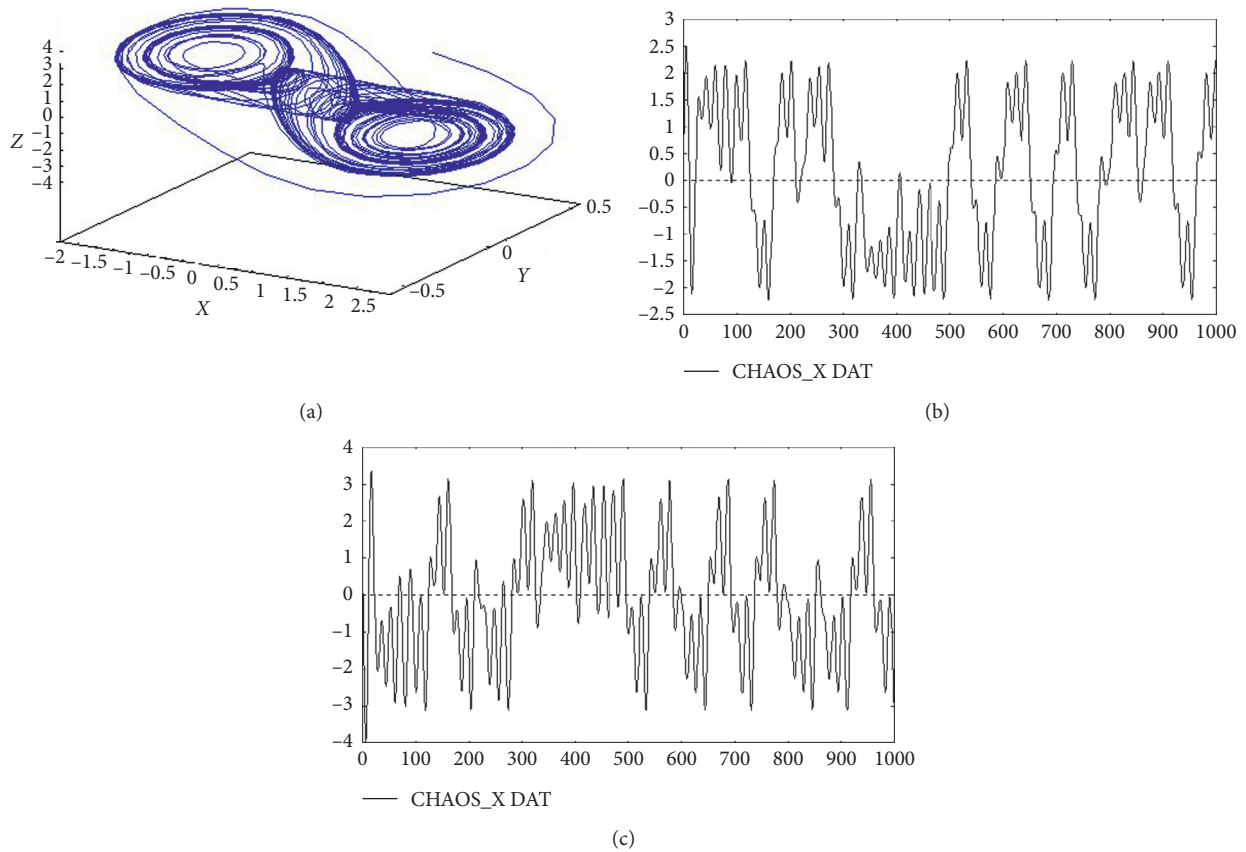


FIGURE 1: Information-bearing signals with chaotic signals. (a) Double scroll chaotic signal. (b) X channel. (c) Z channel.

### 2.3. Decryption

- (1)  $\hat{Z}$  arrives at the receiver.
- (2) The receiver generates a state variable  $Z$  by using the same parameters as in step 1.
- (3) The ciphertext  $[m] = \hat{Z} - Z$  is calculated.
- (4)  $[C] = [m]H_n^{-1}$  is defined.
- (5) If this cipher message  $[C]$  contains noninteger numbers and the difference between the noninteger numbers to their nearest integers is larger than a threshold, then the transmitted message may have been interfered with by noise disturbances or communication error. In this case, the receiver requests the transmitter to send the message again. Otherwise, the transmission is considered to be successful.
- (6) The receiver calculates the chaotic parameters as in equation (1).

The Haar wavelet transform is carried by one of the chaotic signal states ( $x(t)$ ,  $y(t)$ , or  $z(t)$ ) in Chua's circuit (Figure 4). Then, it is sent to a public channel, decrypted at the receiver, and demultiplexed by using the decoder matrix. The changing private key alters the transmitted messages in the public channel and contains the parameters of Chua's circuit and the rank of the encoder matrix. As the plaintext data  $[C]_n$  and the encoder matrix  $H_n$  are both integers, the

ciphertext  $[m]$  contains only integer numbers. This property allows a convenient detection of redundancy when the masked message  $Z$  is corrupted during transmission. For example, network disturbances in computers of heavy load and frequent on-off operations and external electromagnetic fields may contaminate the messages.

### 3. Results and Discussion

We use the seven chaotic parameters ( $\alpha$ ,  $\beta$ ,  $a$ ,  $b$ ,  $x_0$ ,  $y_0$ , and  $z_0$ ) and the dimension of matrix  $n$  as the "encryption keys." The cyber attacker cannot decrypt the encrypted message unless the chaotic behavior is understood as the original signals are carried by the chaotic signals during transmission. The control parameters of chaotic behavior constantly change in the collision-free one-way function. As a result, the security property results in a high sensitivity of synchronization with the parameter change. Therefore, understanding the chaotic behavior of the chaotic parameters that change in each transmission is required for decryption. In other words, the system is secured as long as the first chaotic parameters are kept secret. To decrypt the encrypted data, the encryption key of the system is demanded to synchronize the signal [14].

In the other words, the modulation-demodulation requires the system to spend much time, and the message is not decrypted without a correct key. For updating other parameters such as the encoder matrix order, initial



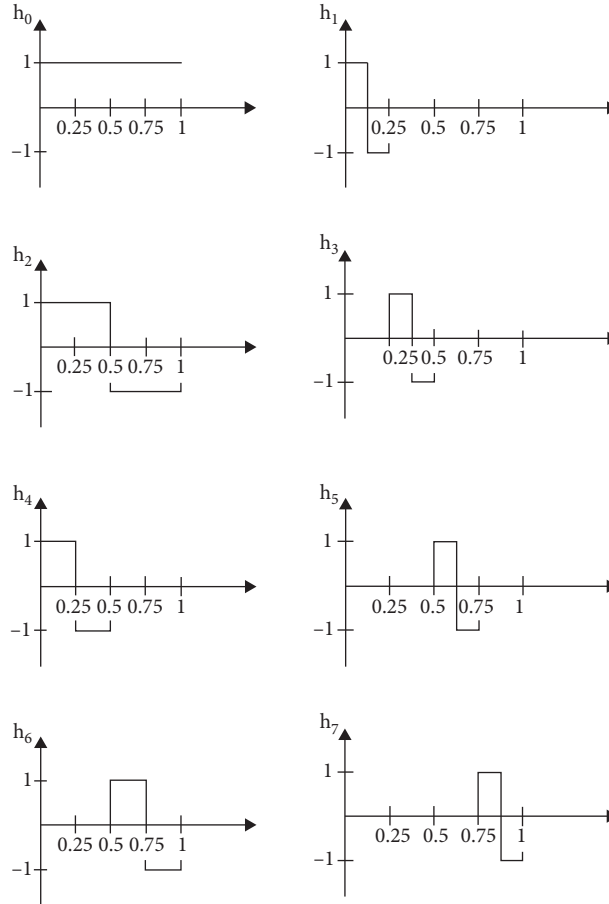


FIGURE 2: The Haar wavelet transform.

conditions of chaotic circuits, coefficients, and prime number in equation (1), communication security needs enhancement, which is realized in this research. Besides the safety of secret messages, this system also enhances communication efficiency and improved performance for the secured communication. For any disturbance or communication error, the received message contains noninteger numbers that are easily detected, which improves the capability for error detection. It is important to exactly estimate the unknown parameters of chaotic systems in chaos control and synchronization. Hu et al. presented a method for estimating a one-dimensional discrete chaotic system based on the mean value method (MVM) by exploiting the ergodic and synchronization features of the chaos. This research proposed a method that estimates the parameter value more accurately than the MVM [15].

The suggested chaotic parameters can be any integers between  $-32767$  and  $32767$ , and the possible combination of keys is  $(32767)^7 \times 2^7 \times [(I+1)!-1]$ . As it takes  $10^{-9}$  seconds for one calculation, this is beyond the capability of the existing supercomputers. The total time needed for solving the message is up to  $2.83 \times 10^{22}$  years. The number of keyspace reaches  $1.88 \times 10^{39}$  if the rank of encoder/decoder matrix is set to be eight including seven independent variables and one dependent function. The variables are chaotic parameters ( $\alpha$ ,  $\beta$ ,  $a$ , and  $b$ ), the initial conditions ( $x_0$ ,  $y_0$ , and

$z_0$ ) on which the rank of the encoder matrices is based. This private key that contains the parameters of Chua's circuit and the rank of the encoder matrix changes constantly to alter the appearances of the transmitted messages in the public channel. As integers, the plaintext data  $[C]_n$  and the encoder matrix  $H_n$  result in ciphertext  $[m]$  of only integer numbers. This property offers a convenient way to detect whether the masked messages  $Z$  are corrupted during transmission. JAVA codes of the proposed algorithm were tested successfully on two remote machines. Of course, the ideal encryption should be robust so that the transmitted messages in the public channel are not decrypted by an unauthorized person.

The results are shown in Figure 5 based on JAVA codes of the proposed algorithm. Heavy-loaded computer networks, on-off operations of computers, and external electromagnetic fields cause disturbances to corrupt messages. If the disturbances are large, the messages decrypted by the receiver contain nonintegers. The receiver then becomes aware of obtaining a corrupted message and requests retransmission immediately. However, decrypted messages with nonintegers need caution; they are false transmissions. In the algorithm, the chaotic signals with nonintegers of floating parts can be introduced during the masking and unmasking. Also, when computers or the operating systems of the transmitter and the receiver are not the same, this

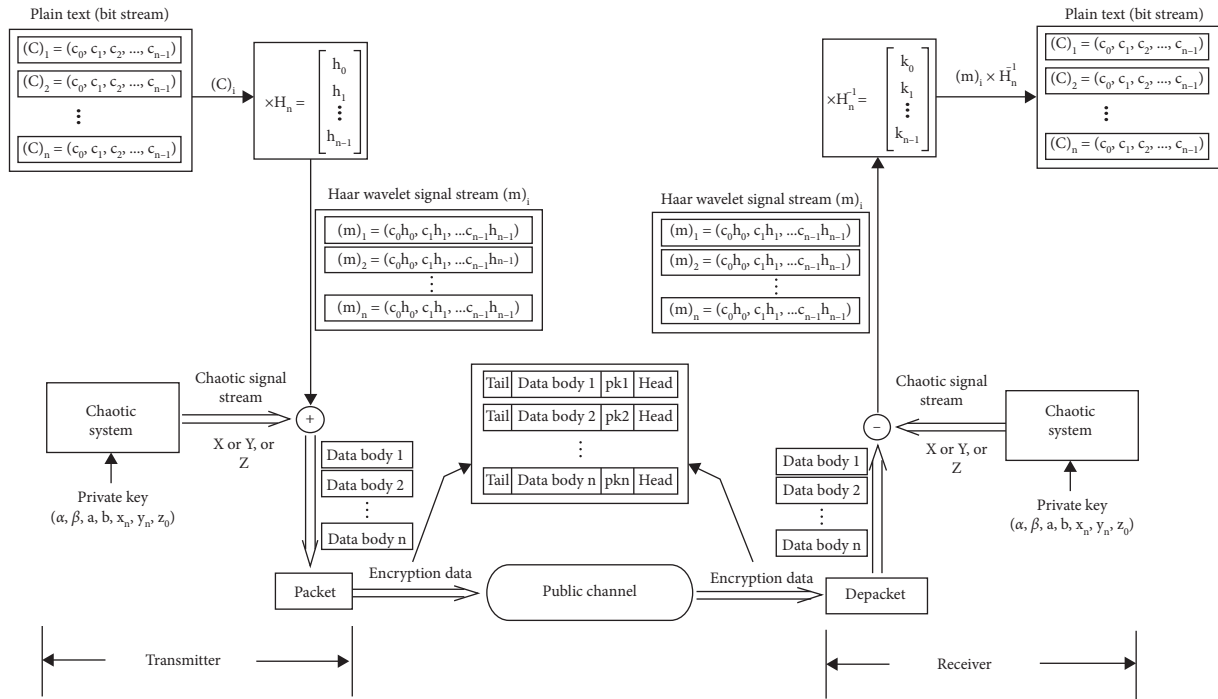


FIGURE 3: The diagram of a chaotic cryptosystem.

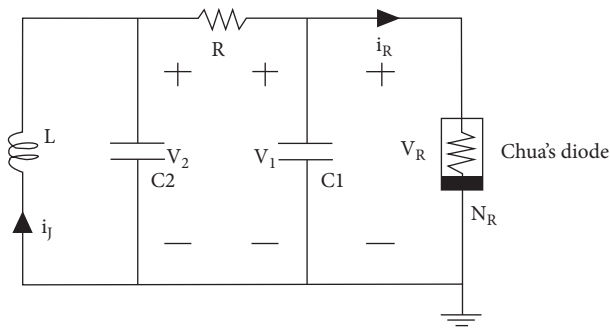


FIGURE 4: Chua's circuit.

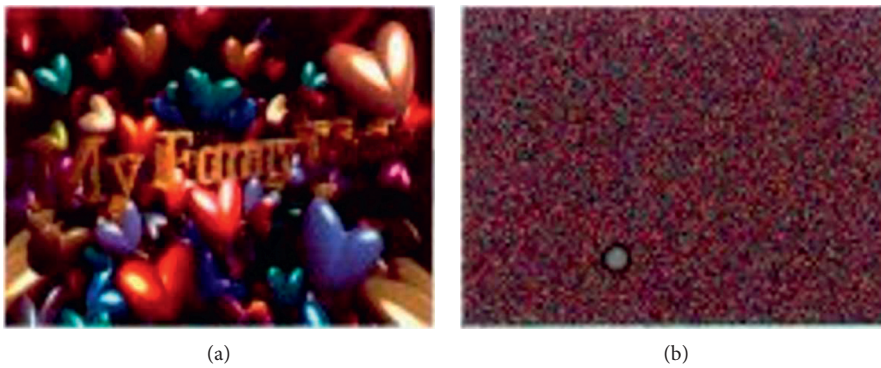


FIGURE 5: Continued.

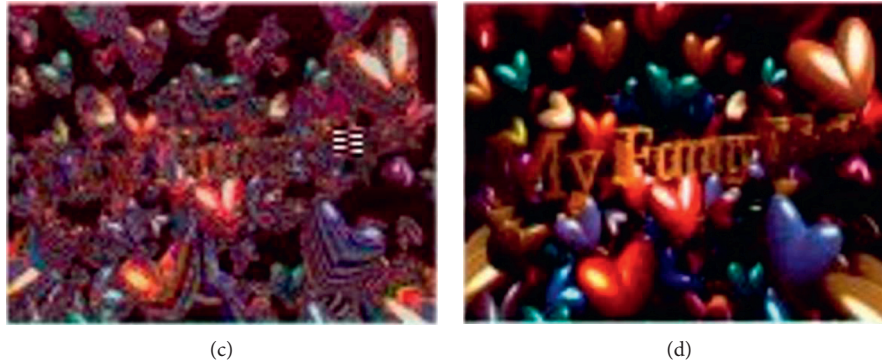


FIGURE 5: The results of “many hearts” signals transmitted by chaotic parameters. (a) Original signal, (b) chaotic signal, (c) encryption signal, (d) recovery signal.

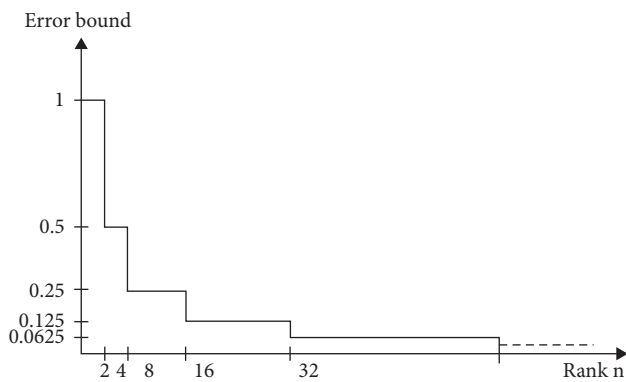


FIGURE 6: Threshold of the error bound (the  $x$ -axis) and the rank of the encoder matrix (the  $y$ -axis).

creates decimal parts. Therefore, to remedy the possible situation for misinterpreting, a threshold regarding the magnitude of the decimal part is required for error detection.

The behavior of the Haar wavelet form is aperiodic. It is not complex like a random signal or a noise that needs time and an algorithm to distinguish signals and noises. The plaintext message is converted into a Haar wavelet form by the encoder matrix and sets the criterion for filtering. That is, there is a convenient criterion from the Haar wavelets to check and filter the transmitted error. Since a fitting error of approximating any function by the Haar wavelet form is a reciprocal of the highest-order Haar function, we selected a threshold (or an error bound) as the reciprocal of the rank of the encoder matrix. In this way, the transmission becomes acceptable and all the noninteger numbers are rounded to their nearest integers if most of the decimal parts are less than the threshold. With this simple criterion, the error-detection function in the proposed system is established. In this case, the threshold is equal to  $1/n$  as shown in Figure 6.

#### 4. Conclusions

The importance of communication security is becoming critical as the number of satellites is increasing. Thus, preventing transferred information from eavesdropping or

wiretapping has been attracting much interest. New cryptographic technology for the security of the satellite communication network is proposed by using a chaotic signal as a carrier and the Haar wavelets for multiplexing and demultiplexing. The proposed system allows secure encryption of messages and easy detection of errors. Three pictorial examples were tested in the system, and the result validated the performance and security of the system. The system has the following four advantages: (1) simplicity and low cost as it runs on PCs by implementing the algorithm, (2) high security, (3) secure authentication, and (4) easy detection of transmission errors. The JAVA code of the proposed algorithm was also tested and operated successfully on two remote machines. The result shows that the proposed system is available for individual, academic, or industrial purposes conveniently. The result of the system leads to further research on the encryption and decryption of messages including plaintexts, voice, pictures, or their combination for multimedia purposes.

#### Data Availability

The data used to support the findings of this study are restricted by Jai-Houng Leu in order to protect PATIENT PRIVACY. Data are available from Jai-Houng Leu (jahonleu@yahoo.com.tw) for researchers who meet the criteria for access to confidential data.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### References

- [1] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
- [2] T. E. Gamal, “Design of universal test sequences for VLSI,” *IEEE Trans. Inf. Theory*, vol. 31, p. 469, 1985.
- [3] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

- [4] M. O. Rabin, MIT Laboratory of Computer Science, Technical Report MIT/LCS/TR-212, 1979, [https://www.scirp.org/\(S\(i43dyn45teexjx455qlt3d2q\)\)/reference/ReferencesPapers.aspx?ReferenceID=45711](https://www.scirp.org/(S(i43dyn45teexjx455qlt3d2q))/reference/ReferencesPapers.aspx?ReferenceID=45711).
- [5] E. L. Richardson, E. O. Vetter, B. Ancker-Johnson, and E. Ambler, Data Encryption Standard. FIPS PUB National Bureau of Standards, Washington D.C., 1977, <https://csrc.nist.gov/CSRC/media/Publications/fips/46/archive/1977-01-15/documents/NBS.FIPS.46.pdf>.
- [6] W. M. Daley and R. G. Krammer, *Data Encryption Standard (DES)*, National Institute of Standards and Technology, Gaithersburg, Maryland, 1999, <https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>.
- [7] X. Lai and J. L. Massey, *Proc, Advances in Cryptology Eurocrypt*, 1992, [https://link.springer.com/chapter/10.1007/3-540-47555-9\\_5](https://link.springer.com/chapter/10.1007/3-540-47555-9_5).
- [8] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.
- [9] Zhang Chengxin, 1997, <https://www.elsevier.com/books/computer-and-information-security-handbook/vacca/978-0-12-803843-7>.
- [10] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 67, no. 3, p. 644, 1976.
- [11] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, p. 654, 1976.
- [12] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [13] K.-S. Hwang, "Long-yeu chung, isaac yuchih chen , reducing the cost of spacecraft ground systems and operations," *Part of the Space Technology Proceedings Book Series*, vol. 3, p. 421, 1988.
- [14] Y. Zherly, T. Matsumoto, and H. Imai, "Impossibility and optimality results on constructing pseudorandom permutations," *Lecture Notes in Computer Science, Advances in Cryptology-EUROCRYPT*, vol. 89, pp. 412–422, 1990.
- [15] J. Hu, H. Li, and J. Li, "Parameter Estimation of a Class One-Dimensional Discrete Chaotic System," *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 696017, 9 pages, 2011.

## Research Article

# An Optimized and Efficient Routing Protocol Application for IoV

**Kiran Afzal,<sup>1</sup> Rehan Tariq<sup>1</sup>,<sup>2</sup> Farhan Aadil,<sup>2</sup> Zeshan Iqbal<sup>1</sup>,<sup>2</sup> Nouman Ali<sup>1</sup>,<sup>3</sup> and Muhammad Sajid<sup>4</sup>**

<sup>1</sup>Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan

<sup>2</sup>Department of Computer Science, University of Engineering and Technology Taxila, Taxila 47050, Pakistan

<sup>3</sup>Department of Software Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, Pakistan

<sup>4</sup>Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, Pakistan

Correspondence should be addressed to Nouman Ali; nouman.ali@live.com

Received 19 March 2021; Revised 23 April 2021; Accepted 5 May 2021; Published 19 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Kiran Afzal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IoV is the latest application of VANET and is the alliance of Internet and IoT. With the rapid progress in technology, people are searching for a traffic environment where they would have maximum collaboration with their surroundings which comprise other vehicles. It has become a necessity to find such a traffic environment where we have less traffic congestion, minimum chances of a vehicular collision, minimum communication delay, fewer communication errors, and a greater message delivery ratio. For this purpose, a vehicular ad hoc network (VANET) was devised where vehicles were communicating with each other in an infrastructureless environment. In VANET, vehicles communicate in an ad hoc manner and communicate with each other to deliver messages, for infotainment purposes or for warning other vehicles about emergency scenarios. Unmanned aerial vehicle (UAV)-assisted VANET is one of the emerging fields nowadays. For VANET's routing efficiency, several routing protocols are being used like optimized link state routing (OLSR) protocol, ad hoc on-demand distance vector (AODV) routing protocol, and destination-sequenced distance vector (DSDV) protocol. To meet the need of the upcoming era of artificial intelligence, researchers are working to improve the route optimization problems in VANETs by employing UAVs. The proposed system is based on a model of VANET involving interaction with aerial nodes (UAVs) for efficient data delivery and better performance. Comparisons of traditional routing protocols with UAV-based protocols have been made in the scenario of vehicle-to-vehicle (V2V) communication. Later on, communication of vehicles via aerial nodes has been studied for the same purpose. The results have been generated through various simulations. After performing extensive simulations by varying different parameters over grid sizes of  $300 \times 1500$  m to  $300 \times 6000$  m, it is evident that although the traditional DSDV routing protocol performs 14% better than drone-assisted destination-sequenced distance vector (DA-DSDV) when we have number of sinks equal to 25, the performance of drone-assisted optimized link state routing (DA-OLSR) protocol is 0.5% better than that of traditional OLSR, whereas drone-assisted ad hoc on-demand distance vector (DA-AODV) performs 22% better than traditional AODV. Moreover, if we increase the number of sinks up to 50, it can be clearly seen that the DA-AODV outperforms the rest of the routing protocols by up to 60% (either traditional routing protocol or drone-assisted routing protocol). In addition, for parameters like MAC/PHY overhead and packet delivery ratio, the performance of our proposed drone-assisted variants of protocols is also better than that of the traditional routing protocols. These results show that our proposed strategy performs better than the traditional VANET protocols and plays important role in minimizing the MAC/PHY and enhancing the average throughput along with average packet delivery ratio.

## 1. Introduction

IoV is the new form of VANET and is the alliance of Internet and IoT. VANET is a type of wireless network where vehicles interact with each other as well as with roadside units within

a short distance [1]. For the avoidance of human loss and to minimize the time being waste, everyone wants a traffic environment that has fewer chances of accidents and collision, with a more reliable path that could help us to avoid any delay caused by the traffic congestion [2]. Moreover, a



reliable and quick communication is also an ample demand in disaster or emergency scenarios [3]. Some of the critical issues that make such communication difficult are physical hindrance including on road obstacles, mobility issues, limited range of vehicles, and cost of infrastructure installation. Such factors not only result in unreliable communication, but also in some cases totally make it impossible for vehicles to communicate efficiently. For an efficient communication in a vehicular environment, we must keep in mind some of the factors like the following:

- (i) No. of possible paths
- (ii) Turns
- (iii) Intersections
- (iv) Traffic congestion
- (v) The nearest route to the destination

Several routing techniques like ant colony optimization have been used for this purpose [4]. The selected optimized path, that is, the shortest one, is tested again and again using route planning software available. Optimization can be gained based on heuristics which are gained through experience and provide us with efficient solutions. VANET comes under the category of mobile ad hoc network (MANET) that is a subclass of wireless ad hoc networks. Moving vehicles in VANET operate in two basic architecture modes: V2V (vehicle-to-vehicle) communication and V2I (vehicle to infrastructure) communication [5]. In the former architecture, the vehicles communicate with each other, to exchange information, through Dedicated Short Range Communication (DSRC) protocol, while in the latter architecture the communication between vehicles is via roadside units [6]. Vehicular ad hoc network has a highly dynamic topology with varying the speed of the vehicle, the number of vehicles, and the direction changed by the vehicles [2]. Due to such issues and those mentioned previously, a new class of ad hoc networks has been devised "Internet of Vehicles." It makes use of unmanned aerial vehicles which proved to be helpful in efficient communication between vehicles. In this paper, we have devised an optimized solution for enhancing the network efficiency in terms of better throughput, average packet delivery ratio, and less MAC/PHY overhead. Such proposed scheme will not only help in having better network experience in traffic, but also enhance the medicine and healthcare, agriculture, disaster, and emergency scenarios and provide environmental and surrounding information and a better solution for communication over a congested road. The topological constraint changes made differentiate our proposed scheme from those proposed earlier.

The remainder of the paper consists of the following sections: Section 2 involves introduction and brief explanation of the field of IoV. Section 3 discusses the routing problems and challenges, mobility models, application, and related work done by the researchers in the past. The proposed methodology is discussed in Section 4. The results are presented in Section 5. Section 6 throws light on the comparative analyses of the scenarios used in the proposed

research. Lastly, the whole research is concluded under Section 7 along with intended future work.

## 2. Internet of Vehicles (IoV)

IoV is a special class that falls under the category of VANETs and IoT. This class constitutes the framework of vehicles that interact with each other for the sake of exchanging useful information about the traffic, roads, and environment around them. The interaction can be through the infrastructure using RSU (roadside units) which is vehicle to infrastructure communication, or the exchange is directly between the vehicles themselves adapting the vehicle-to-vehicle communication mode. In IoV, vehicles communicate not only with other vehicles but also with the infrastructure, the handheld devices being carried by the pedestrians, the cloud servers, and the sensors deployed in the environment or within the vehicles themselves.

Due to traffic problems like traffic congestion, delays, and route optimization, there is a need to find some vehicle mobility pattern or routing protocols that can resolve these issues. Many routing protocols have been proposed, but not all of them can give our desired results, nor is each protocol best suited for vehicular ad hoc networks. One of the main problems that hinder vehicles from reaching their desired destination is the nonavailability of an optimized route. Due to frequently dynamic topology, there are frequent disconnections between vehicles. Moreover, the hindrance caused by tall buildings and physical objects makes it difficult for vehicles to receive data or to communicate with each other efficiently. One of the possible solutions can be the use of aerial nodes. Such nodes deployed at certain ranges might give us some better results, and performance might increase. To enhance the overall efficiency of a network, researchers work on some fundamental parameters like average throughput, packet delivery ratio, communication delay, MAC/PHY overhead, overall network congestion, and packet drop. By doing so, the coverage of vehicular nodes can be enhanced. Such nodes can be deployed at certain ranges for performance gains. In our intended work, we have proposed a model in which first we have analyzed the efficiency of different routing protocols where the vehicular nodes communicate with each other, scenario (a). Later, the results are generated in scenario (b) where the vehicles communicate with each other indirectly via some aerial nodes deployed at some distance. The results of both scenarios are compared and evaluated to determine which scenario gives us better results. Figure 1 shows a brief description of our desired scenarios. The focus of our research is mainly on the following:

- (i) Utilizing the UAVs to evaluate the performance of traditional VANET routing protocols.
- (ii) Evaluating average packet delivery ratio in traditional VANET by incorporating UAVs.
- (iii) Minimizing the MAC/PHY overhead.
- (iv) Maximizing the average throughput.

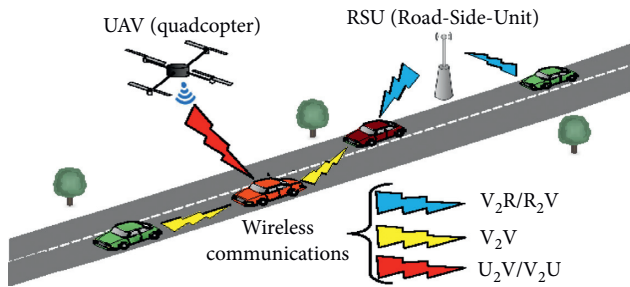


FIGURE 1: Scenarios a and b.

The end results of simulations show that our proposed strategy have better performance in terms of average packet delivery ratio, MAC/PHY overhead, and average throughput for larger grid size involving a greater number of vehicular as well as sink nodes.

### 3. Related Works

For years, the researchers from the academic field as well as from industry are investigating certain possible ways to solve the problems which are being faced in VANETs. Surveys have been done and algorithms have been proposed to provide optimized solutions for data delivery, better throughput, and optimized path. This section will focus on the relevant literature based on problems and challenges faced in vehicular ad hoc networks, Internet of Vehicles, mobility models used, and applications of such networks. We will also discuss some strategies or methodologies proposed by various researchers to minimize the issues being faced in this field.

**3.1. Routing Problems and Challenge.** Despite always ongoing research, certain challenges are still being faced in VANET including security, safety, and low latency. In the following subsections, we will discuss some of the problems which affect the network performance of VANET as well as IoV.

**3.1.1. Route Optimization.** One of the main issues in vehicular ad hoc networks is route optimization. Route optimization is basically about determining the efficient route in terms of less cost and shortest path with less amount of delay. For route optimization, we should keep in mind some of the factors like the no. of possible paths, turns, intersections, traffic congestion, and nearest route to the destination. Due to the highly dynamic topology and unpredictable movement of vehicles, it has become necessary to propose such a routing strategy that can mitigate such issues for better dissemination of information within interacting vehicles and the surrounding environment. As we have a broader range of possibilities, we mostly use algorithms which reduce the possibilities and help us to choose an optimized route (in terms of cost and energy consumption).

**3.1.2. Security Issues.** As the data being transferred in any network are crucial to its users, to have a secure network becomes a necessity. In VANETs, when two or more vehicles are communicating and sharing their information, they may face the interference of any hacker or intruder which could breach the network security by hacking the information flowing in the network (between the vehicular nodes) [7]. It has become a common cyber issue in the modern world as hackers and crackers are utilizing advanced and modern technologies to steal the information flowing in a vehicular environment. These cyber attacks can be active as well as passive nature including DOS attacks, revealing the identity, tracking and tracing of the vehicle's movement in the network, and black hole attacks.

**3.1.3. Network Scalability.** The scalability of VANET raises many critical issues. In the modern era, most people prefer to use their private transport, which results in an extensive and complicated network due to the increased number of vehicular nodes in VANET topology. Such an increase makes it difficult for the routing protocol to fully cover all the moving vehicles [8]. This affects the overall performance of the network where some of the areas of the network are under the control of the routing protocol and work efficiently, whereas, due to intensive network scalability, some of the vehicles are deprived of the efficient routing protocol.

**3.1.4. Fluctuating Node Density.** The vehicular node density in VANET is always unpredictable as the vehicles are always entering or leaving the network. Moreover, one cannot exactly predict in which part of the day the node density will be higher or will be less dense. Some of the routes in VANET are heavily dense due to route characteristics like better road conditions and shortest path to a certain destination. Such a condition can congest the traffic network, increasing network complexity, while the other routes may be sparsely dense which results in uneven node distribution in the network [9].

**3.1.5. High Mobility and Dynamic Topology.** Due to always moving vehicles that are entering the network as well as exiting it, the topology of vehicular ad hoc networks is always changing. Such type of mobility and topological changes make it difficult to have an optimized routing, and routing protocols which are well efficient in handling such types of routing problems in VANET are required.

**3.2. Mobility Models.** Depending upon the network requirements, a variety of mobility models have been proposed for the vehicular ad hoc network, each of which has its characteristics. In the following subsections, we will discuss some of the commonly used mobility models for VANETs. There are certain characteristics which are necessarily needed to build up an efficient mobility model. These characteristics may include the pattern in which mobility within the network is carried out, the average speed with which a vehicle can move in a network, and the mechanisms

which can control the traffic. Depending upon such characteristics, a mobility model is selected and adapted for the intended network [10].

*3.2.1. Random Waypoint Model (RWM).* Random waypoint model is commonly used for ad hoc networks. Its main characteristics include simplicity and availability at a wide range. In RWM, the nodes can move freely without any limitation and restriction. Parameters like speed and direction of the nodes are chosen randomly. Along with its pros, there are two major issues of the random waypoint model: sudden stop and rapid change of directions [11].

*3.2.2. Stop Sign Model (SSM).* In the stop sign model, the moving vehicles make their movement relative to the traffic sign when they reach any type of intersection on the road. When a moving vehicular node reaches the intersection, that node must wait for a certain specified interval of time before heading towards its next destination. The vehicular node keeps distance from the node that is moving in front of it [7].

*3.2.3. Probabilistic Traffic Sign Model (PTSM).* The probabilistic traffic sign model uses traffic lights instead of utilizing the stop sign on the road when it reaches the junction. When a vehicular node approaches the junction, it has to wait for a randomly selected amount of time interval. In the same way, the vehicle that reaches this node has to wait again for a second, which increases the delay. The described model is useful as it decreases the excessive wait [10].

*3.2.4. Manhattan.* Manhattan mobility model works on the maps and is mostly preferred for urban environments. The maps used in the Manhattan mobility model use roads with different lanes, and each of these lanes has further two directions. Therefore, overall a node can move in four possible directions, that is, from north to south, from south to north, from east to west, and from west to east. Even a vehicular node can change its direction from left to right or from right side to left one when it will reach any kind of intersection. There is a 50% possibility that a moving vehicle will stay on the road, while the possibility of taking a turn is even half of it [12].

*3.2.5. Freeway Mobility Model.* The freeway mobility model operates on the behavior in which the vehicular nodes are moving on different types of freeways. As we know, there are several lanes on any freeway, and even those lanes have two types of directions separately for incoming and outgoing vehicles. In this model, each of the vehicular nodes is restricted to its specified lane. The speed with which nodes are moving is dependent on the speed of the previous node for a short time [13].

*3.3. Applications.* Vehicular ad hoc networks have a wide range of applications in different fields. With the advancement in modern technology, researchers have been

adopting different methods to increase the utilization and applications of VANET. Such a network can be used for gaining information, for emergency scenarios, for entertainment, for safety, and for better utilization of roads in an efficient manner. Some of such characteristics have been listed below whereas more is yet to come.

*3.3.1. Safety Purposes.* As with the increasing number of vehicles on the road, there is an increasing risk of road accidents and vehicle collisions. Researchers have been working to deduce improved technologies for better traffic conditions. As in VANET the vehicles are communicating with each other, in case of any accidents the vehicles can generate warning or alerts so that the upcoming vehicles can be alerted. The drivers of vehicular nodes can easily be informed in advance about mishaps taken place on road.

*3.3.2. Infotainment.* Sometimes driving a car can be so boring, especially if you are moving on the same road on daily basis. Moreover, it could be difficult to travel around in an area if you do not have any information about that area. In such a case you will need to know about your location and nearest places or where your specified destination is. All problems like these can now be addressed by the vehicular ad hoc environment where the vehicular nodes are always in interaction with each other. Moreover, the interactive billboards and hoardings, downloads, notifications for the points, or things you are interested in can be appealing in VANET scenarios [14].

*3.3.3. Emergency Scenarios.* VANET is also well suited for disaster scenarios, as in such emergency scenarios, where any calamity has taken place like an earthquake or flooding, the infrastructure of the network deployed in that area partially or completely becomes inactive. Therefore, in such cases, the VANETs are helpful for communication with one and other as well as calling for help and services. Moreover, if there is something that could be dangerous for the upcoming vehicles like any wild animal that is present on the road and could be harmful, the drivers of those vehicles can be warned and stopped by sending them a warning or alert messages in a vehicular ad hoc network [14].

*3.3.4. Management of Congested Traffic.* For a smooth and safe traffic environment, the management of traffic is an important parameter. Consequently, to avoid congestion because of high node density, certain methods have been suggested for vehicular ad hoc networks like developing the application that can keep track of location information of the vehicles. This information is then shared with the drivers of the vehicles if they have installed the application [15]. Based on this received information, the driver can leave the congested road or may turn to any other route.



3.3.5. *Environmental Information.* Like certain other applications, VANET provides the facility of dissemination of real-time data which may include alternative paths and weather conditions. In weather information, the driver can be provided with the information on the weather forecast and the possible adverse effects of the weather which may help to reduce the delays occurring due to adverse effects of weather like fog and rain [16].

3.4. *Related Work.* Several routing techniques like ant colony optimization have been used for this purpose [4]. The selected optimized path is tested again and again using route planning software available. Optimization can be gained based on heuristics which are gained through experience and provide us with efficient solutions.

Chen and coauthors proposed an efficient protocol that is designated to disseminate the data packet in the scenarios of urban areas while keeping various parameters in consideration like road traffic, topology, and information related to the specified geographical areas [17]. The protocol utilized the artificial spider web technology to discover the route between the source node and the destination, and it performs better in terms of end-to-end delay and packet delivery ratio. Nazib and Moh Reviewed various routing protocols that are most commonly used in vehicular environment with the assistance of aerial nodes [18]. The review has been done based upon the working mechanism and the principles adopted to design these protocols. The optimization and effectiveness of the protocols mentioned in this survey have also been discussed in detail.

Oubbati et al. proposed a reactive routing scheme which also involves the prediction method to select an efficient path to the desired or destination nodes. They have suggested the use of unmanned aerial vehicles for enhancing the effectiveness of the proposed scheme [19].

The usage of drone serving as relay node has been adopted in [20] by Lin et al. The aerial nodes have been distributed after predicting the number of vehicles participating in the ongoing traffic. This strategy considers several aspects of on-road traffic like non-line-of-sight and load on the network. Moreover, a new algorithm named as multi-modal nomad algorithm is also proposed as an efficient solution to the problems in the vehicular environment involving aerial. The proposed model has slight loss in end-to-end delay. Integrating the suggested model with other networks, such as the software-defined network, can enhance management and network control.

Kumar et al. suggested a heuristic algorithm for providing QoS in smart transportation system [21]. Although the proposed method enhances the network performance, it cannot be applied for a larger smart network. Lu et al. introduced an enhanced scheme for the city scenarios based upon geographical routing. The IGR scheme presented by the researchers works on two modes involving the greedy approach to forwarding data packets [22].

Bhatt et al. [23] suggested a model that uses the Bat algorithm to communicate with the destination by performing three stages using an optimized path. The first step

of the proposed model is to predict where the destination is. In the second step, unnecessary or useless nodes are discarded, and a region is formed. In the last step, an optimized path among the multiple paths is selected. ACO was proposed by the Mexican researchers Dorigo et al. [24].

The central theme of ACO was taken from the social behavior of ants. Each ant in ACO represents one solution, and a group of multiple solutions or ants form the swarm. ACO encodes the real-world problem into a graph. Vertices of the graph correspond to a component of a candidate solution, and ants create a trail by traversing an edge. While traversing, ants diffuse some chemical substance, pheromone. The quantity of pheromone on the edge of the graph determines its quality. Ants add the component to its candidate solution by evaluating each edge of the graph. If the quality of the edge is better than others, ant traverses the edge and adds that vertex to the candidate solution. After some repetition of this procedure, the algorithm converges towards some candidate solution. Farhanchi et al. also proposed a model to figure out the shortest and optimized path [25]. Prakash used variant of two protocols [26]: the first protocol that has been used is P-OLSR for avoiding congestion, and the second is E-OLSR for balancing load and optimizing path.

In [27], Bao et al. arranged the nodes in clusters and determined an optimized path. A hybrid routing protocol that is road/path-aware and is assisted by the infrastructure has been discussed in [28]. It provides some key aspects like duration of the path, velocity of the moving vehicle, and transmission range. The model performs better in terms of packet delivery ratio and reduces delay with the help of predicting the duration of the path [29].

Jindal and Bedi combine the benefits of MACO and PSO algorithms to reduce travel time in VANET. West and Bowman uses ant colony optimization algorithm (ACO) to select the optimum path with better network connectivity in [3]. Zhang et al. applied the Q-learning algorithm to the parameter of link reliability, and its performance was analyzed. Based on these evaluations, a new strategy was proposed which performs better in terms of packet delivery ratio, transmission time, and frequent change of topology in VANET [30].

Tian et al. proposed a new model based on bioinspiration and is a unicast-routing protocol. It guarantees the efficiency of message delivery and the robustness of the overall system compared to prior conventional routing protocols [31]. Elhoseny and Shankar presented a model in which they utilized the K-Medoid Clustering for arranging the vehicles in the form of clusters. The nodes which have efficient energy are distinguished by utilizing the metaheuristic algorithm. Afterward, these nodes are used for communication [32].

Nayyar analyzed different protocols like AODV, OLSR, DSDV, DSR, AOMDV, and HWMP to evaluate their performance in the FANETs scenarios to use them in real operations [33]. Leonov tried to examine various approaches that are based on the bee colony algorithm. Results were analyzed and a new strategy, BeeAdHoc, that is comparatively better than traditional VANET protocols (AODV, DSDV, and DSR) was proposed [34]. Majumdar and

coauthors tried to overcome the problem of high latency and unsuccessful delivery of data to the destination [35]. For this purpose, the advantages of the ant colony optimization technique were utilized. The use of artificial ants and artificial neighbors was considered to enhance the discovery of new paths.

Considering various parameters, the pros and cons of different protocols which are being utilized in FANETs have been discussed by Oubbati et al. [36]. Furthermore, future challenges have been discussed as well, which could be considered for research study and work. The focus of this paper is mainly on position-based routing protocols for FANETs. Saritha et al. proposed a new algorithm based on particle swarm optimization, leapfrog, and learning automata in [37].

The proposed strategy is supposed to find multiple paths for the data delivery considering the link stability. Leapfrog helps in determining the link failure in advance for avoiding any data loss. The results thus gathered show that the proposed algorithm performs better in terms of a better packet delivery ratio.

In [38], Bravo-Torres et al. proposed a virtual node layer which lies between the link layer and Internet layer which can enhance the work of AODV. The newly adapted AODV is termed as VNAODV which can give better results in a vehicular ad hoc environment. Dixit et al. surveyed the VANET architecture in [39]. They provided the research challenges and details about different routing protocols being adopted. Application and algorithms for VANET scenarios have also been discussed.

Maistrenko et al. tried to compare AODV, DSDV, and DSR with AntHocNet routing protocol [40]. After performing simulations, it was concluded that AntHocNet performs better, as the other three experimented protocols have low performance with highly mobile nodes.

For maximizing the throughput, Zeng and others presented a novel technique of embedding the sink or relay nodes over the aerial nodes. Due to such projection, the relay nodes were able to fly with great speed. This technique of utilizing the mobile relay nodes has enhanced the throughput gained as compared to the traditional relaying where the nodes acting as relay are static in wireless communications [41].

A novel technique has been proposed by Mozaffari for collection of data from the IoV that has been deployed or are being used on the ground via unmanned aerial vehicles deployed or moving in all the three dimensions. The technique resulted in better transmission power and data collection as compared to the conventional stationary aerial nodes deployed at a height from the ground [42].

For better performance in intelligent transportation systems, Yasser and coauthors have proposed a new strategy that can help the people living in developing countries or areas where there is lack of roadside units. They have utilized the vehicle-to-vehicle communication as freestanding system for intelligent transportation system. Different proactive and reactive routing protocols have been tested without the usage of RSU. The real-world simulations were performed with the utilization of OPNET simulator, and finally the

simulation results showed that utilizing such standalone system without roadside units has better performance for developing areas with utilization of AOD protocol [43].

## 4. Proposed Methodology

The proposed model's framework is shown in Figure 2. At the start, the network has been created by deploying only the vehicles with a certain transmission range. Certain parameters were set as per our requirement. We implemented our proposed strategy after evaluation of traditional VANET routing protocols, i.e., OLSR, AODV, and DSDV. We called this phase scenario a and its steps are listed below:

- (i) Routing protocols' selection
- (ii) Direct vehicular communication without the assistance of aerial nodes
- (iii) Results' generation

All the above-described steps followed the routing procedure in the traditional VANET. Here, scenario a of our proposed strategy ends, and we move to scenario b which is communication via aerial nodes. The steps involved in the routing of scenario b are as follows:

- (i) Changing altitude of sinks and deploying them as aerial nodes
- (ii) Communication between vehicles via these deployed aerial nodes
- (iii) Generation of results

A detailed discussion of both scenarios is as follows.

### 4.1. Routing by Using Traditional Vehicular Ad Hoc Network (Scenario A)

**4.1.1. Selection of Routing Protocol.** In our simulation, we considered OLSR, AODV, and DSDV protocol to check their efficiency in our first scenario. Each protocol depicted different results in terms of average throughput, average packet drop ratio, and MAC/PHY overhead.

**4.1.2. Direct Vehicular Communication.** In the first phase of our simulation, the deployed vehicles communicate with each other without the assistance of any aerial node. The vehicles have certain transmission ranges, some speed, and basic service message route from node to node to be delivered to our desired destination. These messages are initiated by a certain source.

**4.1.3. Generation of the Result.** Once the protocols have been selected, the simulations were performed, and the results were generated in the form of a graph for later comparison.

### 4.2. Routing via Aerial Nodes (Scenario B)

**4.2.1. Changing Altitude of Sink Nodes.** After all the procedures described above, we proceed ahead towards scenario b where we have changed the height of sink nodes to deploy



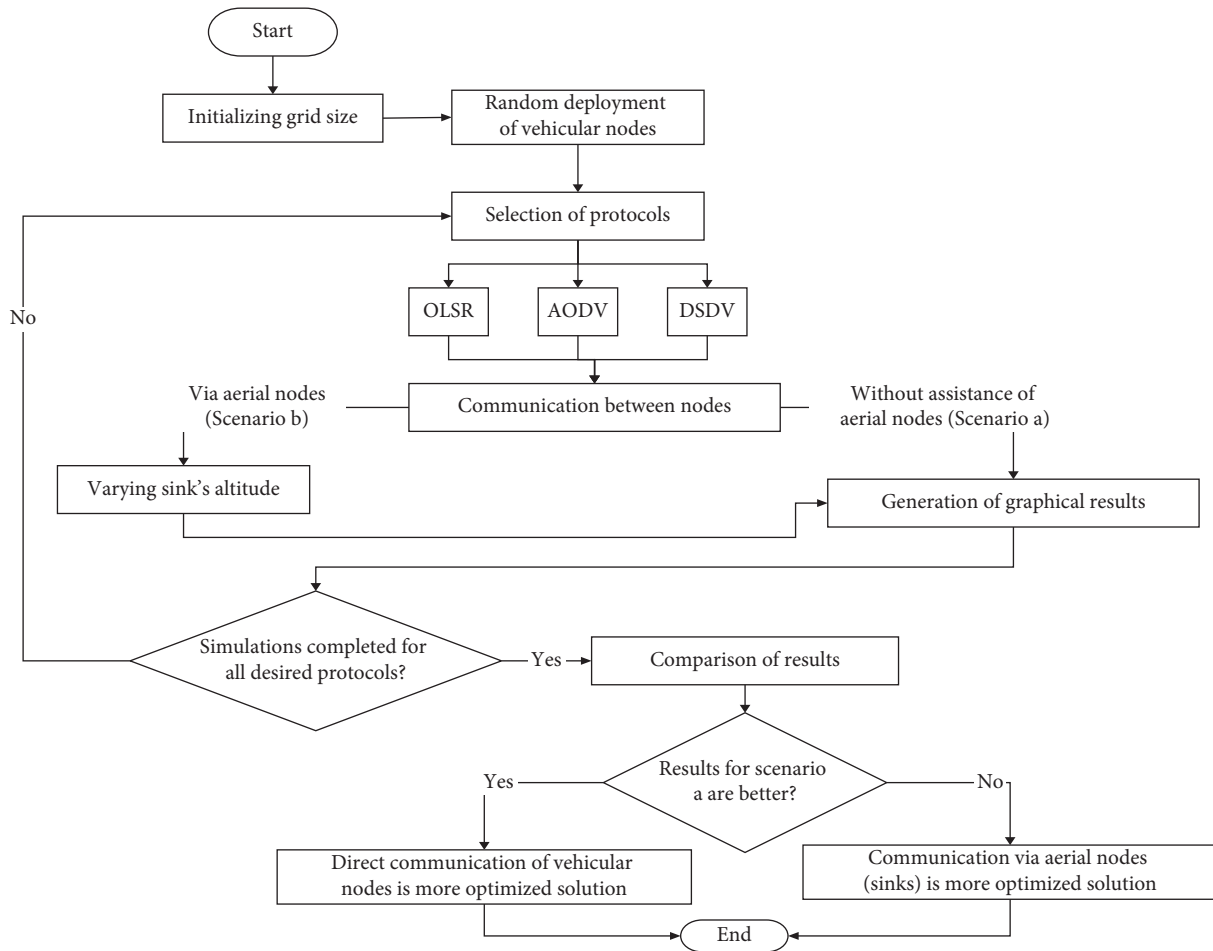


FIGURE 2: Framework of the proposed methodology.

them as aerial vehicles at some range and some altitude. The number of aerial nodes can be different for different scenarios. These aerial nodes have their specifications and parameters.

4.2.2. *Communication via Aerial Nodes.* Once the sink nodes have been deployed as aerial nodes, vehicles start communicating with each other indirectly via aerial vehicles. Such indirect communication can help us in an environment where vehicles are at such a distance from each other that they are unable to communicate directly with each other. Hence, such distanced vehicles can deliver their messages indirectly via these elevated sink nodes. These aerial vehicles have certain number and certain specifications like speed and altitude power consumption.

4.2.3. *Generation of Results in Graphical Form.* Once the simulations have been performed keeping scenario b under consideration, we have generated graphs. These graphs help us in comparing our two scenarios.

4.3. *Simulation Scenario.* All the simulations have been carried out in synthetic highway scenarios. For the performance evaluation of our proposed strategy, simulations

have been carried out in the NS3 simulator. For the analysis of the performance of OLSR, AODV, and DSDV protocols, in both scenarios, i.e., traditional vehicular ad hoc network and our proposed scenario of the Internet of Vehicles, the density of vehicles has been varied from low to high. It is done so that we can track the performance in an environment where we have congested traffic and where there is less vehicular traffic. The routing protocols that have been considered are OLSR, AODV, and DSDV. The rest of the parameters and their specifications are described in Table 1. The step-wise explanation of Figure 2 is mentioned as Algorithm 1.

4.4. *Description of Key Parameters.* The key parameters involved in the simulation are described in Table 2. These parameters helped us in evaluating and analyzing our proposed strategy to decide whether the traditional vehicular ad hoc network performs better or the assistance of aerial nodes would be beneficial.

4.5. *Evaluation Metrics.* The evaluation of our proposed strategy has been done by keeping in mind the following metrics.

TABLE 1: Simulation setup.

Parameters	Specification
Operating system	Ubuntu-18.04.3
MATLAB	R2015a
Simulator	NS3-3.30.1
Scenario	VANET (802.11p)
Mobility model	Random way point
Speed of vehicles	20 m/s
Pause time	300.01 s
Grid size	300 × 1500 m, 300 × 3000 m, 300 × 4500 m, 300 × 6000 m
Number of vehicular nodes	100,200,300,400,500
WiFi	802.11p
Control channel	10 MHz
Number of sink nodes	25, 50
Loss model	Two-ray ground
Transmission power	20 dBm
Transmission range	145 m
Total simulation time	300.01 s
Antenna height along z-axis	1.5 m (in scenario a), 50 m (in scenario b)

```

(1) START
(2) Define grid size
(3) Random deployment of Vehicular nodes
(4) Set up routing protocol
(5) Switch for the choice of protocol
(6) Case choice = "0" protocol = "none"
(7) Break
(8) Case choice = "1" protocol = "OLSR"
(9) Break
(10) Case choice = "2" protocol = "AODV"
(11) Break
(12) Case choice = "3" protocol = "DSDV"
(13) Break
(14) Otherwise protocol = "No such protocol"
(15) Assign IP addresses
(16) Setting up routing transmissions
(17) Configuring the values using VanetRouting Experiment
(18) Creating a WiFi channel
(19) Where WiFi channel = "WiFi-802.11p"
(20) Create c number of nodes where c is equal to 100, 200, 300, 400 or 500 and adding mobility
(21) Setting up routing messages
(22) Setting up one source as source node and the other as sink node message routing
(23) Create var as object for specifying the number of stream
(24) FOR i=0, I should be less than the number of sinks used
(25) If Choice of protocol is not equal to zero
(26) Get the address of sink
(27) Start routing with var equal to 1.0, 2.0 seconds
(28) Stop simulation when total simulation time ends.
(29) END_IF
(30) Iteration ++;
(31) END_FOR
(32) Print Received routing packets
(33) If Ipv4Address of received message matches with the source address
(34) Print one message received from this Ipv4Address
(35) ELSE
(36) One packet received
(37) END_IF

```

ALGORITHM 1: Continued.

```

(38) Set receive call back to acknowledge packet received
(39) Logging
(40) VanetRoutingExperiment experiment ();
(41)   WiFiApp ()
(42)   SetDefaultAttributeValues (); set default values to all attributes
(43)   ConfigureNodes (); configure all the nodes
(44)   ConfigureChannels (); for configuration of channels
(45)   ConfigureMobility (); configure the mobility
(46)   ConfigureApplications (); for configuration of applications
(47)   RunSimulations (); start and end simulations from zero seconds to the total simulation time
(48)   ProcessOutputs (); process the results obtained as output
(49)   CourseChange (); set up sinks' velocity and position
(50) main ();
(51)   VanetRoutingExperiment experiment ();
(52) END
    
```

ALGORITHM 1: UAV-assisted VANET routing protocol.

TABLE 2: Key parameters for simulation.

Parameters	Description
Transmission range	It defines the vehicle's range in which it would be able to communicate with other vehicles in the network. Varying the transmission range may impact the overall performance of the network.
Rate of transfer	It specifies how much data can be transferred in the given time.
Packet received	This parameter describes the number of successful packets received by the destination. It greatly impacts the performance of any network.
Packet size	Packet size may vary from network to network, but we will have a fixed size of packets that can be routed in the network.
Pause time	It is the controlling parameter specifying how much time a sink node will stay in a specified grid.
Simulation time	The total time taken for one whole simulation is considered in this parameter.
Received rate	This parameter tells us the amount of data received in bytes/kilobytes.
Packet loss	The number of packets that were not received by the destination due to communication error is calculated under the packet loss parameter.
No. of sinks	Sinks are the nodes that help us to gather and preprocess the data collected from the surroundings via sensor node. The number of deployed sink nodes for the desired scenario is specified under this parameter.
Grid size	The overall size of the grid in which the experimentation is done.
Node speed	This is the movement speed of the node.
Node direction	This is the direction of the vehicle on the road on which it is moving, it may include intersections and left, right, or straight lane.
Basic safety messages (BSM)	These messages do not help in routing, but rather they provide other useful information, yet they consume bandwidth of the network.

4.5.1. *Average Packet Delivery Ratio.* The average packet delivery ratio in any scenario tells us about the ratio of the number of packets received by the destination to the total number of packets sent by the source. It is an important parameter as it helps us to evaluate the performance of any network. The higher the average packet delivery ratio is, the higher the reliability of that network will be.

$$\text{average PDR} = \frac{\sum \text{no. of the packets received}}{\sum \text{no. of the packets sent}} \quad (1)$$

4.5.2. *Average Throughput.* Average throughput specifies, at any time, the amount of data sent from the source to the desired destination successfully. If we have a higher value of throughput, then the performance of our network will be enhanced. It can be calculated as

$$\text{throughput} = \frac{(\sum \text{no. of successful packets}) * (\text{average packet size})}{\text{transmission time}} \quad (2)$$

4.5.3. *MAC/PHY Overhead.* In vehicular ad hoc network, we use BSM that help us to share information in the network, whereas the information related to the updates of routing is disseminated by the routing packets. However, the routing packets do not provide any useful information related to the application. BSM as well as routing packets consume the bandwidth of the network, which affects the overall performance of the network. Hence, we call these routing packets causing an overhead on the network bandwidth as MAC/PHY overhead, and for their calculation, we need to know the total number of physical bytes and we should have

information about the total application byte. The MAC/PHY overhead in scenario a and in scenario b can be calculated as

$$\text{MAC/PHY Overhead} = \frac{(\text{totalPhyBytes} - \text{totalAppBytes})}{(\text{totalPhyBytes})} \quad (3)$$

## 5. Simulated Results

### 5.1. For Traditional Vehicular Ad Hoc Network

**5.1.1. MAC/PHY Overhead with 25 Sink Nodes.** For all the graphs presented in Figure 3, we have kept the number of sink nodes initially equal to 25, whereas vehicular nodes increase from 100 to 500.

In Figure 3(a), we have grid size =  $300 \times 1500$  m; MAC/PHY overhead ascends for OLSR as the number of nodes ascends/increases. In the case of AODV, MAC/PHY overhead descends when the number of nodes ascends from 100 to 200. It becomes constant when nodes increase from 200 to 300. Again, it ascends when the number of nodes ascends from 300 to 400, and after that it becomes constant. In DSDV, when the number of nodes increases, MAC/PHY overhead firstly increases and then becomes constant. Again, it increases and after that shows constant behavior.

It is clear from Figure 3(b), where we have a grid size of  $300 \times 3000$  m, that MAC/PHY overhead increases for OLSR with the increasing number of nodes. In the case of DSDV, MAC/PHY overhead ascends when the number of nodes ascends. In AODV, MAC/PHY overhead ascends when the number of nodes ascends from 100 to 200, it becomes constant for 200 to 300 nodes, and it ascends onwards.

Figure 3(c) shows that MAC/PHY overhead ascends for all the protocols of OLSR, AODV, and DSDV when the number of nodes increases while grid size is  $300 \times 4500$  m. The MAC/PHY overhead is the highest for AODV and is the lowest for OLSR, whereas for DSDV it lies in between them.

Figure 3(d) presents a scenario where the grid size is increased to  $300 \times 6000$  m. In the case of OLSR, MAC/PHY overhead ascends when the number of nodes is up to 300 and becomes constant when the number of nodes increases from 400 to 500. In DSDV, MAC/PHY overhead ascends with increasing the number of nodes; in the case of AODV, MAC/PHY overhead increases when the number of nodes increases from 100 to 200; then, it becomes constant when the number of nodes ascends from 200 to 300; and after that it ascends gradually.

**5.1.2. MAC/PHY Overhead with 50 Sink Nodes.** For all the graphs presented in Figure 4, we have kept the number of sink nodes equal to 50 whereas vehicular node increases from 100 to 500.

It is clear from Figure 4(a) that, in case of OLSR, MAC/PHY overhead descends gradually with the ascending number of nodes keeping the grid size equal to  $300 \times 1500$  m. While in the case of DSDV, MAC/PHY overhead ascends from 200 to 300, and there is a sudden increase when the number of nodes ascends from 200 to 300. After that,

MAC/PHY overhead descends with an ascending number of nodes. In the case of AODV, MAC/PHY overhead ascends when the number of nodes ascends from 100 to 200, after that it descends with ascending number of nodes till 400, and from 400 nodes onwards MAC/PHY overhead also ascends.

Figure 4(b) indicates that MAC/PHY the overhead of AODV is greater than those of the other two protocols, while OLSR has the least MAC/PHY overhead. The DSDV, the same as in the previous cases, lies between AODV and OLSR routing protocol. Grid size, in this case, has been increased from  $300 \times 1500$  m to  $300 \times 3000$  m.

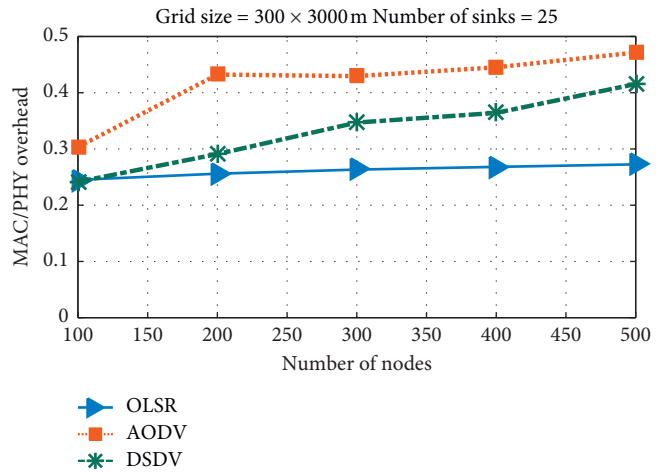
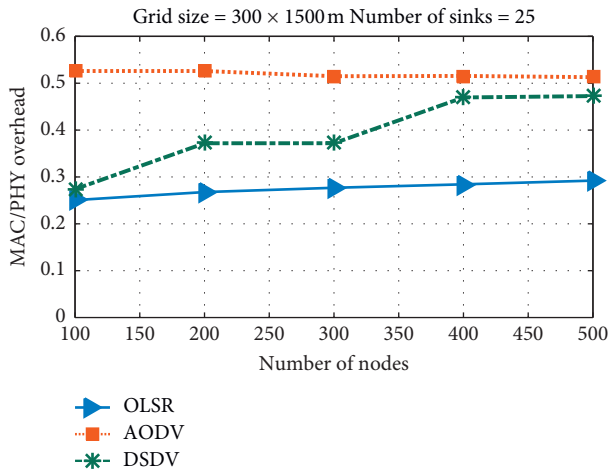
Figure 4(c) shows the results with a grid size equal to  $300 \times 4500$  m. The results demonstrate that MAC/PHY overhead ascends with the ascending number of nodes in both OLSR and DSDV, but its behavior is different in the case of AODV. In AODV, MAC/PHY overhead ascends when the number of nodes increases from 100 to 300 and, after that, it descends with the ascending number of nodes.

From Figure 4(d), in grid size  $300 \times 6000$  m, the MAC/PHY overhead for OLSR ascends slowly, with the ascending number of nodes. In DSDV, MAC/PHY overhead increases with the increasing number of nodes. In AODV, MAC/PHY overhead ascends when the number of nodes ascends from 100 to 300, then becomes constant when nodes increase from 300 to 400, and again increases.

**5.1.3. Average Throughput with 25 Sink Nodes.** For all the graphs in Figure 5, we keep a constant number of sink nodes, i.e., 25, and the vehicular nodes increase from 100 to 500. Grid size is incremented by 1500 m each time along the y-axis.

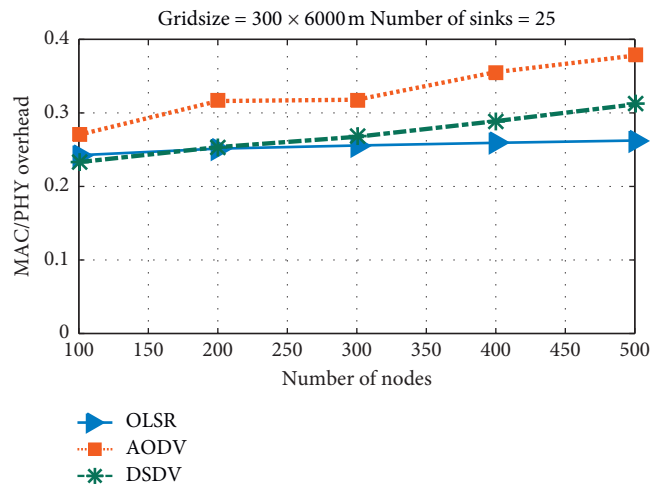
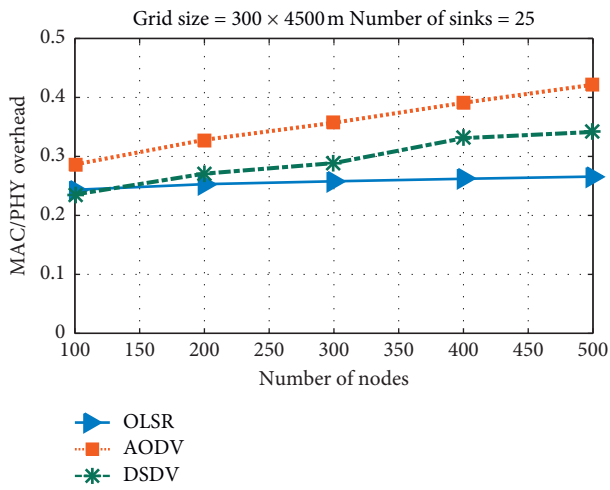
Figure 5(a), where the grid size is  $300 \times 1500$  m, represents the average throughput of three protocols (OLSR, AODV, and DSDV). Throughput for OLSR rapidly increases from 100 to 200 and then gradually decreases with nodes from 200 to 500. Protocol AODV shows average throughput on the first 100 nodes; then, it decreases rapidly; after 200 nodes, it shows average performance for 400 nodes; and then its performance is enhanced from 400 to 500 nodes. In the case of DSDV, throughput increases as the number of nodes ascends from 100 to 200 rapidly, and then it starts descending as the number of nodes reaches from 200 to 300. It increases as the number of nodes increases, and then again it starts descending.

In Figure 5(b), we considered the grid size equal to  $300 \times 3000$  m. OLSR shows the minimum average throughput as compared to AODV and DSDV. It remains constant from 100 to 200 nodes and then shows a gradual decrease as the number of nodes ascends, and after that it remains constant from 300 to 400 nodes. It shows a decrease in average throughput with nodes from 400 to 500. AODV shows a rapid increase in average throughput as the nodes ascend from 100 to 200, and then its throughput decreases from 200 to 300. After that, it shows a gradual decrease as the number of nodes goes up. DSDV shows a good increase in throughput from 100 to 300, but it decreases as the nodes ascend from 300 to 400. Then again, it shows an increase in the average throughput from 400 to 500.



(a)

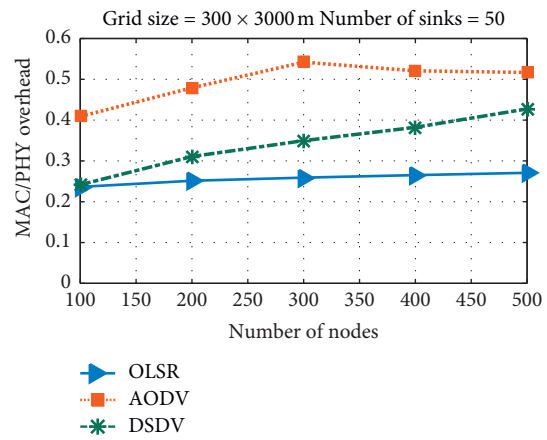
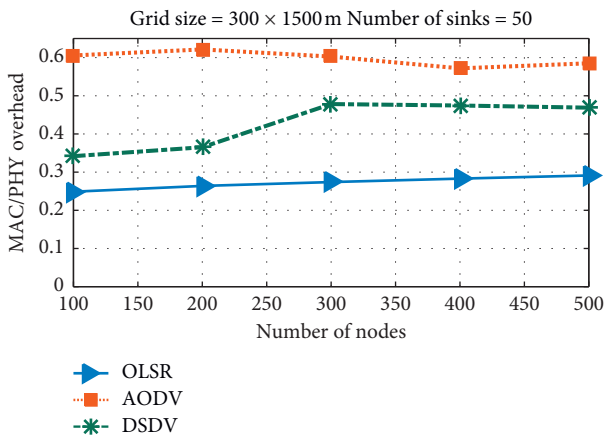
(b)



(c)

(d)

FIGURE 3: MAC/PHY overhead in traditional VANET with no. of sinks = 25.



(a)

(b)

FIGURE 4: Continued.



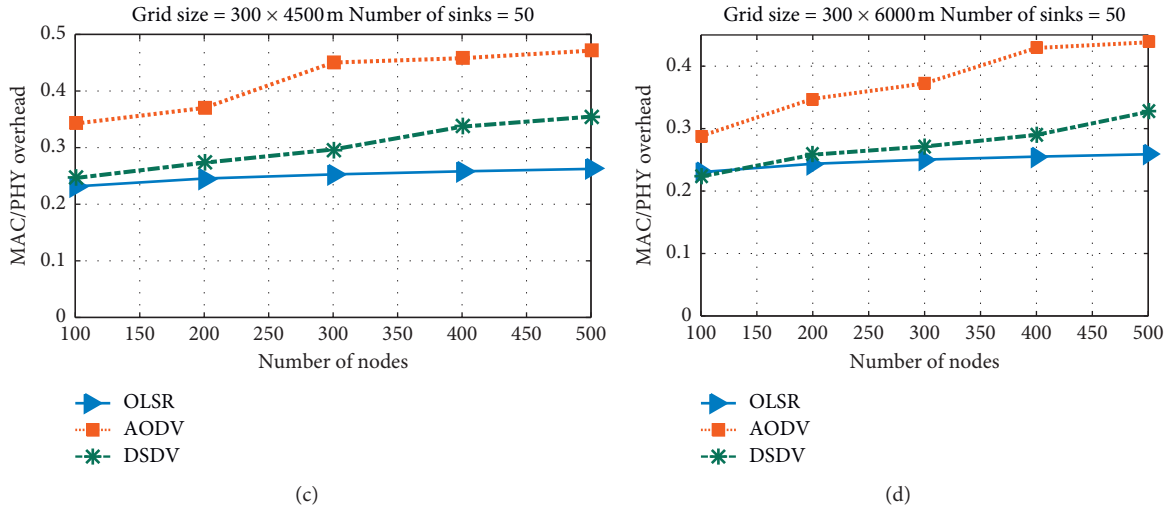


FIGURE 4: MAC/PHY overhead in traditional VANET with no. of sinks = 50.

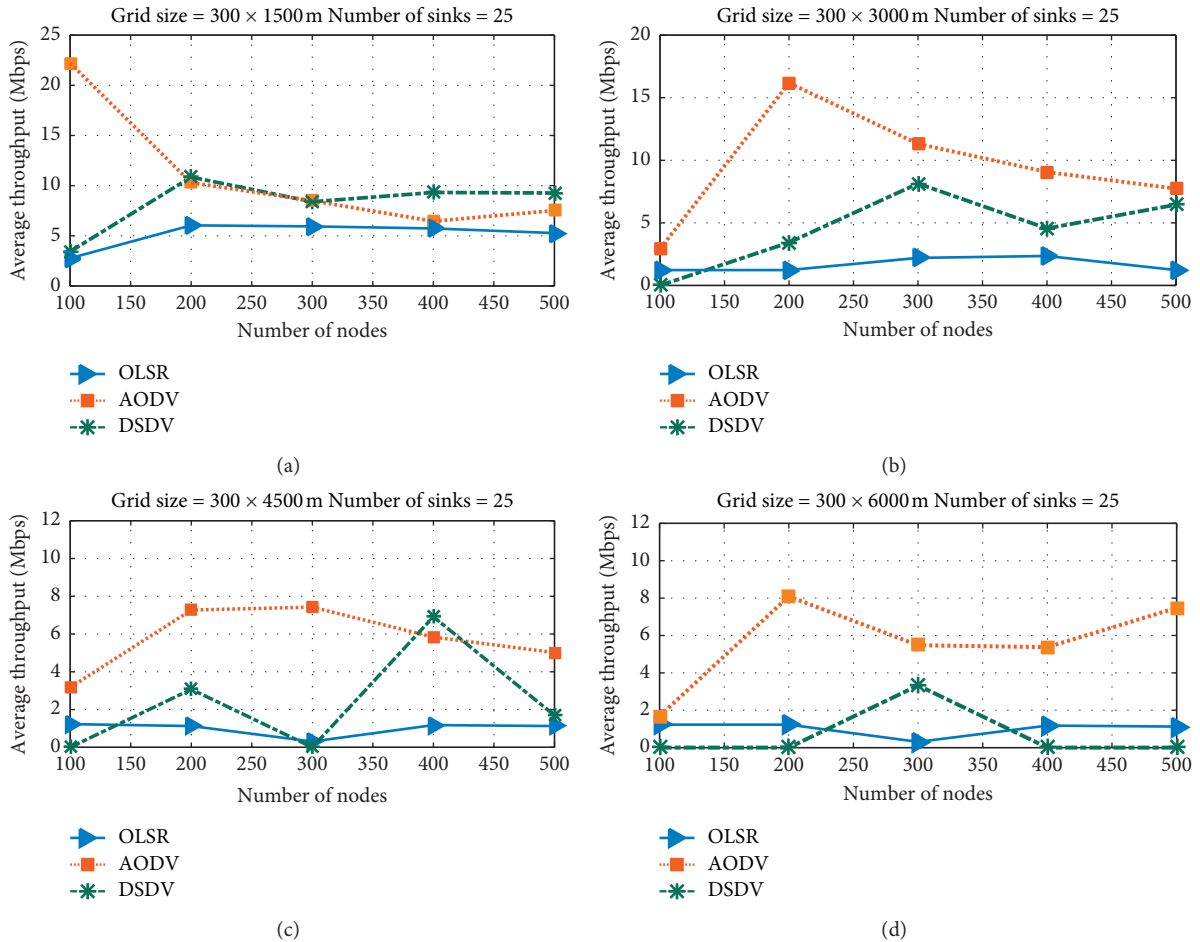


FIGURE 5: Average throughput in traditional VANET with no. of sinks = 25.

From Figure 5(c), we can see that OLSR shows minimum throughput; it starts decreasing as the number of nodes ascends from 100 to 300 and then gradually increases as the

number ascends. AODV shows high throughput overall as it increases from 100 to 300 but then starts decreasing from 300 to 500. DSDV exhibits a rise in average throughput from

100 to 200 nodes, then there is a fall in average throughput as nodes ascend, and its average throughput again starts rising with nodes from 300 to 400. After 400 nodes, its average throughput decreases as the nodes ascend. The grid size in this scenario is  $300 \times 4500$  m.

When we have a grid size of  $300 \times 6000$  m, as shown in Figure 5(d), the DSDV protocol shows the lowest average throughput with nodes from 100 to 200 and from 400 to 500. On the other hand, the highest average throughput is depicted in the AODV protocol. OLSR operates in reverse to the DSDV protocol. Where there is a decrease in average throughput in the case of DSDV, the throughput for OLSR ascends. In the rest of the scenario, throughput for OLSR lies between the other two protocols.

**5.1.4. Average Throughput with 50 Sink Nodes.** For all graphs presented in Figure 6, the number of sinks is equal to 50, and the number of nodes increases from 100 to 500, whereas the initial grid size is  $300 \times 1500$  m which has an increase of 1500 m along the  $y$ -axis in the rest of the cases.

Figure 6(a) indicates that OLSR has the least average throughput when we have a grid size of  $300 \times 1500$  m. AODV shows good performance on the first 100 nodes but then gradually decreases as the nodes ascend from 200 to 500. DSDV has a good average throughput as the nodes increase from 200 to 400. It has a comparatively less average throughput for the rest of the nodes.

OLSR almost remains constant with little rise and fall from 100 to 500 nodes within a grid of size  $300 \times 3000$  m as shown in Figure 6(b). AODV shows a decrease in average throughput with nodes from 100 to 200; then, there is an increase in average throughput as the number of nodes increases. It again slopes down with nodes from 300 to 400. Its average throughput remains constant with nodes from 400 to 500. DSDV shows high average throughput as compared to AODV and OLSR with nodes from 100 to 500.

Figure 6(c) represents simulations in grid size of  $300 \times 4500$  m. OLSR gives low average throughput and almost remains constant with nodes from 100 to 500, with little rise and fall. AODV shows a high average throughput with nodes from 100 to 400, and then it gradually decreases as the nodes ascend. DSDV remains constant with nodes from 100 to 400, with a little increase, and decreases after that with nodes from 400 to 500. Anyhow, the best performance in this scenario is depicted in AODV, and OLSR protocol is least performing.

Figure 6(d) shows the average throughput of three protocols in the grid size of  $300 \times 6000$  m. OLSR almost remains constant and gives the least throughput throughout the simulation with nodes from 100 to 500, with a little increase and decrease. DSDV shows average performance with the number of nodes from 100 to 300 and then increases as the number increases from 300 to 500. AODV performs best, and its throughput increases as the number of nodes increases from 100 to 300 and then gradually decreases with nodes from 300 to 500. Still, it behaves better than the other two protocols.

**5.1.5. Average Packet Delivery Ratio with 25 Sink Nodes.** For all the four grid sizes, i.e.,  $300 \times 1500$  m to  $300 \times 6000$  m, we have deployed 25 sink nodes for each case. The grid size increases with an equal interval of 1500 m along the  $y$ -axis each time. We have kept the number of vehicular nodes constant, that is, from 100 to 500. Figures 7(a)–7(d) show the average packet delivery ratio of the three protocols, OLSR, AODV, and DSDV. In grid size of  $300 \times 1500$  m, the performance of AODV is less than the other two protocols, whereas OLSR and DSDV show very close results. At 100 nodes, the three protocols give the highest average packet delivery ratio, but as we increase the number of vehicular nodes, there is a decrease in average packet delivery ratio in all the four grid sizes. However, it is obvious from Figures 7(a)–7(d) that the three protocols are giving a better average packet delivery ratio as we increase the grid size each time. The performance of the three protocols is better at grid size  $300 \times 6000$  m than that at  $300 \times 1500$  m. Accordingly, we can conclude that the average packet delivery ratio increases as we increase the grid size.

**5.1.6. Average Packet Delivery Ratio with 50 Sink Nodes.** For all the four grid sizes, i.e.,  $300 \times 1500$  m to  $300 \times 6000$  m, we have deployed 25 sink nodes for each case. The grid size increases with an equal interval of 1500 m along the  $y$ -axis each time. We have kept the number of vehicular nodes constant, that is, from 100 to 500. From Figure 8(a), we can see that, at grid size of  $300 \times 1500$  m, the OLSR and DSDV have the highest performance with 100 vehicular nodes, but as we ascend towards vehicular nodes equal to 500, the average packet delivery ratio decreases. The performance of AODV at this point is less than that of the other two protocols. Later, when we increase the grid size up to  $300 \times 3000$  m,  $300 \times 4500$  m, and  $300 \times 6000$  m, respectively, the DSDV protocol behaves better than the other two, as could be seen from Figures 8(b)–8(d).

**5.2. For Drone-Assisted Vehicular Ad Hoc Network.** After completing the simulations for traditional VANET, we performed extensive simulations for scenario b where we have made use of aerial vehicles. Results are computed against MAC/PHY overhead, average throughput, and average packet delivery ratio.

**5.2.1. MAC/PHY Overhead with 25 Sink Nodes.** Simulated results for MAC/PHY overhead in our scenario b with the assistance of aerial nodes are presented in Figures 9(a)–9(d). Here, the number of sink nodes is equal to 25, and vehicular nodes are taken from 100 to 500. It is clear from Figure 9(a) that MAC/PHY overhead for DA-DSDV increases when the number of nodes increases from 100 to 300. It shows a decrease in performance with nodes from 300 to 400 and, later on, there is an enhancement in its performance once again. MAC/PHY overhead for DA-AODV ascends when the number of nodes ascends from 100 to 200, and it descends when the number of nodes ascends from 200 to 400 and becomes constant after that.

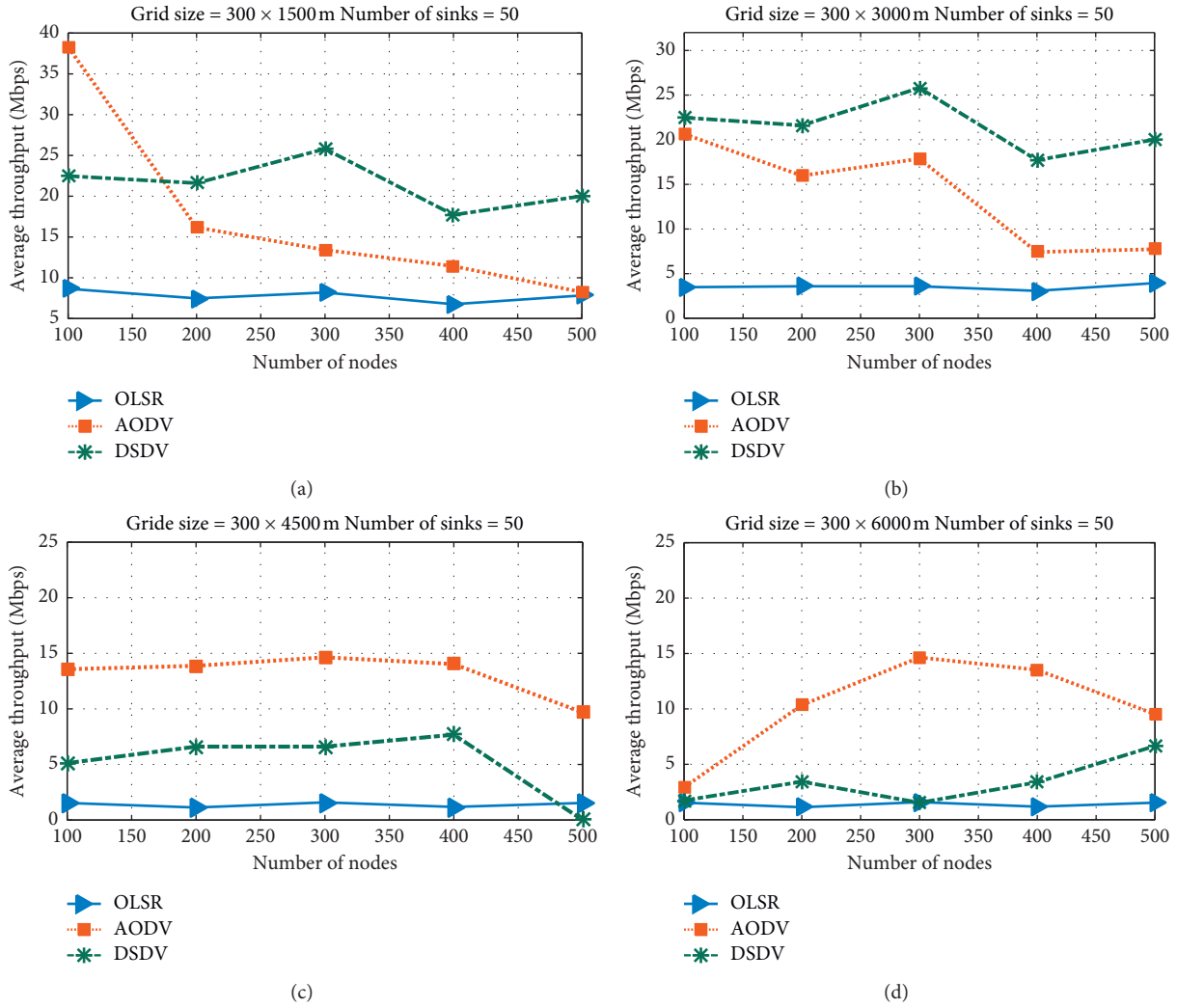


FIGURE 6: Average throughput in traditional VANET with no. of sinks = 50.

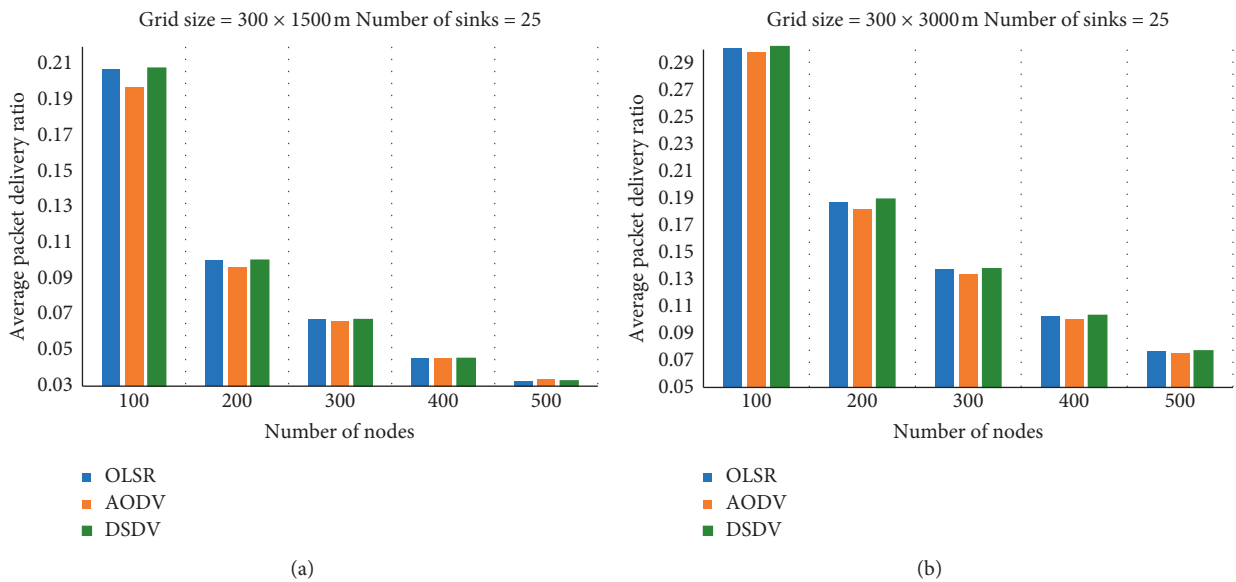


FIGURE 7: Continued.

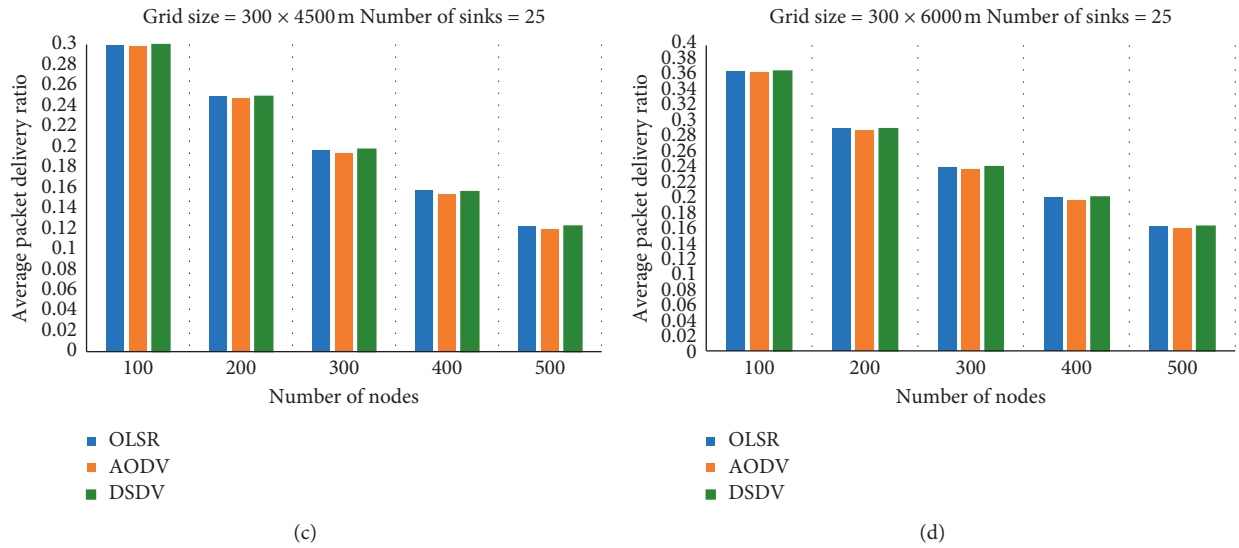


FIGURE 7: Average packet drop ratio in traditional VANET with no. of sinks = 25.

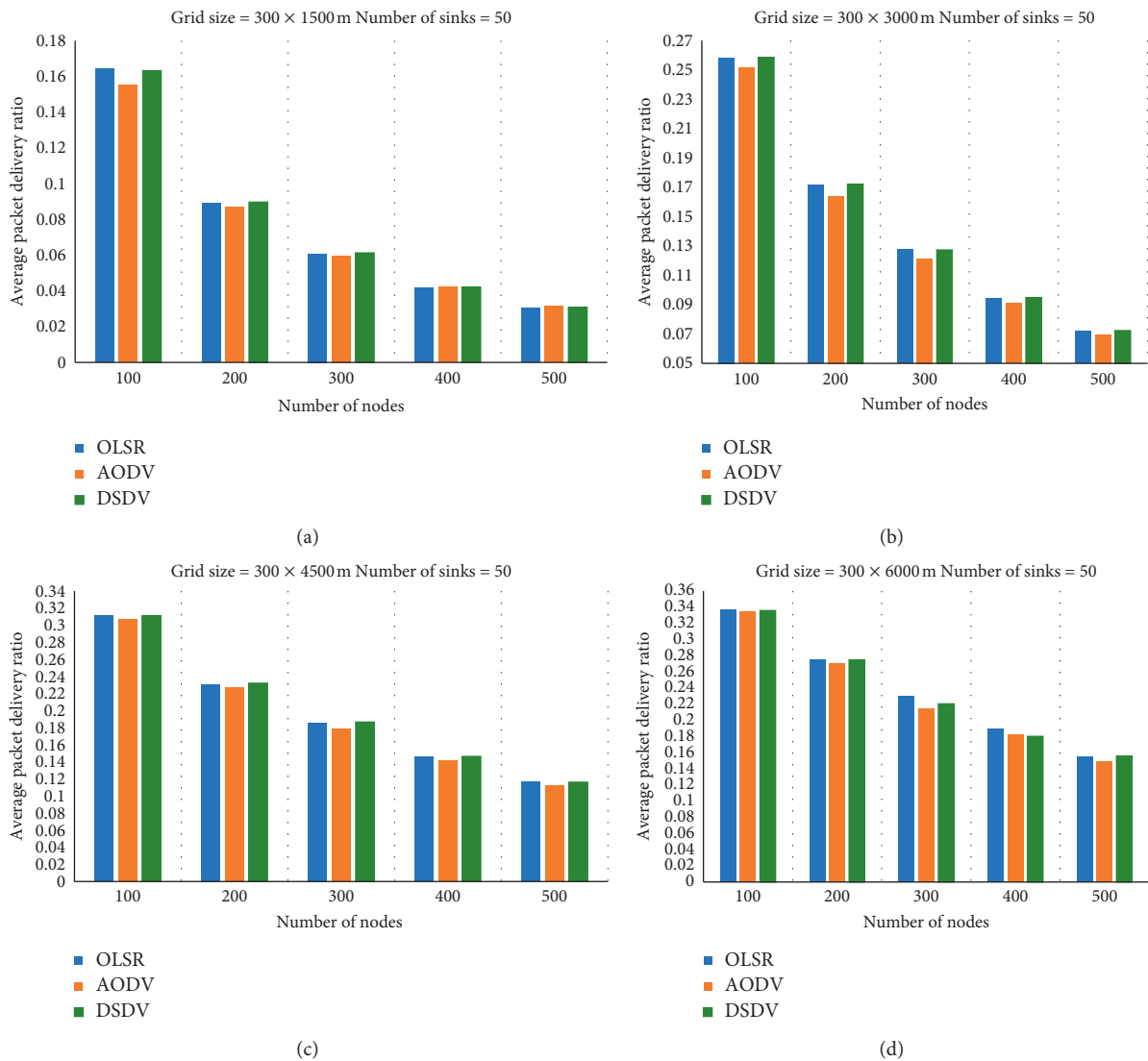


FIGURE 8: Average packet drop ratio in traditional VANET with no. of sinks = 50.

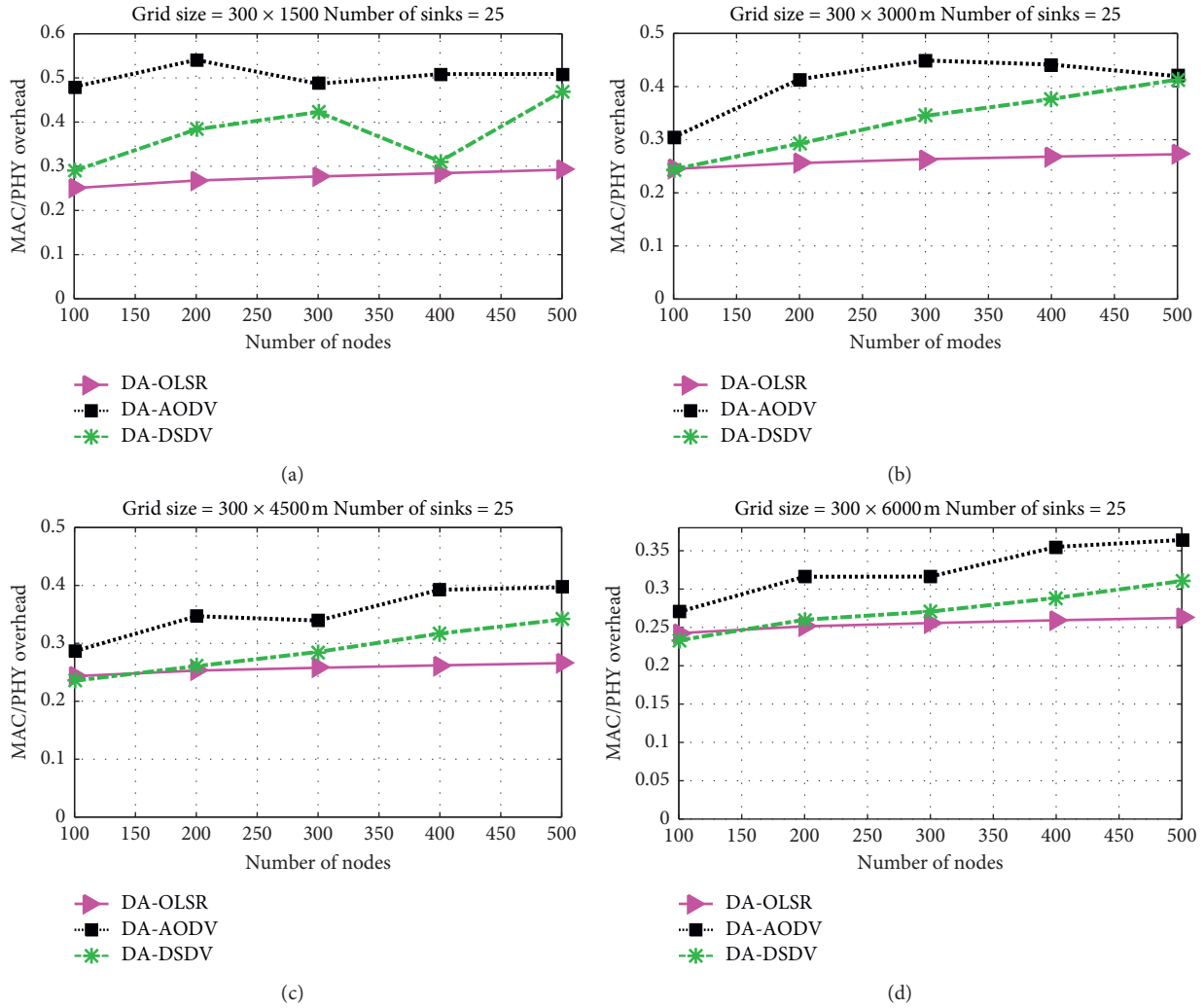


FIGURE 9: MAC/PHY overhead with drone assistance and no. of sinks = 25.

Figure 9(b) represents results for drone-assisted protocol within grid size of  $300 \times 3000$  m. It is clear from Figure 10 that MAC/PHY overhead for DA-OLSR ascends slightly with the ascending number of nodes. MAC/PHY overhead for DA-DSDV ascends when the number of nodes ascends. MAC/PHY overhead for DA-AODV ascends when the number of nodes ascends from 100 to 300, and it decreases when we have 300 to 500 vehicular nodes.

It is clear from Figure 9(c) that MAC/PHY overhead for DA-OLSR ascends with the ascending number of nodes. MAC/PHY overhead for DA-DSDV increases when the number of nodes increases. MAC/PHY overhead for DA-AODV ascends when the number of nodes ascends from 100 to 200, and it becomes constant when the number of nodes ascends from 200 to 300. Its performance is enhanced when the number of nodes ascends from 300 to 400 and becomes constant after that. Here, the grid size is  $300 \times 4500$  m.

Figure 9(d) shows results simulated in a grid size of  $300 \times 6000$  m. The MAC/PHY overhead for DA-OLSR

ascends linearly with the ascending number of nodes. MAC/PHY overhead for DA-DSDV increases when the number of nodes increases. It can be seen easily that MAC/PHY overhead for DA-AODV is higher than that of all the other protocols, whatever the number of nodes we have.

**5.2.2. MAC/PHY Overhead with 50 Sink Nodes.** The number of sink nodes is kept constant for all the scenarios shown in Figure 10. The grid size increases from  $300 \times 1500$  m to  $300 \times 6000$  m whereas the number of vehicular nodes is from 100 to 500. The protocols involved in the simulations are OLSR, AODV, and DSDV with the assistance of aerial nodes. Figure 10(a) shows the simulated results generated for grid size  $300 \times 1500$  m. The performance of DA-OLSR decreases with a slight change at every point throughout the simulations. The DA-DSDV performance is low at the start and up to node 300, after that its performance is neither



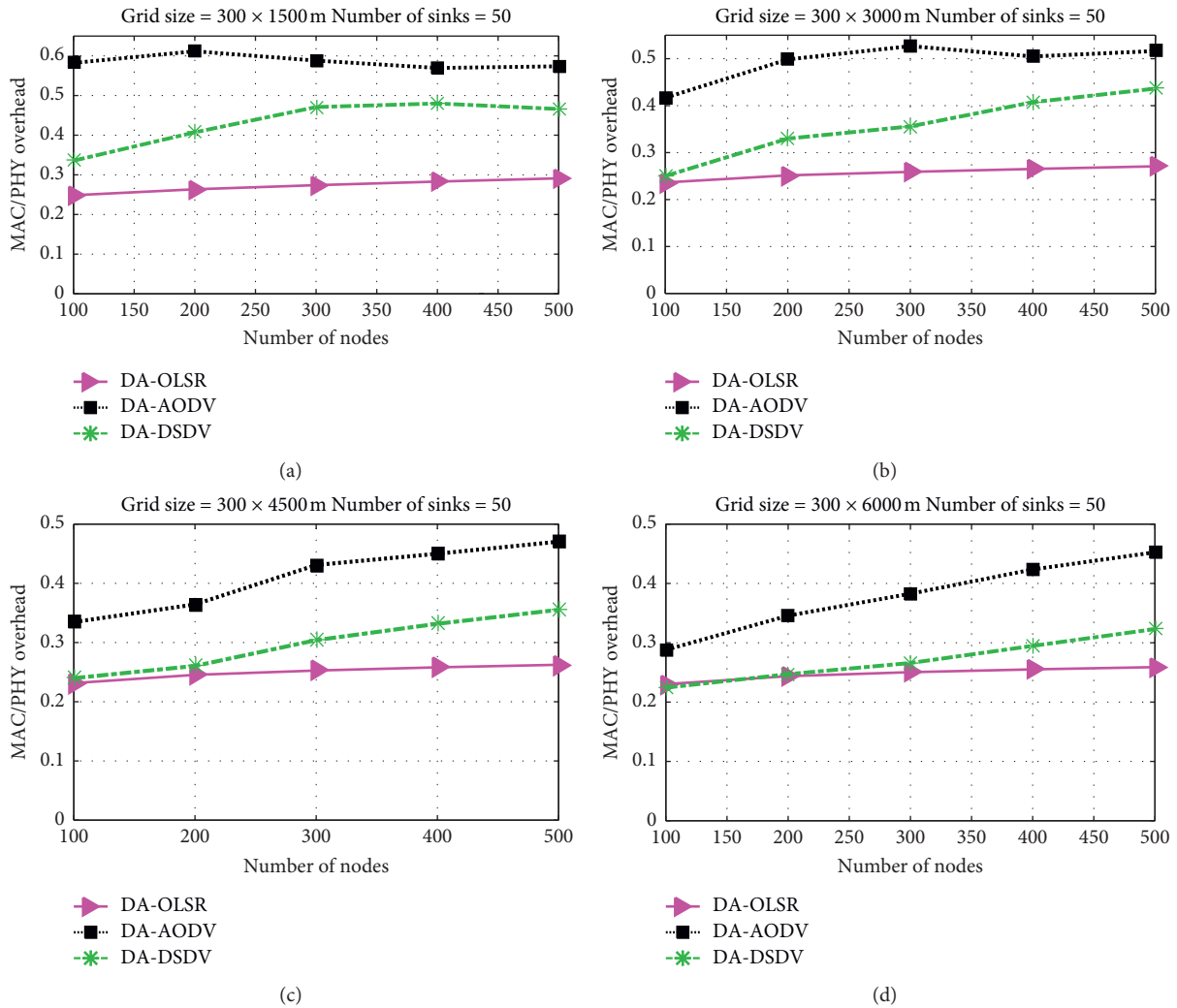


FIGURE 10: MAC/PHY overhead with drone assistance and no. of sinks = 50.

increasing nor decreasing, but as the number of nodes reaches 400, the performance of DA-DSDV starts to enhance. From Figure 10(b), we can conclude that the performance of DA-OLSR is the best one while the DA-AODV has the least performance within grid size of 300 × 3000 m. The DA-DSDV lies between the other two protocols.

Results simulated in grid sizes of 300 × 4500 m and 300 × 6000 m are shown in Figures 10(c) and 10(d), respectively. We can see clearly that the DA-AODV protocol has the highest MAC/PHY overhead throughout the simulations as compared to the other two protocols, while DA-OLSR has the least MAC/PHY overhead. In the case of DA-DSDV, MAC/PHY overhead increases with the increase in the number of vehicular nodes.

**5.2.3. Average Throughput with 25 Sink Nodes.** The average throughput for DA-OLSR, DA-AODV, and DA-DSDV calculated with 25 sink nodes is represented in Figure 11. The size of the grid increases from 300 × 1500 m to 300 × 6000 m

with an interval of 1500 m each time. The number of vehicular nodes ranges from 100 to 500. Figure 11(a) shows that the average throughput for DA-OLSR increases as nodes ascend from 100 to 200 and then remains constant with vehicular nodes from 200 to 400. It shows a gradual decrease with nodes from 450 to 500. The second protocol, DA-AODV, shows excellent throughput on the first 100 nodes but then shows a rapid decrease with nodes from 100 to 500. In the case of DA-DSDV, average throughput increases as the number of nodes increases from 100 to 200 and then remains constant with nodes from 200 to 300; after that, its performance gets worse with nodes from 300 to 400. There is an increase once again in the throughput as the number of nodes reaches from 400 to 500.

From Figure 11(b), it is clear that the DA-OLSR shows the minimum average throughput as compared to the other ones. It remains constant with nodes from 100 to 200, and then its throughput decreases as the number of nodes ascends from 200 to 400. Its performance degrades with nodes from 400 to 500. DA-AODV shows a high average

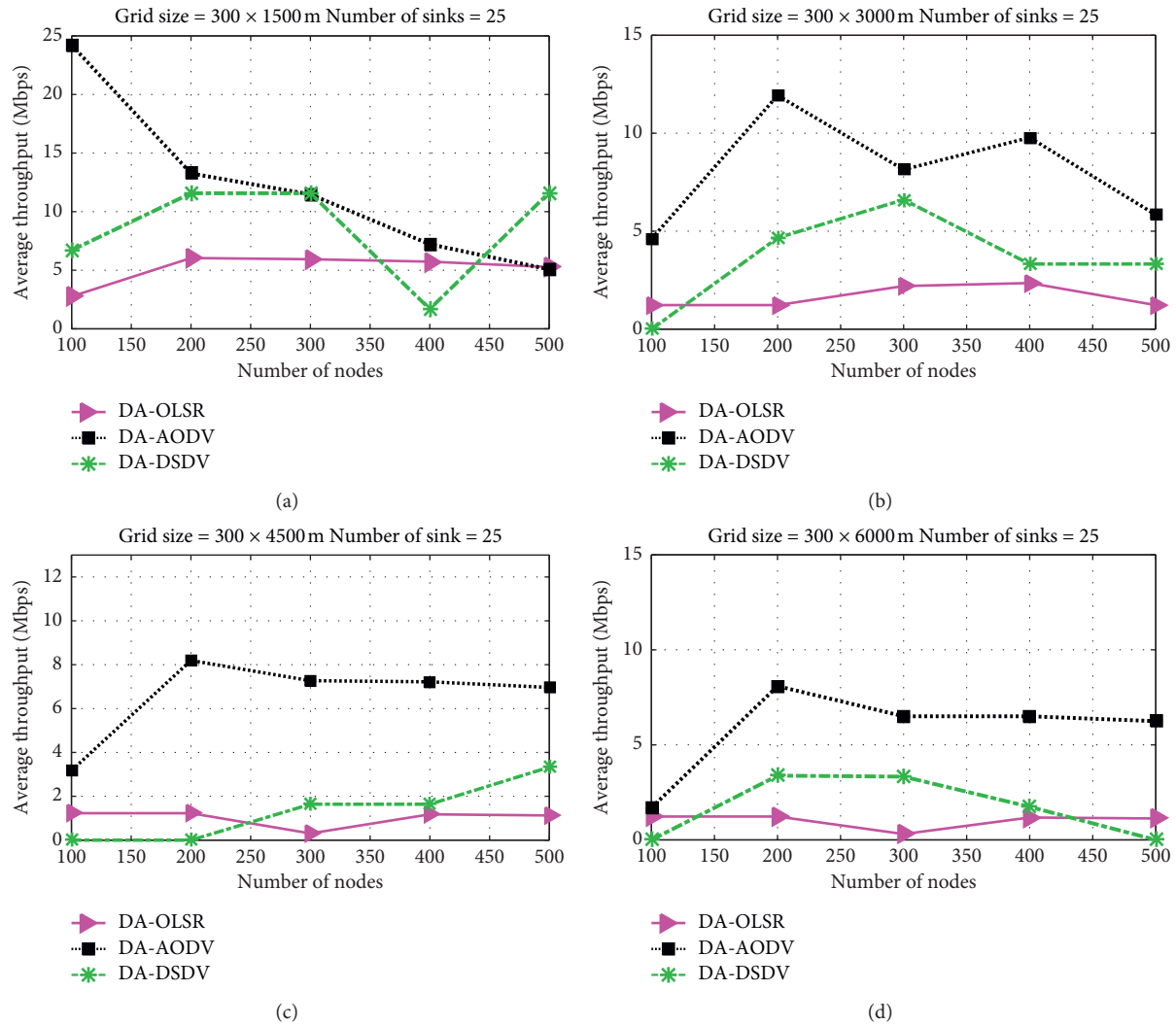


FIGURE 11: Average throughput with drone assistance and no. of sinks = 25.

throughput as compared to the other ones. It increases rapidly as the number of nodes ascends from 100 to 200, and then it starts decreasing with nodes from 200 to 300. Then, again it rises with nodes from 300 to 400, and after that it again starts decreasing as the number of nodes ascends from 400 to 500. DA-DSDV shows an increase as the number of nodes ascends from 100 to 300, and then it decreases with nodes from 300 to 400; after that, it remains constant with nodes from 400 to 500.

As shown in Figure 11(c) DA-OLSR shows minimum throughput throughout, as it remains constant with nodes from 100 to 200, and then it decreases as the nodes ascend and then rises with nodes from 300 to 400. Then, it remains constant again with nodes from 400 to 500. DA-AODV shows good throughput as it increases as the number of nodes ascends, and then it gradually decreases with nodes from 200 to 300; after that, it almost remains constant from 300 to 500 nodes. DA-DSDV shows minimum throughput with nodes from 100 to 200. The performance of DA-DSDV is enhanced as we increase the number of vehicular nodes from 200 to 500.

Figure 11(d) demonstrates the average throughput in grid size of  $300 \times 4500$  m. As can be seen, DA-OLSR remains constant but has less throughput at the start. Its throughput decreases as nodes ascend, and then again it starts rising with nodes from 300 to 400. The throughput is constant with nodes from 400 to 500. DA-AODV shows a rapid increase in throughput as nodes ascend from 100 to 200, and then its performance degrades with nodes from 200 to 300. The average throughput has a constant value with nodes from 300 to 500. DA-DSDV shows an increase in throughput as nodes ascend from 100 to 200; then, with nodes from 200 to 300, it remains constant; and after that it decreases with nodes from 300 to 500.

**5.2.4. Average Throughput with 50 Sink Nodes.** For the scenario shown in Figures 12(a)–12(d), we have the number of sink nodes equal to 50. The grid size increases from  $300 \times 1500$  m to  $300 \times 6000$  m. We have taken the vehicular nodes from 100 to 500.

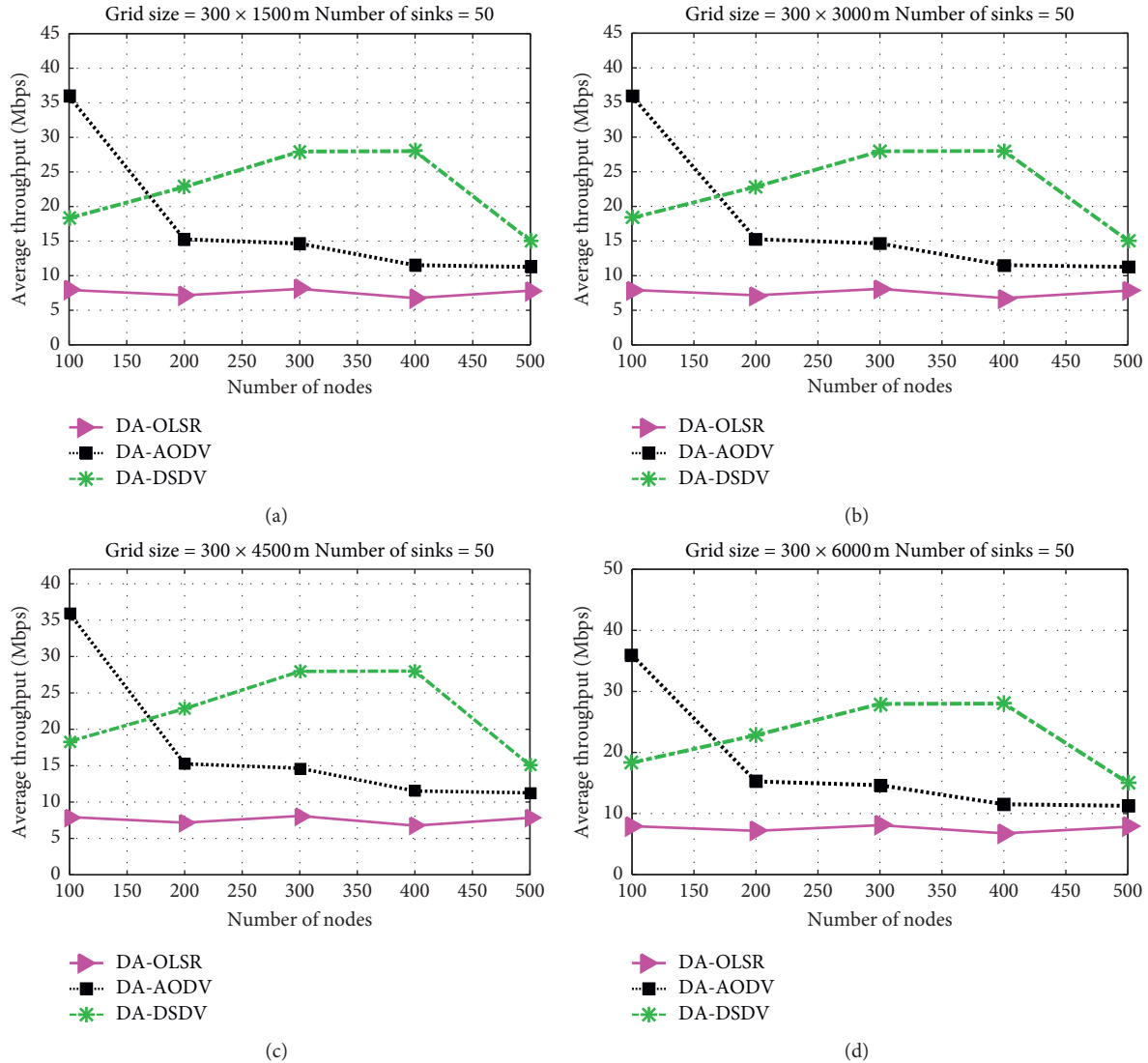


FIGURE 12: Average throughput with drone assistance and no. of sinks = 50.

Figures 12(a) and 12(b) show the average throughput within grid sizes of  $300 \times 1500$  m and  $300 \times 3000$  m. Average throughput in the case of DA-OLSR remains constant as the number of nodes increases with a little rise and fall with nodes from 100 to 500. DA-AODV has the highest performance at 100 nodes, and then it decreases as the number of nodes increases up to 500. DA-DSDV performance increases as the number of nodes ascend from 100 to 400 and then remains constant with nodes from 300 to 400. With nodes from 400 to 500, the throughput decreases.

Figures 12(c) and 12(d) represent average throughput in grid sizes of  $300 \times 4500$  m and  $300 \times 6000$  m, respectively. DA-OLSR almost remains constant with a little increase and decrease as the nodes ascend. DA-AODV decreases rapidly as the number of vehicular nodes decreases from 100 to 500. DA-DSDV shows an increase as the number of nodes increases from 100 to 300, and then it shows a constant value with nodes between 300 and 400. With nodes from 400 to 500, the average throughput decreases.

**5.2.5. Average Packet Delivery Ratio with 25 Sink Nodes.** For the scenario shown in Figures 13(a)–13(d), we have the number of sink nodes equal to 25. The grid size increases from  $300 \times 1500$  m to  $300 \times 6000$  m. We have taken the vehicular nodes from 100 to 500. Figures 13(a)–13(d) show the average packet delivery ratio of the three protocols, DA-OLSR, DA-AODV, and DA-DSDV. In grid size of  $300 \times 1500$  m, the performance of DA-AODV is less than that of the other two protocols, whereas DA-OLSR and DA-DSDV show very close results. At 100 nodes, the three protocols give the highest average packet delivery ratio, but as we ascend the number of vehicular nodes, there is a decrease in average packet delivery ratio in all four grid sizes. However, it is obvious from Figures 13(a)–13(d) that the three protocols are giving a better average packet delivery ratio as we ascend the grid size each time. The performance of the three protocols is better at grid size of  $300 \times 6000$  m than that at  $300 \times 1500$  m. Moreover, the DA-DSDV surpasses the DA-OLSR and DSDV at vehicular nodes 100 to

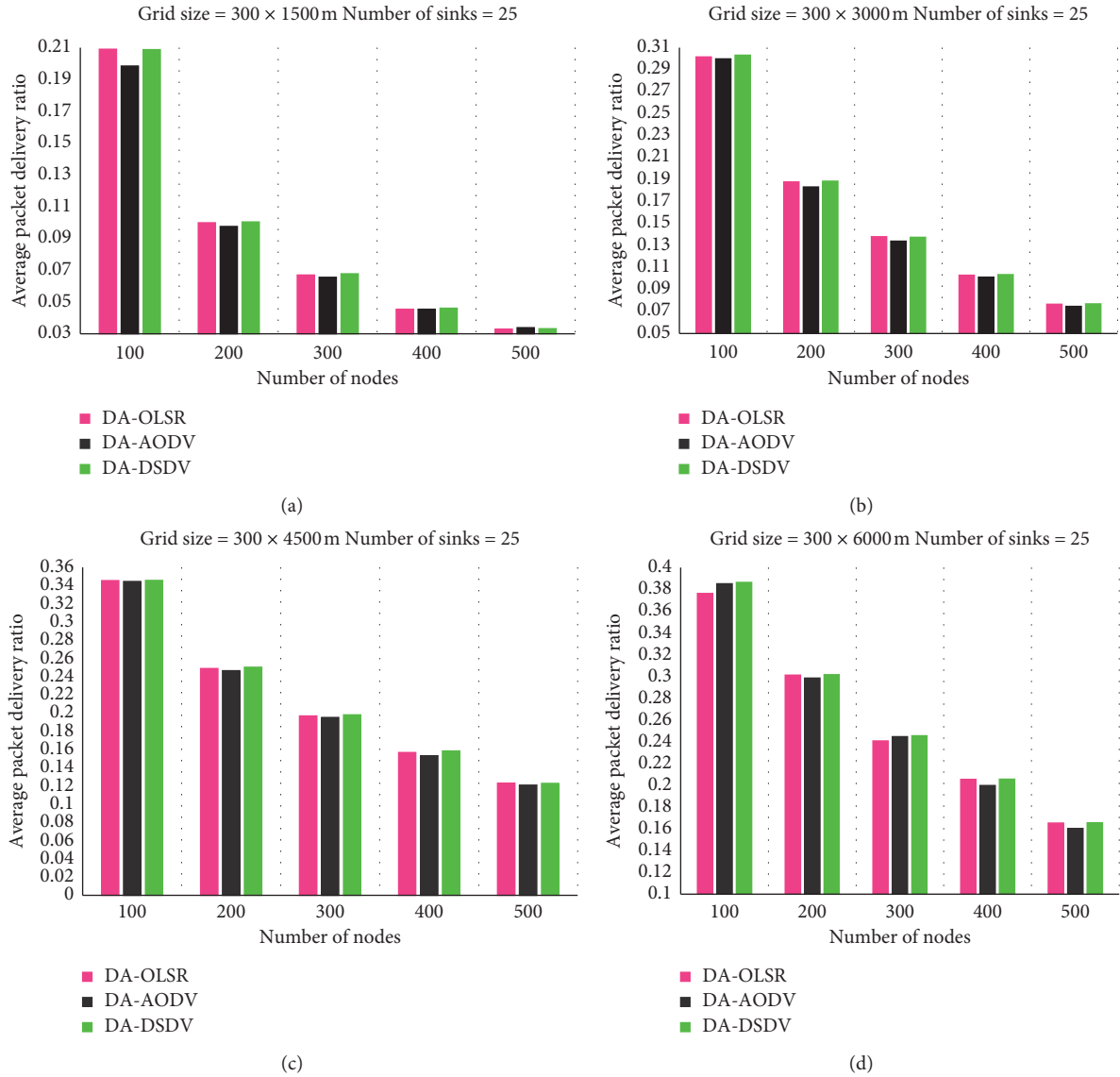


FIGURE 13: Average packet delivery ratio with drone assistance and no. of sinks = 25.

500. Thus, we can conclude that average packet delivery ratio increases as we increase the grid size.

**5.2.6. Average Packet Delivery Ratio with 50 Sink Nodes.** For the scenario shown in Figures 14(a)–14(d), we have the number of sink nodes equal to 50. The grid size increases from  $300 \times 1500$  m to  $300 \times 6000$  m. We have taken the vehicular nodes from 100 to 500. Figures 15(a)–15(d) show the average packet delivery ratio of the three protocols, DA-OLSR, DA-AODV, and DA-DSDV. In grid size of  $300 \times 1500$  m, the performance of DA-AODV is less than that of the other two protocols, whereas DA-OLSR and DA-DSDV show very close results. At 100 nodes, the three protocols give the highest average packet delivery ratio, but as we ascend the number of vehicular nodes, there is a decrease in average packet delivery ratio in all four grid sizes. However, it is obvious from Figures 14(a)–14(d) that the three protocols are giving a better average packet delivery

ratio as we ascend the grid size each time. The performance of the three protocols is better at grid size of  $300 \times 6000$  m than that at  $300 \times 1500$  m. Furthermore, the DA-DSDV surpasses the DA-OLSR and DSDV at vehicular nodes 100 to 500. Consequently, we can conclude that average packet delivery ratio increases as we increase the grid size.

## 6. Comparative Analysis of Traditional VANET and Drone-Assisted VANET

For detailed analysis to figure out which scenario is better for the IoV environment, we have combined the traditional VANET routing protocols and drone-assisted VANET protocols. These combined graphs will help in a deep insight into the conducted simulations.

**6.1. MAC/PHY Overhead with 25 Sink Nodes.** Figure 15 illustrates that, for all the grid sizes, the OLSR and DA-OLSR have the least MAC/PHY overhead, whereas the highest

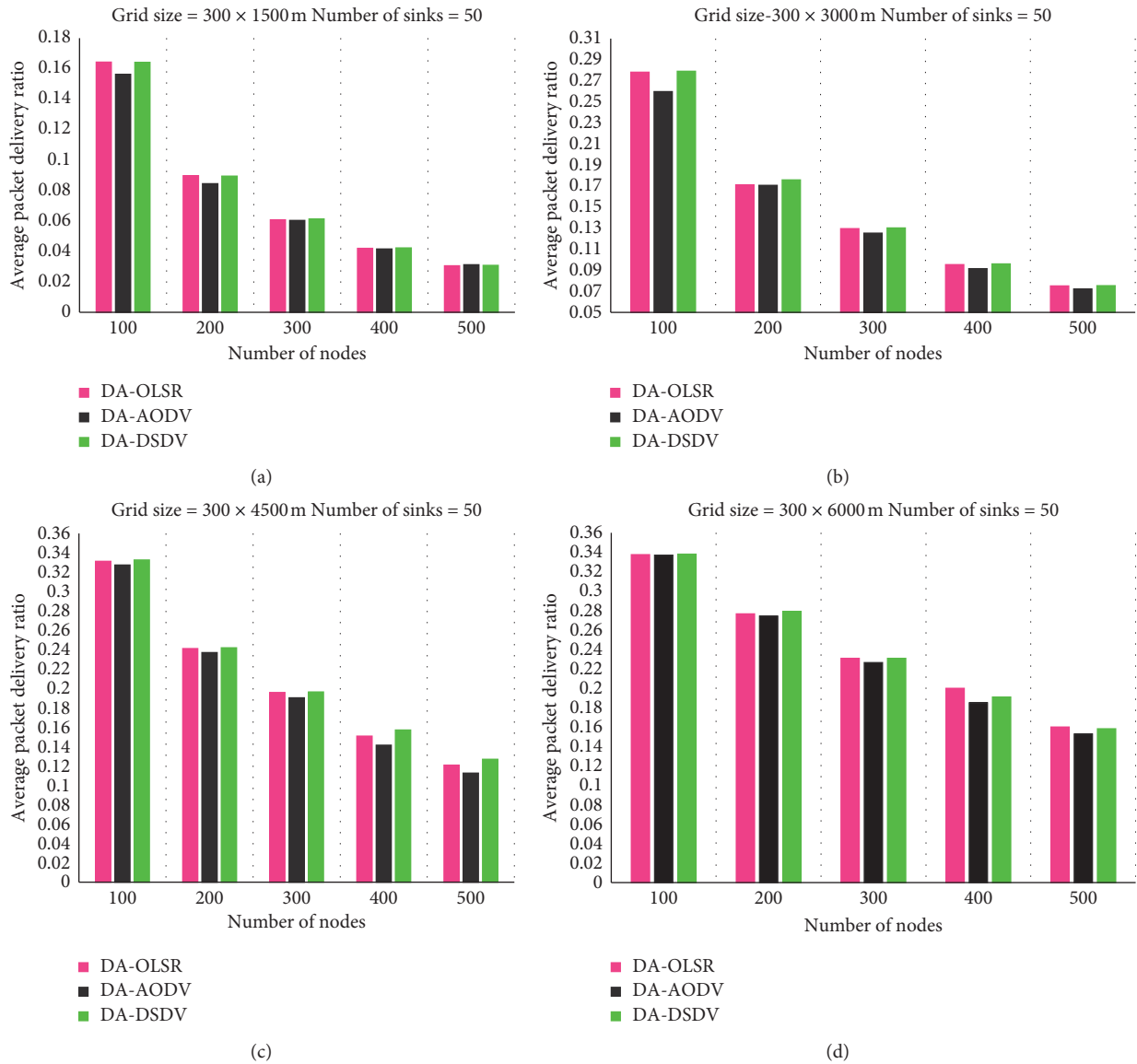


FIGURE 14: Average packet delivery with drone assistance and no. of sinks = 50.

MAC/PHY overhead is depicted in AODV, and the rest of the protocols lie between both. However, if we compare the performance of traditional VANET routing protocols to that of the drone-assisted ones, it will be clear that drone-assisted protocols show less MAC/PHY overhead. When we have a smaller grid, the MAC/PHY overhead for all the six protocols has greater values, but as we ascend towards a bigger grid, this MAC/PHY overhead decreases. The results presented in Figures 15(a)–15(d) are for the same number of sink nodes, i.e., 25, and vehicular nodes for the presented four graphs are from 100 to 500. The grid size is initially  $300 \times 1500$  m and reaches up to  $300 \times 6000$  m with a constant interval of 1500 m along the  $y$ -axis.

6.2. *MAC/PHY Overhead with 50 Sink Nodes.* Figure 16 clearly shows that, for all grid sizes, the OLSR and DA-OLSR have the least MAC/PHY overhead except for

$300 \times 6000$  m where DA-OLSR has less MAC/PHY overhead even compared to OLSR. This means that at a bigger grid size the performance of drone-assisted OLSR is far better than that of the rest of the protocols. On the other hand, the highest MAC/PHY overhead is depicted in AODV. The rest of the protocols lie between DA-OLSR and AODV. When we compare the performance of traditional VANET routing protocols to that of the drone-assisted ones, it becomes clear that drone-assisted protocols show less MAC/PHY overhead for most of the cases. When we have a smaller grid, the MAC/PHY overhead for all the six protocols has greater values, but as we ascend towards a bigger grid, this MAC/PHY overhead decreases. It can be concluded from Figure 16 that, for a bigger grid size like  $300 \times 6000$  m, the drone-assisted protocols outperform the traditional ones. The results presented in Figures 16(a)–16(d) are for the same number of sink nodes, i.e., 50, and vehicular nodes for the presented four graphs are from 100 to 500. The grid size is



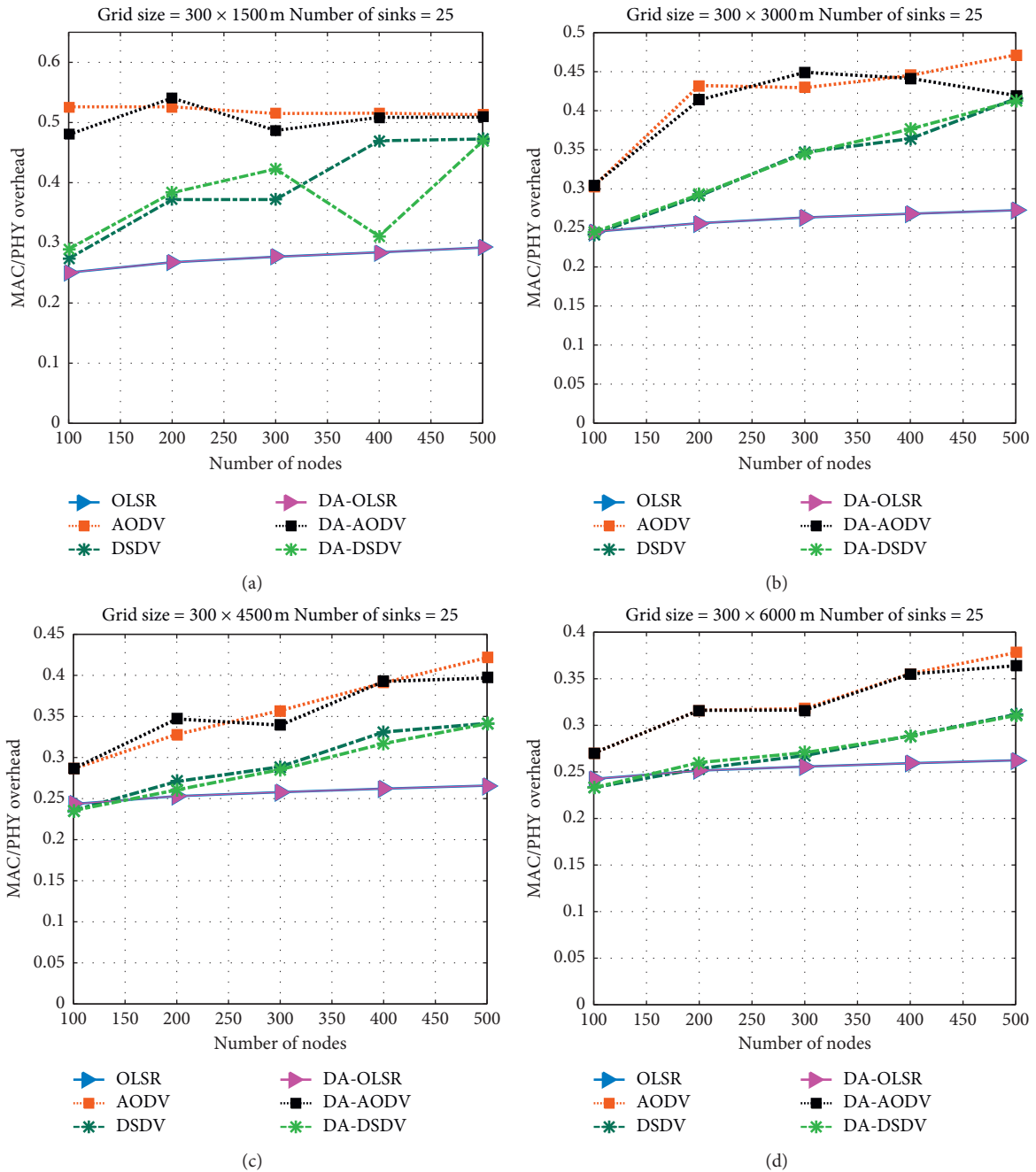


FIGURE 15: Comparison of MAC/PHY overhead of scenarios a and b with 25 sinks.

initially  $300 \times 1500$  m and reaches up to  $300 \times 6000$  m with a constant interval of 1500 m along the  $y$ -axis. One more thing to be noted is that the greater the number of sink nodes is, the higher the performance of protocols will be. As can be seen from Figures 15 and 16, all the protocols have better performance when we have a number of sinks = 50, especially the drone-assisted protocols.

**6.3. Average Throughput with 25 Sink Nodes.** Figure 17 shows that, for all the grid sizes, the OLSR and DA-OLSR have the least average throughput, whereas the highest average

throughput is depicted in DA-AODV when we have the least number of vehicular nodes, and the rest of the protocols lie between both. If we compare the performance of traditional VANET routing protocols to that of the drone-assisted ones, it will be clear that drone-assisted protocols show less throughput when we have a greater number of nodes. When we have a smaller grid, the MAC/PHY overhead for all the six protocols has greater values, but as we ascend towards a bigger grid, this average throughput decreases. This is because of the dissemination of vehicular nodes at a great distance due to an increase in grid size. The vehicular nodes are unable to communicate with each other, hence resulting

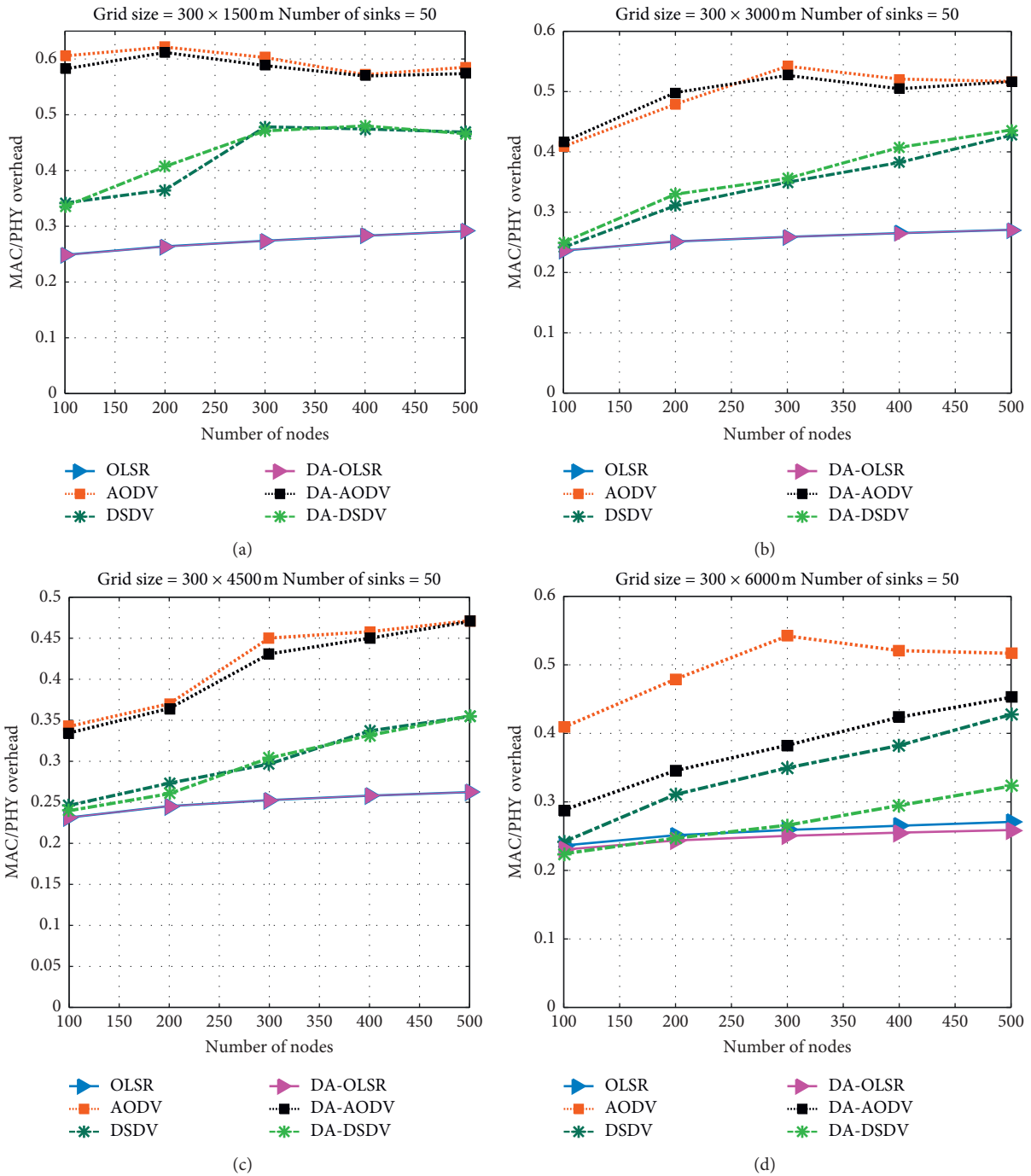


FIGURE 16: Comparison of MAC/PHY overhead of scenarios a and b with 50 sinks.

in less throughput. The results presented in Figures 17(a)–17(d) are for the same number of sink nodes, i.e., 25, and vehicular nodes for the presented four graphs are from 100 to 500. The grid size is initially  $300 \times 1500$  m and reaches up to  $300 \times 6000$  m with a constant interval of 1500 m along the y-axis.

6.4. Average Throughput with 50 Sink Nodes. Figure 18 clearly shows that, for all grid sizes, the OLSR and DA-OLSR have the least average throughput, whereas the highest average throughput is shown by DSDV in smaller grid sizes, but when we have larger grid sizes, the performance of DA-AODV is better for a greater number of nodes. The rest of

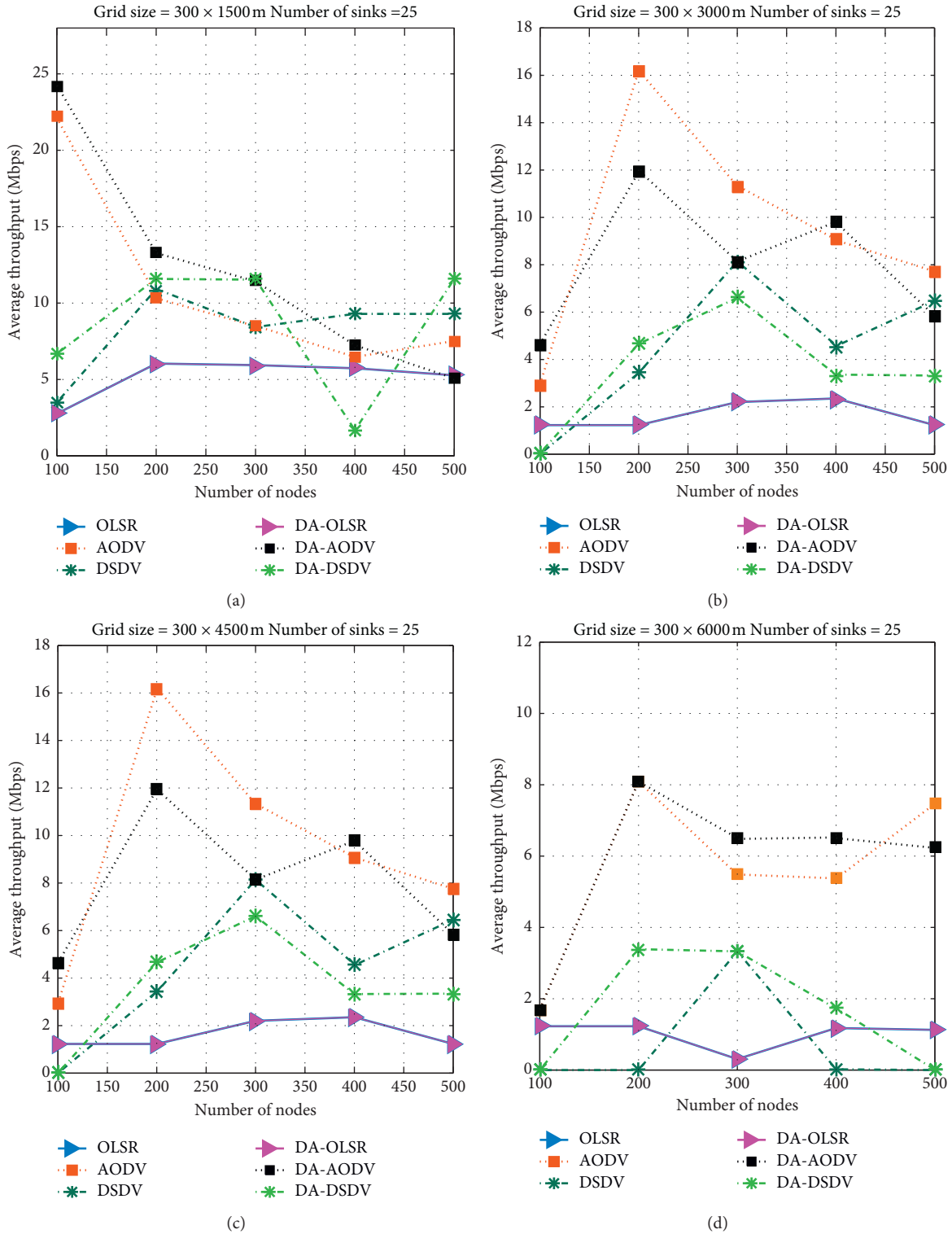


FIGURE 17: Comparison of average throughput of scenarios a and b with 25 sinks.

the protocols lie between DSDV and DA-AODV. When we compare the performance of traditional VANET routing protocols to that of the drone-assisted ones, it becomes clear that drone-assisted protocols show less average throughput for smaller grids, but these protocols have a comparatively enhanced performance for larger grids. The results presented

in Figures 18(a)–18(d) are for the same number of sink nodes, i.e., 50, and vehicular nodes for the presented four graphs are from 100 to 500. The grid size is initially 300 × 1500 m and reaches up to 300 × 6000 m with a constant interval of 1500 m along the y-axis. One more thing to be noted is that the greater the number of sink nodes is, the

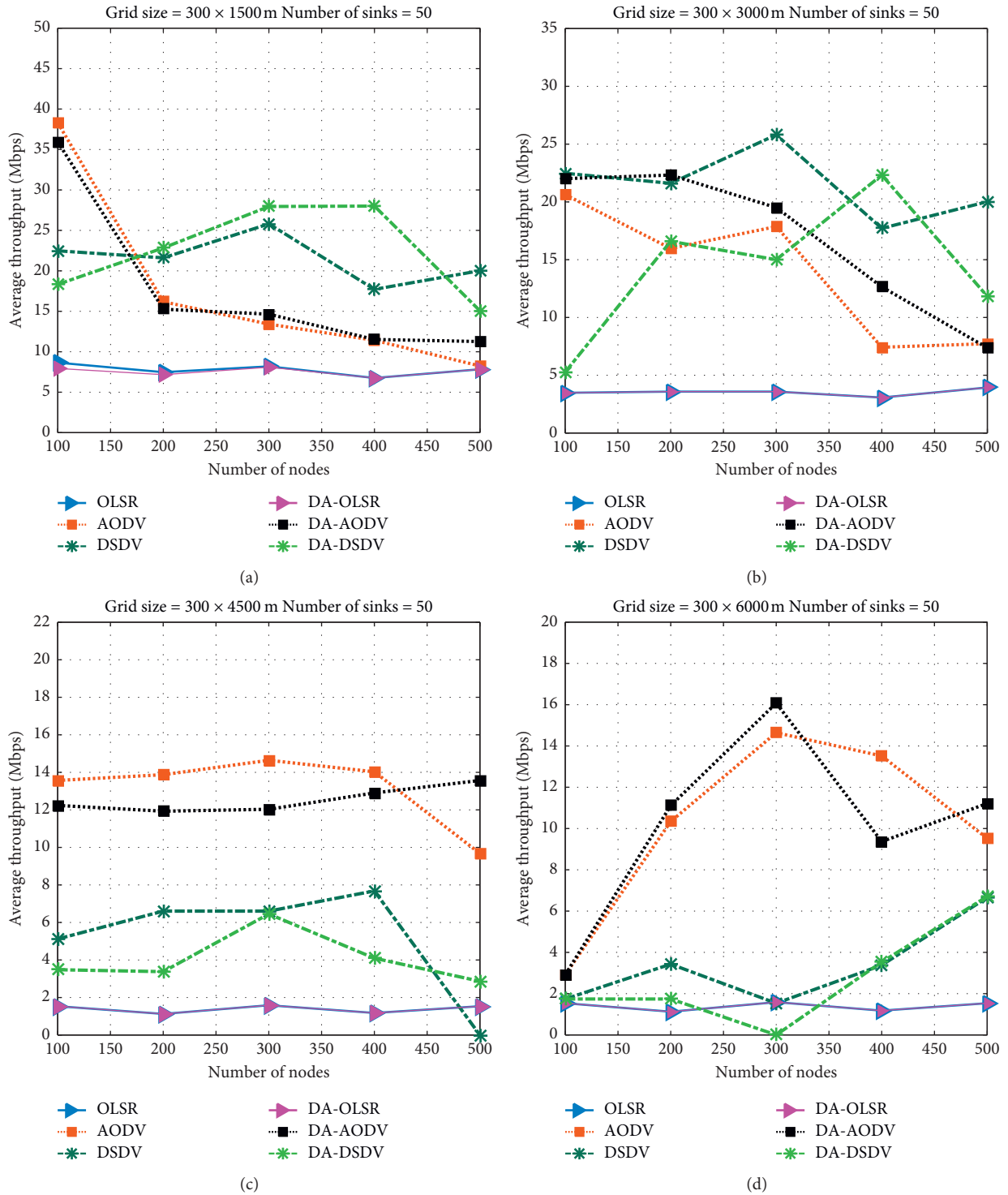


FIGURE 18: Comparison of average throughput of scenarios a and b with 50 sinks.

higher the performance of protocols will be. As can be seen from Figures 17 and 18, all the protocols have better performance when we have the number of sinks = 50, especially the drone-assisted protocols.

6.5. Average Packet Delivery Ratio with 25 Sink Nodes. Figures 19(a)–19(d) present the comparative analysis of average packet delivery ratio in traditional VANET protocol and our proposed strategy at grid size of 300 × 1500 m to

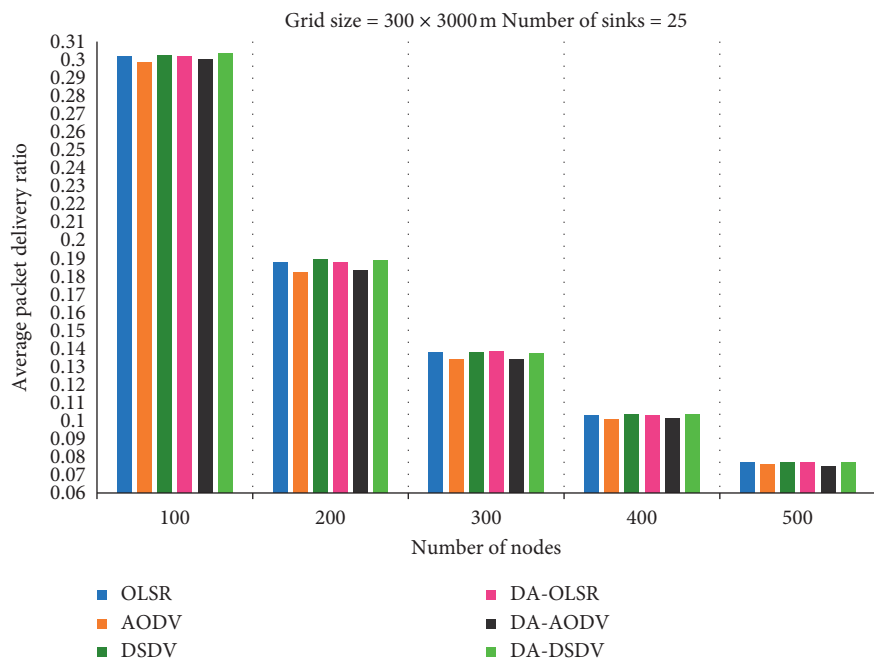
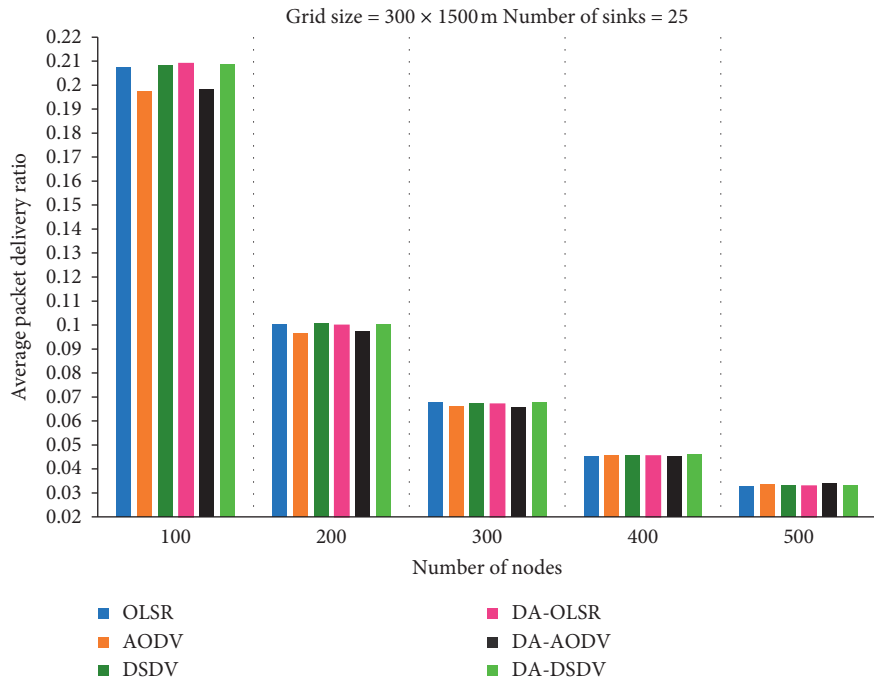
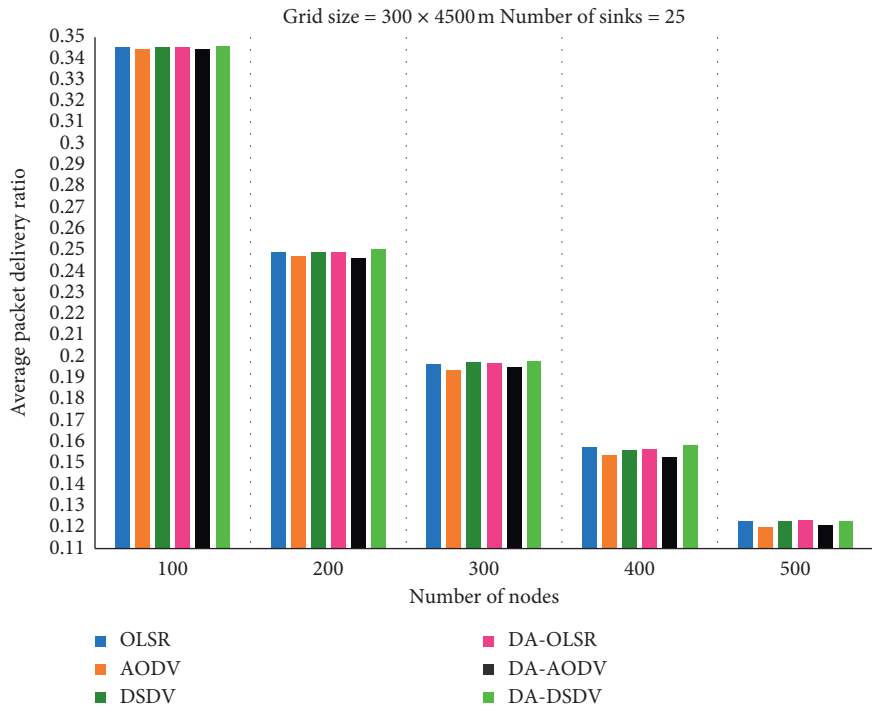
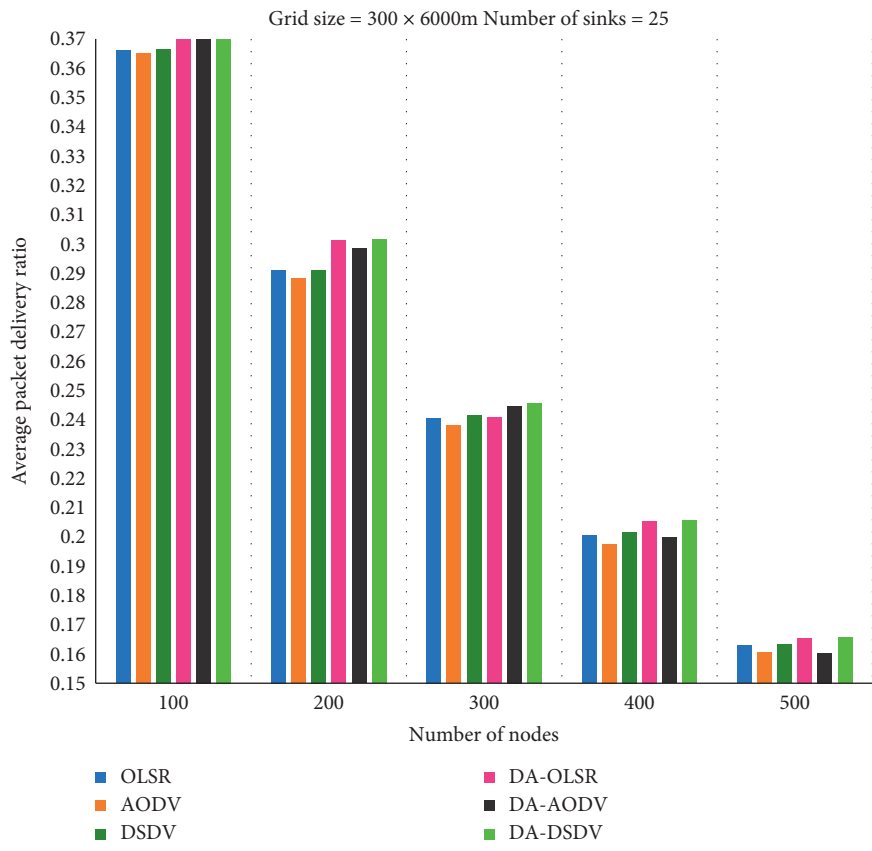


FIGURE 19: Continued.





(c)



(d)

FIGURE 19: Comparison of Average PDR of scenarios a and b with 25 sinks.

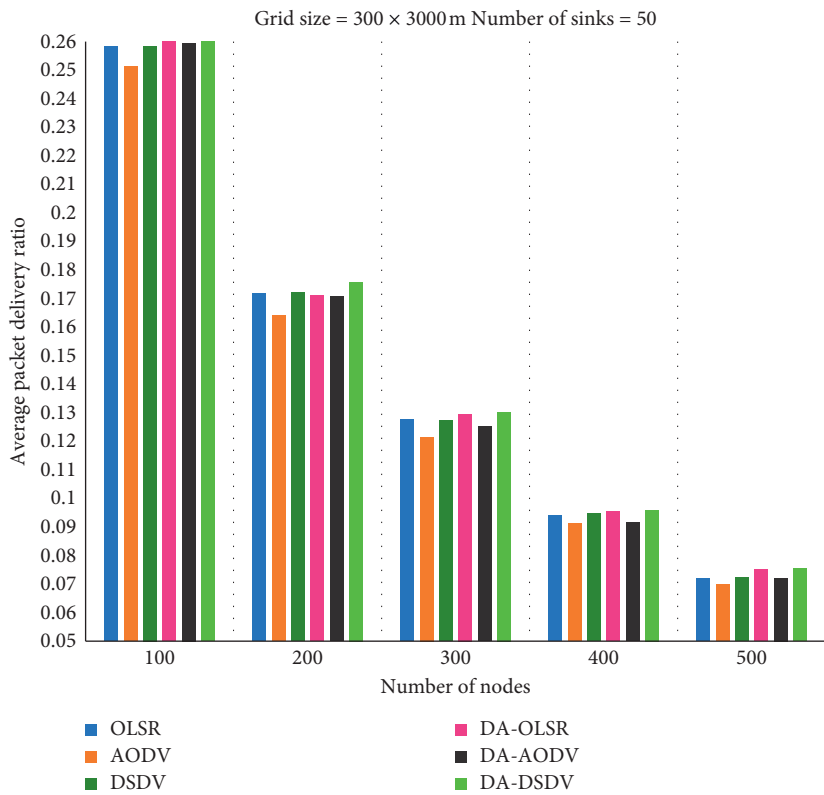
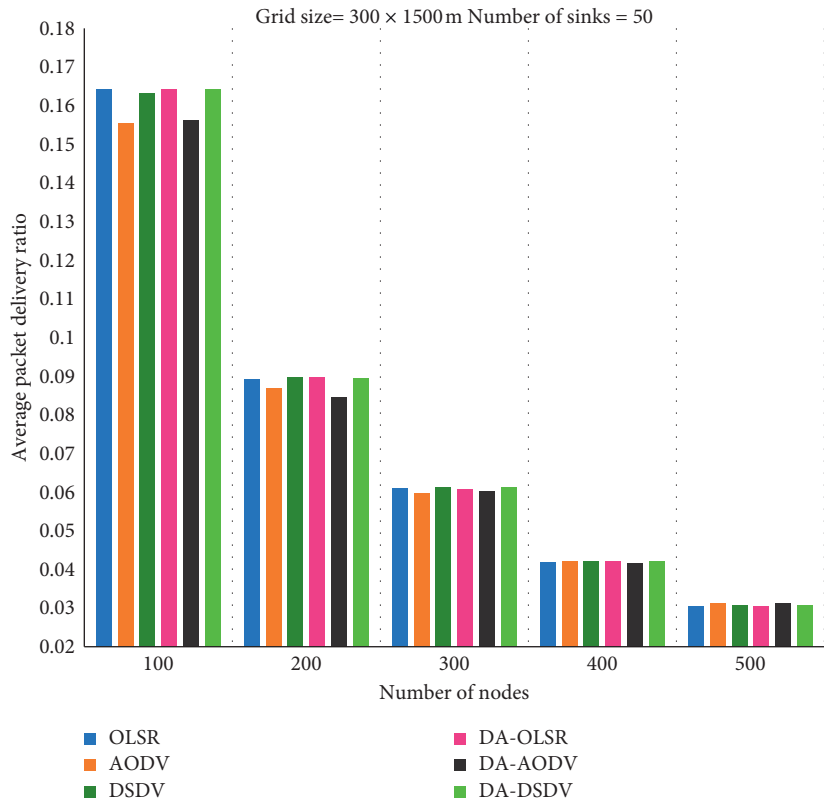


FIGURE 20: Continued.

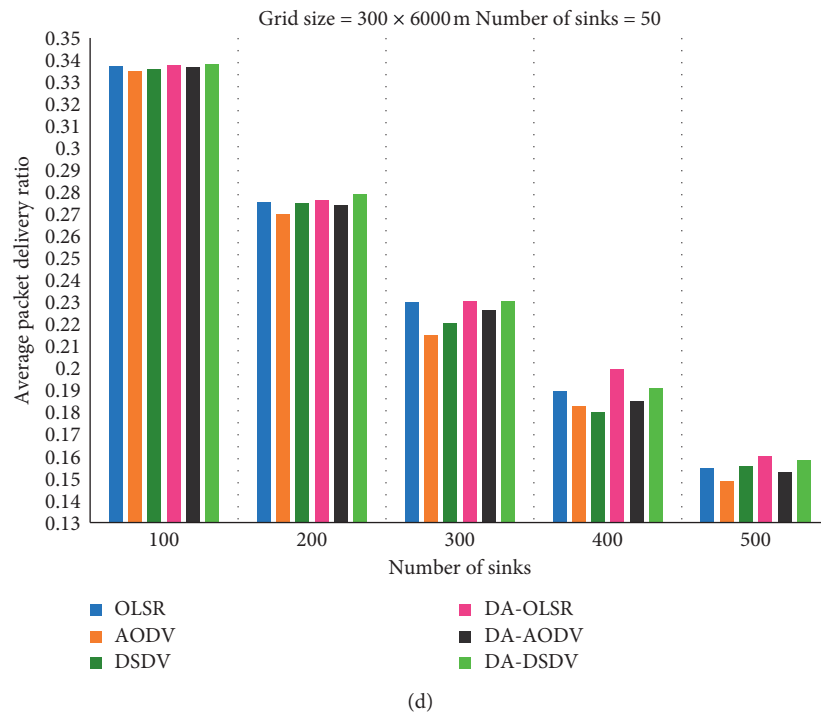
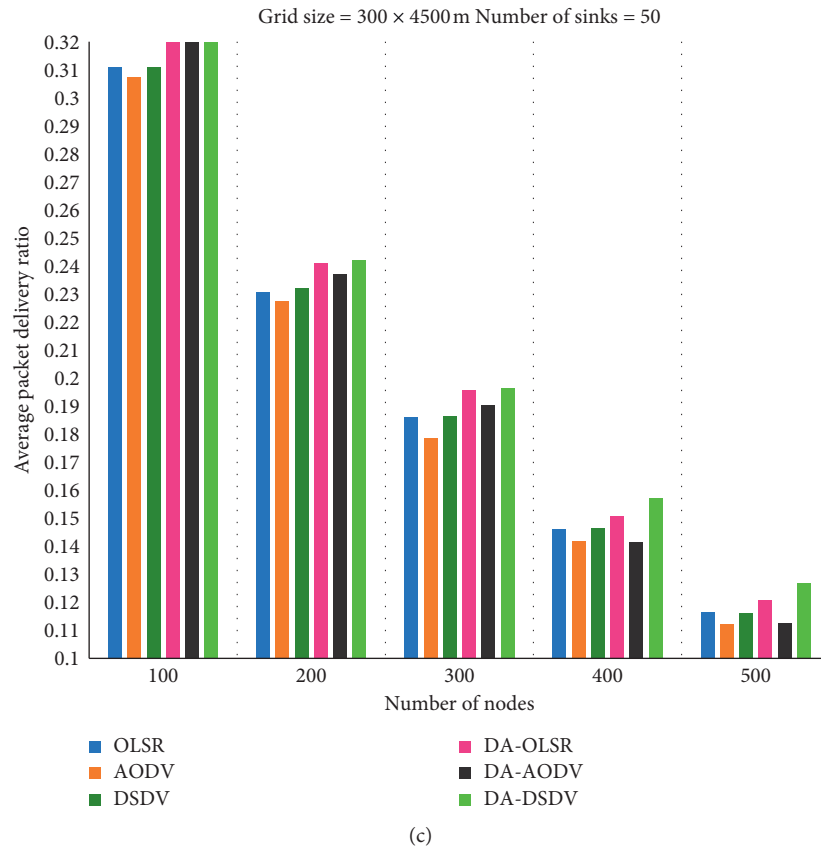


FIGURE 20: Comparison of average PDR of scenarios a and b with 50 sinks.

300 × 6000 m with 25 sink nodes in each case. In smaller grid size, the performance of the traditional routing protocol and that of our drone-assisted protocol are very close to each other, but as we move further towards greater grid size, the

performance of our proposed drone-assisted protocols begins to enhance. As can be seen from Figures 19(a)–19(d), the performance of DA-OLSR and DA-DSDV is better than that of traditional OLSR and DSDV. Though AODV and

DA-AODV could not perform significantly, the overall performance of our proposed strategy is better at a bigger grid size.

**6.6. Average Packet Delivery Ratio with 50 Sink Nodes.** Figures 20(a)–20(d) present the comparative analysis of average packet delivery ratio in traditional VANET protocol and our proposed strategy at grid size of  $300 \times 1500$  m to  $300 \times 6000$  m with 50 sink nodes in each case. In smaller grid size, the performance of the traditional routing protocol and that of our drone-assisted protocol are very close to each other, but as we move further towards greater grid size, the performance of our proposed drone-assisted protocols begins to enhance. As can be seen from Figures 20(a)–20(d), the performance of DA-OLSR, DA-AODV, and DA-DSDV is better than that of traditional OLSR, AODV, and DSDV. Therefore, we can conclude that the performance of our desired strategy is even better with 50 sinks as compared to traditional VANET.

## 7. Conclusion and Future Work

IoV is the new form of VANET and is the alliance of Internet and IoT. Internet of Vehicles is emerging as an important class of networks in the modern era, because of the immense traffic on the road, congested vehicular environment, and increased chance of vehicular collision. Many strategies have been proposed. The main concern of the researchers is improving the overall efficiency of IoT. The efficiency parameters may be greater average throughput, enhanced packet delivery ratio, less MAC/PHY overhead focus, less end-to-end delay, and minimum packet drop ratio. Our focus in this research is on providing such an efficient routing protocol that can help us in providing greater average throughput, enhanced packet delivery ratio, and less MAC/PHY overhead. For this purpose, we have made use of aerial nodes. We did so by elevating the sink nodes to a height greater than that we have in traditional routing protocols.

Extensive simulations have been carried out for traditional VANET routing protocols and drone-assisted routing protocol. The results have been generated and presented in graphical form. The output results have been analyzed one by one. Later, these results have been compared for both the traditional VANET and the one deployed using aerial nodes. This comparison helped us to understand that the assistance of aerial nodes helped us to enhance network efficiency. We have changed the grid size from  $300 \times 1500$  m to  $300 \times 6000$  m with an interval of 1500 m each time. We have also experimented with different numbers of sink nodes, i.e., 25 and 50. From all the experimentation and results gathered, we conclude that our proposed strategy performs well in terms of average throughput and average packet drop ratio when we have a bigger grid. Moreover, the number of sinks also affects these parameters; that is, the greater the number of aerial nodes is, the greater the performance of these parameters will be.

In the case of MAC/PHY overhead, although it increases with the increase in the number of vehicular nodes, its values

are less with a greater number of aerial nodes than those of traditional routing protocols. In the future, we will implement our strategy by varying the transmission ranges. The grid sizes may also be increased along the  $x$ -axis as well as along the  $y$ -axis. Such proposed scheme will not only help in having better network experience in traffic, but also enhance the medicine and healthcare, agriculture, disaster, and emergency scenarios and provide environmental and surrounding information and a better solution for communication over a congested road. The topological constraint changes made produce novelty in our suggested scheme. We intend to find the average packet drop ratio and the end-to-end delay in the future and to analyze the performance of our proposed strategy.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] M. Houmer and M. L. Hasnaoui, "A qualitative assessment of VANET routing protocols under different mobility models," *Journal of Computer Science*, vol. 15, no. 1, pp. 161–170, 2019.
- [2] J. Wang, X. Xiao, and P. Lu, "A survey of vehicular ad hoc network routing protocols," *Journal of Electrical and Electronic Engineering*, vol. 7, no. 2, pp. 46–50, 2019.
- [3] J. P. West and J. S. Bowman, "The domestic use of drones: an ethical analysis of surveillance issues," *Public Administration Review*, vol. 76, no. 4, pp. 649–659, 2016.
- [4] S. K. Lakshmanaprabu, K. Shankar, S. Sheeba Rani et al., "An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: towards smart cities," *Journal of Cleaner Production*, vol. 217, pp. 584–593, 2019.
- [5] N. Melauouene and R. Romadi, "An enhanced routing algorithm using ant colony optimization and VANET infrastructure," *MATEC Web of Conferences*, vol. 259, Article ID 02009, 2019.
- [6] S. R. Yahiabadi, B. Barekatin, and K. Raahemifar, "TIHO: an enhanced hybrid routing protocol in vehicular ad-hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 192, 2019.
- [7] N. K. Chaubey, "Security analysis of vehicular ad hoc networks (VANETs): a comprehensive study," *International Journal of Security and Its Applications*, vol. 10, no. 5, pp. 261–274, 2016.
- [8] A. Ahamed and H. Vakilzadian, "Issues and challenges in VANET routing protocols," in *Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 0723–0728, IEEE, Rochester, MI, USA, May 2018.
- [9] A. Awang, K. Husain, N. Kamel, and S. Aissa, "Routing in vehicular ad-hoc networks: a survey on single- and cross-layer design techniques, and perspectives," *IEEE Access*, vol. 5, pp. 9497–9517, 2017.
- [10] N. Chowdhary and P. D. Kaur, "Addressing the characteristics of mobility models in IoV for smart city," in *Proceedings of the 2016 International Conference on Computing, Communication*

- and Automation (ICCCA), pp. 1298–1303, IEEE, Greater Noida, India, April 2016.
- [11] R. C. Manurung, D. Perdana, and R. Munadi, “Performance evaluation Gauss-Markov mobility model in vehicular ad-hoc network with spearman correlation coefficient,” in *Proceedings of the 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 351–356, IEEE, Lombok, Indonesia, July 2016.
  - [12] P. E. Ramadhani, M. D. Setiawan, M. A. Yutama, D. Perdana, and R. F. Sari, “Performance evaluation of hybrid wireless mesh protocol (HWMP) on VANET using VanetMobiSim,” in *Proceedings of the 2016 International Conference on Computational Intelligence and Cybernetics*, pp. 41–46, IEEE, Makassar, Indonesia, November 2016.
  - [13] B. Ramakrishnan, M. Selvi, and R. B. Nishanth, “Efficiency measure of routing protocols in vehicular ad hoc network using freeway mobility model,” *Wireless Networks*, vol. 23, no. 2, pp. 323–333, 2017.
  - [14] M. Arif, G. Wang, M. Zakirul Alam Bhuiyan, T. Wang, and J. Chen, “A survey on security attacks in VANETs: communication, applications and challenges,” *Vehicular Communications*, vol. 19, Article ID 100179, 2019.
  - [15] C. Jayapal and S. S. Roy, “Road traffic congestion management using VANET,” in *Proceedings of the 2016 International Conference on Advances in Human Machine Interaction (HMI)*, pp. 1–7, IEEE, Bangalore, India, March 2016.
  - [16] S. Singh, S. Negi, S. K. Verma, and N. Panwar, “Comparative study of existing data scheduling approaches and role of cloud in VANET environment,” *Procedia Computer Science*, vol. 125, pp. 925–934, 2018.
  - [17] C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, “ASGR: an artificial spider-web-based geographic routing in heterogeneous vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1604–1620, 2018.
  - [18] R. A. Nazib and S. Moh, “Routing protocols for unmanned aerial Vehicle-Aided vehicular Ad Hoc networks: a survey,” *IEEE Access*, vol. 8, pp. 77535–77560, 2020.
  - [19] O. Sami Oubbati, N. Chaib, A. Lakas, S. Bitam, and P. Lorenz, “U2RV: UAV-assisted reactive routing protocol for VANETs,” *International Journal of Communication Systems*, vol. 33, no. 10, Article ID e4104, 2020.
  - [20] N. Lin, L. Fu, L. Zhao, G. Min, A. Al-Dubai, and H. Gacanin, “A novel multimodal collaborative drone-assisted VANET networking model,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4919–4933, 2020.
  - [21] A. Kumar, K. Rajalakshmi, S. Jain, A. Nayyar, and M. Abouhawwash, “A novel heuristic simulation-optimization method for critical infrastructure in smart transportation systems,” *International Journal of Communication Systems*, vol. 33, no. 11, Article ID e4397, 2020.
  - [22] T. Lu, S. Chang, and W. Li, “Fog computing enabling geographic routing for urban area vehicular network,” *Peer-to-Peer Networking and Applications*, vol. 11, no. 4, pp. 749–755, 2018.
  - [23] M. Bhatt, S. Sharma, A. K. Luhach, and A. Prakash, “Nature inspired route optimization in vehicular adhoc network,” in *Proceedings of the 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 447–451, IEEE, Noida, India, September 2016.
  - [24] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
  - [25] M. Farhanchi, R. Hassanzadeh, I. Mahdavi, and N. Mahdavi-Amiri, “A modified ant colony system for finding the expected shortest path in networks with variable arc lengths and probabilistic nodes,” *Applied Soft Computing*, vol. 21, pp. 491–500, 2014.
  - [26] K. Prakash, P. C. Philip, R. Paulus, and A. Kumar, “A packet fluctuation-based OLSR and efficient parameters-based OLSR routing protocols for urban vehicular ad hoc networks,” in *Recent Trends in Communication and Intelligent Systems*, pp. 79–87, Springer, Berlin, Germany, 2020.
  - [27] X. Bao, H. Li, G. Zhao, L. Chang, J. Zhou, and Y. Li, “Efficient clustering V2V routing based on PSO in VANETs,” *Measurement*, vol. 152, Article ID 107306, 2020.
  - [28] M. T. Abbas, A. Muhammad, and W.-C. Song, “Road-aware estimation model for path duration in Internet of vehicles (IoV),” *Wireless Personal Communications*, vol. 109, no. 2, pp. 715–738, 2019.
  - [29] V. Jindal and P. Bedi, “An improved hybrid ant particle optimization (IHAPO) algorithm for reducing travel time in VANETs,” *Applied Soft Computing*, vol. 64, pp. 526–535, 2018.
  - [30] D. Zhang, T. Zhang, and X. Liu, “Novel self-adaptive routing service algorithm for application in VANET,” *Applied Intelligence*, vol. 49, no. 5, pp. 1866–1879, 2019.
  - [31] D. Tian, K. Zheng, J. Zhou, Z. Sheng, Q. Ni, and Y. Wang, “Unicast routing protocol based on attractor selection model for vehicular ad-hoc networks,” in *Proceedings of the International Conference on Internet of Vehicles*, pp. 138–148, Springer, Nadi, Fiji, December 2016.
  - [32] M. Elhoseny and K. Shankar, “Energy efficient optimal routing for communication in VANETs via clustering model,” in *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*, pp. 1–14, Springer, Berlin, Germany, 2020.
  - [33] A. Nayyar, “Flying adhoc network (FANETs): simulation based performance comparison of routing protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP,” in *Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–9, IEEE, Durban, South Africa, August 2018.
  - [34] A. V. Leonov, “Application of bee colony algorithm for FANET routing,” in *Proceedings of the 2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, pp. 124–132, IEEE, Erlagol, Altai, June 2016.
  - [35] S. Majumdar, P. R. Prasad, S. S. Kumar, and K. S. Kumar, “An efficient routing algorithm based on ant colony optimisation for VANETs,” in *Proceedings of the 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 436–440, IEEE, Bangalore, India, May 2016.
  - [36] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, “A survey on position-based routing protocols for flying ad hoc networks (FANETs),” *Vehicular Communications*, vol. 10, pp. 29–56, 2017.
  - [37] V. Saritha, P. V. Krishna, S. Misra, and M. S. Obaidat, “Learning automata based optimized multipath routing using leapfrog algorithm for VANETs,” in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, Paris, France, January 2017.
  - [38] J. F. Bravo-Torres, M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, M. Ramos-Cabrer, and A. Gil-Solla, “Optimizing reactive routing over virtual nodes in VANETs,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2274–2294, 2015.



- [39] M. Dixit, R. Kumar, and A. K. Sagar, "VANET: architectures, research issues, routing protocols, and its applications," in *Proceedings of the 2016 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 555–561, IEEE, Greater Noida, India, April 2016.
- [40] V. A. Maistrenko, L. V. Alexey, and V. A. Danil, "Experimental estimate of using the ant colony optimization algorithm to solve the routing problem in FANET," in *Proceedings of the 2016 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–10, IEEE, Moscow, Russia, May 2016.
- [41] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, 2016.
- [42] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.
- [43] A. Yasser, M. Zorkany, and N. Abdel Kader, "VANET routing protocol for V2V implementation: a suitable solution for developing countries," *Cogent Engineering*, vol. 4, no. 1, Article ID 1362802, 2017.

## Research Article

# Nonpreferential Attachment Leads to Scale-Free or Not

Chuankui Yan , Nan Meng , and Yu Yang 

College of Mathematics and Physics, Wenzhou University, Wenzhou, China

Correspondence should be addressed to Chuankui Yan; [yanchuankui@163.com](mailto:yanchuankui@163.com)

Received 6 January 2021; Revised 15 April 2021; Accepted 19 April 2021; Published 6 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Chuankui Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many studies have considered the preferential attachment mechanism to cause scale-free networks. On the contrary, a network evolution model based on nonpreferential attachment is proposed to explain some non-scale-free network, and the existence of a stable degree distribution of the model is theoretically proven. Three methods are suggested to estimate the distribution. The model's significance shows that preferential attachment is not the only mechanism of tail power-law distribution, which gives a reasonable explanation to non-scale-free phenomenon. Our results provide a new train of thought for understanding the degree distribution of network.

## 1. Introduction

Network research is an effective way to study complex systems. Since the emergence of two landmark network models, such as the WS model [1] and the BA model [2], network models have been widely studied. Such studies are performed in many fields, including social interactions between individuals, protein or gene interactions in living organisms, synaptic connections, communication between computer networks, and various transportation systems. Generally, in network research, it is said that most or all real-world networks are scale-free [3–8]. That is, the node degree  $k$  of the network follows power-law distribution. Besides, in network science, studies on the application of a scale-free network have been extensively performed [2, 9, 10]. Various studies have evaluated how the existence of a scale-free structure affects the running of networks [7–12]. Scale-free networks have been used as the basis for numerical simulation and experiments. Studies have also investigated the generation mechanisms of scale-free networks [2, 5, 13–15]. Network models can describe many systems, and it is reported that preferential attachment can lead to scale-free [16]. In addition to the BA model, some representative development models have been used to describe this mechanism, for example, the earlier price model with an adjustable power rate [17, 18], the HK model with an

adjustable clustering coefficient [19], the fitness model that is based on individual differences [20], and the local-world evolving network model that is based on local world evolution [21] among many others. Many real networks have some standard features, including power-law degree distribution, small average shortest path length, and high clustering. Some networks following the WS model (not all) have a small average shortest path length and high clustering with no power-law degree distribution, which contrasts with the BA network model [18–21].

The universality of scale-free networks has not been established. Some studies have reported that scale-free is universal [6, 10, 11, 22, 23]. ER random graph can also be “scale-free,” and classical random graphs with unbounded expected degrees are locally scale-free. The others hold the opposite view, based on data and statistical theory [4, 5, 24–30]. Broido and Clauset tested 1000 real network datasets [31], and they concluded that scale-free networks are very rare. Besides, they established that only about 15% of the network showed a strong or strongest scale-free structure features.

The generation mechanism is commonly discussed in scale-free network studies, particularly preferential attachment [2, 3, 17, 18]. One of the most famous connection mechanisms is that the probability of obtaining a connection is proportional to current node degrees. It is called

preferential attachment, which means that the new node has a greater preference to connect to an old node with a bigger degree. Many studies are based on the connection mechanisms.

Many previous network generation models were based on scale-free assumptions. However, empirical data show that not all network distributions strictly conform to power-law distribution [31]. Occasionally, when lognormal distribution is used to fit the real network data, the result is the same as the power-law distribution, or even better [32–34]. Preference connections explain the mechanism of scale-free generation. Therefore, the question is as follows: are there other mechanisms to explain these distributions?

## 2. Network Evolution Model

The initial number of nodes in the network is  $N_0$ , the average degree is  $\langle k \rangle$ , and the final network size is  $N$ . The following were the network evolution rules:

- (i) Initial moment: the network contained  $N_0$  nodes, random connections, and  $(N-N_0)$  isolated potential nodes
- (ii) Nonpreferential disconnection (NPD): for node  $v_i$  in the network, a neighbor  $v_j$  is randomly selected to disconnect  $(v_i, v_j)$
- (iii) Nonpreferential attachment (NPA): we randomly selected a node  $v_k$  from the network and connected  $(v_k, v_j)$
- (iv) NPD and NPA for each point in network
- (v) Steps (ii)–(iv) were repeated

The evolutionary steps take place one at a time, with only a rewiring involving two edges being involved. It seems to be counterintuitive to the objective of applying a fair and completely unbiased connectivity, as it restricts multiple such rewirings happening concurrently. However, when the network size is very large, the probability of simultaneous occurrence is very small, so the results of the sequential evolution of nodes and simultaneous evolution are very close. The sequential evolution brings great convenience to our computer simulation. In the following theoretical analysis, the evolution of some nodes has no sequence at all and is completely unbiased.

Preferential attachment exists in human relations. Human social contacts can be biased to a certain extent. However, some network nodes cannot be biased due to a lack of subjective consciousness, such as neural connections, metabolic networks, and protein regulatory networks. Compared to the BA model's preferential attachment, disappearance and generation of edges between nodes in this model are fair and completely unbiased to every node. A mechanism of edge fading is presented in this model, while a mechanism of new edge generation is also proposed. This design is in tandem with the real network, where many node relationships fade and emerge over time. Node relationships of the network are not unchanged after generation. For example, a friend relationship of a social network will break down or make new friends [35–37], and synaptic plasticity of

neural networks leads to the loss of synaptic connections or are newly formed and strengthened [38, 39]. This reconnection mechanism is a connection transfer mechanism, which has a particular practical significance in some networks, such as the trade and debt lending networks [40–43], where the trade volume and debt relationship are transferred between nodes. Under this mechanism, some old nodes are disconnected and leave the network, while some new nodes join the network and they are connected without preference. This is in tandem with the fact that aging nodes exit the network, whereas new nodes join the network.

## 3. Degree Distribution Analysis

**3.1. Degree Distribution and Network Evolution.** For time  $t$ , the probability of the node with degree  $k$  is  $P_t(k)$ . Consequently, the distribution is  $P_t = (P_t(1), P_t(2), \dots, P_t(N-1))^T$ . “ $T$ ” stands for transpose.

The probability that the node degree changes from  $i$  to  $j$  is  $p_{ij}$ . Obviously, in our model, it is impossible for degree  $i$  to become  $j$  when  $j < i - 1$ . Considering node  $v$  with degree  $i > 1$  and  $j \geq i - 1$ , in addition to the node that was randomly selected in NPD, the other neighbor nodes and the node itself in the process did not change the degree. The degree is  $(i - 1)$  after NPD. To change its degree to  $j$ , it has to be selected  $(j - i + 1)$  times by the rest of the  $(N - i)$  nodes in NPA, with no preference to the probability, which is  $1/(N - 1)$ .

Thus, one could get

$$p_{ij} = \begin{cases} C_{N-i}^{j-i+1} \left(\frac{1}{N-1}\right)^{j-i+1} \left(1 - \frac{1}{N-1}\right)^{N-j-1}, & j \geq i - 1, \\ 0, & j < i - 1, \end{cases} \quad (1)$$

where  $C_N^i$  is the number of combinations of  $N$  choose  $i$ .

Therefore, the probability function of node degree at time  $t$  is

$$P_t(k) = \sum_{i=1}^{N-1} p_{ik} P_{t-1}(i), \quad (2)$$

Notably  $A = (p_{ij})_{(N-1) \times (N-1)}^T$ , then  $P_t = AP_{t-1} = A^t P_0$ .

This iterative process dynamically describes node degree distribution of the development model at any given time.

**3.2. Existence of Stability Distribution.** We prove that stability distribution exists as the time approaches infinity. The lemmas and definitions to be used later are introduced first.

**Definition 1.** For a square matrix  $M = (m_{ij})_{m \times m}$ ,  $G_j(M) = \{z \mid |z - m_{jj}| \leq R_j\}$ ,  $j = 1, 2, \dots, n$  is called column Gerschgorin circle, where  $R_j = \sum_{i=1, i \neq j}^n |m_{ij}|$ .

**Definition 2.** For a square matrix  $M = (m_{ij})_{m \times m}$ , it is called a primitive matrix if there is a positive integer  $n$ ,  $M^n > 0$ .

**Lemma 1** (Gerschgorin disk theorem [44]). *If  $M$  is a square matrix, any eigenvalue  $\lambda$  of  $M$  belongs to at least one column Gerschgorin circle  $G_j(M)$ ,  $\lambda$  belongs to union of them,  $\lambda \in G = \cup_{j=1}^n G_j$ .*

**Lemma 2** (Perron–Frobenius theorem [44]). *If  $M$  is a primitive matrix, then the spectrum radius  $\rho(M)$  is a single root, and  $\lim_{k \rightarrow +\infty} (\rho(M)^{-1}M)^k = vw^T$ , where  $w$  and  $v$  are left and right Perron vectors.*

*Then, we prove that Lemmas 3 and 4 are true.*

**Lemma 3.** *Matrix  $A = (p_{ij})_{(N-1) \times (N-1)}^T$  is the primitive matrix.*

*Proof.* Let  $P = A^T = (p_{ij})_{(N-1) \times (N-1)}$ ,  $P^n = (p_{ij}^{(n)})_{(N-1) \times (N-1)}$ .  
Because

$$p_{ij} = \begin{cases} C_{N-i}^{j-i+1} \left(\frac{1}{N-1}\right)^{j-i+1} \left(1 - \frac{1}{N-1}\right)^{N-j-1} > 0, & j \geq i-1, \\ 0, & j < i-1, \end{cases} \quad (3)$$

one can get

$$p_{ij}^{(n)} = \sum_{k=1}^n p_{ik}^{(n-1)} p_{kj}^{(n-1)} = \begin{cases} > 0, & \text{else,} \\ 0, & j < i-n. \end{cases} \quad (4)$$

We take  $n = N - 2$ , obviously  $j \geq i - n$ .  
Therefore,  $A^n = (P^T)^n = (P^n)^T > 0$ .

We get that  $A$  is the primitive matrix.  $\square$

**Lemma 4.**  $\lambda = 1$  is an eigenvalue of  $A = (p_{ij})_{(N-1) \times (N-1)}^T$ .

*Proof.* Substituting  $\lambda = 1$  into the eigenpolynomial  $f(\lambda) = |A - \lambda E|$ , one gets

$$f(1) = |A - E| = \begin{vmatrix} p_{11} - 1 & p_{11} & 0 & \dots & 0 & 0 \\ p_{12} & p_{12} - 1 & p_{32} & \dots & 0 & 0 \\ p_{13} & p_{13} & p_{33} - 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{1,N-2} & p_{1,N-2} & p_{3,N-2} & \dots & p_{N-2,N-2} - 1 & p_{N-1,N-2} \\ p_{1,N-1} & p_{1,N-1} & p_{3,N-1} & \dots & p_{N-2,N-1} & p_{N-1,N-1} - 1 \end{vmatrix}. \quad (5)$$

We add the first  $N - 2$  rows to the last row.

Then,

$$f(1) = \begin{vmatrix} p_{11} - 1 & p_{11} & 0 & \dots & 0 & 0 \\ p_{12} & p_{12} - 1 & p_{32} & \dots & 0 & 0 \\ p_{13} & p_{13} & p_{33} - 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{1,N-2} & p_{1,N-2} & p_{3,N-2} & \dots & p_{N-2,N-2} - 1 & p_{N-1,N-2} \\ 0 & 0 & 0 & \dots & 0 & 0 \end{vmatrix} = 0. \quad (6)$$

Its shown that  $\lambda = 1$  is an eigenvalue of  $A$ .

Therefore, we prove the existence theorem for stable distributions.  $\square$

**Theorem 1.** *It states that the network evolution model has a stable degree distribution, and the stable distribution is the eigenvector corresponding to the eigenvalue 1.*

*Proof.* For column Gerschgorin circle

$$G_j(M) = \left\{ z \mid |z - m_{jj}| \leq R_j \right\}, \quad j = 1, 2, \dots, n, \quad (7)$$

where  $R_j = \sum_{i=1, i \neq j}^n |p_{ji}| = \sum_{i=1, i \neq j}^n p_{ji} = 1 - p_{jj}$ .

Thus,  $|z - p_{jj}| \leq 1 - p_{jj}$ , namely, any eigenvalue satisfies  $|z| \leq 1$ .

From Lemma 4, we have  $\lambda = 1$  is an eigenvalue of  $A$ .

Therefore, 1 is the greatest eigenvalue and spectrum radius  $\rho(A) = 1$ .

From Lemmas 2 and 3, spectrum radius  $\rho(A)$  is a single root, and  $\lim_{k \rightarrow +\infty} (\rho(A)^{-1}A)^k = vw^T$ , where  $w$  and  $v$  are left and right Perron vectors, namely,  $\lim_{k \rightarrow +\infty} A^k = vw^T$ .

Then,  $\lim_{t \rightarrow \infty} P_t = \lim_{t \rightarrow \infty} A^t P_0 = vw^T P_0$ . Let  $\lim_{t \rightarrow \infty} P_t = P$ .

Taking the limit of the distribution iterating  $P_t = AP_{t-1}$  over time  $t \rightarrow +\infty$ , then  $P = AP$ . That is,  $P$  is the eigenvector corresponding to the eigenvalue 1.

Remark: conclusion of the theorem shows that network evolution has a stable distribution. This distribution is the eigenvector corresponding to the eigenvalue of 1, and this distribution is independent of the initial state of the network. Though this conclusion provides the distribution computing method, it is very difficult to solve the matrix eigenvectors when the network size is large.  $\square$

**3.3. Estimation of Stability Distribution.** As the size of the matrix increases, the  $C_N^i$  becomes unusually large, its computation is difficult, and accurate calculation is impossible. We used an approximate calculation:

$$\begin{aligned} & \lim_{N \rightarrow \infty} C_{N-i}^{j-i+1} \left( \frac{1}{N-1} \right)^{j-i+1} \left( 1 - \frac{1}{N-1} \right)^{N-j-1} \\ &= \lim_{N \rightarrow \infty} \frac{(N-i)!}{(N-j-1)!(j-i+1)!} \left( \frac{1}{N-1} \right)^{j-i+1} \lim_{N \rightarrow \infty} \left( 1 - \frac{1}{N-1} \right)^{N-i} \lim_{N \rightarrow \infty} \left( 1 - \frac{1}{N-1} \right)^{i-j-1} \\ &= \lim_{N \rightarrow \infty} \frac{(N-i)!}{(N-j-1)!} \left( \frac{1}{N-i} \right)^{j-i+1} \frac{1}{(j-i+1)!} \left( \frac{N-i}{N-1} \right)^{j-i+1} e^{-((N-i)/(N-1))} \\ &= \lim_{N \rightarrow \infty} \frac{(N-i)(N-i-1) \dots (N-j)}{(N-i)(N-i) \dots} \lim_{N \rightarrow \infty} \frac{1}{(j-i+1)!} \left( \frac{N-i}{N-1} \right)^{j-i+1} e^{-((N-i)/(N-1))} \\ &= \lim_{N \rightarrow \infty} \frac{1}{(j-i+1)!} \left( \frac{N-i}{N-1} \right)^{j-i+1} e^{-((N-i)/(N-1))}. \end{aligned} \quad (8)$$

It shows that when  $N$  is large,  $p_{ij} = C_{N-i}^{j-i+1} (1/(N-1))^{j-i+1} (1-1/(N-1))^{N-j-1} \approx 1/(j-i+1)! ((N-i)/(N-1))^{j-i+1} e^{-((N-i)/(N-1))}$ . Note that  $p_{ij} \approx \tilde{p}_{ij}$ .

The approximate results  $A \approx \tilde{A} = (\tilde{p}_{ij})_{(N-1) \times (N-1)}^T$ .

It should be noted that because of the approximation, the maximum eigenvalue of  $A$  is not 1, just approximate 1, and the eigenvector cannot be a positive vector. The negative component's absolute value is minimal, which can be treated as 0. Therefore, three estimation methods are proposed.

Method 1: calculate the eigenvectors corresponding to the maximum eigenvalue of  $\tilde{A}$ , ignore the negative components, and normalize it

Method 2: solving equations  $\begin{cases} \tilde{A}P = P \\ \|P\|_1 = 1 \end{cases}$

Method 3: iterative estimation  $P_t = \tilde{A}P_{t-1}$

#### 4. Comparisons between Model and Real Network

A simulation experiment is operated. The value of  $N$  needs to be large enough because too few nodes will affect two results. First, it will influence the statistical accuracy of the degree distribution. Second, the estimation of stable distribution

requires Poisson distribution to approximate binomial distribution, and the error is large due to too few nodes.

If no parameters are specified, the simulation result is the average of 10 networks, the number of nodes in each network  $N = 10000$ , and the average degree  $\langle k \rangle = 10$ .

Statistical analysis of real network data established that not all networks have strict scale-free degree distribution. A study [31] adopted mathematical methods to ascertain whether the real network meets the scale-free threshold. However, findings were greatly different from previous cognition. It was found that scale-free networks are rare. In this study, as illustrated in Figures 1(e)–1(h), we have shown four representative real network degree distributions. These networks are not strictly scale-free distributions, but a tail with approximate scale-free characteristics. Such distribution is not individual, but a large number. Broido et al. reported that, about 96% of networks are not strictly scale-free [31]. In particular, the form of pressure-head and heavy-tail in Figure 1(g) and degree distribution of tail approximate power law is prevalent and is significantly different from the BA model [2] in terms of head characteristics. In the simulation experiment, our model exhibits a strong network reproduction ability. Four different degree distribution patterns (Figures 1(a)–1(d)) correspond to degree



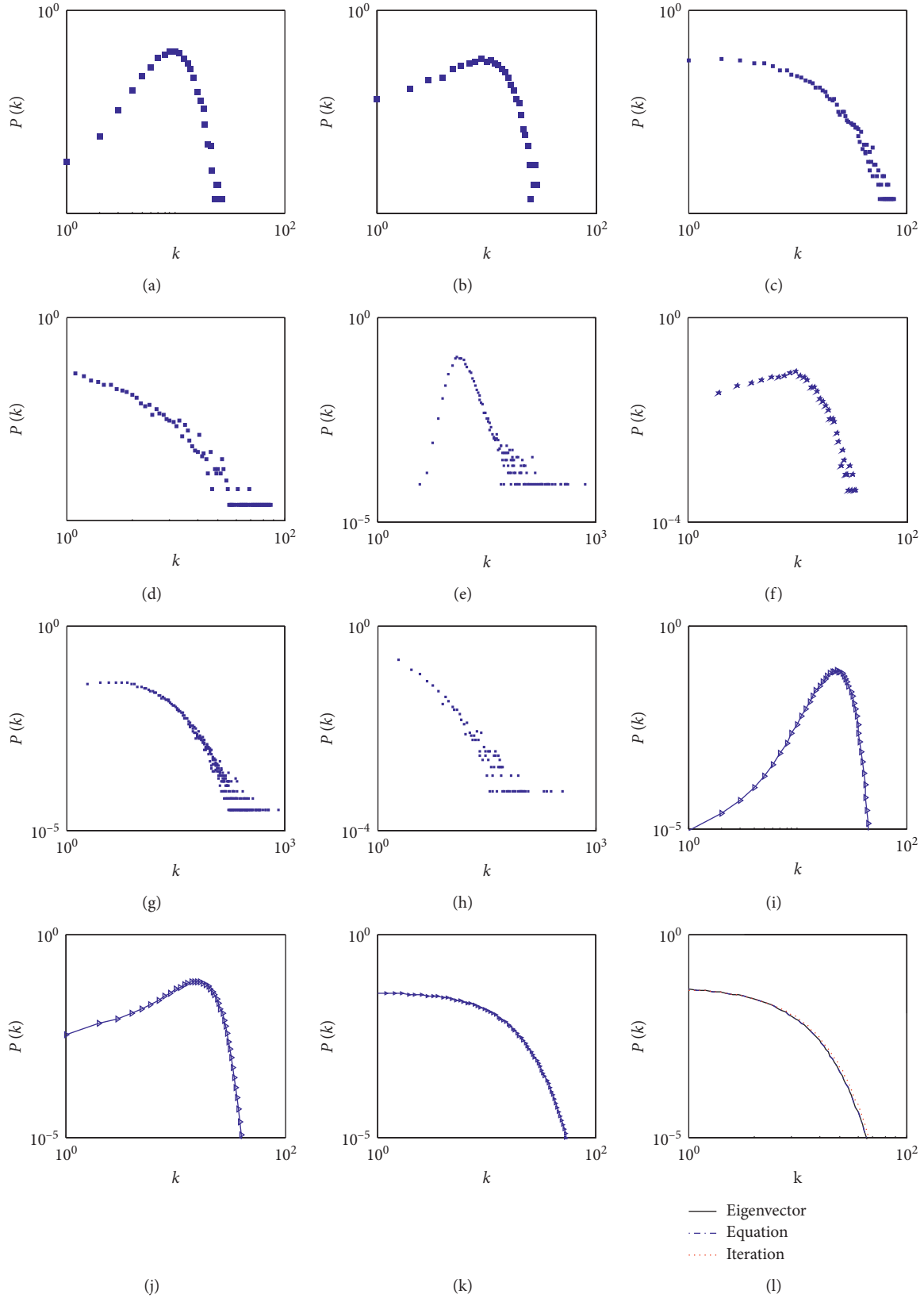


FIGURE 1: Comparison of degree distributions obtained by a simulation experiment, real network, and iteration estimation. (a–d) Degree distribution in different stages of simulation evolution; (e–h) Degree distribution in real network, data from online dictionary entry network [45], adolescent social friend network [35], publication citation network [15], and Bible vocabulary network [46]; (i–l) Degree distribution of different evolutionary stages given by the iterative estimation, (l) Degree distribution obtained using the three estimation methods.

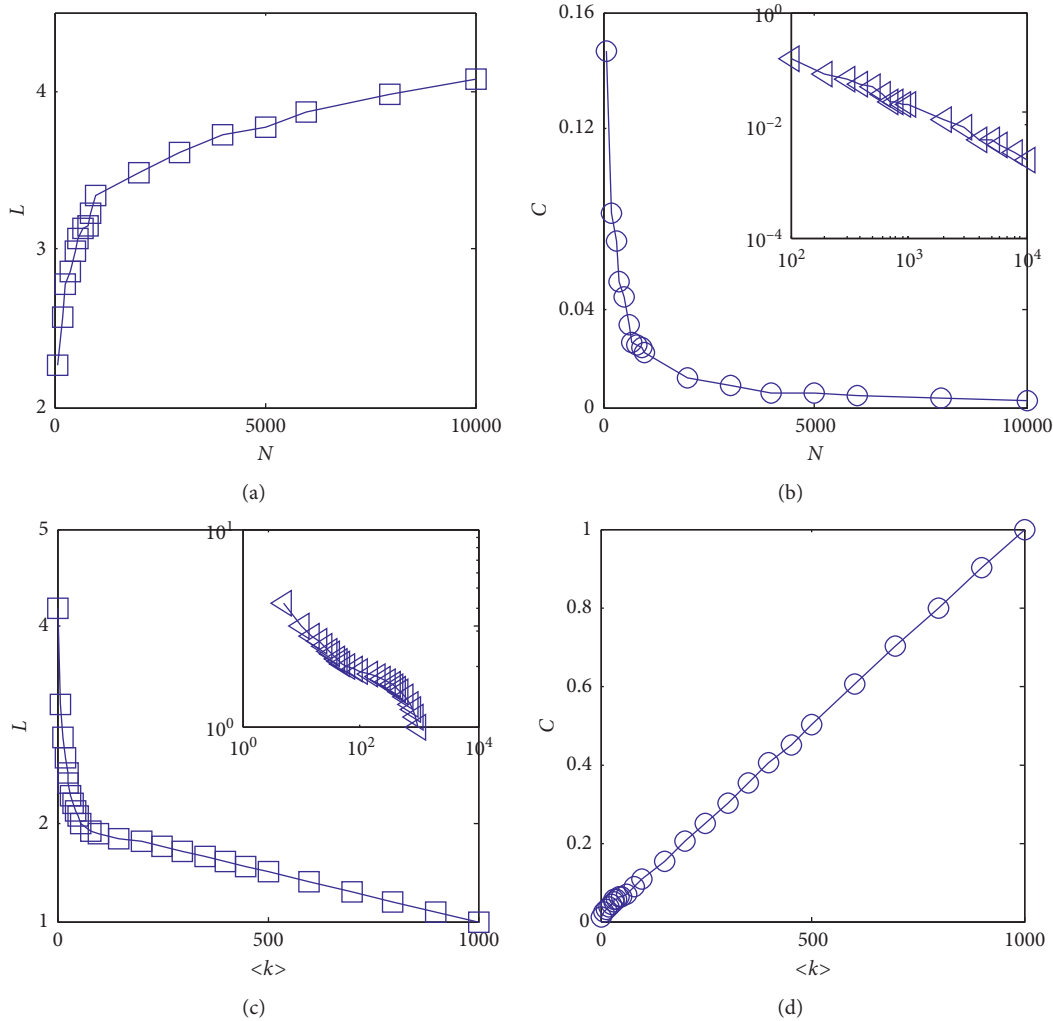


FIGURE 2: Average path length and network average clustering coefficient. (a) Average path length versus the number of nodes. (b) Network average clustering coefficient versus the number of nodes. (c) Average path length versus average degree. (d) Network average clustering coefficient versus average degree.

distributions of four real networks in different evolution stages. Figures 1(i)–1(l) show the results of three degree distribution estimation methods after stability distribution analysis. They are consistent with the real network and simulation results, and it is also the approximate power-law distribution of the tail. Head distribution is slightly different and is due to differences between theoretical analysis, simulation experiment, and real network evolution mechanisms, namely, the reconnection of edges appears sequentially in simulation and real networks, but not in theoretical analysis.

As shown in Figure 2, we studied the average path length ( $L$ ) and the average clustering coefficient ( $C$ ) of the model. Performance of the model on the average path is consistent with that of the real network. It shows the small-world characteristic. With an increase in  $N$ ,  $L$  is approximately proportional to the  $\ln N$ , which is close to ER, WS, and BA models [47, 48]. With the increase in average network degree

$\langle k \rangle$ ,  $L$  descends rapidly, especially in the early stage, the descending speed is a power law, and in the middle and later stages, it approaches 1. Table 1 compares the average path length of some real networks with simulation networks of the same size and average degree. Real network data are from literature [15, 46, 49–55], and the results show that the model can well describe the small-world characteristics of real networks. However, the clustering coefficient changes in precisely the opposite way. With an increase in network size,  $C$  rapidly decreases to 0 by a power law, like in the BA model, while with an increase in  $\langle k \rangle$ ,  $C$  increases to 1 by a linear law. These findings suggest that the model network in the more massive network average degrees has a useful node aggregation. The network model can generate a high clustering coefficient of the network, and the clustering coefficient is adjustable. But it has obvious gaps when compared to the real network. Some smaller  $\langle k \rangle$  of the real network also possesses a high clustering coefficient.

TABLE 1: Comparisons of average path length between real networks and simulation networks.

Network	$N$	$\langle k \rangle$	$L(\text{real})$	$L(\text{model})$
Dolphins [49]	62	5.12903	3.45433	2.7345
Gene fusion [50]	291	1.91753	3.87139	4.0270
Yeast [51]	1870	2.43529	7.06974	8.2569
Route views [15]	6474	4.29255	3.66686	5.5543
Sister cities [46]	14274	2.88258	7.65392	7.2581
Network science [52]	1461	3.75359	6.28528	5.0790
Food web [53]	183	27.2568	2.12782	1.9885
Political blogs [54]	1224	54.6242	2.74667	2.1063
Infectious [55]	410	84.3805	1.78397	1.7953

## 5. Conclusion

We propose a new network evolution mechanism that is unbiased for all network nodes. It is believed that the scale-free network is caused by the preferred connection mechanism [16–21], and our study shows that scale-free results of real networks could have other mechanisms. Model results revealed that the head's degree distribution is pressure-head, slightly smaller than that of the power-law distribution, and the empirical data is consistent. The pressure-head phenomenon is common in real network data [15, 35, 45, 46]. Models such as the BA model can only exhibit power-law distribution. They do not provide results of the pressure-head. However, this phenomenon was produced in our model. The simulation shows that it exhibits the tail power-law distribution and the pressure-head phenomenon. Another *nonpreferential attachment* mechanism was proposed [56]. The connection between two nodes depends asymmetrically on their types. The model results based on graph limit theory, in the sense that the number of copies of any fixed subgraph converges when network size tends to infinity, while network distribution converges when time tends to infinity in our work. However, their results do not involve scale-free distribution and the shortest path discussed.

The BA network model [2, 48] shows that the network's degree distribution conforms to the power law  $P_k \sim k^{-\alpha}$ ,  $\alpha = 3$ . Our model's simulation implies that it has a broader range of  $\alpha$  and is a power-law adjustable model, consistent with the real network. In verifying small-world characteristics, simulation results showed excellent performance when compared with some real networks. The average path length of the network is very close to the real network. As the network size increases, the average path length is proportional to  $\ln N$ , and it rapidly decreases as the average degree increases. However, the performance of the clustering coefficient is not consistent with that of real networks. As the network size increases and the average degree decreases, the clustering coefficient tends to approach zero. In contrast, many real networks have a high clustering coefficient at a larger scale. When the triangle connection mechanism [19] is employed in this model, the clustering coefficient could be quickly improved, but it will be challenging to theoretically prove stable distribution.

As mentioned above, many previous studies concluded that degree distribution of the network should be scale-free.

However, some studies contradict this conclusion. Broido and Clauset reported that scale-free networks are rare [31], and only 4% of networks have the most vital scale-free characteristic. These are two seemingly opposite conclusions, but they may not be contradictory from a different perspective. This is because scale-free properties referred to tail distribution but strictly speaking is not on the whole range. This reason accounts for 'scale-free networks are rare' [31]. It is reported that very different networks may have the same degree distribution [57]. The degree distribution is *not* the only important thing in network. Even networks with identical degree distributions have completely different properties.

Pressure-head and heavy-tail distribution in the real network data seriously affects acceptance of scale-free distribution. The model proposed in this paper provides a unified explanation. Figures 1(a)–1(d) shows that various distributions can appear in the process of network evolution, with apparent non-power-law distribution and tail approximate power-law distribution, consistent with multiple distributions in real network data (Figures 1(e)–1(h)). In other words, network distribution can be scale-free or non-scale-free. Degree distribution of a network is always in the process of random evolution. It is a particular stage in the evolution process for all real network structures, rather than the network's final state. Maybe the limit state of network evolution is scale-free as observed in many real networks with tail power laws. But many networks are not strictly scale-free because they have not yet achieved maturity or stability in their evolution. Two viewpoints can be unified into our model, which provides a new idea for the understanding degree distribution in network research. The real data employed are rather restricted and insufficient to support the strong claims of the paper. More extensive comparison with real networks should be performed regarding the degree distribution. The amount of data used in this paper is relatively limited, and the more the data, the better it can support the viewpoints of this paper.

## 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.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 11502062.

## References

- [1] J. W. Duncan and H. S. Steven, "Collective dynamics of small world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

- [3] R. Albert, H. Jeong, and A.-L. Barabási, "Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130-131, 1999.
- [4] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, pp. 177-183, 2007.
- [5] G. Lima-Mendez and J. van Helden, "The powerful law of the power law and other myths in network biology," *Molecular BioSystems*, vol. 5, no. 12, pp. 1482-1493, 2009.
- [6] M. T. Agler, J. Ruhe, S. Kroll et al., "Microbial hub taxa link host and abiotic factors to plant microbiome variation," *PLoS Biology*, vol. 14, pp. 1-31, 2016.
- [7] G. Ichinose and H. Sayama, "Invasion of cooperation in scale-free networks: accumulated versus average payoffs," *Artificial Life*, vol. 23, no. 1, pp. 25-33, 2017.
- [8] L. Zhang, M. Small, and K. Judd, "Exactly scale-free scale-free networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 433, pp. 182-197, 2015.
- [9] M. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, no. 5, pp. 323-351, 2005.
- [10] K.-I. Goh, E. Oh, H. Jeong, B. Kahng, and D. Kim, "Classification of scale-free networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 20, pp. 12583-12588, 2002.
- [11] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of Modern Physics*, vol. 87, no. 3, pp. 925-979, 2015.
- [12] M. E. J. Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, Article ID 016128, 2002.
- [13] H. A. Simon, "On a class of skew distribution functions," *Biometrika*, vol. 42, no. 3-4, pp. 425-440, 1955.
- [14] R. Pastor-Satorras, E. Smith, and R. V. Solé, "Evolving protein interaction networks through gene duplication," *Journal of Theoretical Biology*, vol. 222, no. 2, pp. 199-210, 2003.
- [15] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 1-40, 2007.
- [16] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47-97, 2002.
- [17] D. J. Price, "Networks of scientific papers," *Science (New York, N.Y.)*, vol. 149, no. 3683, pp. 510-515, 1965.
- [18] D. J. S. Price, "A general theory of bibliometric and other cumulative advantage processes," *Journal of the American Society for Information Science*, vol. 27, no. 5-6, pp. 292-306, 1976.
- [19] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical Review E*, vol. 65, Article ID 026107, 2002.
- [20] G. Bianconi and A. L. Barabási, "Bose-Einstein condensation in complex networks," *Physical Review Letters*, vol. 86, no. 24, pp. 5632-5635, 2000.
- [21] X. Li and G. R. Chen, "A local-world evolving network model," *Physica A-Statistical Mechanics & Its Applications*, vol. 328, no. 1-2, pp. 274-286, 2003.
- [22] E. B. Elizabeth and M. E. J. Newman, "Aspirational pursuit of mates in online dating markets," *Science Advances*, vol. 4, Article ID eaap9815, 2018.
- [23] T. House, J. M. Read, L. Danon, and M. J. Keeling, "Testing the hypothesis of preferential attachment in social network formation," *EPJ Data Science*, vol. 4, no. 13, 2015.
- [24] W. Willinger, D. Alderson, and J. C. Doyle, "Mathematics and the internet: a source of enormous confusion and great potential," *Mathematics and the Internet: A Source of Enormous Confusion and Great Potential. Notices of the American Mathematical Society*, vol. 56, no. 5, pp. 586-599, 2009.
- [25] R. Tanaka, "Scale-rich metabolic networks," *Physical Review Letters*, vol. 94, no. 1-4, 2005.
- [26] M. P. H. Stumpf and M. A. Porter, "Critical truths about power laws," *Science*, vol. 335, no. 6069, pp. 665-666, 2012.
- [27] M. Golosovsky, "Power-law citation distributions are not scale-free," *Physical Review Letters*, vol. 96, no. 3, pp. 1-12, 2017.
- [28] M. P. H. Stumpf, C. Wiuf, and R. M. May, "Subnets of scale-free networks are not scale-free: sampling properties of networks," *Proceedings of the National Academy of Sciences*, vol. 102, no. 12, pp. 4221-4224, 2005.
- [29] M. O. Jackson and B. W. Rogers, "Meeting strangers and friends of friends: how random are social networks?" *American Economic Review*, vol. 97, no. 3, pp. 890-915, 2007.
- [30] R. Khanin and E. Wit, "How scale-free are biological networks," *Journal of Computational Biology*, vol. 13, no. 3, pp. 810-818, 2006.
- [31] A. D. Broido and A. Clauset, "Scale-free networks are rare," *Nature Communications*, vol. 10, no. 1, 2019.
- [32] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661-703, 2009.
- [33] S. Redner, "Citation statistics from 110 years of physical review," *Physics Today*, vol. 58, no. 6, pp. 49-54, 2005.
- [34] F. Radicchi, S. Fortunato, and C. Castellano, "Universality of citation distributions: toward an objective measure of scientific impact," *Proceedings of the National Academy of Sciences*, vol. 105, no. 45, pp. 17268-17272, 2008.
- [35] J. Moody, "Peer influence groups: identifying dense clusters in large networks," *Social Networks*, vol. 23, no. 4, pp. 261-283, 2001.
- [36] A. Mehdi and B. R. Lotfi, "An efficient two-phase model for computing influential nodes in social networks using social actions," *Journal of Computer Science & Technology*, vol. 33, no. 2, pp. 286-304, 2018.
- [37] R. A. Rossi and N. K. Ahmed, "Role discovery in networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1112-1131, 2015.
- [38] G. Sperling, "The information available in brief visual presentations," *Psychological Monographs: General and Applied*, vol. 74, no. 11, pp. 1-29, 1960.
- [39] G. Rainer and E. K. Miller, "Effects of visual experience on the representation of objects in the prefrontal cortex," *Neuron*, vol. 27, no. 1, pp. 179-189, 2000.
- [40] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Communications*, vol. 6, p. 6864, 2015.
- [41] R. Hausmann, *The Atlas of Economic Complexity: Mapping Paths to Prosperity*, MIT Press, Cambridge, MA, USA, 2014.
- [42] U. Redmond and P. D. Cunningham, *A Temporal Network Analysis Reveals the Unprofitability of Arbitrage in the Prosper Marketplace*, Pergamon Press, Oxford, UK, 2013.
- [43] G. Iosifidis, Y. Charette, E. M. Airoidi, G. Littera, L. Tassioulas, and N. A. Christakis, "Cyclic motifs in the Sardex monetary network," *Nature Human Behaviour*, vol. 2, no. 11, pp. 822-829, 2018.
- [44] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [45] V. Batagelj, A. Orvar, and M. Zaversnik, "Network analysis of texts," *Language Technologies*, vol. 40, pp. 143-148, 2002.
- [46] J. Kunegis, "KONECT. The koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web Companion*, 2013, pp. 1343-1350, Rio de Janeiro, Brazil, May 2013.

- [47] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Physical Review Letters*, vol. 90, no. 5, Article ID 058701, 2003.
- [48] K. Klemm and V. M. Eguíluz, "Growing scale-free networks with small-world behavior," *Physical Review E*, vol. 65, no. 5, Article ID 057102, 2002.
- [49] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Sloaten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [50] M. Höglund, A. Frigyesi, and F. Mitelman, "A gene fusion network in human neoplasia," *Oncogene*, vol. 25, no. 18, pp. 2674–2678, 2006.
- [51] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.
- [52] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, 2006.
- [53] N. D. Martinez, J. J. Magnuson, T. Kratz, and M. Sierszen, "Artifacts or attributes? effects of resolution on the little rock lake food web," *Ecological Monographs*, vol. 61, no. 4, pp. 367–392, 1991.
- [54] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: divided they blog," in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, Chicago, IL, USA, August 2005.
- [55] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, "What's in a crowd? Analysis of face-to-face behavioral networks," *Journal of Theoretical Biology*, vol. 271, no. 1, pp. 166–180, 2011.
- [56] Y. Shang, "Limit of a nonpreferential attachment multitype network model," *International Journal of Modern Physics B*, vol. 31, no. 5, Article ID 1750026, 2016.
- [57] Y. Shang, "Distinct clusterings and characteristic path lengths in dynamic small-world networks with identical limit degree distribution," *Journal of Statistical Physics*, vol. 149, no. 3, pp. 505–518, 2012.



## Research Article

# Cross-Platform Drilling 3D Visualization System Based on WebGL

Shanshan Liu <sup>1</sup>, Yueli Feng <sup>1,2</sup>, Xiaoqiu Wang <sup>1</sup>, and Pengyin Yan <sup>1</sup>

<sup>1</sup>College of Petroleum Engineering, China University of Petroleum, Beijing 102249, China

<sup>2</sup>State Key Laboratory of Petroleum Resources and Prospecting, China University of Petroleum, Beijing, China

Correspondence should be addressed to Xiaoqiu Wang; 13466396559@126.com

Received 10 January 2021; Revised 23 February 2021; Accepted 23 April 2021; Published 5 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Shanshan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study develops a novel drilling 3D visualization solution based on WebGL, termed as WebDrillingViz, and introduces the system architecture design and software programming implementation. The software is part of the Engineering Technology Internet of Things (IoT) System, interfacing with other software, and also capable of direct hardware interfacing for data retrieval and system control. It is fully web-based, used real time, and used in RTOC (Real-Time Operating Center) of IoT system, which is a software system for drilling process remote monitor and decision. WebDrillingViz uses the most frontier HTML5 technology to realize a brand-new drilling 3D visualization system. The front end is designed in single-page application (SPA) mode and adopts technologies such as angular, bootstrap, and WebGL. The front-end uses single page application (SPA) mode, Angular, Bootstrap, WebGL and other technologies are used. The back-end data services provide data interface support for front-end visualization applications based on HTTP protocol which uses NodeJS, a lightweight development platform suitable for cloud platform, and Restify to realize a REST JSON API. Both sides are using the same object-oriented development language—TypeScript. The front-end develops an easy-to-extend 3D visualization class library based on WebGL for drilling. It is encapsulated as Angular modularization to form an Angular component, which can be used standalone or integrated into other Angular applications. At the same time, the back-end microservice architecture combined with container and cloud technology is easy to maintain, deploy, and expand and has the advantages of being lightweight, cross-platform, flexible, and efficient. Using HTML5 standard and Bootstrap's responsive layout achieves cross-platform, which can support different operating systems and screen sizes. The system has better robustness and maintainability, thanks to the object-oriented and strong typing characteristics of TypeScript. Practical application shows that WebDrillingViz is efficient, capable of visualization of large drilling 3D scene, and compatible with mainstream devices, such as Windows, Linux, macOS, iOS, and Android. The use of open standards-based modern web technologies and data format enables a more lightweight and economical solution. WebGL, Angular, NodeJS, and TypeScript formed a powerful technology stack, which can be used as an excellent reference for other browser-based visualization development.

## 1. Introduction

In November 2005, the International Telecommunication Union (ITU) released a report entitled “the Internet of things,” which formally puts forward the Internet of things (IOT), which attracted the attention of governments and industries. Petroleum industry is an important industry of IOT application, which has been applied in some enterprises. The Internet of things system of engineering technology includes the automatic data acquisition, remote data transmission, data center, and remote operation support center (RTOC). Through the remote operation support center, technical experts can view the data and video

information of the well pad on the computer of the center for remote analysis and decision-making. RTOC aims at building data acquisition and application, a multiprofessional collaborative service platform to improve the integrated service capabilities, realize intelligent operation support, the integration of engineering and technology wellbore business, and information technology [1–3]. RTOC includes the following parts: automatic collection, integrated storage, and remote transmission of field data; real-time data detection, early warning and analysis optimization; remote technical support; and decision-making. It provides a powerful guarantee for making and transmitting drilling decision quickly and effectively, as shown in Figure 1.

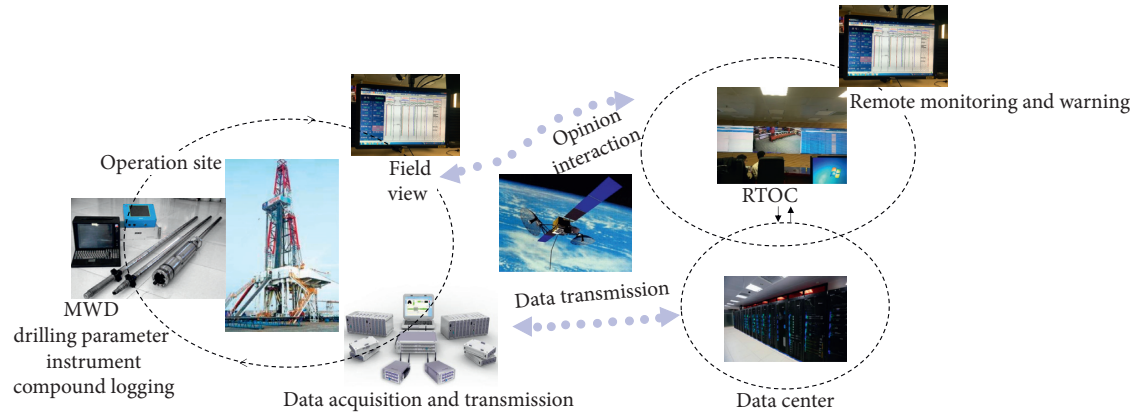


FIGURE 1: Remote operation center architecture.

Drilling 3D visualization system is an important task for building web applications in ROTC, which provides a highly visualized working environment to view real-time 3D drilling scene for drilling process remote monitor and decision. It can put borehole trajectories, LWD, MWD, mud logging, well-logging data, seismic slice, and formation interfaces, reservoir geological model all into a 3D scene, realizing the information sharing of logging, geology, and engineering; thus, experts can get realistic underground scenes. Cross-platform, lightweighted, and open are three basic rules when building drilling 3D visualization systems. Currently, some efforts have been made on drilling 3D visualization; however, most of them are traditional desktop software which cannot provide online service for ROTC. At present, some companies built such visualization products: 3D visualization software enables drilling engineers, geophysical engineers, geological engineers, and reservoir engineers to work with the same data volume environment, breaking professional and geographical constraints, and making multidisciplinary [4–9]. Combination really becomes a reality. Most of the existing 3D visualization technologies adopt the client-server (C/S) mode, such as DecisionSpace®, and WellViz3D are released as standalone software, which must be installed and used in a local operation system (OS). Discovery Web of Kongsberg adopts the B/S mode. However, based on ActiveX plug-in technology, special plug-ins need to be installed, which can only support IE browser on PC Windows and cannot support mobile devices (Android and IOS) [10]. Undoubtedly, a web-based drilling 3D visualization system has flexibility to perform real-time monitoring of drilling activities through a browser platform in ROTC. However, there is no 3D visualization technology of drilling based on standard browser at present. A cross-platform online drilling 3D visualization system to describe the underground geological environment and drilling state using WebGL (termed as WebDrillingViz) based on HTML5 and WebGL standard is developed in this paper, which can support all mainstream operating systems and browsers. By using standards-based technologies, this visualization can operate on a multitude of devices removing both the technical and logistical restraints on a remote collaboration. The online drilling 3D visualization system is

different from that on local drilling 3D visualization in the following aspects:

- (i) Lightweight: HTML5 and WebGL enable the browsers realize excellent 3D effects without any plugins, the REST server built with Node.js and Restify in microservice architecture, which can easily be deployed on cloud.
- (ii) Cross-platform: rich, interactive, browser-based, cross-platform visualization tools can serve both PCs and mobile devices that run different OSs, which support WebGL, such as Windows, Linux, IOS, and Android.
- (iii) Open: the software supports the data formats commonly used in oil and gas industry and can easily import geological and related data into the system's database.
- (iv) Economic: the software is all based on HTML5 and open-source technology stack, which is not bundled with expensive software packages and has no tedious licensing restrictions.

## 2. 3D Visualization Technology on the Web

The idea is to update the experts of ROTC with live data, i.e., detailed 3D real-time insights into the ongoing drilling operation. In the past few years, a great progress has been made in modern web browsers and network standards [11, 12]. Cross-browser and cross-platform standardization greatly simplifies the design and implementation of rich interactive Web applications. In the past, third-party plug-ins needed to realize complex interaction or graphics in browsers. Java applets, Flash, and Silverlight were used for browser-based rich and interactive applications. Many security issues, as well as poor support and performance on mobile devices, made these technologies gradually abandoned. Instead, a powerful set of Web standards (commonly known as HTML5) has evolved into a framework on which real browser-based and cross-platform visualization tools can be developed. Special interest in drilling and geological visualization is HTML5, HTML5 canvas, and WebGL.

**2.1. HTML5.** HTML5 is a markup language used for structuring and presenting content on the web pages. It is the fifth and current major version of the HTML standard, and it was developed by World Wide Web Consortium (W3C, a broad coalition of organizations).

Compared with the previous version, the 5th version has made a qualitative leap in processing graphics and images. On the premise of not relying on third-party plug-ins, a new standard for graphics and image applications is proposed, which supports dynamic display and interaction on various mobile platforms and further improves the flexibility and security of web applications. After discarding all kinds of plug-ins, it combined with JavaScript makes the expansion of website functions easier to achieve and the development of website more efficient and safer.

**2.2. HTML5 Canvas.** The <canvas/> HTML tag provides a container that programmatically draws graphical objects on the screen. In short, a canvas tag is similar to an image but provides the ability to draw raster-based dynamic graphics. The canvas tag has been existing for many years and was eventually integrated into the HTML5 specification. At present, all the major browsers and devices platforms have implemented canvas tag, which makes it a good and feasible way to draw dynamic drilling scenarios. Although the canvas is essentially a 2D space, 3D effects can be simulated in 3D games. The rendering engine calculates the perspective on the 3D object and maps it to the 2D view for display on a flat screen. No matter what the underlying technology is, this general method is suitable for graphic rendering of all 3D objects in the 2D environment. The canvas tag simply provides a standards-based and well-supported tool for rendering dynamic graphics in Web browsers.

**2.3. WebGL.** WebGL has been introduced as one of the powerful web features for developing 3D content, and it is based on OpenGL ES 2.0 and allows rendering 3D scene in the browser. WebGL provides a general interface for accessing the 3D graphics hardware of the underlying system in the browser and can realize the rendering of complex 3D scene efficiently. Currently, all major browsers provide support for WebGL, making it the ideal technology to render 3D drilling and geology scene across platform and devices.

### 3. System Architecture

The system was built on HTML5 standards and open-source frameworks.

**3.1. Overall Framework.** Under the environment of HTML5, the system adopts the B/S architecture building a network platform using the TypeScript development language. With the help of the language's full object-oriented and strong typing characteristics, the code has strong expansibility, reusability, security, robustness, and stability. The system uses an object-oriented design method, builds a scalable 3D

visualization class library, and encapsulates it as an Angular module, which has strong reusability and inheritance.

Through Angular's powerful compile ability, the compiled program can run efficiently in the browser. It accesses back-end RESTful data interface API to obtain user data and provides a friendly user interface by using the responsive layout, which can support both desktop and mobile systems.

The software realizes the unified management of visual objects, displays the coordinates of 3D scene, and performs the fundamental functions including zooming, rotating, and translating. Figure 2 illustrates software architecture including data layer, service layer, and view layer.

The data layer is Geological Engineering Warehouse, and it uses real-time and relational databases to receive and preserve static and dynamic drilling geological data. Static data include oilfield information, adjacent well data, drilling design data, formation tops, seismic slices, and other geological engineering data. Dynamic data include drilling, MWD, LWD, and mud logging data.

The service layer based on Node.js and Restify framework provides a RESTful API. It is the data interface between front-end application and the back-end data warehouse. All communications between the data warehouse and the visualization program will also be performed through the servers.

The view layer is a single page Web application (SPA), based on HTML5 standards and TypeScript language, using Angular front-end framework and Bootstrap UI library to realize responsive SPA. Through the WebGL interface of browser, 3D visualization components are implemented to realize web-based cross-platform interactive 3D visualization application for drilling and geological.

**3.2. Technology Stack.** This software uses a series of open-source technologies to build an economical and efficient technology stack.

**3.2.1. TypeScript.** TypeScript is an open-source programming language developed and maintained by Microsoft [13]. It is a strict syntactical superset of JavaScript and adds optional static typing to the language and offers a module system, classes, interfaces, and a rich gradual-type system. TypeScript developers sought a solution that would not break compatibility with the standard and its cross-platform support. TypeScript may be used to develop applications for both client-side and server-side execution. The client and server of this software are all developed with TypeScript.

**3.2.2. Angular.** Single Page Application (SPA) was built on expanding reach via the browser, reducing round-tripping, and enhancing User Experience (UX). Compared with traditional web applications, SPA application loads the html, CSS, and JavaScript programs of a single page at one time and dynamically updates the content when the user interacts with it. It enables Web application to realize complex dynamic interaction in one page. Advanced SPA application framework will compile, package, and compress all source

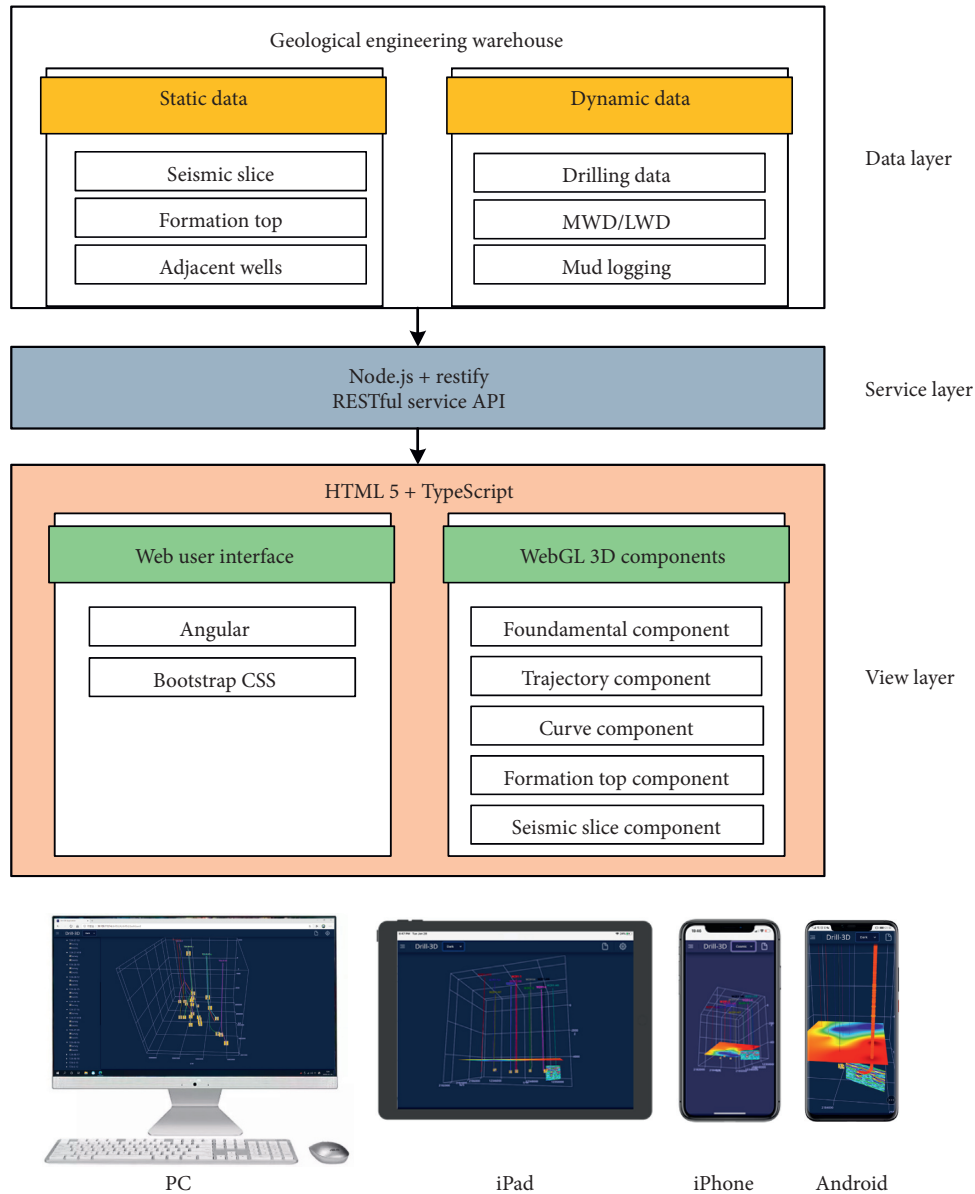


FIGURE 2: System architecture.

codes. The final application has only a small number of compiled, optimized, and compressed .js files. The execution efficiency is much higher than the traditional way through the browser one-time loading. The core of SPA technology is the combination of front-end technology and back-end REST data service. The interactive operation of the application is completed through JavaScript programs in the browser without the participation of the server. The server only provides the support of data services, so it can get closer to the interactive effect of desktop C/S applications.

A clear picture of typical SPA architecture contains a server and client, as shown in Figure 3. Angular has become an increasingly popular choice for rapid development of dynamic HTML pages [14]. The major front-end technologies are Angular, VUE, and React. Angular, an open-source TypeScript framework, has been developed to enable and

give an extreme freedom to client-side developers building powerful SPA. This software adopts the MVVM architecture and latest version 7.0 of Angular to develop.

**3.2.3. Responsive Web Design.** Although Angular implements SPA, it mainly solves the problems of data binding and page interaction. At the UI level, it needs to use additional framework. Responsive web design (RWD) is a method for web page construction to detect the user's screen size and orientation and dynamically change the layout accordingly for multiple devices [15], so the site produces the output, which is viewable and navigable with the devices and web software of the intended site users. It employs the use of flexible layouts (columns), scalable images, and CSS media queries. Thus, responsive web pages display their elements

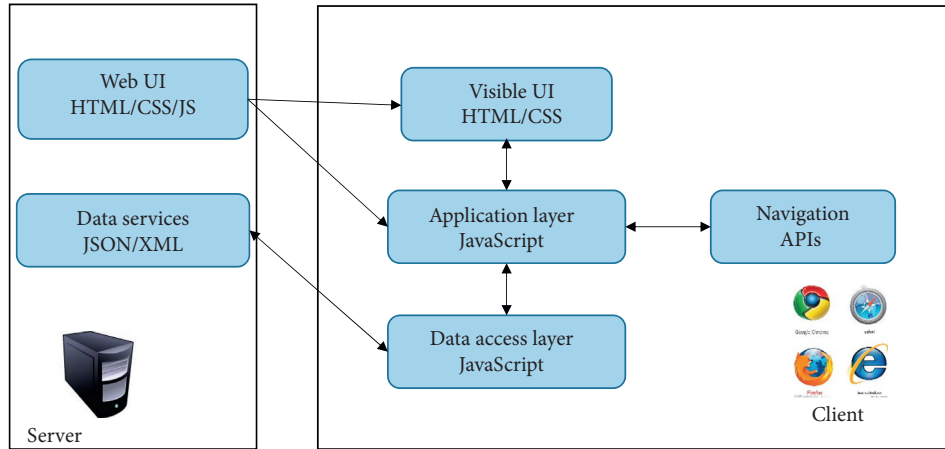


FIGURE 3: High-level architecture of SPA.

differently for different screen sizes. They change the sizes, shapes, and arrangements of their elements or make some elements invisible on small-sized screens using DOM APIs. An adaptive layout system for different platforms (devices) is needed in front-end page layout of the software, as is shown in Figure 4. Each of these devices has unique display dimensions that Web elements will adapt to, maintaining a consistent user experience.

3.2.4. *Three.js.* WebGL provides an API to create hardware-accelerate 3D graphics program, but we need to have an in-depth knowledge of how WebGL works internally to work with the API. Luckily, there are several JavaScript libraries available that hide the complexity of WebGL and provide an easy-to-use API to create 3D applications. Currently, the best of these libraries is Three.js, and it is a third-party open-source library of WebGL written in JavaScript, which provides a lightweight and easy-to-use 3D class library.

3.2.5. *Restify.* Restify is a middleware framework for developing REST-style APIs, which enable rapid development of robust REST service interfaces. Restify is used by some of the industry’s most respected companies to power some of the largest deployments of Node.js on planet Earth.

3.3. *3D Visualization Components.* According to the type of information to be visualized, corresponding components are developed to realize detailed display of well, borehole, real-time data and geological model, which makes visualization more useful for drilling experts, for example, the gamma ray curve related to formation lithology; the curve component can show the curve along the wellbore, and the change in formation lithology can be intuitively displayed, as shown in Figure 5.

The main 3D components include 3D visualization fundamental component, well-trajectory component, curve

component, formation tops component, seismic slice component, and reservoir model component.

3.3.1. *Fundamental Component.* The fundamental components are the most important part of the whole software. In addition to realizing the basic functions of 3D drawing, including zooming, rotation, illumination, object management, and coordinate display of graphics, an extensible 3D drawing framework is realized by using object-oriented technology. All other 3D drawing components are extended on this basis, and new components can be added any time according to the requirements, as shown in Figure 6.

Axis box (AxeBox class) is the most important foundation component, it can display X, Y, and Z grids and labels, and as a container manages all other 3D components, it can dynamically change label’s position and always displays the labels at appropriate positions.

There are eight color-map classes implemented IColorMap interface, and they can produce eight different color maps. It is also very easy to add new color-map class into software, as shown in Figure 7.

The basic component is responsible for the interaction with the mouse, realizing the translation, rotation, and scaling operations of the three-dimensional scene, depth alignment, coordinate system drawing, and graphics object management. Rotation, translation, and scaling are three basic operations for spatial three-dimensional objects, which enable users to view drilling three-dimensional scenes from different angles and positions. Although formulas can be used to represent each transformation, the formulas can become very complex when representing multiple continuous transformations. Using the coordinate transformation matrix, various basic transformations and combinational transformations can be realized more conveniently.

(1) Basic transformation of three-dimensional graphics

Rotation: spatial rotation can be decomposed into two-dimensional rotation around three coordinate axes. The



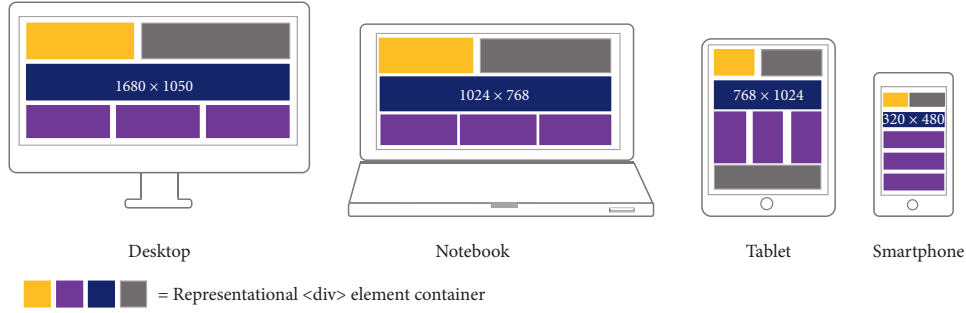


FIGURE 4: Responsive web design for multiple devices.

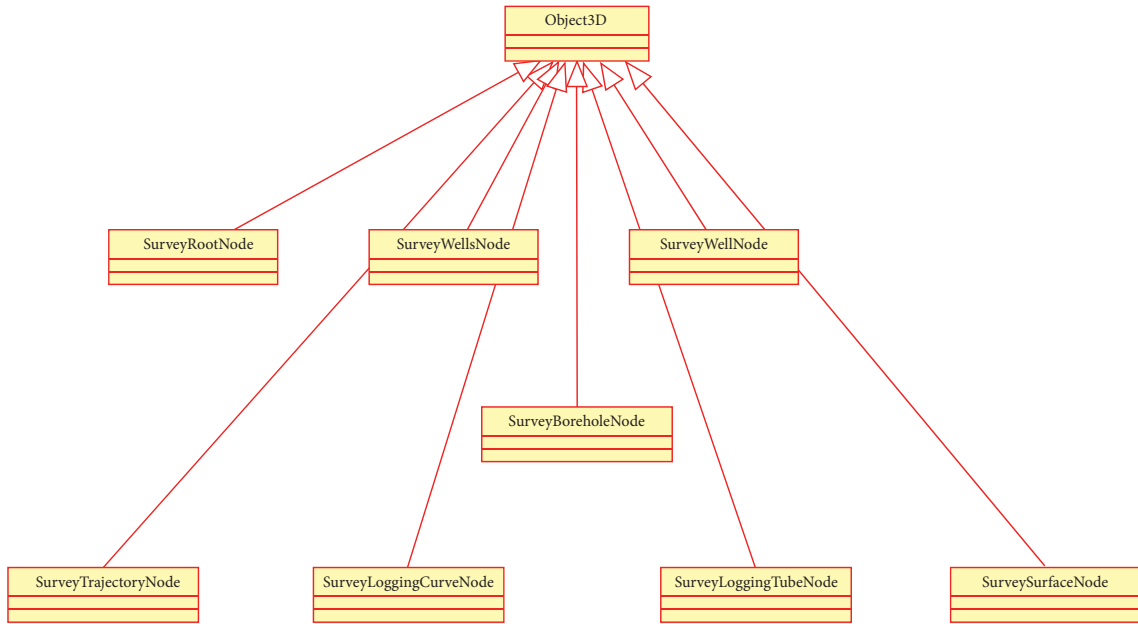


FIGURE 5: Class diagram of 3D visualization components for drilling.

rotation angles  $\psi$ ,  $\varphi$ , and  $\theta$  around the X, Y, and Z axes are transformed into

$$\begin{aligned}
 R_x(\psi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix}, \\
 R_y(\varphi) &= \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix}, \\
 R_z(\theta) &= \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.
 \end{aligned} \tag{1}$$

The complete transformation is

$$R = R_x R_y R_z. \tag{2}$$

Translation: spatial translation is the movement of an object in any distance and direction. The transformation of the moving distances  $t_x$ ,  $t_y$ , and  $t_z$  is as follows:

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}. \tag{3}$$

Scaling: scale the object. In the direction of X, Y, and Z, the scaling factors  $s_x$ ,  $s_y$ , and  $s_z$  are converted into

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} s_x x & s_y y & s_z z & 1 \end{pmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Based on the above basic transformation, any complex three-dimensional transformation can be decomposed into a combination of three basic transformations; for example, the object is rotated around any point  $(x, y, z)$  in space. The transformation steps are as follows: first, the center point  $(x, y, z)$  is translated to the origin  $(0, 0, 0)$  and then rotated around the origin, and finally, the origin is translated to the

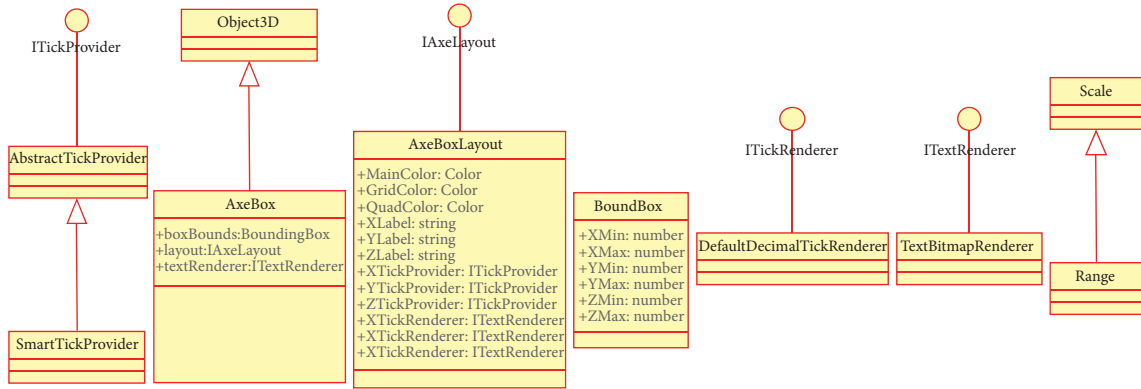


FIGURE 6: Class diagram of 3D visualization fundamental components.

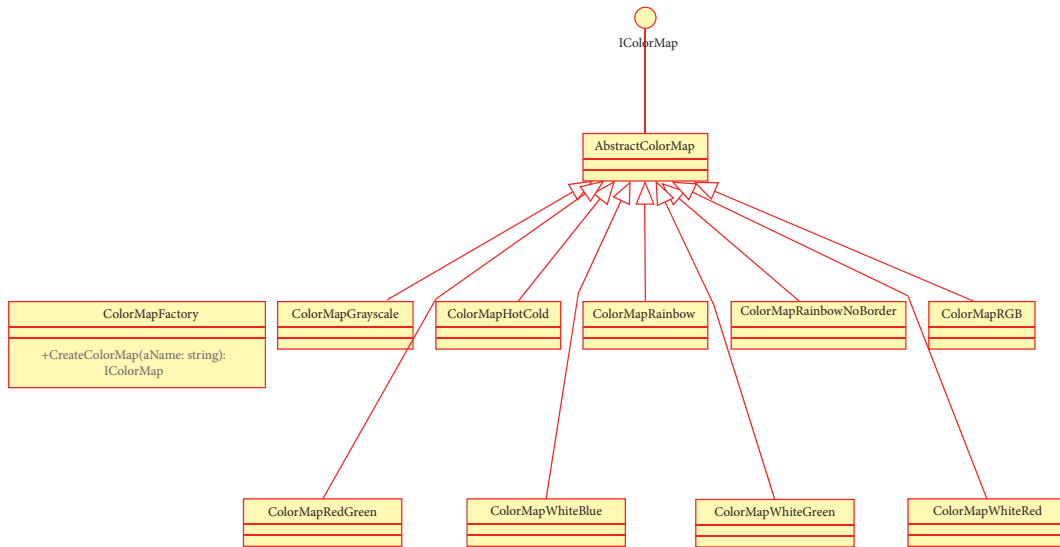


FIGURE 7: Color-maps classes diagram.

center point  $(x, y, z)$ . The final transformation matrix can be obtained by multiplying the matrices in the transformation process. In practical application, only the final matrix is calculated, and the complex process in the middle is not concerned:

$$M = T(-x, -y, -z)R(\psi, \varphi, \theta)T(x, y, z),$$

$$\begin{pmatrix} x' & y' & z' & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} M. \quad (5)$$

### (2) Depth alignment

The NS, EW, and TVD for a single well are relative to the wellhead. For different wells, the map-north, map-east, and elevation of the well are different. If multiwell data are placed in the same coordinate system, all wellhead data must be corrected to a unified coordinate system. The coordinate Z

takes the sea level as zero and the upward direction as positive:

$$(u_E, u_N, u_z) = (W_E + W_i, W_N + N_i, W_{bh} - TVD_i), \quad (6)$$

where  $(u_E, u_N, u_z)$  are the uniform coordinates, m;  $(E_i, N_i, TVD_i)$  are well data coordinates, m;  $W_E$  are eastern coordinates of wellhead, m; and  $W_N$  are borehole north coordinates, m.

### (3) Perspective transformation and mouse interaction

The nature of perspective is the same as photographic imaging. It is projecting a 3D shape onto a 2D plane. Perspective is very close to human vision; for example, when the human eye looks at a road, the road in the distance looks narrower. The basic principles of perspective are shown in Figure 8.

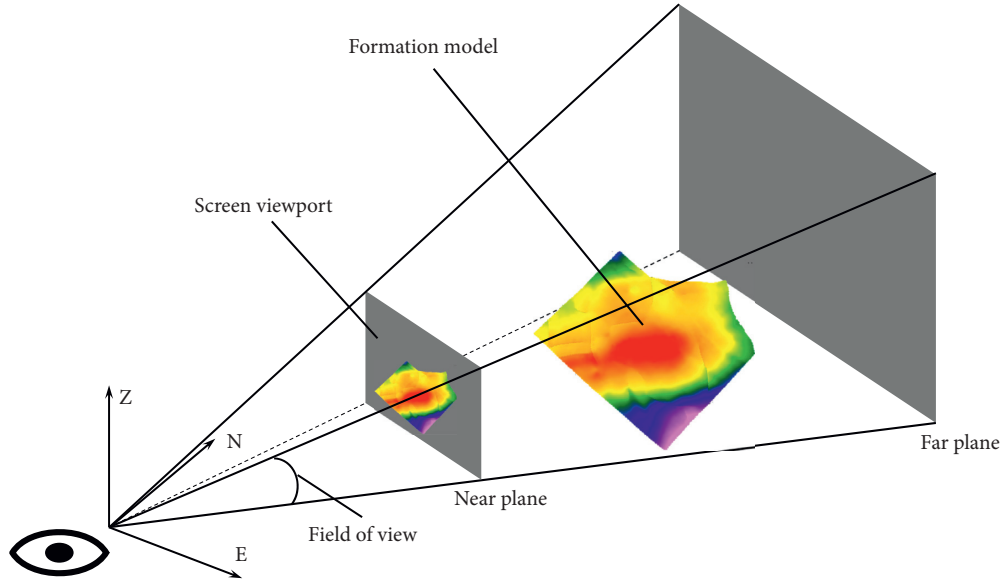


FIGURE 8: Perspective schematic diagram.

Through perspective transformation, the 3D graphics are presented on the 2D screen. When users want to observe the 3D scene with different distances and angles, they can move the camera position through the mouse. Mouse movement is a 2D action, so it is necessary to unproject the movement of the mouse to the 2D space according to the position of the camera. The users can achieve the effect of rotation and scaling by moving the position of the camera.

The software can trace the position of the camera in the process of mouse movement and dynamically adjust the display content; for example, in the coordinate system, only displaying three grid-faces of the axe-box located far-end can achieve better display effect. By calculating the distance between six planes of the axe-box and the camera, three distant planes are selected to display. The relative relationship between the ticks' text and the sight on the screen is calculated to reasonably determine the position of the calibration text and obtain better visual effect.

**3.3.2. Trajectory Component.** The trajectory component is used to display the design and drilling trajectories of the wellbore including static and real-time data.

The calculation of the trajectory is the basis of the well visualization. The track data, returned by the measurement tool mainly including measure depth (MD), deviation angle (Dev), azimuth angle (Azi), are shown in Table 1. The space coordinates of the trajectory are obtained by calculation.

The spatial coordinate parameters of the trajectory are defined as follows: North/South coordinate (NS), East/West coordinate (EW), and the true vertical depth (TVD), which represent the displacement of a point on the trajectory relative to the North/South, East/West, and the vertical direction of the wellhead, respectively, corresponding to the

TABLE 1: Trajectory measurement data.

MD (m)	Inc (deg)	Azi (deg)
0	0	0
153.32	0.45	3.21
180.8	0.39	21.9
208.48	0.46	26.04
...	...	...

Y, X, and Z axes of the direct coordinate system, in which the North, East, and Bottom directions are positive; for example, if the vertical depth of a well is 1000 m, the TVD of the wellhead is 0 and the TVD of the bottom hole is 1000.

The calculation process is as follows: the wellhead is taken as the origin, where  $(TVD, NS, EW) = (0, 0, 0)$ , starting from the wellhead, in turn, recursive calculation of two adjacent points as  $(L_1, \alpha_1, \varphi_1, D_1, N_1, E_1)$ ,  $(L_2, \alpha_2, \varphi_2, D_2, N_2, E_2)$ .

By assuming the 3D curve shape of wellbore section between two measuring points, the vertical increment  $\Delta D$ , North/South increment  $\Delta N$ , and East/West increment  $\Delta E$  of measured section are calculated and then add them up:

$$\begin{aligned}
 D_2 &= D_1 + \Delta D, \\
 N_2 &= N_1 + \Delta N, \\
 E_2 &= E_1 + \Delta E.
 \end{aligned} \tag{7}$$

The wellhead coordinates are known; therefore,  $D_1$  and  $E_1$  are always known during the recursive process. The 3D coordinate values of the trajectory can be obtained by the recursive calculation. The methods of calculation are different due to the different assumed curve types of the well segments. This article supports number of commonly used

calculation methods in the industry. The following are the most commonly used minimum curvature method:

$$\begin{aligned}\Delta D &= \frac{\Delta L}{2} (\cos \alpha_1 + \cos \alpha_2) \frac{2}{r} \tan \frac{r}{2}, \\ \Delta E &= \frac{\Delta L}{2} (\sin \alpha_1 \sin \beta_1 + \sin \alpha_2 \sin \beta_2) \frac{2}{r} \tan \frac{r}{2}, \\ \Delta N &= \frac{\Delta L}{2} (\sin \alpha_1 \cos \beta_1 + \sin \alpha_2 \cos \beta_2) \frac{2}{r} \tan \frac{r}{2}, \\ \gamma &= \cos^{-1} (\cos \alpha_1 - \cos \alpha_2 + \sin \alpha_1 \sin \alpha_2 \cos (\beta_1 - \beta_2)),\end{aligned}\quad (8)$$

where  $L$  is the measuring depth, m;  $\alpha$  is the deviation angle, deg;  $\varphi$  is the azimuth angle, deg;  $D$  is the true vertical depth, m;  $N$  is the North/South coordinate, m;  $E$  is the East/West coordinate, m;  $\Delta D$  is the vertical depth increment, m;  $\Delta N$  is the North/South coordinate increments, m;  $\Delta E$  is the East/West coordinate increments, m;  $\Delta L$  is the length of well depth between two adjacent points, m; and  $\gamma$  is Dogleg, deg.

In the well trajectory model, a new method of natural curve method is proposed. Because of the high complexity of the model, it cannot be solved analytically. It needs to solve the complex implicit equation, and it is easy to encounter the problem of iterative divergence. This paper discusses the natural curve calculation methods under fixed point problem and designs a new solution process.

Fixed point problem (build turn point): given  $\Delta N \Delta E \Delta Z$  solving  $\Delta L$ ,  $K_\alpha$ , and  $K_\phi$ , as shown in Figure 9.

Assume the starting and ending points of the well section are A and B, and the known conditions are  $(L_A, \alpha_A, \phi_A, N_A, E_A, Z_A)$  and  $(N_B, E_B, Z_B)$ , where  $L_A, \alpha_A, \phi_A$  and  $L_B, \alpha_B, \phi_B$  are depth measurement, well deviation angle, and azimuth angle of the two measuring points above and below the well section,  $K_\alpha$  is the rate of deviation change (build-up rate),  $K_\phi$  is the azimuth rate of change (steering rate), and  $\Delta N, \Delta E, \Delta Z$  are the increments of the well section in the North, East, and vertical directions, respectively.

The process of solving the model is as follows:

(1) Calculate the slope of AB line:

$$\alpha_{A,B} = \cos^{-1} \left( \frac{\Delta Z}{L_{A,B}} \right), \quad (9)$$

where

$$\begin{aligned}L_{A,B} &= \sqrt{\Delta N^2 + \Delta E^2 + \Delta Z^2}, \\ \Delta N &= N_B - N_A, \Delta E = E_B - E_A, \Delta Z = Z_B - Z_A.\end{aligned}\quad (10)$$

When  $\alpha_{A,B} > 0$ , the well section is building up. When  $\alpha_{A,B} < 0$ , then the well section is dropping, and when  $\alpha_{A,B} = 0$ , the slope is holding.

(2) Assign an initial value to  $\alpha_B$ :

$$\alpha_B^0 = 2\alpha_{A,B} - \alpha_A. \quad (11)$$

(3) Confirm the initial value of  $\phi_B$  as  $\phi_B^0$ .

According to the azimuth and coordinate offset of point A, the azimuth change direction and initial value of azimuth are solved. Solving the initial value of azimuth, the transformation matrix  $M$  is established according to  $\phi_A$ , and  $N$  and  $E$  are transformed to determine the coordinate increment along the direction of azimuth  $\phi_A$ , and the initial value of azimuth change  $\Delta\phi^0$  is determined:

$$\begin{bmatrix} \Delta N' \\ \Delta E' \end{bmatrix} = \begin{bmatrix} \cos \phi_A & \sin \phi_A \\ -\sin \phi_A & \cos \phi_A \end{bmatrix} \begin{bmatrix} \Delta N \\ \Delta E \end{bmatrix}, \quad (12)$$

$$\Delta\phi^0 = \tan^{-1} \left( \frac{\Delta E'}{\Delta N'} \right),$$

$$\phi_B^0 = \phi_A + \Delta\phi^0.$$

(4) Calculate well section length  $\Delta L$ .

According to the equation,

$$\Delta L = \frac{\Delta Z}{c \cos \bar{\alpha}}. \quad (13)$$

(5) Solve azimuth  $\phi_B$ :

$$\frac{\Delta N}{\Delta E} = \frac{1/2\Delta L ((\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B))/\alpha_B - \alpha_A + \phi_B - \phi_A) + (\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B))/\alpha_B - \alpha_A - \phi_B + \phi_A)}{1/2\Delta L ((\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A))/\alpha_B - \alpha_A - \phi_B + \phi_A) + (\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A))/\alpha_B - \alpha_A + \phi_B - \phi_A)}, \quad (14)$$

$$\alpha_B - \alpha_A - \phi_B + \phi_A = \Delta\alpha - \Delta\phi, \quad (15)$$

$$\alpha_B - \alpha_A + \phi_B - \phi_A = \Delta\alpha + \Delta\phi, \quad (16)$$

$$f(\phi_B) = \Delta N \left( \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{\Delta\alpha - \Delta\phi} - \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{\Delta\alpha + \Delta\phi} \right) - \Delta E \left( \frac{\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} + \frac{\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right) = 0. \quad (17)$$

In equation (17), there is no analyzing expression for  $\phi_B$ .  $\phi_B$  is monotonically differentiable in the critical region, so the Newton iteration method is used to calculate  $\phi_B$ :

$$f'(\phi_B) = \Delta N \left( \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos((\alpha_B + \phi_B))}{\Delta\alpha + \Delta\phi} + \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right) - \Delta E \left( \frac{\cos(\alpha_B + \phi_B) - \cos(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin((\alpha_B + \phi_B))}{\Delta\alpha + \Delta\phi} - \frac{\cos(\alpha_B - \phi_B) - \cos(\alpha_A - \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} \right). \quad (18)$$

By substituting  $\phi_B^0$  into equation (16),  $f[\phi_B^0]$  is obtained. If  $|f[\phi_B^0]|$  is less than the set error, this paper uses 1.0E-6, then  $\phi_B^0$  is the solved azimuth angle  $\phi_B$ ; otherwise, use the following formula to calculate the new  $\phi_B^0$ :

$$\phi_B^0 = \phi_B^0 - \frac{f[\phi_B^0]}{f'[\phi_B^0]} \quad (19)$$

By repeating the above iterative process,  $\phi_B$  can be obtained.

It should be noted that, in formula (17) and formula (18), when there are two molecular terms  $|\Delta\alpha \pm \Delta\phi| \rightarrow 0$ , due to the floating-point calculation error of the computer, the calculation result will not be accurate. When it is equal to zero, there will be an error of dividing zero. In the process of iteration, this situation is avoided. Therefore, the algorithm must consider the limit problem of  $|\Delta\alpha \pm \Delta\phi| \rightarrow 0$ . In this critical region, limit is used instead of calculation to avoid calculation error.

The following is the limit problem in the above formula:

$$\begin{aligned} \lim_{\Delta\alpha + \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{\Delta\alpha + \Delta\phi} &= \cos(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{\Delta\alpha - \Delta\phi} &= \cos(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \sin(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \sin(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} - \frac{\cos(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \frac{1}{2} \sin(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A)}{(\Delta\alpha - \Delta\phi)^2} - \frac{\cos(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \frac{1}{2} \sin(\alpha_A - \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_B + \phi_B) - \cos(\alpha_A + \phi_A)}{(\Delta\alpha + \Delta\phi)^2} + \frac{\sin(\alpha_B + \phi_B)}{\Delta\alpha + \Delta\phi} &= \frac{1}{2} \cos(\alpha_A + \phi_A), \\ \lim_{\Delta\alpha - \Delta\phi \rightarrow 0} \frac{\cos(\alpha_B - \phi_B) - \cos(\alpha_A - \phi_A)}{(\Delta\alpha - \Delta\phi)^2} + \frac{\sin(\alpha_B - \phi_B)}{\Delta\alpha - \Delta\phi} &= \frac{1}{2} \cos(\alpha_A - \phi_A). \end{aligned} \quad (20)$$



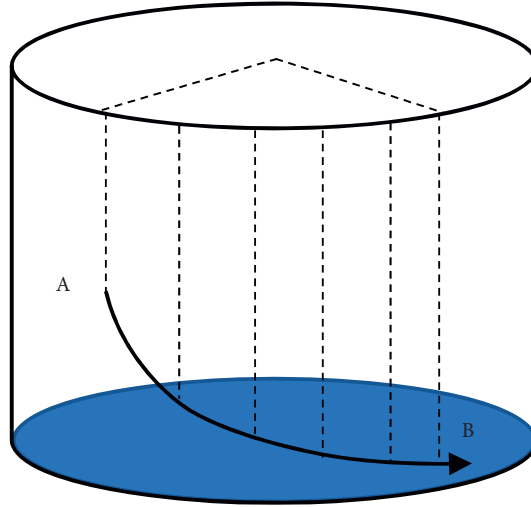


FIGURE 9: Schematic diagram of natural curve method.

(6) Solve well deviation angle  $\alpha_B$ .

There is the following relationship:

$$\Delta L = \frac{2\Delta N}{((\cos(\alpha_A + \phi_A) - \cos(\alpha_B + \phi_B))/(\Delta\alpha + \Delta\phi)) + ((\cos(\alpha_A - \phi_A) - \cos(\alpha_B - \phi_B))/(\Delta\alpha - \Delta\phi))}, \quad (21)$$

$$\Delta L = \frac{2\Delta E}{((\sin(\alpha_B - \phi_B) - \sin(\alpha_A - \phi_A))/(\Delta\alpha - \Delta\phi)) - ((\sin(\alpha_B + \phi_B) - \sin(\alpha_A + \phi_A))/(\Delta\alpha + \Delta\phi))}. \quad (22)$$

Equation (21) or equation (22) is selected to calculate the current  $\Delta L$  according to the absolute values of  $\Delta N$  and  $\Delta E$ . Equation (23) is derived from formula  $\Delta Z = c\Delta L \cos \bar{\alpha}$ , and the new well deviation angle  $\alpha_B^0$  can be calculated as

$$\alpha_B^0 = 2 \cos^{-1} \left( \frac{\Delta Z}{c\Delta L} \right) - \alpha_A. \quad (23)$$

If the error of  $\alpha_B^0$  between the two calculations is small enough (1.0E-6 is adopted in this paper), then the currents  $\alpha_B^0$  and  $\phi_B^0$  are the results of the final required solution, and the build-up slope and steering rate can be calculated by formulas (24) and (25). Otherwise, go to Step 4.

$$K_\alpha = \frac{\Delta\alpha}{\Delta L}, \quad (24)$$

$$K_\phi = \frac{\Delta\phi}{\Delta L}. \quad (25)$$

**3.3.3. Curve Components.** The curve component displays the curve parameter changes along the wellbore trajectory, such as torque, hook load, riser pressure, drilling time, rotating speed of turntable and other engineering data, or logging data, e.g., gamma ray. Each curve is displayed in a different color. Another curve style is displayed in the form of tube. The component represents the change in one parameter with the different colors of the color-map, and the change in the

thickness of the cylinder can represent the change in another parameter (such as well diameter).

**3.3.4. Formation Tops Component.** Formation tops are described using the following data, as shown in Table 2.

Formation tops are defined in the grid mode, including  $M$  columns and  $N$  rows grid vertex. It contains  $(M - 1) \times (N - 1)$  quadrangles, but some of them are not included in the formation tops model. The data only contain the vertexes of visible quadrangles, and each vertex includes north coordinate, east coordinate, altitude, column number, and row number. The formation tops display is mainly realized by creating a Three.js mesh. The method is as follows:

- (1) Create material based on color-map.
- (2) Calculate the depth range of all visible vertexes ( $D_{\min}, D_{\max}$ ).
- (3) Convert all vertex coordinates into a uniform coordinate array position.
- (4) According to the indexes position, each visible quadrangle is treated as two triangles to form an index array of vertex coordinates of the triangle indices.
- (5) Calculate material coordinates of vertexes  $(u, v) = ((Di - D_{\min}) / (D_{\max} - D_{\min}), 0)$ , and form material coordinate array— $uvs$ .

TABLE 2: Formation tops data format.

Map north (m)	Map east (m)	Altitude (m)	Column	Row
18496205.34	3106995.77	2020.84	315	1
18496254.58	3106987.09	2021.12	316	1
18496214.02	3107045.01	2027.11	315	2
18496263.26	3107036.33	2028.75	316	2
18496017.06	3107079.74	2028.87	307	3
18496066.30	3107071.06	2025.83	308	3
⋮	⋮	⋮	⋮	⋮

- (6) Create mesh object and compute normal vectors automatically.

**3.3.5. Tube Curve Component.** Tube curve component is used to display the tube curve along the path of borehole trajectory. The amplitude of the tube curve is displayed by different diameters and colors, which can show the change in a certain parameter along the borehole intuitively. Figure 10 is the basic principle of drawing the component. It divides the wellbore circumferentially into certain parts, such as 20, and takes values at certain depth intervals on the curve. Therefore, the whole barrel curve is transformed into drawing a series of quadrilateral, and each quadrilateral can be drawn with two triangles, as shown in Figure 10.

**3.3.6. Seismic Slice Component.** Generally, the amount of seismic data is very large. The seismic slice data near the drilling profile are processed and extracted by the pre-processing program and then saved as an image file. In this way, the seismic slice component only needs to load a small amount of data from the server to display the slice.

## 4. RESTful Data Service

It is necessary to develop corresponding background data services to provide data interface support for front-end application. In the past, traditional technologies such as Java and .NET were usually used to develop back-end data services. The development and deployment of services were complex, and it was difficult to combine them with advanced cloud and container technologies. For a long time, web service was the most mainstream way to build SOA web application, and it uses a complex and heavy SOAP message format based on XML. The concept of RE-presentational State Transfer (REST) was first proposed by Dr. Roy Thomas Fielding in his doctoral thesis in 2000 [16] as a way to provide interoperability between the computer systems in the Internet. It overcomes difficulties that are associated with conventional web services, realize more efficient data communications on the web. This paper presents a lightweight restful data service solution for the system. The scheme adopts microservice architecture and chooses Node.js—a lightweight development platform suitable for cloud platform [17]. TypeScript development language in Restify development framework is used for developing a highly available REST-style data service. Figure 11 shows the REST request basic process.

## 5. Real-Time Data Transmission Scheme

WITSML (Wellsite Information Transfer Standard Markup Language) is a standard markup language for well site information transmission based on XML, which can realize seamless data exchange between service companies and oil companies [18], but the WITSML standard is huge; the cost of implementing a WITSML server is high. Referring to some design ideas of WITSML, this paper designs a set of real-time data transmission scheme. Data transmission adopts the HTTP REST mode, and data objects are encapsulated in the JSON format. The scheme is simple to implement and can make up for the shortcomings of WITS0. The concept of timestamp is adopted in the design, which can not only acquire real-time data but also historical data, and can expand the interface of other static data at any time.

## 6. Data Service API

**6.1. Data Type Definition.** Visual drilling data are defined below, including well basic data, drilling trajectory data, logging data, layer interface data, geological slice data, and reservoir grid data.

**6.1.1. Trajectory Data.** The inclinometer data are used to describe the drilling trajectory, which comes from the drilling design or actual drilling. After receiving the data, the client needs to calculate  $(x, y, z)$  coordinates of each measuring point according to the drilling trajectory calculation method. These data are an array of objects, and the type definition of array elements is shown in Table 3.

Here are some examples of data:

```
{
  "md": 0, "dev": 0, "azi": 0 },
{ "md": 24.003, "dev": 0.44, "azi": 25.45 },
{ "md": 36.54, "dev": 0.79, "azi": 327.32 }
]
```

**6.1.2. Logging Curve Data.** Logging data are used to describe a set of measurement data along the drilling trajectory. The data consist of a series of measurement depth and measurement values. The data may come from LWD, MWD, geological logging, and logging. Continuous or tubular curves along drilling trajectory can be generated by logging data. Data-type definitions are shown in Table 4.

Here are some examples of data:

```
{
  "name": "GR",
```

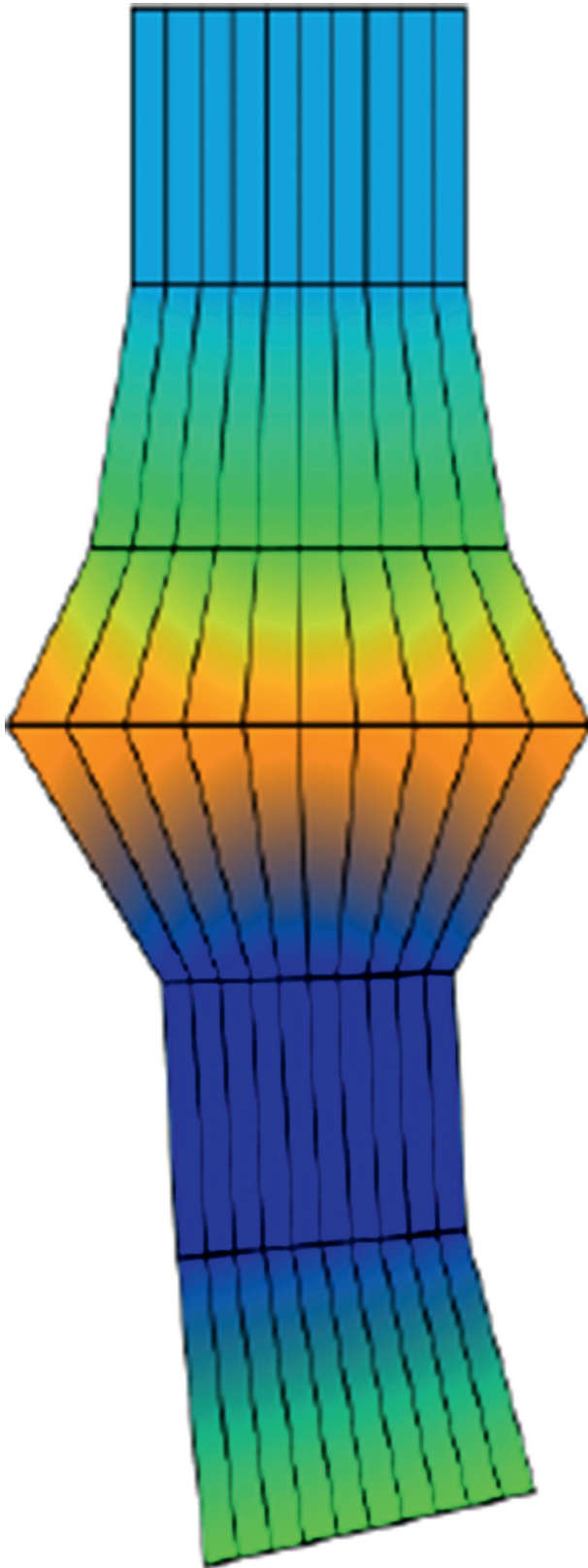


FIGURE 10: Tube curve schematic.

```

“min”: 0,
“max”: 400,
“depth”: [100, 100.5, 101, 101.5, 102],

```

```

“value”: [78.460, 81.242, 75.677, 80.046, 81.659]
}

```

6.1.3. *Formation Tops Data.* Formation tops data are used to describe the top interface of geological horizon. The data volume of layer interface is a two-dimensional array. The definition is shown in Table 5.

Here are some examples of data:

```

{
“nx”:201,
“ny”:101,
“vectors”: [
[18459025, 3269000, 2687.109375],
[18459050, 3269000, 2687.5],
[18459075, 3269000, 2687.109375],
[18459100, 3269000, 2687.890625],
[18459125, 3269000, 2686.71875],
[18459150, 3269000, 2687.5],
[18459175, 3269000, 2686.71875],
[18459200, 3269000, 2687.5],
...
]
}

```

6.1.4. *Geological Slice Data.* The geological data are often very large, such as seismic data volume of a block, which often reaches tens of GB. It is impossible and unnecessary to load the data of the whole data volume into the client for 3D display. In the process of drilling, usually only the geological information near the drilling trajectory is concerned. It is only necessary to display information of some facets to the 3D scene. The geological slice data format defined by this software is shown in Table 6.

Image is slice data. It is a base64 encoding string converted from PNG format pictures in the same format as HTML embedded pictures. Corners is the coordinates of the four corners of the slice  $\{[x_1, y_1, z_1], [x_2, y_2, z_2], [x_3, y_3, z_3], [x_4, y_4, z_4]\}$ .

Here is the sample data:

```

{
“name”: “slice1”,
“image”: “data:image/png; base64,iVBORw0A...”,
“corners”: [
{“x”: 4881500, “y”: 15617500, “z”: 825 },
{“x”: 4881500, “y”: 15621400, “z”: 825 },
{“x”: 4881500, “y”: 15621400, “z”: 2600 },
{“x”: 4881500, “y”: 15617500, “z”: 2600 }
]
}

```

6.2. *Service Interface Design.* According to the definition of drilling visualization data, the corresponding data interface is designed. To access data services, you only need to use HTTP interface address. The address format of the interface is `//host:port/api/interface/{parameter}`; for example, the host is local host, the port is 4000, “the block list,” and the interface name is “block,” no parameters, and the interface

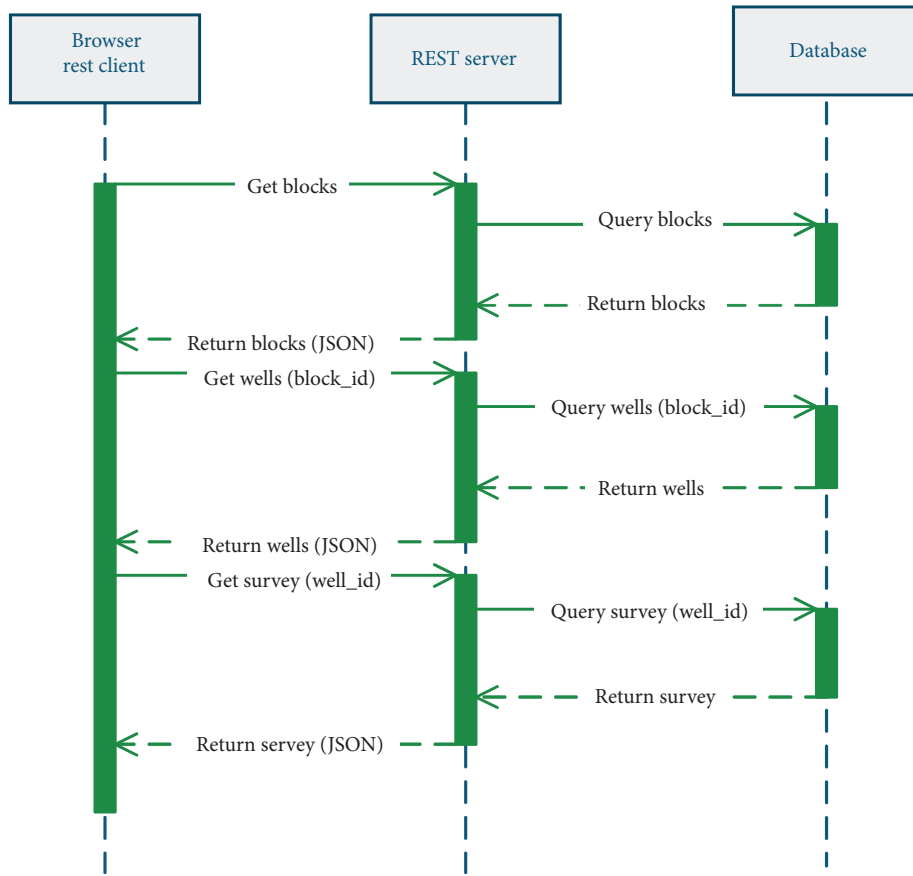


FIGURE 11: REST request basic process.

TABLE 3: Survey data.

Field	Type	Unit	Explanation
Md	Number	m	Measuring well depth
Dev	Number	deg	Well deviation angle
Azi	Number	deg	Azimuth

TABLE 4: Curve data.

Field	Type	Unit	Explanation
Name	String		Curve name
Min	Number		Left scale
Max	Number		Right scale
Color	String		Default color
Depth	Number[]	m	Depth of measurement
Value	Number[]		Curve value

address is `//localhost:4000/blocks`. The result is in the JSON format. Interface definitions are shown in Table 7.

The design has strong expansibility and can easily expand support for other drilling and completion and geological data by adding new data interface.

**6.3. Cross-Domain Problem.** When browsers want to access data service interfaces under different domain names, it needs to solve the problem of cross-domain access considering that data services may not be in the same server with

Web servers. The commonly used method is JSONP. JSONP (JSON with Padding) is a “usage mode” of JSON. For security reasons, browsers have a homologous policy. Generally speaking, web pages located at `server 1.example.com` cannot communicate with servers that are not `server 1.example.com`, but HTML `<script>` element is an exception. Using the open strategy of `<script>` elements, web pages can get JavaScript code generated dynamically from other sources. Through ingenious design, executing this code can dynamically generate JSON data needed by users. This mode of use is called JSONP. The service program is designed to support both JSON and JSONP. It only needs to add “`? Callback=jsonp`” after the interface address. The program returns JavaScript code in the JSONP mode; otherwise, it returns JSON data directly. The web page can get JavaScript code generated dynamically from other sources with the open strategy of `<script>` element. Through ingenious design, executing this code can generate JSON data dynamically required by users. This usage pattern is called JSONP.

**6.4. Service Deployment.** Service program can be deployed conveniently on the cloud server using PM2 and NodeJS, and the update and execution of the program can be monitored to ensure the uninterrupted operation and hot update of the program through the powerful process management function of PM2 [19, 20]. PM2 is a Node.js production environment process management tool, which

TABLE 5: Formation tops data.

Field	Type	Unit	Explanation
Nx	Number		X-direction points
Ny	Number		Y-direction points
Vectors	Number[3]	m	nx * ny group (x, y, z) coordinates

TABLE 6: Geological slice data type.

Field	Type	Unit	Explanation
Name	String		Slice name
Image	String		Slice data
Corners	Vector3[4]	m	Slice corner coordinate data

TABLE 7: Service interface definition.

Interface name	Interface address
Block list	Blocks
Well list	Wells/{block_id}
Formation tops list	Surfaces/{block_id}
Seismic slice list	Slices/{block_id}
Trajectory data	Survey/{well_id}
Curve list	Curves/{well_id}
Events list	Events/{well_id}
Curve data	Curve/{curve_id}
Formation tops data	Surface/{surface_id}

has built-in load balancing function, especially for micro-service-based applications. The use of PM2 guarantees the uninterrupted operation of Node.js services. Once a program error occurs, PM2 will automatically restart the process, which is critical for data service program. The REST-style data service developed by using Node.js development platform is a lightweight technology, which is more suitable for cloud platform at present. Through JSONP, cross-domain access is realized. PM2 is used to deploy the service program. The drilling 3D visualization system developed by this service has the characteristics of high reliability, reliability, and easy expansion. With the advantage of static type of advanced TypeScript language, the software can eliminate most potential errors in the development stage and solve the problem of weak type and poor reliability of JavaScript language. By compiling to JavaScript execution, it can give full play to the advantages of Node.js in cross-platform and lightweight. The lightweight data service solution described in this paper is completely based on open-source technology, does not rely on any company's proprietary technology and platform, can freely choose the server deployment platform, and can be combined with advanced cloud and container technology.

## 7. Applications and Discussion

The software is now installed/used in the RTOC is shown in Figures 12 and 13, for example, block 291, which contains 11 wells of carbonate rock, including 4 vertical wells, 1 directional well, and 6 horizontal wells, among which W291-H9 is the latest horizontal well and the target formation is Ordovician. The drilling trajectory, logging data,

and complex and accidents and seismic slice data along the well trajectory of the target formation are loaded into the database.

The software can display the complex and accidents of drilling in the block, the seismic characteristics of the target layer, the tops of the target formation, drilling trajectories, and LWD data and can be used for auxiliary analysis and decision-making. The drilling and geological experts can not only use the workstation of the Office but also use the mobile equipment to view the data and graphics.

Due to the consistency and efficiency of WebGL on various platforms, the software has good performance and consistency on the main platforms, including Windows, Linux, macOS, Android, and iOS, and the responsive layout can adapt well for the different resolutions of desktop devices and mobile devices; for small mobile screen, the software will automatically hide the left tree menu, making the software more practical. The goal of the development is achieved.

In Figure 14, drilling 3D scene is visualized on iPad, Android, and iPhone different devices. Compared with non-cross-platform software, WebDrillingViz provides a powerful fundamental tool for the drilling visualization applications. Different from current studies on visualization of 3D drilling visualization, WebDrillingViz is lightweighted, cross-platform, and open. We can anticipate that WebDrillingViz can be rewarding to more applications during the life-cycle drilling. At the end, it is verified that WebDrillingViz is efficient, compatible with mainstream devices through extensive real projects' drilling and geological data, and the rest interface of data service software fully meets the actual needs in terms of function, performance, and



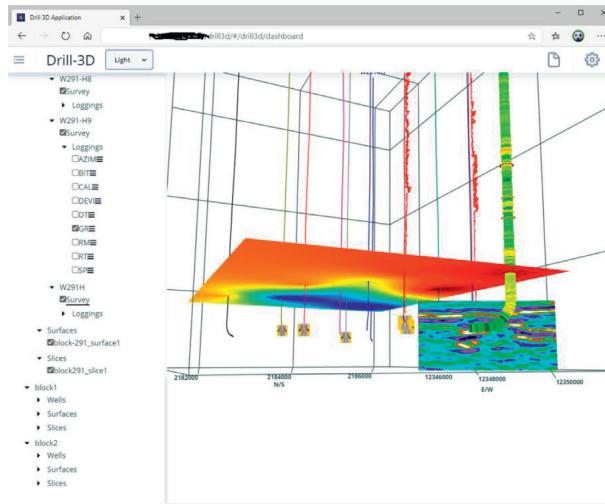


FIGURE 12: Drilling 3D scene: well tracks, surface, and loggings on Windows.

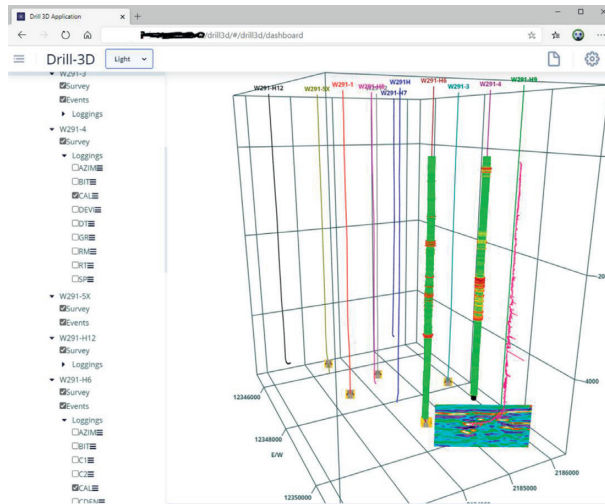


FIGURE 13: Drilling 3D scene: well tracks, loggings, and events.

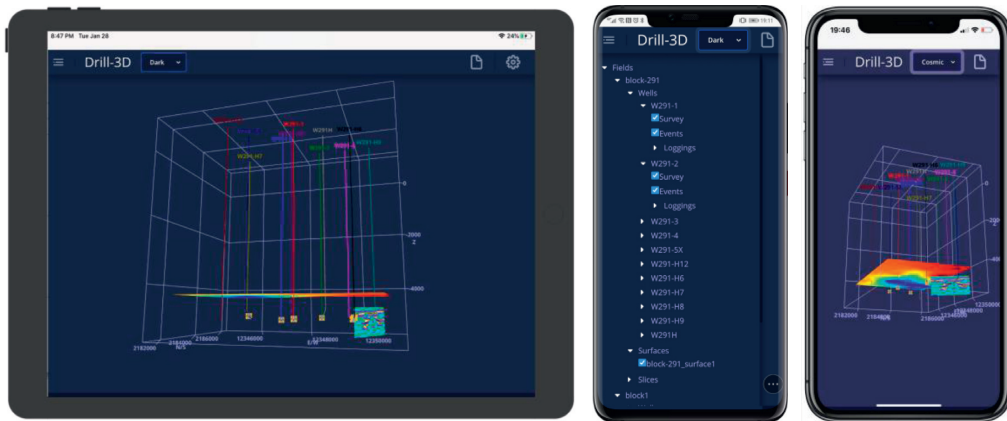


FIGURE 14: Drilling 3D Scene on iPad, Android, and iPhone.

reliability, which proves that the architecture proposed in this paper is a very valuable lightweight solution.

## 8. Conclusions

This paper introduces the structure and development of 3D visualization software for drilling engineering and geology. The software is based on WebGL and developed with TypeScript, an open-source framework. Combined with the real-time transmission scheme proposed, a set of low-cost and practical real-time 3D visualization monitoring software is provided for RTOC. The conclusions can be summarized as follows:

- (1) The use of native Web technologies allows detailed renderings of the drilling and geological data with no additional software to install. 3D drilling scene renderings are processed entirely within the modern Web browser without a significant tradeoff in performance. It is firstly suggested that an online 3D drilling visualization system has to meet three basic requirements: lightweighted, cross-platform, and open. Based on these basic rules, we developed a novel online 3D drilling visualization system based on HTML5 and WebGL, termed as WebDrillingViz. It is now installed in the RTOC, interfacing with other software systems. This software will enable decision-makers to have better insights into the status of the well and formation surrounding the well and thus makes better and quicker decisions.
- (2) WebDrillingViz integrates responsive design and high-performance back-end services, using the latest technologies, such as NodeJS, Angular, and Bootstrap frameworks with excellent compatibility. The front-end components can also quickly be embedding into other web applications and the back-end efficient service based on NodeJS platform. The software is completely implemented in TypeScript language on both sides, and the whole development cycle reduced significantly because using same language. The REST-style data service is a lightweight technology, which is more suitable for cloud platform at present. Through JSONP, cross-domain access is realized. PM2 is used to deploy the service program. The service has the characteristics of high reliability, reliability, and easy expansion.

## Data Availability

The data used to support the findings of this study are available from the first author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors would like to thank China University of Petroleum (Beijing) for the support during this research and permission to publish these results. This work was financially

supported by the National Natural Science Foundation Project (Grant no. 51374222), National Major Project (Grant no. 2017ZX05032004-002), National Key Basic Research and Development Program (Grant no. 2015CB250905), and CNPC's Major Scientific and Technological Project (Grant no. 2017E-0405).

## References

- [1] D. H. Kaminiski, N. M. Pellerin, and J. H. Williams, "A new data integration and work process system for providing online real-time drilling collaboration," in *Proceedings of the European Petroleum Conference*, Aberdeen, UK, October 2002.
- [2] A. L. F. Madaleno, S. L. S. Neto, L. A. Dos Santos, and C. A. L. De Oliveira, "Operational and safety improvements of applying real-time analytics in a drilling contractor RTOC," in *Proceedings of the Annual Offshore Technology Conference*, Houston, TX, USA, May 2018.
- [3] L. J. Ursem, J. H. Williams, N. M. Pellerin, and D. H. Kaminski, "Real time operations centers; the people aspects of drilling decision making," in *Proceedings of the Drilling Conference*, Amsterdam, The Netherlands, February 2003.
- [4] G. A. Dorn, K. Touyinhthiphonexay, J. Bradley, and A. Jamieson, "Immersive 3-D visualization applied to drilling planning," *The Leading Edge*, vol. 20, no. 12, pp. 1389–1392, 2001.
- [5] J. Holt, W. J. Wright, H. Nicholson, A. Kuhn-De-Chizelle, and C. Ramshorn, "Mungo field: improved communication through 3D visualization of drilling problems," in *Proceedings of the SPE/AAPG Western Regional Meetings*, vol. 53, Long Beach, CA, USA, June 2000.
- [6] R. Rommetveit, K. S. Bjørkevold, S. I. Ødegård, M. Herbert, and G. W. Halsey, "Automatic real-time drilling supervision, simulation, 3D visualization and diagnosis on ekofisk," in *Proceedings of the SPE/IADC Drilling Conference* Orlando, FL, USA, March 2008.
- [7] R. Rommetveit, K. S. Bjørkevold, S. I. Ødegård, O. Sandve, B. Larsen, and M. Herbert, "EDrilling: linking advanced models and 3D visualization to drilling control systems in real time," in *Proceedings of the Offshore Mediterranean Conference and Exhibition 2007, OMC 2007*, Ravenna, Italy, March 2007.
- [8] W. C. Sanstrom and M. J. Hawkins, "Perceiving drilling learning through visualization," in *Proceedings of the IADC/SPE Asia Pacific Drilling Technology Conference, APDT*, Limerick, Ireland, September 2000.
- [9] P. Song, L. Shi, Y. Zhou, Q. Zhao, H. Jiang, and P. Cui, "Study and applications on integrated drilling engineering software for drilling engineering design and real-time optimization (Russian)," in *Proceedings of the Society of Petroleum Engineers - SPE Annual Caspian Technical Conference and Exhibition*, Astana, Kazakhstan, November 2014.
- [10] W. Phillips, "Interactive web-based 3D wellbore viewer enables collaborative analysis," *SPE Production & Operations*, vol. 33, no. 02, pp. 345–352, 2018.
- [11] X. Liu, N. Xie, K. Tang, and J. Jia, "Lightweighting for Web3D visualization of large-scale BIM scenes in real-time," *Graphical Models*, vol. 88, pp. 40–56, 2016.
- [12] I. Morozov, G. Chubak, and S. Blyth, "Interactive 3D/2D visualization for geophysical data processing and interpretation," *Computers and Geosciences*, vol. 35, no. 7, pp. 1397–1408, 2009.

- [13] G. Bierman, M. Abadi, and M. Torgersen, *Understanding TypeScript*, vol. 8586, pp. 257–281, ECOOP 2014 - Object-Oriented Programming, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014.
- [14] J. Dickey, *Write Modern Web Apps with the Mean Stack: Mongo, Express, AngularJS, and Node.js (Develop and Design)*, Peachpit Press, Berkeley, CA, USA, 1st edition, 2014.
- [15] T. A. Walsh, P. McMinn, and G. M. Kapfhammer, “Automatic detection of potential layout faults following changes to responsive web pages (N),” in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering*, Lincoln, NE, USA, November 2015.
- [16] R. T. Fielding and R. N. Taylor, “Principled design of the modern web architecture,” *ACM Transactions on Internet Technology*, vol. 2, no. 2, pp. 115–150, 2002.
- [17] M. Cantelon, M. Harter, T. J. Holowaychuk, and N. Rajlich, *Node.js in Action*, American: Manning Publications, Shelter Island, NY, USA, 2013.
- [18] M. Khudiri, J. James, M. Amer, B. Otaibi, M. Nefai, and J. Curtis, “The integration of drilling sensor real-time data with drilling reporting data at Saudi Aramco using WITSML,” in *Proceedings of the Society of Petroleum Engineers - SPE Intelligent Energy International 2014*, Utrecht, The Netherlands, April 2014.
- [19] X. Chenjie, Z. Xiang, P. Xin, and Z. Wenyun, “Microservice system oriented runtime deployment optimization,” *Computer Application Software*, vol. 35, no. 10, pp. 85–93, 2019.
- [20] L. Yongyi, “Research on cross-domain request scheme for large distributed web system,” *J. Chang. Univ.* vol. 35, no. 2, pp. 54–56, 2018.

## Research Article

# A Validated Study of a Modified Shallow Water Model for Strong Cyclonic Motions and Their Structures in a Rotating Tank

Hung-Cheng Chen <sup>1</sup>, Jai-Houng Leu <sup>1</sup>, Yong Liu <sup>1</sup>, He-Sheng Xie <sup>1</sup>,  
and Qiang Chen <sup>2</sup>

<sup>1</sup>School of Intelligent Manufacturing, Shandong Polytechnic, Jinan 250104, China

<sup>2</sup>School of Railway, Shandong Polytechnic, Jinan 250104, China

Correspondence should be addressed to Jai-Houng Leu; jahonleu@yahoo.com.tw

Received 30 January 2021; Revised 29 March 2021; Accepted 20 April 2021; Published 3 May 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Hung-Cheng Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A joint theoretical and numerical study was carried out to investigate the fluid dynamical aspect of the motion of a vortex generated in a rotating tank with a sloping bottom. This study aims at understanding the evolution of strong cyclonic motions on a  $\beta$ -plane in the Northern Hemisphere. The strong cyclonic vortices were characterized by four nondimensional parameters which were derived through a scale analysis of the depth variations of fluid. By simplifying the model flow field and the prototype flow field, respectively, through the conservation of potential vorticity, two sets of dynamic similarity conditions are derived. This study proposed a sophisticated modified shallow water model (MSWM) to investigate the flow features of such strong vortices. A detailed numerical calculation adopted by multidimensional positive definite advection transport algorithm (MPDATA) was carried out to validate those effects considered in the MSWM model, including sloping bottom, parabolic free surface deformation, and viscous dissipation. Close agreements were found between the experimental and numerical results, including the streamlines patterns and the vortex trajectory. Comprehensive simulations for strong cyclonic vortices over different sloping bottoms were investigated to understand the impact of planetary  $\beta$  effect on vortex. The results calculated by MSWM demonstrate a variety of flow features of interactions between the primary vortex and induced secondary Rossby wave wakes that were essential and prominent in environmental geophysical flows.

## 1. Introduction

A joint theoretical and numerical study is used to investigate the fluid dynamical aspect of the motion of a vortex generated in a rotating tank with a sloping bottom. This study is motivated by getting an insight of the evolution of a strong barotropic cyclone on a  $\beta$ -plane in the Northern Hemisphere. The dynamics of barotropic vortices on a  $\beta$ -plane have been studied intensively over the past several decades through analytical, numerical, and laboratory investigations [1–22]. Both analytical and numerical studies have shown that (a) a  $\beta$ -gyre develops in the initial time [1–4], (b) the vortex intensity or structure changed during the vortex propagation [5–9], and (c) a more intense quasi-geostrophic vortex may evolve for a longer time [10]. These dynamic features are basically examined by the advancement of the

fluid cyclonically around the vortex core. It is considered to be a nonlinear, self-propelling motion or the interaction between  $\beta$ -gyre and vortex with time evolution under the effect of Rossby wave radiation.

The above phenomena can be further elucidated from the viewpoint of point vortex dynamics. In many studies, Resnik et al. [10] presented the theory of long-term evolution of singular or point vortex based on the conservation of vortex energy and enstrophy, which is of great significance. They suggest that the evolution of intense vortices on the  $\beta$ -plane can be divided into three stages. In the first stage, the beta-gyres generated by the near-field radiation of Rossby wave and the nonlinear advection of the vortex make the cyclone (anticyclone) move to the northwest (southwest). The study of vortex dynamics at this stage has attracted much attention, including laboratory experiments [11–13],

numerical simulation [3, 4, 18, 19], and theoretical analysis [5–7, 9]. In the second stage, the amplitude and velocity of the vortex are gradually slowed down by other azimuthal harmonics generated by Rossby wave radiation and non-linearity. In the last third stage, the amplitude of the vortex decreases to the background value under the continuous influence of Rossby wave radiation. The above three stages correspond to three characteristic scales of time. They are advection time scale, wave time scale, and distortion time scale. For a detailed review of the vortex dynamics of singular vortices on the  $\beta$ -plane, the reader is referred to the work of Resnik & Kravtsov [20].

In the laboratory experiments, Firing and Beardsley [11] conducted the first laboratory experiment of a barotropic eddy on a  $\beta$ -plane. By generating the eddy with a piston mechanism on an inclined plate with a gentle slope ( $s_y \approx 0.1$ ), these authors observed that the eddy evolved to form a dipole in a short time. Their purpose was to validate the initial northwestward translation of the vortex on a  $\beta$ -plane which was predicted earlier by Adem [1]. To generate a long-lived isolated eddy, Takematsu and Kita [12] applied a locally cooling method to create a monopolar vortex on an inclined bottom in a rotating fluid. The resulting stratified eddy was inherently stable and migrated to the northwest as a recognizable structure like a Gulf Stream ring. Masuda et al. [13] carried out a joint laboratory and numerical study for a strong isolated eddy on an inclined plate with a steep slope ( $s_y \approx 0.33$ ). Satisfactory agreements were obtained on the flow patterns between the laboratory and numerical experiments. They adopted a two-dimensional quasi-geostrophic vorticity equation (QGVE) to numerically simulate the flow specifically for a small Rossby number vortex.

The initial vortex conditions for QGVE simulation were carefully estimated by measuring water column shrinking or stretching in a sink/source vortex generator. The results showed that the vortex evolved into a larger main eddy, and several secondary eddies displayed significant influence of the strong  $\beta$  effect on the cyclonic vortex.

For a long-lived vortex translating on a gentle-slope ( $s_y \approx 0.13$ ) bottom, Carnevale et al. [14] generated isolated vortices using stirred/sink methods and numerically solved a QGVE to simulate their motions. They interpolated both by a Gaussian-type distribution and a Rankine-type distribution as the initial conditions of numerical integrations. The results showed that the essential mechanism of both the stirring-induced and the sink-induced vortices is inviscid, quasi-geostrophic. Flór and Eames [15] investigated the dynamics of a cyclonic monopolar vortex on a topographic  $\beta$ -plane by laboratory experiments and theoretical analysis. They systematically measured the vortex distributions generated by the stirred or the suction method and characterized the initial distribution in terms of a radius  $R_m$ , the maximum azimuthal velocity  $v_\theta$ , and a dimensionless parameter  $\alpha$  which describes the steepness of the velocity profile.

Recently, Chen et al. [22] generated a strong barotropic vortex of large vortex Rossby number  $Ro_v \sim O(1)$  in a rotating tank with a gentle sloping bottom ( $s_y \approx 0.0538$ ) to simulate the movement of a hurricane-like vortex on a

$\beta$ -plane. The cyclonic vortex was generated by a rotating cylinder in a thin-walled hollow cylinder in parallel to the axis of the rotating tank. The most remarkable feature of this study is the use of gradient wind balance vortex model to capture vortex structure. The radial distribution of depression depth on the vortex surface was clearly visualized by the illumination of a laser light sheet perpendicular to the vortex centre. The parameters of the GWB model can be fitted by using the measurement depression depth. The corresponding tangential velocity distribution of the vortex can be calculated accordingly. In their experiment results, the vortices with strong strength (the vortex Rossby number  $Ro_v$  is about 4.3) will produce weak Rossby wakes during their motion. The vortex with weaker strength (the vortex Rossby number  $Ro_v$  is about 1.8) maintains a clean single vortex structure during its movement. These phenomena are mainly related to the relative importance of the vortex  $\beta$  effect and the planetary  $\beta$  effect.

The present paper is the first of a series of works following Chen et al. [22] and is mainly devoted to validation of a proposed modified shallow water model (MSWM) by a joint experimental and numerical study. To improve the ability of capturing the flow features, a dissipative momentum flux term and an effective gravitation term were adopted in MSWM, while an artificial viscosity term was added to ensure numerical stability [23, 24]. We solved this modified shallow water model using a multidimensional positive definite advection transport algorithm (MPDATA) which was proposed and has been well known for simulations of geophysical flow by Smolarkiewicz and his colleagues for decades [25–28]. This study also proposes a theoretical analysis of dynamical similarity conditions to mimic the hurricane-like vortices on a  $\beta$ -plane by vortices generating in a rotating tank with a gently sloping bottom. The paper is organized as follows. First, Section 2 derived the dynamical similarity conditions for the model rotating tank experiment and the prototype hurricane-like motion by the potential vorticity conservation. Section 3 presents the numerical calculations of the strong cyclone motions in the rotating tank by a modified shallow water model. The simulation results by MSWM using the fitted parameters for gradient-wind-balance (GWB) model are shown in Section 4. We also investigate the long-term evolutions of strong cyclonic motion on different bottom slopes. Finally, conclusions are presented in Section 5.

## 2. Governing Principle and Similarity Laws

Let us now proceed with the governing principles of phenomena under consideration. Two important nondimensional parameters concerning the barotropic cyclonic motion must be first introduced. Let  $V_m$  and  $R_m$  be the characteristic velocity and length scales for cyclonic motion, respectively. We define the vortex Rossby number by

$$Ro_v = \frac{V_m}{fR_m}, \quad (1)$$

where  $f$  is the planetary vorticity. The other is the vortex planetary  $\beta$  parameter, defined by



$$\beta_0^* = \frac{\beta R_m^2}{V_m} \quad (2)$$

where  $\beta$  is planetary vorticity gradient. In this study, we are interested in those cyclonic motions which are  $\text{Ro}_v \sim O(1)$  and  $\beta_0^* \sim O(10^{-2} - 10^{-3})$ . In our terminology, such cyclones are called strong ( $\text{Ro}_v \sim O(1)$ ) and intense ( $\beta_0^* \sim O(10^{-2} - 10^{-3})$ ) cyclones. The cyclones are said to be strong because the characteristic relative vorticity  $V_m/R_m$  of the cyclone is of the same order of magnitude with the planetary vorticity  $f$ . The cyclones are said to be intense because variation of the vorticity across the cyclone due to the relative vorticity gradient is much larger than that due to the planetary vorticity gradient in the vicinity of the cyclonic core structure [10]. For example, a tropical cyclone with moderate strength with a maximum tangential speed  $V_m = 40$  m/s and a corresponding radius  $R_m = 150$  km results in  $\text{Ro}_v \sim 5.34$  and  $\beta_0^* \sim 0.0121$ .

For a rotating, shallow water flow, the governing principle is the law of conservation of potential vorticity (PV) [29].

$$\frac{D\Pi}{Dt} = 0, \quad (3)$$

where potential vorticity  $\Pi$  is defined as  $\Pi = (f + \zeta)/H$  where  $H$  is the fluid layer depth.

**2.1. PV Conservation for the Model Problem.** As shown in Figure 1(a), fluid layer depth  $H(x, y, t)$  can be expressed as  $H = H_0 + \eta - h_B$ , where  $H_0$  is the unperturbed depth,  $\eta(x, y, t)$  is the free-surface deviation due to the motion, and  $h_B(x, y)$  represents the bottom topography. The local Cartesian coordinates  $x, y$ , and  $z$  point horizontally inward, westward, and vertically upward, respectively. We can choose a gentle-slope (denoted as  $s_y$ ) bottom topography in a rotating fluid confined in a tank spinning at a constant speed  $\Omega = f_0/2$ , where  $f_0$  is the background vorticity of the rotating fluid. The flow dynamical features in the tank experiment can be understood by the law of conservation of potential vorticity (PV) by assuming small variations of  $\eta$  and  $h_B = s_y y$  with respect to  $H_0$ ; that is,  $\eta/H_0 \ll 1$ ,  $s_y y/H_0 \ll 1$ . This allows (3) to be written as

$$\frac{D}{Dt} \left( \frac{f_0}{H_0} \left( 1 - \frac{\eta}{H_0} + \frac{s_y y}{H_0} \right) + \frac{\zeta}{H_0} \left( 1 - \frac{\eta}{H_0} + \frac{s_y y}{H_0} \right) \right) = 0. \quad (4)$$

Next, we will give an appropriate choice on scaling the cyclonic motion so that the magnitudes of the nondimensional variables are of order unity. Take the maximum tangential speed of the cyclone  $V_m$  as the reference velocity,  $\zeta_m = V_m/R_m$  as the reference vorticity, and  $\zeta_m^{-1}$  as the reference time (the vortex turnaround time). In addition, we choose the maximum vortex depression  $\eta_v$  as the reference free-surface deviation, and thus we have the set of nondimensional variables defined by  $t^* = t/\zeta_m^{-1}$ ,  $y^* = y/R_m$ ,  $\eta^* = \eta/\eta_v$ , and  $\zeta^* = \zeta/\zeta_m$ . Since  $f_0/H_0$  is constant and makes no contribution, we can divide equation (4) by  $\zeta_m^2/H_0$  to obtain

$$\frac{D}{Dt^*} (\beta_y^* y^* - \beta_v^* \eta^* + \zeta^* (1 + s_y^* y^* - s_v^* \eta^*)) = 0. \quad (5)$$

Equation (5) is a nondimensional PV conservation law of a strong cyclonic vortex translating on a sloping bottom in a rotating tank. There are four nondimensional parameters involved from different sources of layer depth variation. They are (i) the bottom slope parameter  $s_y^* = s_y R_m/H_0$ , where  $s_y$  represents the bottom slope, and (ii) the bottom  $\beta$ -parameter,

$$\beta_y^* = \frac{f_0 s_y R_m^2}{V_m H_0} = \frac{s_y^*}{\text{Ro}_v}, \quad (6)$$

(iii) the vortex slope parameter  $s_v^* = s_v R_m/H_0$ , where a characteristic vortex slope is defined as  $s_v = \eta_v/R_m$ , and (iv) the vortex  $\beta$ -parameter

$$\beta_v^* = \frac{f_0 s_v R_m^2}{V_m H_0} = \frac{s_v^*}{\text{Ro}_v}. \quad (7)$$

The physical meaning of (5) can be understood by applying it far away from the cyclonic structure where both the nondimensional relative vorticity and surface depression are small; that is,  $\zeta^* \sim O(\epsilon)$ ,  $\eta^* \sim O(\epsilon)$ . Then, we have

$$\frac{D}{Dt^*} (\beta_y^* y^* + \zeta^*) = 0. \quad (8)$$

Equation (8) is the traditional PV conservation law which explains the generation of Rossby wave in a depth-varying fluid layer. On the other hand, we can rearrange (5) as

$$\frac{D}{Dt^*} (\beta_y^* y^* + (\zeta_e^*)_t) = 0, \quad (9)$$

where  $(\zeta_e^*)_t$  is the equivalent relative vorticity, defined by

$$(\zeta_e^*)_t = \zeta^* (1 + s_y^* y^* - s_v^* \eta^*) - \beta_v^* \eta^*. \quad (10)$$

Here, the subscript  $t$  denotes a tank experiment. Equations (9) and (10) state that the strong cyclonic motion moving towards the shallower region will decrease their equivalent relative vorticity.

**2.2. PV Conservation for Prototypical Problem.** As shown in Figure 1(b), we now consider the potential vorticity conservation for strong cyclonic motion on a  $\beta$ -plane with a constant layer depth. The planetary vorticity  $f$  can be linearized by the  $\beta$ -plane approximation; that is,  $f \sim f_0 + \beta_0 y$ , where  $f_0$  and  $\beta_0$  are constants, and  $y$  is the local northward Cartesian coordinate. Considering small relative variation of  $\eta$  and assuming that no bottom topography is present, that is,  $\eta/H_0 \ll 1$ ,  $h_B \approx 0$ , we can rewrite (4) as

$$\frac{D}{Dt} \left( \frac{f_0}{H_0} \left( 1 - \frac{\eta}{H_0} \right) + \frac{\beta_0 y}{H_0} + \frac{\zeta}{H_0} \left( 1 - \frac{\eta}{H_0} \right) \right) = 0, \quad (11)$$

for a strong cyclonic vortex. Next, we will give an appropriate choice on scaling the cyclone motion so that the magnitudes of the nondimensional variables are of order

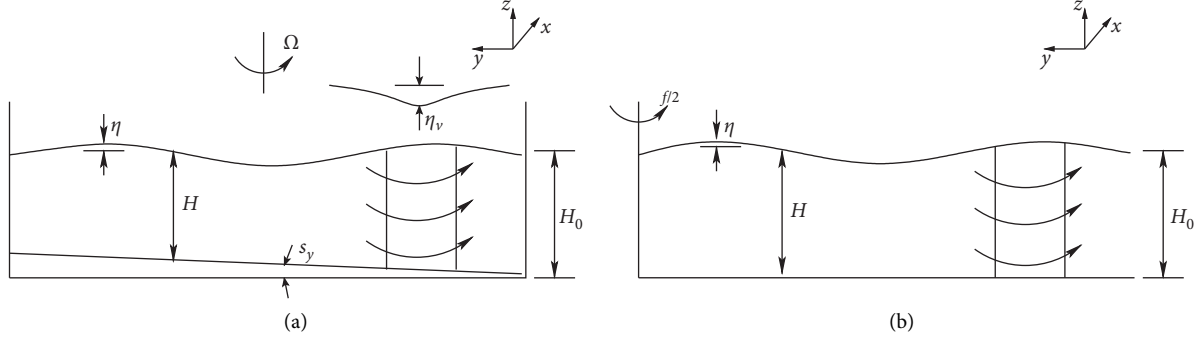


FIGURE 1: (a) Schematic of a cyclonic vortex motion in a rotating tank with a sloping bottom. (b) Schematic of a cyclonic motion on a  $\beta$ -plane.

unity. The reference scales are taken identically as those in the tank experiment. Since  $f_0/H_0$  is constant and makes no contribution, we can divide (11) by  $\zeta_m^2/H_0$  and yield

$$\frac{D}{Dt^*} (-\beta_v^* \eta^* + \beta_0^* y^* + \zeta^* (1 - s_v^* \eta^*)) = 0 \quad (12)$$

where the nondimensional variables are defined the same as in the rotating tank experiment. Equation (12) is a nondimensional PV conservation law of strong cyclonic motions translating on a  $\beta$ -plane with no topographic feature.

Notably, (12) bears a close physical explanation on the potential vorticity dynamics as (5). That is, when the fluid particles move northward (with higher  $f$ ), the relative vorticity decreases and the surface depression varies simultaneously. We can also define an equivalent relative vorticity for the prototype flow field as

$$(\zeta_e^*)_f = \zeta^* (1 - s_v^* \eta^*) - \beta_v^* \eta^*. \quad (13)$$

Although the explanations of the change of equivalent relative vorticity of (5) and (12) are very similar, the mutual adjustment of the relative vorticity and the surface depression of the vortex in the model experiment is more complicated than that in the prototypical flow. An additional term  $\zeta^* (s_y^* y^*)$  is involved in (10) and this makes it difficult to decouple the effects contributed by the northward movement or the surface depression for the change of relative vorticity in the strong cyclonic motion.

**2.3. Dynamical Similarity Conditions.** In order to derive the dynamical similarity conditions, we first apply (5) and (12) far away from the cyclonic core region where the nondimensional quantities  $\zeta^*$  and  $\eta^*$  are small. Comparing those terms on the left-hand sides of (5) and (12), we have the first similarity condition

$$\beta_y^* = \beta_0^*. \quad (14)$$

Then, we apply (5) and (12) in the core region of cyclonic structure with  $\zeta^* \approx O(1)$ . By evaluating both equations on the specific locations that yield unity  $(\zeta^*)_m$  and  $(\zeta^*)_p$ , where the subscripts  $m$  and  $p$  denote, respectively, the model

(rotating tank) and prototypical (field) experiment, also, we assume that the cyclonic vortex is generated on the initial latitude, that is,  $(y^*)_m \sim 0$  and  $(y^*)_p \sim 0$ . Therefore, comparing those terms on the left-hand sides of (5) and (12), we obtain the second similarity condition

$$((\text{Ro}^{-1} + 1)s_v^*)_m = ((\text{Ro}^{-1} + 1)s_v^*)_p. \quad (15)$$

Equations (14) and (15) build a dynamical connection from model experiment and the prototypical flow by sharing the similar dynamics according to the conservation of potential vorticity.

### 3. Numerical Calculations

**3.1. Modified Shallow Water Model.** As shown in Figure 1(a), let  $\vec{u} = (u, v)$  be the horizontal velocity, let  $H$  be the fluid layer depth, and let  $h_B$  be the elevation of the bottom topography, respectively. In order to model the viscous friction in the rotating shallow water flow, the present study incorporates an additional term  $\nu \nabla \cdot \nabla \vec{u}$  in the standard shallow water model as suggested in [23, 24].

$$\frac{\partial H}{\partial t} + \nabla \cdot (\vec{u} H) = 0, \quad (16)$$

$$\frac{D\vec{u}}{Dt} + g_e \nabla (h_B + H) + f \vec{k} \times \vec{u} = \nu \nabla \cdot \nabla \vec{u} \quad (17)$$

where the effective gravity  $g_e$  is defined as [30]

$$g_e = \sqrt{g^2 + \frac{f^4 \rho_r^2}{4}}. \quad (18)$$

In (18),  $g_e$  is defined as the net acceleration directing perpendicular to the free surface and  $\rho_r$  is the distance of the fluid particle to the rotation axis; and  $\nu$  denotes the kinematic viscosity coefficient of the working fluid (in the present study,  $\nu = 1 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$  for water at  $20^\circ$ ). By choosing the same set of reference scales of the vortex motion as in the last section, we can recast (16) and (17) into the following modified shallow water model in flux form:

$$\frac{\partial H^*}{\partial t^*} + \nabla^* \cdot (\vec{u}^* H^*) = \hat{\nu}_H \nabla^* \cdot \nabla^* H^*, \quad (19)$$

$$\begin{aligned} \frac{\partial H^* \vec{u}^*}{\partial t^*} + \nabla^* \cdot (\vec{u}^* H^* \vec{u}^*) &= -\text{Ro}_v^{-1} \vec{k} \times (H^* \vec{u}^*) \\ &- \text{Fr}^{-2} H^* \nabla^* (h_B^* + H^*) + \text{Re}^{-1} \nabla^* \cdot H^* \nabla^* \vec{u}^*. \end{aligned} \quad (20)$$

In (19), an artificial viscosity term is incorporated to ensure the numerical stability. It is defined as  $\hat{\nu}_H = 1 \times 10^{-8} \Delta t^{-1} \Delta x$ , where  $\Delta t$  and  $\Delta x$  are time increment and mesh spacing, respectively. It is noted that, in (20), there are three nondimensional parameters. The vortex Rossby number is defined in (1), the vortex Froude number is defined as  $\text{Fr} = V_m / \sqrt{g_e H_0}$ , and the vortex Reynolds number is defined by  $\text{Re} = R_m V_m / \nu$ . The transport momentum variables  $U_{i,j} = (H^* u^*)_{i,j}$  and  $V_{i,j} = (H^* v^*)_{i,j}$  and layer-depth  $H_{i,j}^*$  are given on a rectangular grid, while the advective velocity components  $u_{i+1/2,j}$  and  $v_{i,j+1/2}$  are staggered by a one-half grid spacing. Here  $(i, j)$  denotes the location in the grid and  $\vec{V} = (U, V)$  denotes the momentum variable. The MSWM were discretized on an Arakawa-C staggered grid. The forcing terms in MSWM are discretized by following the suggestions in the works of Schär and Smith [23, 24].

**3.2. MPDATA Scheme.** The discretized equations of the MSWM were solved by the MPDATA (multidimensional positive definite advection transport algorithm) which was proposed by Smolarkiewicz and his colleagues [25–28]. MPDATA is a procedure that iteratively approximates the advection equation, which uses a donor cell approximation to compensate the truncation error of the original donor cell scheme. This step may be repeated an arbitrary number of times, leading to successively more accurate solutions of the advection equation. Concerning the contributions of forcing terms on the transport variables, we have incorporated the MPDATA scheme by a Strang-splitting method and to implement the predictor-corrector concept for ensuring the time marching accuracy to second order. The transport variables  $\psi_{i,j}^{n+1}$  at time level  $n + 1$  can be evaluated by MPDATA scheme by incorporating the contributions from the forcing terms

$$\psi_{i,j}^{n+1} = \text{MPDATA} \left( \psi_{i,j}^{n+1} + 0.5 \Delta t R_{i,j}^n, \vec{u}_{i+(1/2)}^{n+1/2} \vec{e}_j \right) + 0.5 \Delta t R_{i,j}^{n+1}, \quad (21)$$

where  $\psi_{i,j}^{n+1}$  are the transport variables,  $R_{i,j}^{n+1}$  are the forcing terms, and  $\Delta t$  is the time increment. In the above equation, MPDATA symbolizes the homogeneous transport algorithm. Advecting the auxiliary field  $\psi_{i,j}^n + 0.5 \Delta t R_{i,j}^n$ , not only compensates the truncation error due to the forcing terms but also has the physical interpretation of integrating the forces along a parcel trajectory rather than at the grid point. This makes (21) congruent to semi-Lagrangian approximations and facilitates unified fluid models that integrate the equations of motion, optionally, in the Eulerian (point-wise) or Lagrangian (trajectory-wise) sense.

**3.3. Boundary Conditions Treatment.** On the treatment of boundary conditions, the relaxation boundary concept proposed by Davies is used [31]. The eight grid points nearest to the lateral boundary are a dedicated relaxation zone in which the height and momentum field are relaxed towards the externally specified unperturbed values after every time step. The relaxation coefficients are chosen as 1.0, 0.98, 0.9, 0.75, 0.5, 0.25, 0.1, and 0.02.

## 4. Results and Discussion

**4.1. Initial Vortex Structure.** In the present study, we follow the approach in the previous study [22] to identify and to simulate the strong cyclonic motions using the gradient-wind-balance vortex distribution

$$v = -\frac{fr}{2} + \sqrt{\frac{f^2 r^2}{4} + gAB\eta_v \exp\left(\frac{-Ar^{-B}}{r^B}\right)}, \quad (22)$$

$$h = H_0 - \eta_v (1 - \exp(-Ar^{-B})), \quad (23)$$

where  $A$  and  $B$  are, respectively, the vortex size parameter and the vortex shape parameter. Equations (22) and (23) are derived from an analytic model for radial profiles of sea level pressure and winds in a hurricane which was proposed by Holland [32]. In addition, owing to the velocity distribution of the GWB vortex model being assumed to be axisymmetric, the estimated vertical component of vorticity  $\tilde{\zeta}$  can be derived by

$$\tilde{\zeta} = \frac{1}{r} \frac{\partial \hat{r} v_\theta}{\partial \hat{r}}. \quad (24)$$

Therefore, the estimated vorticity distribution can be expressed as

$$\begin{aligned} \tilde{\zeta} &= -f + \frac{1}{2} \left( \frac{f^2 \hat{r}^{-4}}{4} + gAB\eta_v \hat{r}^{2-B} e^{-A\hat{r}^{-B}} \right)^{-1/2} \\ &\times \left( f^2 \hat{r}^{-2} + gAB\eta_v (2 - B + AB\hat{r}^{-B}) \hat{r}^{-B} e^{-A\hat{r}^{-B}} \right). \end{aligned} \quad (25)$$

In [22], two laboratory vortices  $S$  and  $W$  were generated by different strengths.  $S$  was created to be a larger depression by a rotating oar than  $W$  created by a rotating solid cylinder. Figures 2(a) and 2(b) display the fitted results of the vortex depression and the azimuthal velocity distribution of  $S$  and  $W$ , respectively. It is noted that the measurements of surface depression enable the vortex structure to be fitted with satisfactory confidence. The error bar of the depression measurement is approximately 0.001 cm. The vortex size parameters  $A$  for vortices  $S$  and  $W$  are approximately 2.59 and 2.48, respectively. The vortex shape parameters  $B$  for vortices  $S$  and  $W$  are approximately 1.24 and 0.89, respectively.

Figure 2(c) illustrates the distributions of the estimated vorticity  $\tilde{\zeta}$  in (25) of vortices  $S$  and  $W$  as measured in [22]. The results show that vortices  $S$  and  $W$  are both vortices of large Rossby number, while the maximum value of  $\tilde{\zeta}$  for vortex  $S$  was approximately 15.70, which was about twice

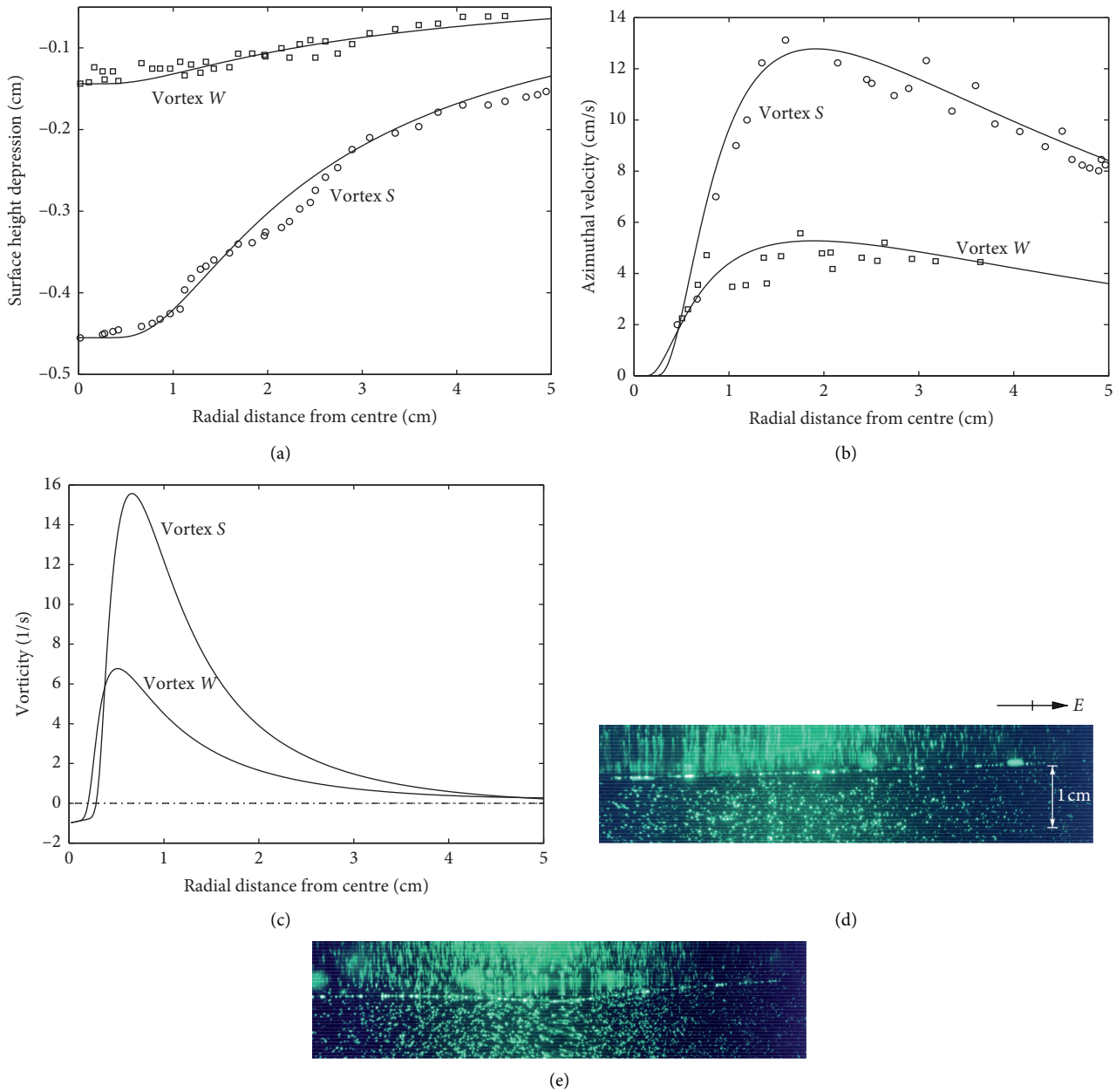


FIGURE 2: The GWB fitted distributions of cyclonic vortices *S* and *W* of (a) surface depression, (b) azimuthal velocity distribution, and (c) vorticity distribution in a radial direction. The circles represent the measured values of vortex *S* and the square symbols show the measured values of vortex *W*. Photographs showing the difference of surface profile at (d) the reference snapshot and (e) the snapshot of vortex centre.

that of the vortex *W* (approximately 6.75). Figure 2(d) shows a reference photograph of a surface profile illuminated by a vertical light sheet taken from the south end of the tank to the north. Figure 2(e) shows a picture of the surface profile of the vortex centre passing through a vertical light sheet. Comparing Figure 2(e) with Figure 2(d), we can extract the actual surface depression depth of the vortex centre.

**4.2. Time Evolutions and Structure Change of Vortices *S* and *W*.** Figures 3 illustrates the calculation results of the relative vorticity of the vortex *S* by MSWM on a sloping bottom in a rotating tank. The domain of numerical calculation is

120 cm  $\times$  120 cm on a 600  $\times$  600 uniform rectangular grid system. The vortex was generated about 45 cm and 40 cm away from the south and the east tank boundaries, respectively, to relax the boundary effect. Compared with the experimental results in the previous study [22], Figures 4 and 5 show a qualitatively close agreement between the experimental and numerical approaches of the vortex *W*. Fairly symmetric isolated vortices were observed experimentally and numerically during their northwest drifts on a sloping bottom. Outside the primary vortex structure, there exists a weak anticyclonic vorticity patch. This flow feature is gradually induced by the Rossby wave radiation accompanied by the primary vortex as indicated in [10] and is observed both in the

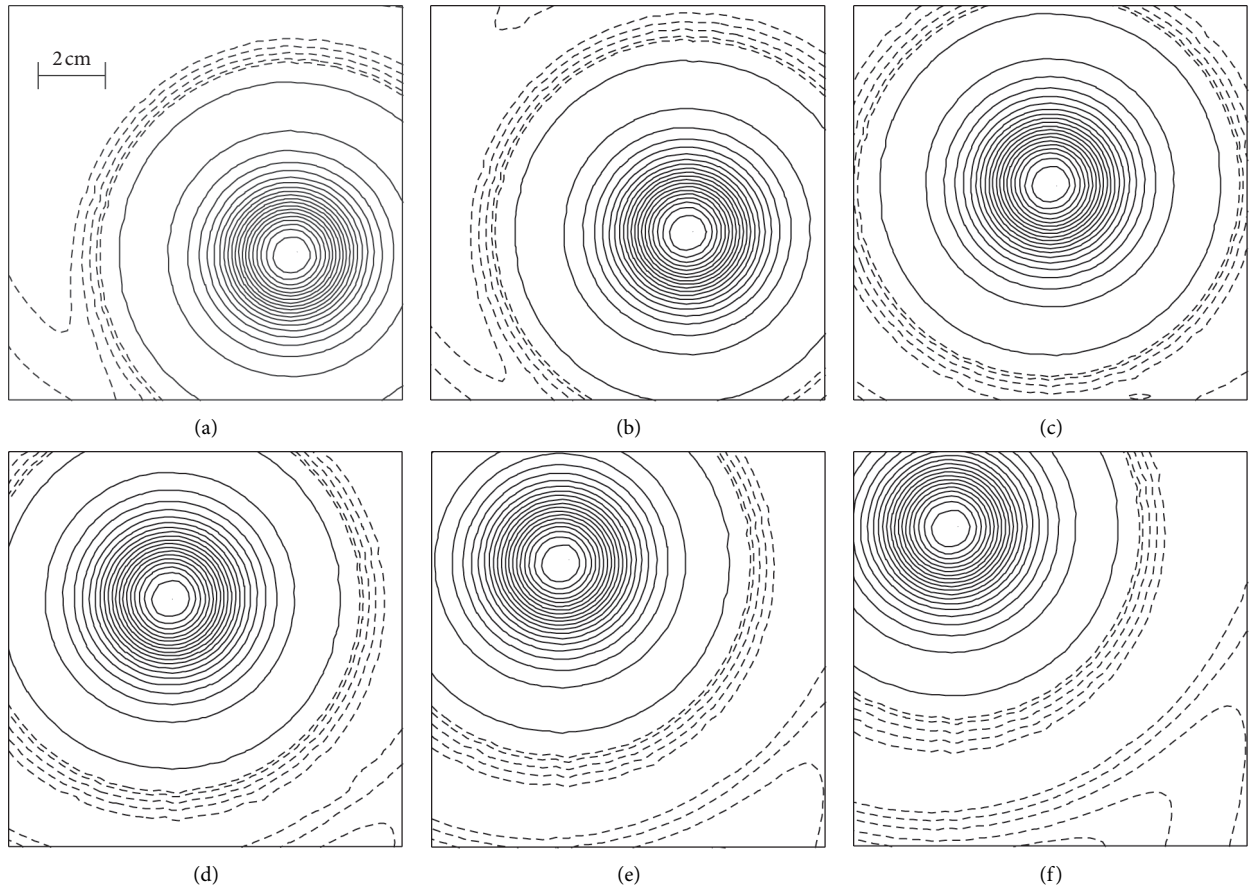
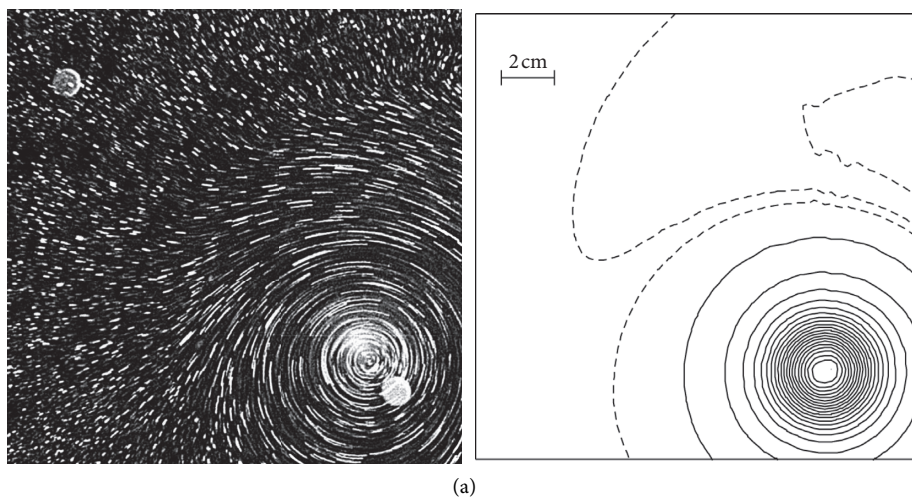
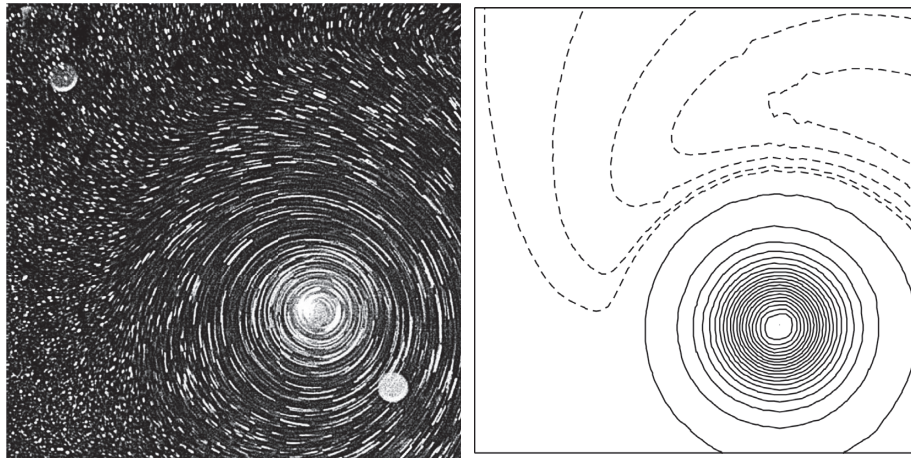


FIGURE 3: Computational results showing the evolution of the relative vorticity of vortex S at the time instances (a) 5 s, (b) 9 s, (c) 11 s, (d) 13 s, (e) 15 s, and (f) 17 s after the oar was lifted. The size of the window is about 12 cm square area. The solid lines represent the positive vorticity, while the dashed lines indicate the negative vorticity.

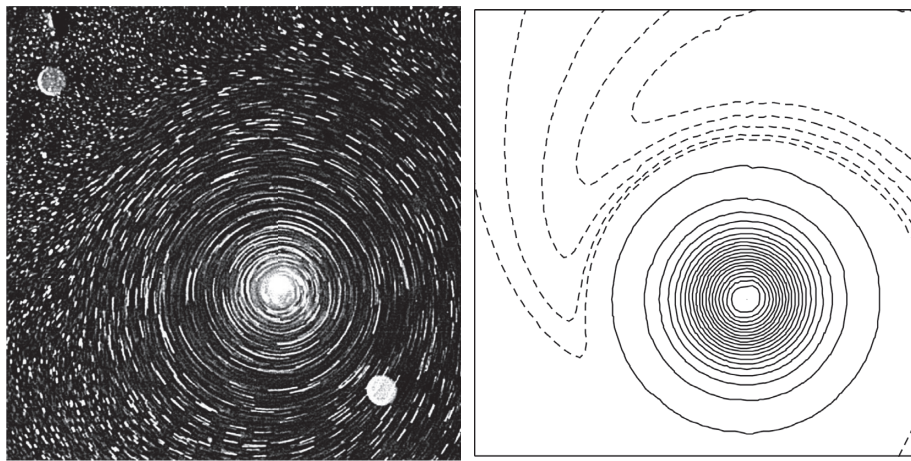


(a)  
FIGURE 4: Continued.



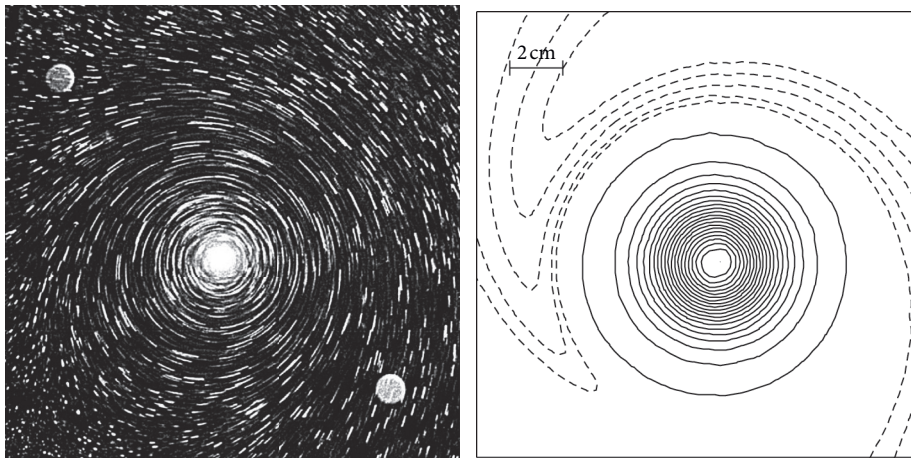


(b)



(c)

FIGURE 4: Comparison of experiment and numerical results for a stirred vortex  $W$  at the time instances (a) 6 s, (b) 10 s, and (c) 14 s after the cylinder was lifted. The size of the window is about 17 cm square area. Note that the experiment photographs are adapted from [22] for comparison with numerical results.



(a)

FIGURE 5: Continued.

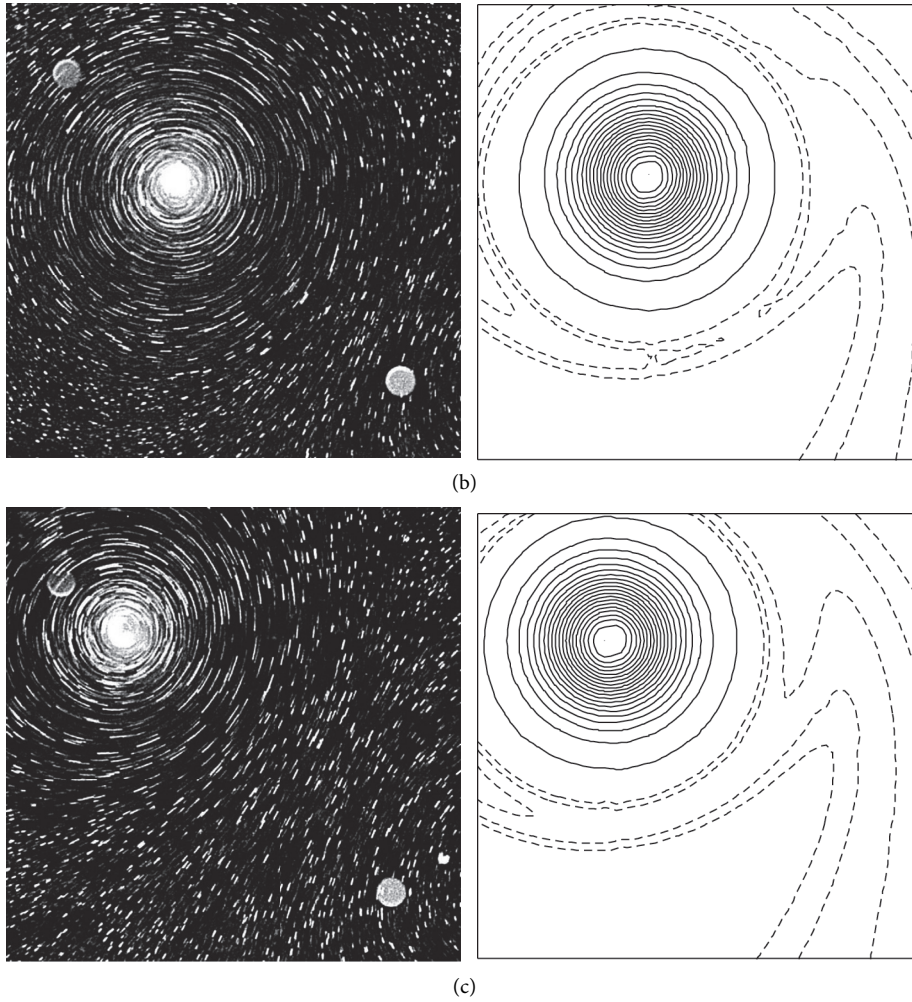


FIGURE 5: Similar to Figure 4 but at the time instances (a) 18 s, (b) 22 s, and (c) 28 s after the cylinder was lifted. The size of the window is about 17 cm square area.

experiments and in the numerical simulations. It is noted that the ratio of the vortex  $\beta$  effect and the planetary  $\beta$  effect  $\gamma = \beta_v^*/\beta_y^*$  reflects the nonlinearity against the linear Rossby wave radiation. In this study, the value of  $\gamma$  corresponding to vortex S and vortex W is about 4.437 and 1.385, respectively. It can be seen that the ability of vortex S to resist Rossby wave radiation is better than that of vortex W.

Figure 6 demonstrates the evolution of vortex structures of vortex W at (a)  $t = 10$  s and (b)  $t = 28$  s after the cylinder was lifted. The solid lines denote the radial distribution of the vertical component of vorticity, the dash-dotted lines indicate the radial distribution of the azimuthal velocity, and the dashed lines show the radial distribution of the surface depression. Notably, these radial distributions of vortex structure are obtained at a cross-sectional plane passing through the vortex with maximum depression. For convenience, only the values at the east of the vortex are plotted. The nondimensional relative vorticity shows a gradually decreasing tendency of its peak value at the vortex central region. The vortex distribution becomes smoother or remains constant at later times as long as the vortex travels to the northwest.

Additionally, the distribution of the azimuthal velocity shows that the maximum value approximately remains constant, while its corresponding radius gradually becomes larger than the initial radius  $R_m$ . Finally, the distribution of the vortex depression also displays a significant decreasing tendency when the vortex evolves to the northwest.

**4.3. Vortex Trajectory and Intensity Change.** Regarding the vortex trajectories, this study used the circle and square symbols in Figure 7 to indicate the measured tracks of vortex S and vortex W, respectively. The solid lines represent the associated MSWM calculated vortex tracks. The dashed lines show that the contour levels of the topographic features  $h_B^*$  calculated from south to north are approximately 0.387, 0.401, 0.415, 0.429, and 0.433, respectively. The nondimensional topographic heights  $h_B^*$  consist of the topographic sloping bottom  $h_s^*$  and free surface deformation  $h_p^*$  owing to the tank rotation as mentioned in [22]. It is worth noting that, compared with the contribution of inclined bottom deformation  $h_s^*$  to the northwest drift of the vortex, the

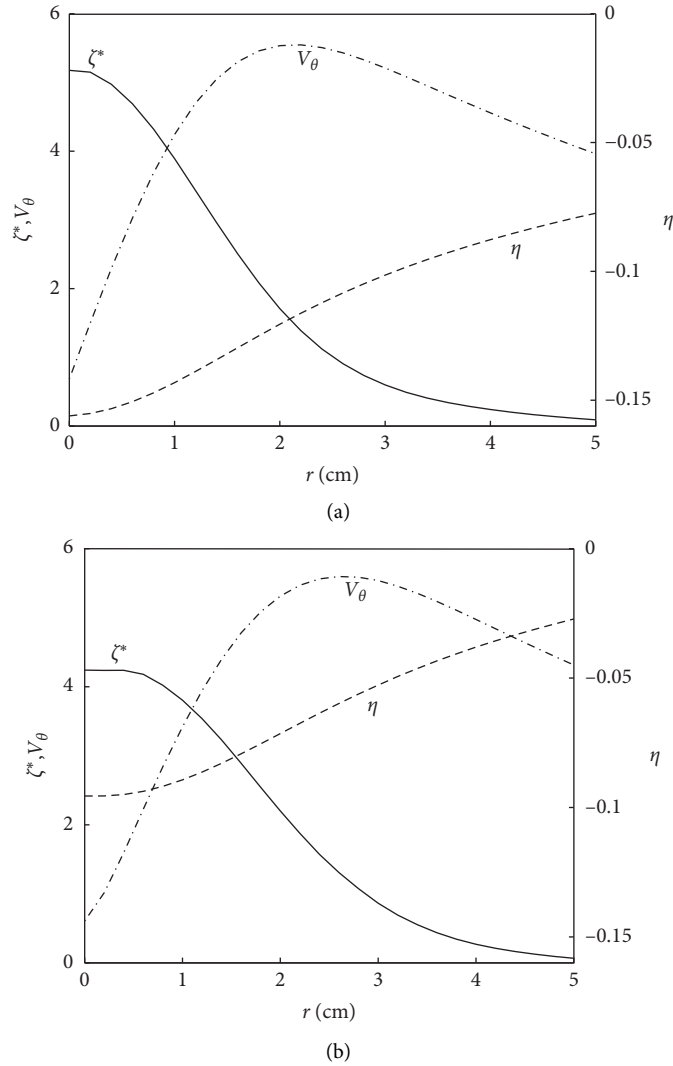


FIGURE 6: Evolution of structures of vortex  $W$  at (a)  $t = 10$  s and (b)  $t = 28$  s after the cylinder was lifted.

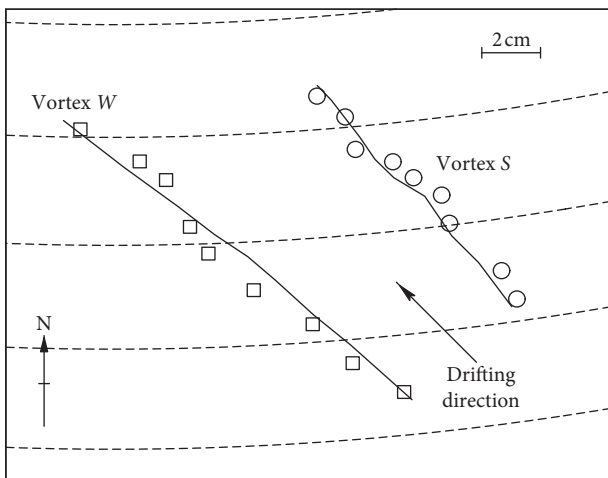


FIGURE 7: Graph showing the trajectories of the cyclonic vortices  $S$  and  $W$ .

contribution of free surface deformation  $h_p^*$  to the northwest drift of the vortex is much smaller.

Figure 8(a) monitors the time variations of the maximum value of the azimuthal velocity  $V_{\theta, \max}$  as intensity change of vortex. They are recorded by the numerical simulation (denoted as a solid line) or the streak photography (represented as circles). The results show that the variations of  $V_{\theta, \max}$  obtained from these two approaches displayed good agreement. It is noted that the calculated  $V_{\theta, \max}$  decayed from its initial value 12.43 cm/s to 9.82 cm/s in 15 seconds. In the laboratory,  $V_{\theta, \max}$  was estimated as decaying from 12.92 cm/s to 9.79 cm/s. Figure 8(b) shows the time variations of the maximum vorticity  $\zeta_{\max}$  that is approximated by assuming that the vortex is axis-symmetric. The measurement results revealed good agreement of the maximum values of azimuthal velocity and vertical component of vorticity between these two approaches.



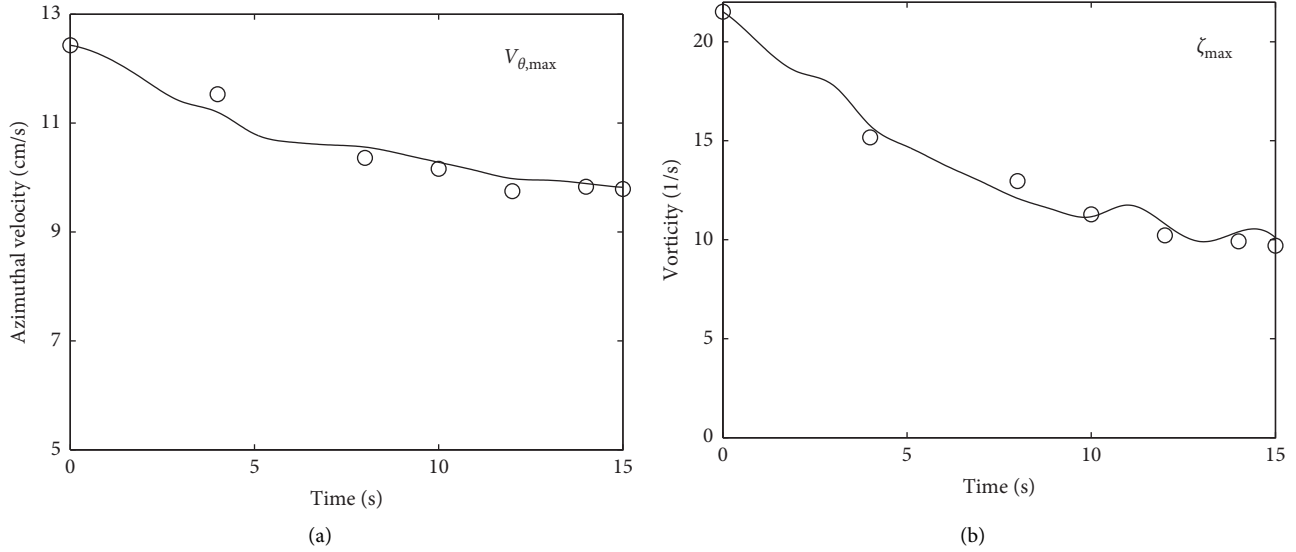


FIGURE 8: Time variations of (a) the maximum azimuthal velocity  $V_{\theta, \max}$  and (b) the maximum vorticity  $\zeta_{\max}$  of vortex S.

#### 4.4. Effect of Bottom Slope on the Strong Cyclonic Motion.

In this section, we would like to investigate the effect of sloping bottom on the strong cyclonic motions. For convenience, we choose a typical case of a moderate tropical cyclone as our prototypical problem and seek for its possible model vortex  $B$  that is dynamically similar in the laboratory. For example, let us choose a strong cyclonic vortex with a maximum azimuthal speed  $V_m = 40$  m/s and its corresponding radius  $R_m = 150$  km initializing at the latitude of  $20^\circ$  in the Northern Hemisphere. The unperturbed layer depth  $H_0$  is about 10 km and a maximum vortex depression  $\eta_v$  is assumed to be 295.56 m. We assume that the vortex shape parameter  $B$  of the prototypical cyclone is 1.5, and the vortex size parameter  $A$  can be determined as 1837 km by (23). On the laboratory side, we choose a typical strong vortex with maximum azimuthal speed  $V_m = 6$  cm/s and a corresponding radius  $R_m = 3$  cm moving on a sloping bottom with  $s_y = 0.0538$  in a rotating tank with an angular speed  $\Omega = 0.785$  rad/s. Under the similarity condition (14), the unperturbed depth  $H_0$  in the tank can be obtained as 10.47 cm. The maximum depression depth  $\eta_v$  of the vortex in the laboratory can also be determined from the similarity condition (15) as 0.206 cm.

Figure 9 shows the numerical results of trajectory of vortex  $B$  translating on a sloping bottom with four different slopes as 0.01076 (Case S1), 0.0538 (Case S2), 0.1345 (Case S3), and 0.269 (Case S4). The trajectories were determined from the calculated streamlines, and the origin of reference was located 60 cm north from the south bound of the tank and 60 cm west from the east bound of the tank. Comparing the above four vortex paths under different bottom slopes, we summarize as follows. First, the vortices in all examples generally move to the northwest. All the examples in the first half of the path show linear motion, when the vortex maintains a single vortex structure. The moving speed of the vortex is proportional to the slope of the bottom. The greater the slope is, the faster the vortex moves. Second, the vortex in

Case S1 moves at the slowest speed and continues to move slowly towards the northwest as a single vortex. On the other hand, for Cases S2, S3, and S4, the vortices accelerate significantly at the early stage and decelerate gradually at the following stage. These trends of intensity change and translating speed are basically coincided with the theory proposed by Resnik et al. [10]. That is to say, in the initial stage, the vortex is between the advection time scale and the wave time scale, so it is accelerated by beta-gyres. However, after wave time scale, beta-gyres will further induce secondary beta-gyres due to nonlinearity. This process will last for a long time to a certain time scale and cause the vortex to slow down.

Figure 10 shows the streamline graphs for Case S3. We can observe that several secondary vortices at the east of the primary vortex were induced consequently by strong planetary  $\beta$  effect. As a result, these secondary vortices in turn interact with the primary vortex and cause a slight meandering and a distortion of the primary vortex. These waves demonstrate an alternative pattern of clockwise and anticlockwise circulation cells translating westward and they are usually referred to as topographic Rossby wave as mentioned in [33].

Figure 11 demonstrates the vorticity contours for Case S4. Prominent Rossby wave wakes following the primary vortex were excited by steep bottom slope. This fact shows that the induced Rossby wave wakes not only alter the trajectory of the primary vortex (as shown in Figure 9) but also stretch the primary vortex from axis-symmetric to axis-asymmetric ( $t = 36$  s to  $t = 56$  s as shown in Figure 10). The understanding of this stage can be explained by the vortex distortion time scale proposed by Resnik et al. [10]. In Figure 11, a negative vorticity patch N1 was generated in the vicinity of the primary vortex P1 and was developed to a vorticity tendril during  $t = 0$  s to 16 s. From  $t = 16$  s to 32 s, this vorticity tendril was influenced and has been stretched by following positive vorticity patch P2 which formed

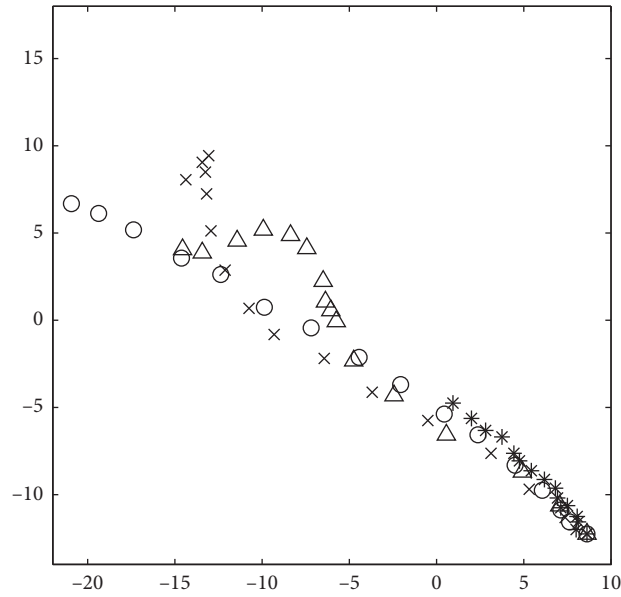


FIGURE 9: Simulation tracks of a benchmark vortex  $B$  translating on a sloping bottom. The vortices were indicated by asterisks, circles, crosses, and triangles for tracks on different bottom slopes of 0.01076, 0.0538, 0.1345, and 0.269, respectively.

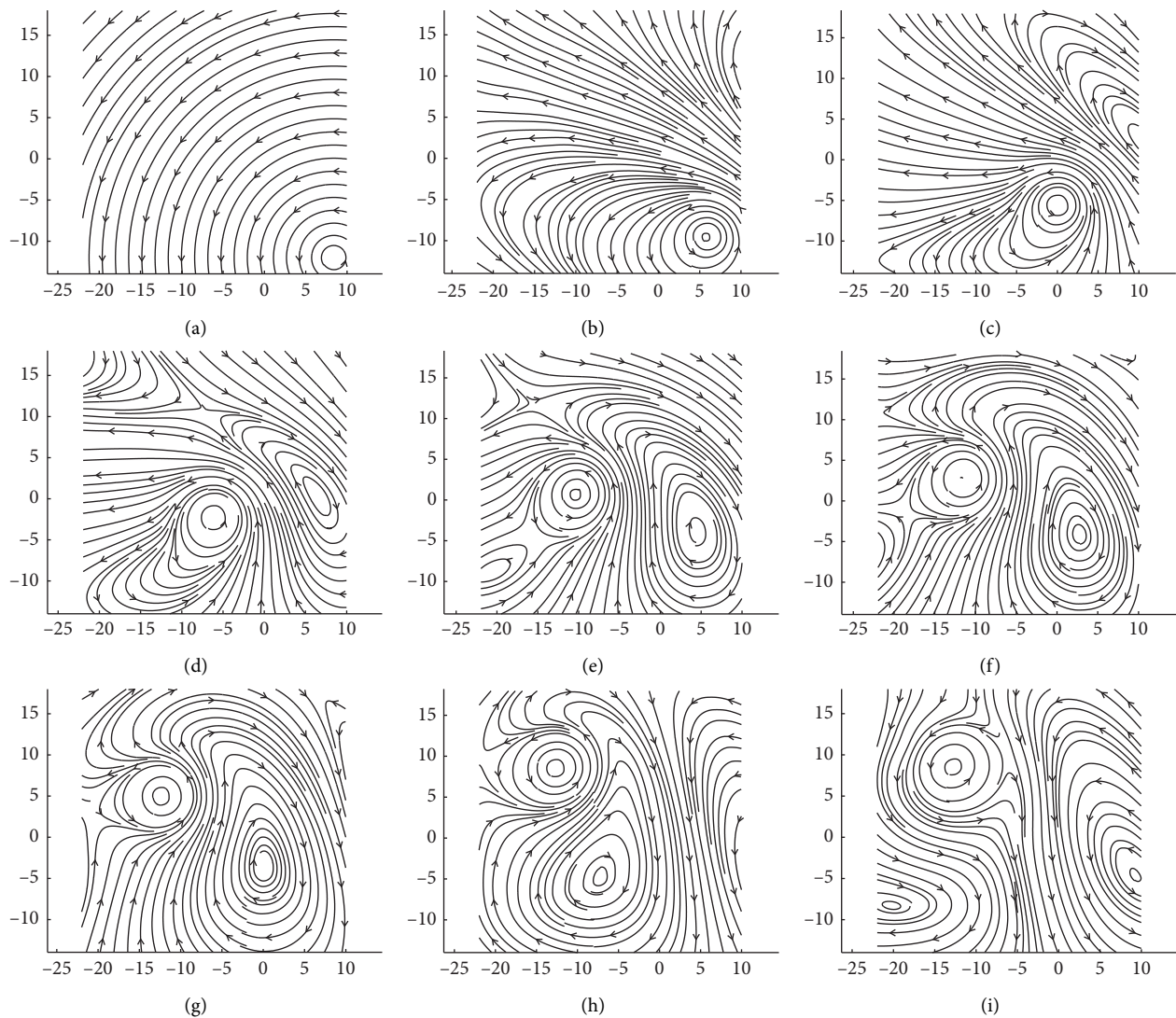


FIGURE 10: Streamlines plots of Case S3 with a bottom slope  $s_y = 0.1345$ . (a)  $t = 0s$ , (b)  $t = 8s$ , (c)  $t = 16s$ , (d)  $t = 24s$ , (e)  $t = 32s$ , (f)  $t = 36s$ , (g)  $t = 40s$ , (h)  $t = 48s$ , and (i)  $t = 56s$ .



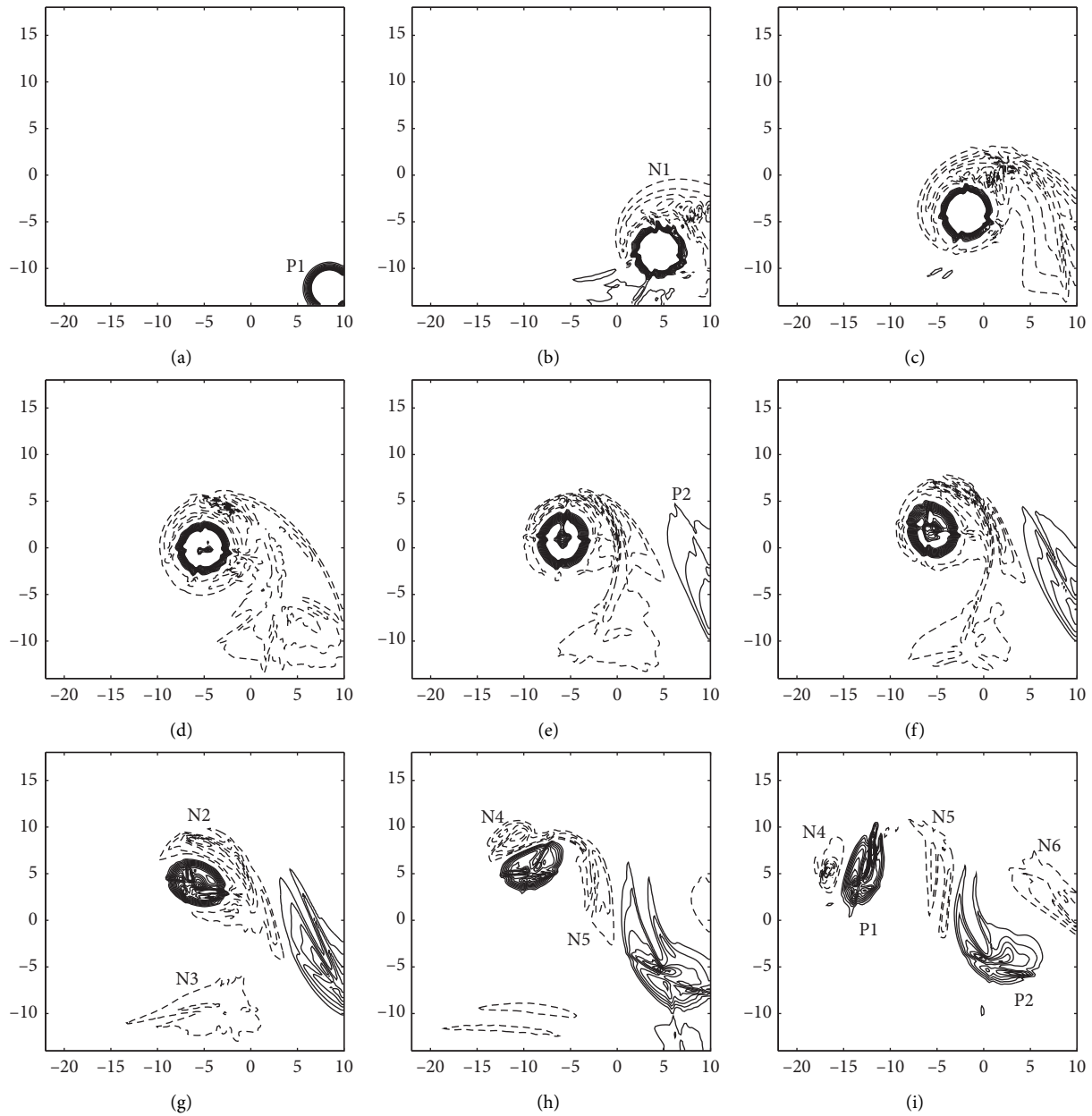


FIGURE 11: Contours of relative vorticity of Case S4 with a bottom slope  $s_y = 0.269$ . (a)  $t = 0s$ , (b)  $t = 8s$ , (c)  $t = 16s$ , (d)  $t = 24s$ , (e)  $t = 32s$ , (f)  $t = 36s$ , (g)  $t = 40s$ , (h)  $t = 48s$ , and (i)  $t = 56s$ .

behind N1. Generally, it is shown that the simulation by MSWM well demonstrates the entire flow features. It also reveals complicated nonlinear interactions of the vortex and its induced Rossby wave wakes that consist of vorticity patches with alternative signs (P for positive and N for negative, for short).

In addition, from the point of view of total energy conservation, it can be observed in Figure 11 that the Rossby waves radiated from the primary vortex provide the energy required for wake development. This process mainly occurs in the near field of vortex, namely, beta-gyres. The wake coexists and remains near the primary vortex and keeps a quasi-steady translation speed similar to that of the primary vortex [10]. This interpretation is different from the

viewpoint held in some literatures [8, 34]. They explained that the Rossby wave wake is the product of quasisresonance between the primary vortex and planetary vorticity gradient. However, the conservation of the total energy of the system does not support this argument and needs further study.

### 5. Conclusions

In this study, a joint theoretical and numerical study is used to investigate the flow features of a strong cyclonic vortex generated in a rotating tank with a sloping bottom. This study clarifies the idea of the dynamical similarity between the prototypical and model flow fields by satisfying the similarity conditions (14) and (15). Calculations by the

proposed modified shallow water model for the strong and intense cyclonic motions show a close agreement with the experimental results in a rotating tank. The present study proposed a modified shallow water model incorporating a gradient-wind-balance (GWB) vortex model for investigating the hurricane-like cyclonic motions on a  $\beta$ -plane in the Northern Hemisphere and their structures in a rotating tank with a gently sloping bottom.

There are two main advantages of the model. (i) Unlike the traditional QGVE model, the MSWM is more suitable to take care of the significant depth depression of the vortex, which is a prominent feature of large Rossby number hurricane-like vortices. From the surface depression measurements, the effect of vortex stretching owing to this vortex depression had the same order of magnitude as that of the vortex stretching caused by the sloping bottom. (ii) Another significant source of vortex stretching that should be considered was the parabolic free surface resulting from the tank rotation. In the present MSWM model, this effect of paraboloidal free surface was conveniently represented by an effective gravity. Regarding the simulation of the large Rossby number vortices, the GWB vortex model pictures the vortex structure more accurately than the traditional Gaussian/Rankine vortex models. It is noted that, in literature, for vortices with both large Rossby number ( $Ro_v \approx O(1)$ ) and large Burger number ( $Bu \gg 1$ ), the fast motion in the flow field, that is, inertia gravity wave (IGW), may be decoupled and can be emitted from the slow vortex motion [34, 35]. The Burger number is defined as  $Bu = (R_d^2/R_m^2)$ , where  $R_d$  is the Rossby deformation radius as  $R_d = \sqrt{gH_0}/f_0$ . Take the vortex S as an example, where  $Ro_v = 4.32$  and  $Bu \approx 1, 154$ . Although the flow field of vortex S satisfies the conditions of IGW generation, there is no direct evidence of the emission of IGW in the current experiment or numerical results. This interesting phenomenon is worthy of further and careful study in the future.

Our major results obtained in this paper were presented in two parts. In the first part, a numerical simulation of a monopolar vortex translating on a gentle-slope bottom was carried out to verify the experimental results in the previous study [22]. Close agreements were found between experiment and simulation, including the streamline patterns and the vortex trajectory. After the long-term evolution, the coherency and monopolar nature of a strong vortex ( $Ro_v \approx O(1)$ ) remained both in the experimental and numerical results. In the second part, the long-term behaviours of vortex motion on different sloping bottoms were investigated numerically. For those gentle-slope cases (Case S1 and Case S2), the vortex moves steadily to the northwest. As the bottom slope goes steeper (Case S3 and Case S4), the trajectory of the primary vortex is being influenced by the associate secondary vortices and altered as meandered, curved motion. In the case of steepest slope (S4), the interaction between the primary vortex and the induced Rossby wave wakes reveals many interesting features such as the deformation of primary vortex, the emergence of a dipolar vortex, and the associated Rossby wave wakes. The interaction between strong cyclonic vortices and the accompanied secondary circulation structures under various

flow conditions requires comprehensive exploration through theoretical analysis, numerical simulation, and laboratory experiments. They may provide invaluable ingredients for understanding the physics of intense oceanic eddies [33, 36] or atmospheric vortices [37] with coherent wave trains [38] and they will be reported elsewhere.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The authors acknowledge Shandong Polytechnic, China, for the partial support of this research. The first author thanks Professor C. C. Chu and Professor C. C. Chang of National Taiwan University for their supports in the early stage of this work.

## References

- [1] J. Adem, "A series solution for the barotropic vorticity equation and its application in the study of atmospheric vortices," *Tellus*, vol. 8, no. 3, pp. 364–372, 1956.
- [2] G. R. Flierl, "The application of linear quasigeostrophic dynamics to Gulf Stream rings," *Journal of Physical Oceanography*, vol. 7, no. 3, pp. 365–379, 1977.
- [3] J. C. McWilliams and G. R. Flierl, "On the evolution of isolated, nonlinear vortices," *Journal of Physical Oceanography*, vol. 9, pp. 1155–1182, 1979.
- [4] M. Fiorino and R. L. Elsberry, "Some aspects of vortex structure related to tropical cyclone motion," *Journal of the Atmospheric Sciences*, vol. 46, pp. 975–990, 1989.
- [5] G. M. Reznik, "Dynamics of singular vortices on a beta-plane," *Journal of Fluid Mechanics*, vol. 240, pp. 405–432, 1992.
- [6] G. M. Reznik and W. K. Dewar, "An analytical theory of distributed axisymmetric barotropic vortices on the  $\beta$ -plane," *Journal of Fluid Mechanics*, vol. 269, pp. 301–321, 1994.
- [7] G. G. Sutyryn and G. R. Flierl, "Intense vortex motion on the beta plane: development of the beta gyres," *Journal of the Atmospheric Sciences*, vol. 51, no. 5, pp. 773–790, 1994.
- [8] G. K. Korotaev and A. B. Fedotov, "Dynamics of an isolated barotropic eddy on a beta-plane," *Journal of Fluid Mechanics*, vol. 264, pp. 277–301, 1994.
- [9] S. G. Llewellyn Smith, "The motion of a non-isolated vortex on the beta-plane," *Journal of Fluid Mechanics*, vol. 346, pp. 149–179, 1997.
- [10] G. M. Reznik, R. Grimshaw, and E. S. Benilov, "On the long-term evolution of an intense localized divergent vortex on the beta-plane," *Journal of Fluid Mechanics*, vol. 422, pp. 249–280, 2000.
- [11] E. Firing and R. C. Beardsley, "The behavior of a barotropic eddy on a  $\beta$ -plane," *Journal of Physical Oceanography*, vol. 6, no. 1, pp. 57–65, 1976.
- [12] M. Takematsu and T. Kita, "The behavior of isolated free eddies in a rotating fluid: laboratory experiment," *Fluid Dynamics Research*, vol. 3, no. 1–4, pp. 400–406, 1988.

- [13] A. Masuda, K. Marubayashi, and M. Ishibashi, "A laboratory experiment and numerical simulation of an isolated barotropic eddy in a basin with topographic  $\beta$ ," *Journal of Fluid Mechanics*, vol. 213, no. 1, pp. 641–655, 1990.
- [14] G. F. Carnevale, R. C. Kloosterziel, and G. J. F. van Heijst, "Propagation of barotropic vortices over topography in a rotating tank," *Journal of Fluid Mechanics*, vol. 233, pp. 119–139, 1991.
- [15] J.-B. Flór and I. Eames, "Dynamics of monopolar vortices on a topographic beta-plane," *Journal of Fluid Mechanics*, vol. 456, pp. 353–376, 2002.
- [16] E. J. Hopfinger and G. J. F. V. Heijst, "Vortices in rotating fluids," *Annual Review of Fluid Mechanics*, vol. 25, no. 1, pp. 241–289, 1993.
- [17] G. J. F. V. Heijst and H. J. H. Clercx, "Laboratory modelling of geophysical vortices," *Annual Review of Fluid Mechanics*, vol. 41, pp. 143–164, 2008.
- [18] J. S. Lam and D. G. Dritshel, "On the beta-drift of an initial circular vortex patch," *Journal of Fluid Mechanics*, vol. 436, pp. 107–129, 2001.
- [19] S. Kravtsov and G. Reznik, "Numerical solutions of the singular vortex problem," *Physics of Fluids*, vol. 31, Article ID 066602, 2019.
- [20] G. M. Reznik and S. V. Kravtsov, "Singular vortices on a beta-plane: a brief review and recent results," *Physical Oceanography*, vol. 27, pp. 659–676, 2020.
- [21] M. M. Jalali and D. G. Dritschel, "Stability and evolution of two opposite-signed quasi-geostrophic shallow-water vortex patches," *Geophysical & Astrophysical Fluid Dynamics*, pp. 1–27, 2020.
- [22] H. C. Chen, J. H. Leu, Y. L. Lin et al., "Cyclonic motion and structure in rotating tank: experiment and theoretical analysis," *Sensors and Materials*, 2021.
- [23] C. Schär and R. B. Smith, "Shallow-water flow past isolated topography. Part I: vorticity production and wake formation," *Journal of the Atmospheric Sciences*, vol. 50, no. 10, pp. 1373–1400, 1993.
- [24] C. Schär and R. B. Smith, "Shallow-water flow past isolated topography. Part II: transition to vortex shedding," *Journal of the Atmospheric Sciences*, vol. 50, no. 10, pp. 1401–1412, 1993.
- [25] P. K. Smolarkiewicz, "A fully multidimensional positive definite advection transport algorithm with small implicit diffusion," *Journal of Computational Physics*, vol. 54, no. 2, pp. 325–362, 1984.
- [26] P. K. Smolarkiewicz and T. L. Clark, "The multidimensional positive definite advection transport algorithm: further development and applications," *Journal of Computational Physics*, vol. 67, no. 2, pp. 396–438, 1986.
- [27] P. K. Smolarkiewicz and L. G. Margolin, "MPDATA: a finite-difference solver for geophysical flows," *Journal of Computational Physics*, vol. 140, no. 2, pp. 459–480, 1998.
- [28] P. K. Smolarkiewicz, "Multidimensional positive definite advection transport algorithm: an overview," *International Journal for Numerical Methods in Fluids*, vol. 50, no. 10, pp. 1123–1144, 2006.
- [29] J. Pedlosky, *Geophysical Fluid Dynamics*, Springer, Berlin, Germany, 2nd edition, 1986.
- [30] M. V. Nezlin and E. N. Snezhkin, *Rossby Vortices, Spiral Structures, Solitons: Astrophysics and Plasma Physics in Shallow Water Experiments*, Springer-Verlag, Berlin, Germany, 1993.
- [31] H. C. Davies, "Limitations of some common lateral boundary schemes used in regional NWP models," *Monthly Weather Review*, vol. 111, no. 5, pp. 1002–1012, 1983.
- [32] G. J. Holland, "An analytic model of the wind and pressure profiles in hurricanes," *Monthly Weather Review*, vol. 108, no. 8, pp. 1212–1218, 1980.
- [33] L. Zavala Sansón and G. J. F. V. Heijst, "Laboratory experiments on flows over bottom topography," *Modeling Atmospheric and Oceanic Flows: Insights from Laboratory Experiments and Numerical Simulations*, pp. 139–158, 2015.
- [34] V. Zeitlin, "Decoupling of balanced and unbalanced motions and inertia-gravity wave emission: small versus large Rossby numbers," *Journal of the Atmospheric Sciences*, vol. 65, no. 11, pp. 3528–3542, 2008.
- [35] V. Zeitlin, *Geophysical Fluid Dynamics: Understanding (Almost) Everything with Rotating Shallow Water Models*, Oxford University Press, Oxford, UK, 2018.
- [36] G. R. Flierl, "Rossby wave radiation from a strongly nonlinear warm eddy," *Journal of Physical Oceanography*, vol. 14, no. 1, pp. 47–58, 1984.
- [37] J. M. Cosgrove and L. K. Forbes, "Nonlinear behaviour of interacting mid-latitude atmospheric vortices," *Journal of Engineering Mathematics*, vol. 104, no. 1, pp. 41–62, 2017.
- [38] K. D. Krouse, A. H. Sobel, and L. M. Polvani, "On the wavelength of the Rossby waves radiated by tropical cyclones," *Journal of the Atmospheric Sciences*, vol. 65, no. 2, pp. 644–654, 2008.

## Research Article

# Relationship between Bitcoin Exchange Rate and Other Financial Indexes in Time Series

Chien-Yun Chang,<sup>1</sup> Chien-Chien Lo,<sup>2</sup> Jui-Chang Cheng,<sup>3</sup> Tzer-Long Chen ,<sup>4</sup>  
Liang-Yun Chi,<sup>5</sup> and Chih-Cheng Chen <sup>6,7</sup>

<sup>1</sup>Department of Fashion Business and Merchandising, Ling Tung University, Taichung, Taiwan

<sup>2</sup>Department of International Business, Providence University, Taichung 43301, Taiwan

<sup>3</sup>Department of Leisure and Recreation Management, National Taichung University of Science and Technology, Taichung, Taiwan

<sup>4</sup>Department of Finance, Providence University, Taichung, Taiwan

<sup>5</sup>Department of Finance, National Taichung University of Science and Technology, Taichung, Taiwan

<sup>6</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>7</sup>Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan

Correspondence should be addressed to Tzer-Long Chen; [tlchen1976@pu.edu.com](mailto:tlchen1976@pu.edu.com) and Chih-Cheng Chen; [ccc@gm.cyut.edu.tw](mailto:ccc@gm.cyut.edu.tw)

Received 29 October 2020; Revised 21 February 2021; Accepted 5 April 2021; Published 29 April 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Chien-Yun Chang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Bitcoin exchange rate (BER) is influenced by many variables such as human speculation and policies and, thus, is dependent on the financial system. The fluctuation of BER submitted has been extensively investigated. However, the correlation analysis of the short- and long-term effects by indicators of online sentiment is unexplored. Therefore, this study establishes a VAR model for BER which provides a framework to the Google search volume index (SVI), the investor fear gauge (VIX), and the S&P500 Index. The findings of the analysis suggest that BER and Google SVI have a Granger causality feedback relationship in both the short- and long-term co-integration equilibrium, and the VIX is significantly related to BER in the long-term co-integration.

## 1. Introduction

The Bitcoin exchange rate (BER) is extremely volatile. As shown in Figure 1, BER to the US dollar from October 01, 2013, to June 22, 2018, increased by nearly 2 million times from less than USD 0.01 to 19,345.49 on December 16, 2017. Kurka concluded that BER is independent of any financial asset classes in the system, but its spillover effect affected the traditional financial markets [1].

The pricing models of stock price or exchange rate have assumptions of transactions by rational investors. However, researchers pointed out that investment decisions made by investors are not necessarily rational. Thus, behavioral financial theories emerged to explain irrational decisions. For example, Barber and Odean proposed the “Attention Theory” that explained investors’ intention of buying stocks without having time to interpret the disclosure of massive

information on the stocks [2]. Merton suggested the “Investor Recognition Hypothesis” that stated the information spillover effect of investors’ stock buying on the firm’s visibility [3]. The higher the visibility of a firm, the more the interest investors have in the stock. It yields higher price returns and greater trading volume. Cai et al. proved that Google SVI of Bitcoin had a significant impact on its price and transaction volume as the proxy explanatory variable of “investor focus” [4].

This research employs the vector autoregression (VAR) model to explore the long- and short-term relationship between the Google SVI and BER. The key variables are studied by applying the theoretical basis of “investor attention” that influences BER. Based on the findings of the current research, the important variables of traditional financial assets are selected in consideration of the volatility index (VIX), the S & P500 index, and the Google SVI. The

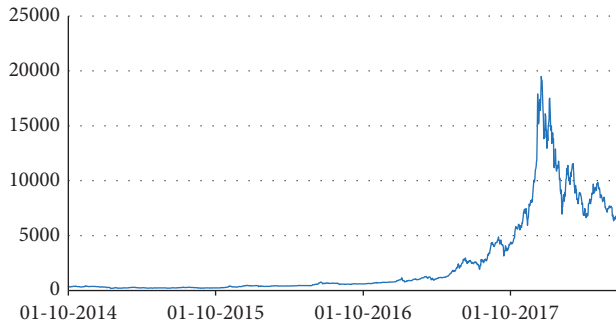


FIGURE 1: Bitcoin exchange rate chart (10/01/2014–06/22/2018).

vector autoregression (VAR) analysis of the short- and long-term impacts and Granger causal relationship ([5–7]) are employed to develop a prediction model for BER change for providing a reference for Bitcoin hedging operation to investors. The BER is a new investment tool with higher volatility to the traditional financial product. Investors have little hedge and others reference target to avoid the risk of BER. The forecasting model of BER can offer mostly relation financial products and be applied for hedge, arbitrage for investors. The result expected the SVI should have higher relationship in the short and long periods.

This article is organized as follows: Section 2 reviews previous references and discusses the relation of the Google SVI and VIS, in variable explanation to Bitcoin. Section 3 presents the VAR for the data analysis as the research method. Section 4 describes the analysis results. Finally, Section 5 concludes this article.

## 2. Literature Review

**2.1. Google SVI.** Urquhart [8] and Bleher and Dimpfl [9] adopted Google SVI as a proxy explanatory variable for “investor attention” for studying 12 cryptocurrencies including Bitcoin. The other researchers also employed Google SVI to measure investor attention [10, 11].

Since “Google Trends” was first introduced, the number of searches was queried by entering the targeted word. Later on, Google launched “Google Insights for Search” which divided the data into detailed categories such as time and geographical regions. “Google Insights for Search” became the current Google Trend in September 2012. The time range for searching is customized or selected in the past 1, 4 hours, 1, 7, 30, 90, and 5 years. Up to 5 keywords are analyzed simultaneously. The search items are searched for their trends in a single country or all over the world. The categories are selected by the industry of interest or all categories. After determining the keywords, region, periods, and categories, Google Trends generate a trend chart with a standard quantitative range from 0 to 100 and calculate the SVI data based on the average time series, but the data frequency is a week.

SVI calculation formula is as follows:

$$\text{SVI} = \frac{\text{total search volume}}{\text{maximum total search volume in a given time period}} \times 100. \quad (1)$$

The SVI may yield different results for the same keywords according to different query, time, date, and geographic location. When Google compiles SVI data, it does not search all data but randomly selects samples. However, different results did not change the research outcomes [12]. Takeda and Wakao present the SVI has a positive impact on the stock price and volume [13]. Aouadi et al. using Google search volume to evidence the relationship among the attention of French stock market investors and trading volume, stock market illiquidity, and volatility [10].

Zhang et al. pointed out that the long-term prices converged to the mean reversion when stock prices rose solely with increased investor attention [14]. To understand whether the mean reversion of BER is related to Google SVI, we selected Google SVI as the proxy interpretation variable for “investor focus”. Details are shown in Figure 2.

**2.2. VIX.** The Volatility Index (VIX) is compiled by the COBE Exchange that implies volatility of future options on the S&P500 and reflects the degree of risk in the stock market over the next 30 days. The VIX greater than 40 indicates that market investors expect a strong fluctuation of stock index in the future, that is, irrational panic. When the index is less than 15, it means the investors believe the volatility in the future stock market to be mild which shows irrational exuberance. As it reflects the traders’ expectations on future stock price change, it is also called the “Fear Index” or “The Investor Fear Gauge.”

Qadan et al. believed that VIX reflected the investors’ sentiment. In a period of extreme uncertainty, investors tend to be more risk-averse and therefore need higher idiosyncratic volatility (IVOL) premium at a high level of VIX [15]. Under the circumstance, investors avoid stocks of high IVOL, which results in lower investment returns with high IVOL than with low IVOL. Tsai et al. also found the reversed prices in the future led to significant negative investment loss when investors overestimated stock prices positively [16]. Simon and Wiggins showed that the VIX index was used to predict the future stock market [17]. Lee pointed the new sentiment of the current period has a positive relationship with investment reports, and the new sentiment of the lagged period has a negative relationship with investment aspirations [18]. Copeland and Copeland noted that COBE Volatility Index (VIX) can be treated as leading index of stock price index [19]. Dennis et al. treated VIX is a proxy variable of stock return volatility [20]. When the VIX soars and the market shows extreme fear, it would be the best opportunity to enter the stock market.

This research includes the VIX by Chicago Board Options Exchange (COBE) as a proxy explanatory variable for investor sentiment in an empirical study to understand the impact of investor sentiment on the BER.

**2.3. Bitcoin.** Kurka studied the interaction between Bitcoin and traditional finance assets [1]. The results showed that BER was independent of any financial system assets. Though traditional financial assets had little impact on the BER



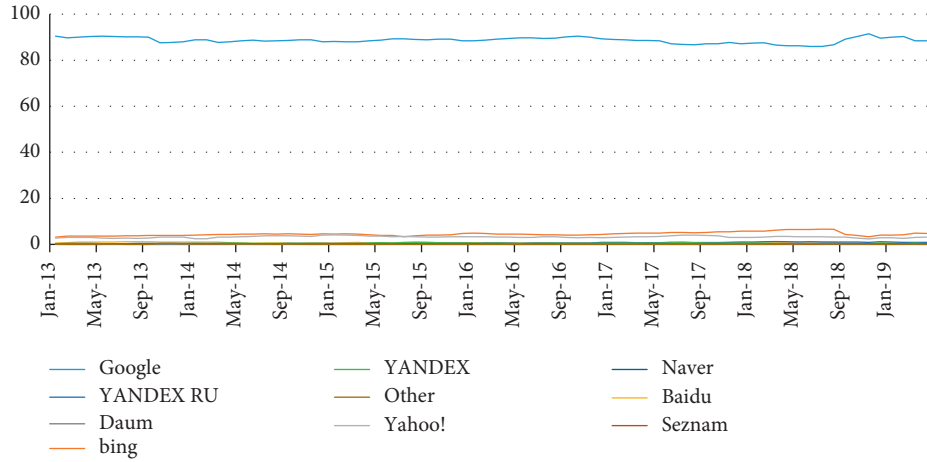


FIGURE 2: Google search engine market share worldwide (2013/1–2019/4, Source: <http://gs.statcounter.com>).

trend, the spillover effect of BER was significant in the traditional financial market.

Cai et al. adopted the Google SVI as the proxy explanatory variable for investors’ decisions [4]. The results supported that the SVI had a vital influence on the price and volume of Bitcoin. Dastgir et al. surveyed Google Trends search flow on Bitcoin [21]. It employed the Granger causality to test the causal relationship between the interest in Bitcoin and investment returns. The conclusion supported the existence of bidirectional causality between Bitcoin and investment returns.

This study analyzed the short- and long-term relationship between Google SVI and BER by the VAR model. The long-term relationship among BER, VIX, and the S&P500 index was also investigated for a co-integration phenomenon.

### 3. Methodology and Model

The research process was divided into two aspects as shown in Figure 3. For the short-term impact analysis, the VAR model and Granger causality were employed to test the explanatory variables of BER and verify their relationship, respectively. For the long-term impact, Nelson and Plosser argued that the variable difference was lost in the implicit information on the long-term equilibrium through the stationary sequence [22]. Therefore, the co-integration analysis and vector error correction model (VECM) validated the long-term relationship between the variables.

3.1. VAR Model and VECM. The VAR model in this study is defined as follows:

$$y_t = c + \sum_{i=1}^n \varphi_i y_{t-1} + \varepsilon_t, \quad (2)$$

where  $y_t$  is a  $(n \times 1)$  vector of endogenous variables,  $c = (c_1, \dots, c_n)$  is the  $(n \times 1)$  intercept vector of the VAR,  $\varphi_i$  is the  $i$ -th  $(n \times n)$  matrix of autoregressive coefficients for  $i = 1, 2, \dots, n$ , and  $\varepsilon_t = (\varepsilon_{1t} \dots \varepsilon_{nt})$  is the  $(n \times 1)$  generalization of a white noise process [24].

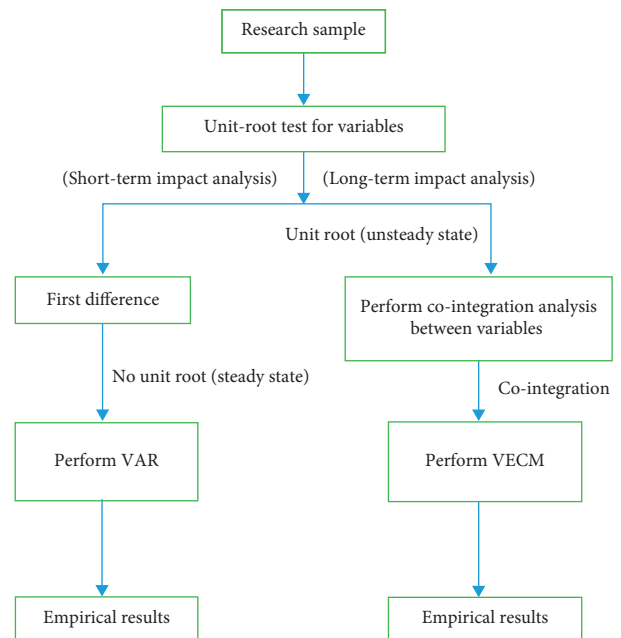


FIGURE 3: Flow chart of this research [23].

The following equation is the co-integration transformation of equation (2).

$$\Delta y_t = \Pi y_{t-1} + \sum_{i=1}^{p-1} \Phi \Delta y_{t-i} + \mu_t, \quad (3)$$

where  $\Pi = \sum_{i=1}^p \varphi_i - I$  and  $\Phi = \sum_{i=1}^{p-1} \varphi_{i+1}$ .

If  $y_t$  has a co-integration relationship, then  $\Pi y_{t-1} \sim I(0)$  and equation (3) is expressed as follows:

$$\Delta y_t = \alpha \beta' y_{t-1} + \sum_{i=1}^{p-1} \Phi \Delta y_{t-i} + \mu_t, \quad (4)$$

where  $\beta' y_{t-1} = ECM_{t-1}$  is the error correction term for a long-term equilibrium relationship between variables. Equation (4), then, becomes

$$\Delta y_t = \alpha \text{ECM}_{t-1} + \sum_{i=1}^{p-1} \Phi \Delta y_{t-i} + \mu_t. \quad (5)$$

Equation (5) is used for the vector error correction model (VECM) in which each equation belongs to an error correction model [25].

Before establishing the VAR model, it is necessary to choose the optimal lagged periods. The criteria include Akaike's Information Criterion (AIC), Hannan-Quinn Criterion (HQ), Schwarz Criterion (SC) likelihood ratio (LR), and so on. However, the Akaike proposed the AIC method and SC method which are most used [26]. The equations of the AIC and SC criteria are defined as follows:

$$\text{AIC} = \ln\left(\frac{\text{SSE}}{T}\right) + \left(\frac{2k}{T}\right), \quad (6)$$

$$\text{SC} = \ln\left(\frac{\text{SSE}}{T}\right) + \left(\frac{k}{T} \ln T\right), \quad (7)$$

where  $k$  is the number of all parameters to be estimated by the VAR model. The number of periods corresponding to the minimum value of the measurement of AIC or SC is the optimal number of lagged terms. The AIC is more consistent than SC. However, as the estimated parameters of the AIC are less and the number of samples is larger than the SC, the SC is better than the AIC for this research.

**3.2. Multivariate Granger Causality Analysis.** Multivariate Granger causality analysis is performed by fitting a VAR model to the time series. Let  $X(t) \in \mathfrak{R}^{d \times 1}$  ( $t = 1, \dots, T$ ) be a  $d$ -dimensional multivariate time series. Multivariate Granger causality is measured by fitting a VAR model to  $L$  time lags as follows:

$$X(t) = \sum_{\lambda=1}^L \Phi_{\lambda} X(t-\lambda) + \nu_t, \quad (8)$$

where  $\nu_t$  is a white Gaussian random vector and  $\Phi_{\lambda}$  is a matrix for every  $\lambda$ . A time series  $X_i$  is called a Granger cause of another time series  $X_j$ , if at least one of the elements  $\Phi_{\lambda}(i, j)$  is significantly larger than zero.

**3.3. Data Source and Sample Collection Period.** This research mainly discussed the key factors that affect the changes in BER by referring to Cai et al. [4], Kurka [1], and Hsieh [27]. The main variables included Google SVI, the VIX of Chicago Board of Exchange, S&P500 index, gold prices, US dollar index, and Japanese exchange rate. The period for sampling was from April 29, 2013, to June 22, 2018. The data sources were Yahoo Finance and Bloomberg. The data frequency was a day, and the data processing method did not affect the final results. If there was no observation value for any variable on the same day, the data of that day were deleted [28]. A total of 1,299 daily observation values for each variable were used.

## 4. Analysis Results and Discussion

The analysis was carried out in two processes as shown in Figure 4. We used the VAR model and Granger causality to test the explanatory stationary time series variables for the short-term impact. Then we used the co-integration analysis and vector error correction model (VECM) to validate the long-term relationship between the nonstationary variables. The descriptive statistics, VAR model analysis, and the co-integration relationship were performed and illustrated below.

**4.1. Descriptive Statistics.** Table 1 shows standard deviation, and maximum and minimum BER, gold price (GOLD), USD/JPY exchange rate (JPY), S&P500 index (SP500), Google SVI, US dollar index (USD), and VIX during the study period.

### 4.2. Short-Term VAR Analysis

**4.2.1. Unit Root Verification.** The Augmented Dickey-Fuller (ADF) and the Phillips-Perron (PP) tests ([29, 30]) were applied to the unit root sequence in this study. By taking the natural logarithm and first-order difference, all the variables were in a significant stationary sequence as shown in Table 2.

**4.2.2. Optimal Lagging Period Selection.** The research employed the AIC to determine the optimal number of two lagged periods. Table 3 shows the details of the optimal lagged periods.

**4.2.3. Results of Analysis of the VAR Model.** The first-order difference in the variables presents a stationary sequence and is used to analyze the short-term influence of the variables with the VAR model. The results are shown in Table 4. BER of the two lagged periods had a positive impact (0.108278) on the overall BER at a significant level of 1%. The Google SVI had a negative effect (−0.100983) on the BER at a significant level of 1%, while Google SVI has a positive impact (0.064947) on the BER at a significant level of 1%. The S&P500 index had a negative impact (−0.108234) on the leading two-period S&P500 index at a significant level of 1%.  $p < 1, 5, \text{ or } 10\%$  of the significant level indicates that the coefficient is not equal to zero, that is, the S&P500 index Granger causality causes the lag two-period S&P500 index.

The Granger causality diagram of the short-term variables affected the BER. The results demonstrated that the feedback relationship appeared only in the BER and the Google SVI among the Granger causality as shown in Figure 4.

### 4.3. Long-Term Co-Integration Analysis

**4.3.1. Johansen Maximum Likelihood Co-Integration Test.** Nelson and Plosser supposed that a different process caused the time series to lose long-term information [22]. That is, all nonstationary time series become stationary after the

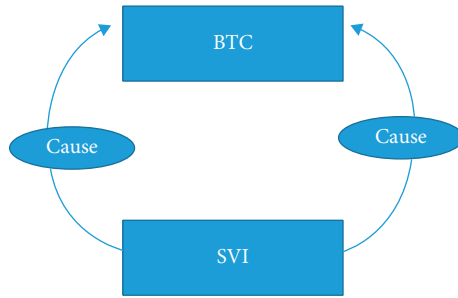


FIGURE 4: Granger causality between short-term variables.

TABLE 1: Variable raw data statistics.

Variables	Average value	Standard deviation	Minimum value	Maximum value
BTC	1916.304	3281.969	69.66000	18972.32
SVI	7.727833	13.16746	1.000000	100.0000
VIX	14.62402	3.864276	9.140000	40.74000
SP500	2130.599	303.6529	1573.090	2872.870
GOLD	1252.709	76.27864	1049.400	1469.250
USD	86.37288	6.655887	74.52690	96.86650
JPY	110.0013	7.632938	94.21000	125.2200

TABLE 2: The natural logarithm of each variable and the first-order difference unit root verification results.

Variables	ADF		PP	
	P	T	P	T
BTC	0.0000***	-18.13386***	0.0000***	-38.34844***
GOLD	0.0000***	-37.81181***	0.0000***	-37.76925***
JPY	0.0000***	-35.90829***	0.0000***	-35.95482***
SP500	0.0000***	-36.56911***	0.0000***	-36.69663***
SVI	0.0000***	-35.99486***	0.0000***	-36.22507***
USD	0.0000***	-34.30820***	0.0000***	-34.47007***
VIX	0.0000***	-37.13990***	0.0001***	-42.94997***

Significant level of \*\*\*1%.

TABLE 3: The optimal lagged periods selected by stationary sequence.

Lagged periods	Information criterion	
	AIC	SC
0	-35.68981	-35.66180*
1	-35.87884	-35.65473
2	-35.88120*	-35.46100
3	-35.86150	-35.24522
4	-35.83668	-35.02430

\*Selected lagged periods.

first-order difference operation, so it is used to study the short-term influence between variables.

Johansen co-integration test was used to test long-term relationship among the variables, and the Trace test to obtain the number of co-integration vector (CE) equation sets for variables of no difference ([31, 32]). As shown in Table 6, the hypothesis assumes that there is none CE, and the trace statistics shows 226.9521 and larger than the critical value 134.6780 which means reject that there is none CE. Therefore, Table 5 shows that there are two sets of co-integration equations under 5% critical value that are

most relevant to the goodness-of-fit test because the trace statistics shows 75.8209 and smaller than the critical value 76.9727 at most 2, which means that we cannot reject that there are two co-integration vectors.

4.3.2. *Vector Error Correction Estimates Model.* Through the Johansen Co-integration test, two sets of co-integration vector equations were obtained by using the trace test method. The co-integration vector equation and the VECM are shown in Table 6. The co-integration vector equations

TABLE 4: Results of analysis by the VAR model.

		BTC	GOLD	JPY	SP500	SVI	USD	VIX
BTC (-2)	Coefficient	0.108278***	-0.005290	-0.000852	-0.001416	-0.100983***	-8.92E-05	-0.001668
	Standard deviation	0.02758	0.00859	0.00217	0.00294	0.04277	0.00089	0.02952
	t value	3.92650	-0.61583	-0.39197	-0.48229	-2.36124	-0.09973	-0.05650
GOLD (-2)	Coefficient	-0.089612	-0.064190**	-0.005214	-0.013245	0.048425	0.005545*	0.073877
	Standard deviation	0.09084	0.02830	0.00716	0.00967	0.14088	0.00295	0.09724
	t value	-0.98646	-2.26838	-0.72842	-1.36920	0.34373	1.88266	0.75975
JPY (-2)	Coefficient	-0.533897	-0.106320	0.018735	-0.068497	-0.098955	0.014686	0.520942
	Standard deviation	0.37632	0.11723	0.02965	0.04007	0.58362	0.01220	0.40282
	t value	-1.41873	-0.90697	0.63185	-1.70925	-0.16955	1.20365	1.29325
SP500 (-2)	Coefficient	0.133665	0.068168	0.007063	-0.108234**	0.384718	-0.001683	0.375067
	Standard deviation	0.45540	0.14186	0.03588	0.04850	0.70627	0.01477	0.48747
	t value	0.29351	0.48053	0.19685	-2.23182	0.54472	-0.11399	0.76942
SVI (-2)	Coefficient	0.064947***	0.002600	-0.000551	-0.002003	0.006269	0.000247	0.026340
	Standard deviation	0.01803	0.00562	0.00142	0.00192	0.02797	0.00058	0.01930
	t value	3.60159	0.46282	-0.38802	-1.04325	0.22417	0.42245	1.36461
USD (-2)	Coefficient	0.599314	-0.362583	0.019072	0.057129	0.452971	0.003090	-1.319591
	Standard deviation	0.90763	0.28273	0.07151	0.09665	1.40761	0.02943	0.97154
	t value	0.66031	-1.28242	0.26669	0.59107	0.32180	0.10501	-1.35825
VIX (-2)	Coefficient	-0.010045	-0.004217	-0.000970	-0.007474	0.007971	-0.000770	-0.026743
	Standard deviation	0.04539	0.01414	0.00358	0.00483	0.07039	0.00147	0.04858
	t value	-0.22130	-0.29825	-0.27119	-1.54626	0.11323	-0.52347	-0.55045
C	Coefficient	0.002676	1.58E-05	9.02E-05	0.000486**	0.001907	1.74E-05	-0.000270
	Standard deviation	0.00205	0.00064	0.00016	0.00022	0.00318	6.7E-05	0.00220
	t value	1.30416	0.02475	0.55767	2.22614	0.59935	0.26144]	-0.12296]
R squared		0.025671	0.007505	0.001804	0.009159	0.004947	0.004295	0.008862
Adj. R squared		0.020376	0.002111	-0.003621	0.003774	-0.000461	-0.001116	0.003476

Significant level of \*\*\*1%, \*\*5%, \*10% and the coefficients means the estimated coefficients.

TABLE 5: Johansen maximum likelihood co-integration test result.

Hypothesized no. of CE(s)	Eigenvalue	Trace statistics	Critical value at $\alpha=0.05$	Probability
None <sup>@</sup>	0.065359	226.9521	134.6780	0.0000***
At most 1 <sup>@</sup>	0.04826	139.6904	103.8473	0.0000***
At most 2	0.0282	75.8209	76.9727	0.0610
At most 3	0.0144	38.8515	54.0790	0.5287
At most 4	0.0077	20.0129	35.1927	0.7259
At most 5	0.0055	9.9263	20.2618	0.6471
At most 6	0.0020	2.6882	9.16454	0.6404

\*\*\*Significant level of 1%; <sup>@</sup>the number of co-integrated equations.

show that equations (2) and (3) are based on the BER and Google SVI, respectively. The coefficients of the equation variables are highly significant to each other.

**4.3.3. Long-Term Granger Causality.** Granger causality is a statistical concept of causality that is based on VAR. The coefficient of granger causality test is assumed that does not influence between two variables under null hypothesis as

shown in Table 7. If the probability is larger than 0.1, then we do not reject the null hypothesis, and these two variables do not have granger causality. The results of the granger causality test are shown in Table 7. The BER influenced the S&P500 index and Google SVI. The Google SVI Granger caused the BER index. Therefore, the BER and SVI presents intergranger causality. The S&P500 SVI Granger influenced Google SVI. The VIX Granger contributed to the S&P500 index, the BER index, and the Google SVI.

TABLE 6: Co-integration vector equation and the VECM.

Co-integrating equation		Equation (2)	Equation (3)	
BTC		1.0000	0.0000	
SVI		0.0000	1.0000	
	Coefficient	16.0725***	11.3168***	
VIX	Standard deviation	2.2836	1.5958	
	t value	7.0381	7.0912	
C		-49.2158	-31.4336	
Error correction:		D(BTC)	D(SVI)	D(VIX)
	Coefficient	-0.0139***	0.0208***	-0.0051
Co-integrating equation (2)	Standard deviation	0.0037	0.0057	0.0039
	t value	-3.7744	3.6354	-1.2972
	Coefficient	0.0186***	-0.0333***	0.0021
Co-integrating equation (3)	Standard deviation	0.0052	0.0081	0.00560
	t value	3.5346	-4.0969	0.3765
	Coefficient	0.1037***	-0.1032***	-0.0081
BTC	Standard deviation	0.0274	0.0423	0.0291
	t value	0.7826	-2.4376	-0.2786
	Coefficient	0.0527***	0.0208	0.0192
SVI	Standard deviation	0.0181	0.0280	0.0193
	t value	2.8992	0.7410	0.9942
	Coefficient	-0.0098	0.0028	-0.0388
VIX	Standard deviation	0.0262	0.04047	0.02785
	t value	-0.3752	0.0698	-1.3941
	Coefficient	0.0027	0.0020	-5.13E-05
C	Standard deviation	0.0020	0.0031	0.0021
	t value	1.3436	0.6529	-0.0237
R squared		0.0353	0.0225	0.0325
Adj. R squared		0.0316	0.0187	0.0288

Significant level of \*\*\*1%, \*\*5%, and \*10%.

TABLE 7: Granger causality test results for the variables in natural logarithm.

Null hypothesis	Prob.
GOLD does not influence BTC	0.9397
BTC does not influence GOLD.	0.3118
JPY does not influence BTC	0.5978
BTC does not influence JPY	0.9319
SP500 does not influence BTC	0.2092
BTC does not influence SP500	0.0325**
SVI does not influence BTC	4.E-05***
BTC does not influence SVI	0.0018***
USD does not influence BTC	0.3888
BTC does not influence USD	0.8713
VIX does not influence BTC	0.0373**
BTC does not influence VIX	0.1173
SVI does not influence SP500	0.4206
SP500 does not influence SVI	0.0469**
VIX does not influence SP500	0.0098***
SP500 does not influence VIX	0.2837
VIX does not influence SVI	0.0074***
SVI does not influence VIX	0.3751

Significant level of \*\*\*1%, \*\*5%, \*10%.

Figure 5 shows all variables that affect BER in the long-term. The “feedback relationship” of the Granger causality appears in BER and the Google SVI. The VIX has a moderately significant Granger causality on the Bitcoin price in the long term.

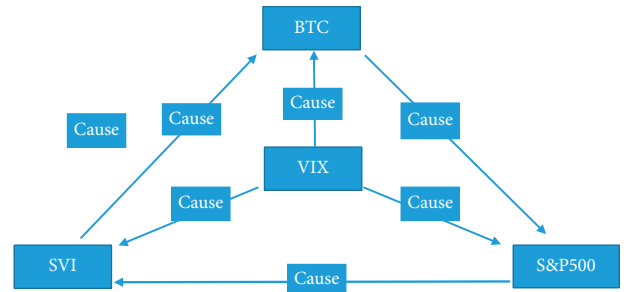


FIGURE 5: Granger Causality diagram among long-term variables.

### 5. Conclusions

The analysis results of this research suggested the co-integrated equilibrium and feedback relationship of the Granger causality between BER and Google SVI in the long term. The VIX influences mostly the BER in the long-term co-integration.

In addition to confirming the results of Cai et al. [4] and Dastgir et al. [21], the result confirmed the Granger causality between Google SVI and BER and the existence of the short- and long-term feedback between Google SVI and the BER. The result also showed that there is a long-term co-integration relationship between the BER, VIX, and Google SVI. By observing Google SVI and the VIX, investors predict the future trend of the BER as BER has a long-term leading relationship with the S&P500 index. The



crypto-economy affects the real economy as the spillover effect of BER influences the S&P500 [1]. When BER rises, the S&P index should be viewed with optimism. Also, the VIX has a long-term leading relationship with BER as it drops with the rise of BER. The co-integration equilibrium between BER and Google SVI in the long-term leads to the feedback relationship of the Granger causality, which impacts Bitcoin price and causes the following: the prohibition of the Bitcoin trading exchange, the issuer's implementation of hard fork policy that damages investors' rights and confidence with no compensation, and information security issues. The price of Bitcoin fell back to the equilibrium value, which verifies the investor attention theory. The results help investors hedging or arbitrating engaged Bitcoin-related risks.

### Data Availability

The data used to 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 conflicts of interest.

### References

- [1] J. Kurka, "Do cryptocurrencies and traditional asset classes influence each other?" *Finance Research Letters*, vol. 31, pp. 38–46, 2019.
- [2] B. M. Barber and T. Odean, "All that glitters: the effect of attention and news on the buying behavior of individual and institutional investors," *Review of Financial Studies*, vol. 21, no. 2, pp. 785–818, 2008.
- [3] R. C. Merton, "A simple model of capital market equilibrium with incomplete information," *The Journal of Finance*, vol. 42, no. 3, pp. 483–510, 1987.
- [4] Z. Cai, A. Liu, E. Lim, C.-W. Tan, and Z. Zheng, "Unraveling the effects of google search on volatility of cryptocurrencies," in *Proceedings of the 39th International Conference on Information Systems (ICIS)*, Association for Information Systems. AIS Electronic Library (AISeL), San Francisco, CA, USA, December 2018.
- [5] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.
- [6] C. W. J. Granger, "Some properties of time series data and their use in econometric model specification," *Journal of Econometrics*, vol. 16, no. 1, pp. 121–130, 1981.
- [7] C. W. J. Granger and P. Newbold, "Spurious regressions in econometrics," *Journal of Econometrics*, vol. 2, no. 2, pp. 111–120, 1974.
- [8] A. Urquhart and H. Zhang, "Is bitcoin a hedge or safe haven for currencies? an intraday analysis," *International Review of Financial Analysis*, vol. 63, pp. 49–57, 2019.
- [9] J. Bleher and T. Dimpfl, "Today I got a million, tomorrow, I don't know: on the predictability of cryptocurrencies by means of Google search volume," *International Review of Financial Analysis*, vol. 63, pp. 147–159, 2019.
- [10] A. Aouadi, M. Arouri, and F. Teulon, "Investor attention and stock market activity: evidence from France," *Economic Modelling*, vol. 35, pp. 674–681, 2013.
- [11] N. Vlastakis and R. N. Markellos, "Information demand and stock market volatility," *Journal of Banking & Finance*, vol. 36, no. 6, pp. 1808–1821, 2012.
- [12] C. H. Huang, "Can google predict the stock return in Taiwan?" Master's Thesis for the Finance Dept, College of Management, National Taiwan University, Taiwan, China, 2013.
- [13] F. Takeda and T. Wakao, "Google search intensity and its relationship with returns and trading volume of Japanese stocks," *Pacific-Basin Finance Journal*, vol. 27, pp. 1–18, 2014.
- [14] Y. Zhang, W. Song, D. Shen, and W. Zhang, "Market reaction to internet news: information diffusion and price pressure," *Economic Modelling*, vol. 56, pp. 43–49, 2016.
- [15] M. Qadan, D. Kliger, and N. Chen, "Idiosyncratic volatility, the VIX and stock returns," *The North American Journal of Economics and Finance*, vol. 47, pp. 431–441, 2019.
- [16] P. R. Tsai, Y. C. Wang, and C. Z. Chang, "Study on the investor sentiment, firm characteristics, and stock returns in Taiwan," *Taipei Economic Inquiry*, vol. 45, pp. 273–322, 2017.
- [17] D. P. Simon and R. A. Wiggins, "S & P futures returns and contrary sentiment indicators," *Journal of Futures Markets*, vol. 21, no. 5, pp. 447–462, 2001.
- [18] A. S. Lee, "Time-varying relationship of news sentiment, implied volatility and stock returns," *Applied Economics*, vol. 48, pp. 4942–4960, 2016.
- [19] M. M. Copeland and T. E. Copeland, "Market timing: style and size rotation using the VIX," *Financial Analysts Journal*, vol. 55, no. 2, pp. 73–81, 1999.
- [20] P. Dennis, S. Mayhew, and C. Stivers, "Stock returns, implied volatility innovations, and the asymmetric volatility phenomenon," *Journal of Financial and Quantitative Analysis*, vol. 41, no. 2, pp. 381–406, 2006.
- [21] S. Dastgir, E. Demir, G. Downing, G. Gozgor, and C. K. M. Lau, "The causal relationship between bitcoin attention and bitcoin returns: evidence from the copula-based granger causality test," *Finance Research Letters*, vol. 28, pp. 160–164, 2019.
- [22] C. Nelson and C. Plosser, "Trends and random walks in macroeconomic time series: some evidence and implications," *Journal of Monetary Economics*, vol. 10, no. 2, pp. 139–162.
- [23] G. Hondroyannis and E. Papapetrou, "Macroeconomic influences on the stock market," *Journal of Economics and Finance*, vol. 25, no. 1, pp. 33–49, 2001.
- [24] C. Sims, "Macroeconomics and reality," *Econometrica*, vol. 48, pp. 1–47, 1980.
- [25] R. F. Engle and C. W. J. Granger, "Co-integration and error correction: representation, estimation, and testing," *Econometrica*, vol. 55, no. 2, pp. 251–276, 1987.
- [26] H. Akaike, "Information theory as an extension of the maximum likelihood principle," in *Springer Series in Statistics Book Series*, B. Petrov, F. Csaki, and A. Kiado, Eds., Springer, Budapest, Hungary, 1973.
- [27] C. J. Hsieh, "An empirical analysis of bitcoin exchange rate," Master's Thesis for the EMBA Program, Department of Economics, National Taiwan University, Taiwan, China, 2017.
- [28] Y. Hamao, R. W. Masulis, and V. Ng, "Correlations in price changes and volatility across international stock markets," *Review of Financial Studies*, vol. 3, no. 2, pp. 281–307, 1990.
- [29] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366a, pp. 427–431, 1979.

- [30] P. C. B. Phillips and P. Perron, "Testing for a unit root in time series regression," *Biometrika*, vol. 75, no. 2, pp. 335–346, 1988.
- [31] S. Johansen, "Statistical analysis of cointegration vectors," *Journal of Economic Dynamics and Control*, vol. 12, no. 2-3, pp. 231–254, 1988.
- [32] S. Johansen and K. Juselius, "Maximum likelihood estimation and inference on cointegration - with applications to the demand for money," *Oxford Bulletin of Economics and Statistics*, vol. 52, no. 2, pp. 169–210, 1990.

## Research Article

# Integrated Image Sensor and Light Convolutional Neural Network for Image Classification

Cheng-Jian Lin <sup>1,2</sup>, Chun-Hui Lin,<sup>3</sup> and Shyh-Hau Wang<sup>3,4</sup>

<sup>1</sup>Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan

<sup>2</sup>College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan

<sup>3</sup>Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan

<sup>4</sup>Intelligent Manufacturing Research Center, National Cheng Kung University, Tainan 701, Taiwan

Correspondence should be addressed to Cheng-Jian Lin; [cjlin@ncut.edu.tw](mailto:cjlin@ncut.edu.tw)

Received 27 January 2021; Revised 1 March 2021; Accepted 10 March 2021; Published 17 March 2021

Academic Editor: Teen-Hang Meen

Copyright © 2021 Cheng-Jian Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning has accomplished huge success in computer vision applications such as self-driving vehicles, facial recognition, and controlling robots. A growing need for deploying systems on resource-limited or resource-constrained environments such as smart cameras, autonomous vehicles, robots, smartphones, and smart wearable devices drives one of the current mainstream developments of convolutional neural networks: reducing model complexity but maintaining fine accuracy. In this study, the proposed efficient light convolutional neural network (ELNet) comprises three convolutional modules which perform ELNet using fewer computations, which is able to be implemented in resource-constrained hardware equipment. The classification task using CIFAR-10 and CIFAR-100 datasets was used to verify the model performance. According to the experimental results, ELNet reached 92.3% and 69%, respectively, in CIFAR-10 and CIFAR-100 datasets; moreover, ELNet effectively lowered the computational complexity and parameters required in comparison with other CNN architectures.

## 1. Introduction

Convolutional neural network (CNN) was firstly introduced in the 1980s. At that time, Lecun et al. [1] proposed a simply constructed CNN architecture which contains three convolutional layers, two subsampling layers, and a fully connected layer. LeNet was mainly used for handwriting recognition in the MNIST dataset and obtained the lowest error rate. However, the hardware equipment was not advanced, and graphics processing units had not been invented which led to the development of CNN being greatly restricted. In 2012, Krizhevsky et al. [2] developed AlexNet and won the first place in the ImageNet large-scale visual recognition competition by achieving a top-5 error of 15.3%. Compared with LeNet, AlexNet uses rectified linear unit (ReLU) to replace the conventional sigmoid activation function in order to resolve the vanishing gradient problem. Moreover, the dropout [3] regularization technique was also

introduced to reduce overfitting in neural networks. In general, AlexNet extends its network architecture resulting in the requirement of nearly 60 million parameters, and the floating-point operations (FLOPs) have reached 0.7 giga FLOPs. Subsequently, researchers have continued to deepen networks to improve the accuracy such as VGGNet [4].

Instead of deepening the CNN architecture, some researchers expand the width of the network architectures. For instance, Szegedy et al. [5] firstly came up with a concept of inception block in the CNN which encapsulates different sizes of kernels for extracting global and local features. It adjusts the computations by adding a bottleneck layer of a  $1 \times 1$  convolutional filter before applying large-size kernels. Furthermore, Srivastava et al. [6] designed a new architecture to moderate gradient-based training of very deep networks which is called highway network. This network imitates the horizontal expansion concept using the gating function to adaptively bypass the input so that the network can go deeper. In addition,

He et al. [7] proposed ResNet by taking inspiration from the bypass and bottleneck layer approaches for reducing the amount of operations. Many improved designs of network architectures are proposed and applied in many applications, such as object detection [8] and semantic analysis [9]. However, regardless of deepening or widening the network architectures, high computational cost and memory requirement are the two main concerns observed with these architectures.

To further alleviate these two primary concerns of the network, designing a lightweight architecture without compromising the performance is necessary, especially when the CNN model is implemented in resource-constrained hardware. Howard et al. [10] adopted depthwise separable convolution in the MobileNet to reduce the model parameters so that the model can be embedded in portable devices for mobile and embedded vision applications. Juefei-Xu et al. [11] proposed the local binary convolutional neural network which adopts local binary convolution (LBC) as a substitute for the conventional CNN. The experimental results showed that the LBC module performs a good approximation of a conventional convolutional layer and results in a major reduction in the number of learnable parameters while training the network. Iandola et al. introduced SqueezeNet [12] which replaces  $3 \times 3$  filters with  $1 \times 1$  filters and decreases the number of input channels to  $3 \times 3$  filters. These strategies are desirable to decrease the quantity of parameters in a CNN while attempting to maintain accuracy. According to the experimental results reported by Iandola et al., the parameters used in SqueezeNet are 50x fewer than those in AlexNet; besides, it preserves AlexNet-level accuracy on ImageNet. Others such as parameter pruning and quantization can reduce redundant parameters which reduces the network complexity and addresses the overfitting problem. Furthermore, without decreasing accuracy, more improvements of YOLO were also proposed [13, 14] to prove that light CNN can reduce training time and make applications more diverse without being limited by hardware.

The three modules provide capabilities and advantages: saving computations when kernel size and the number of kernels are large using depthwise separable convolution, expanding the field of view (FOV) of filters without increasing parameters by atrous convolution, and extracting local and global features simultaneously adopted by the inception module to reduce the parameters and operations of the CNN. In this study, the proposed model, efficient light convolutional neural network (ELNet) with the three modules, is no longer limited by memory and computational constraints.

The rest of the paper is organized as follows. In Section 2, the conventional CNN architecture is briefly reviewed. The ELNet is introduced in Section 3. The experimental results using CIFAR-10 and CIFAR-100 datasets are revealed in Section 4 and compared with other state-of-the-art CNN architectures such as GoogLeNet, ResNet-50, and MobileNet. Lastly, Section 5 draws conclusions.

## 2. Convolutional Neural Network (CNN)

The concept of neural networks mainly comes from biological neural network systems; however, neural networks

are connected in a fully connected manner which causes a great amount of calculations when the input size is large. Therefore, in the 1980s, convolution kernel was first introduced and then was widely applied in image processing. There are four main parts of the CNN: convolutional layer, pooling layer, activation function, and fully connected layer. The function of feature extraction depends on the first three parts, and the fully connected layer is used to classify the obtained features. More descriptions of these parts are explained as follows.

**2.1. Convolutional Layer.** A convolutional layer consists of a set of learnable filters (or kernels) which have a small receptive field; however, feature extraction can be acquired by extending filters through the full depth of the input volume. The formula is as follows:

$$O_{r,c} = \sum_{k=1}^n \sum_{i=1}^{k_h} \sum_{j=1}^{k_w} I_{r-k_h+i, c-k_w+j}^k \times W_{i,j}^k + b, \quad (1)$$

where  $r$  and  $c$  represent the row and column of the feature map,  $n$  is the number of input channels,  $k_w$  and  $k_h$  are the width and height of a convolution kernel,  $W_{i,j}^k$  is the weight of the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column convolution kernel in the  $k^{\text{th}}$  channel,  $I_{i,j}^k$  is the input of the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column in the  $k^{\text{th}}$  channel, and  $b$  is the bias.

**2.2. Pooling Layer.** In order to effectively extract features, most of the moving strides are set as 1; yet, this setting causes relatively more operations. Therefore, pooling layer is usually added in the CNN for effectively reducing the amount of operations. Equation (2) shows the calculation of max pooling and average pooling:

$$O_{r,c} = \begin{cases} \text{max pooling} \left( \max(I_{i,j}) \mid \begin{array}{l} r \leq i < r + P_h \\ c \leq j < c + P_w \end{array} \right), \\ \text{average pooling} \left( \frac{\sum_{i=r}^{r+P_h} \sum_{j=c}^{c+P_w} I_{i,j}}{P_w \times P_h} \right), \end{cases} \quad (2)$$

where  $O_{r,c}$  is the output row and column,  $I_{i,j}$  is the row and column of the input image, and  $P_w$  and  $P_h$  are the width and height of the pooling kernel.

**2.3. Activation Function.** The conventional operation of the convolution kernel is a linear operation; LeNet adopts sigmoid function as an activation function to solve nonlinear problems. Along with the development of deeper network, researchers found out that gradient disappearance occurs when the sigmoid function approaches to 0 in the saturation region. Then, ReLU is introduced in AlexNet to address this problem. Moreover, the operations using ReLU are simpler than those of the sigmoid function. Later, many scholars made various improvements based on ReLU and sigmoid functions. For instance, Leaky ReLU [15] solves the problem

that ReLU is not activated when  $x$  is less than 0, PReLU [16] adds a parameter to make ReLU more accurate when  $x$  is less than 0, and RReLU [17] learns parameters automatically via the neural network. Here, PReLU is selected as the activation function which is shown in Figure 1, and its equation is given as follows:

$$f_{\beta}(x) = \begin{cases} \beta \cdot x, & x \leq 0, \\ x, & x > 0. \end{cases} \quad (3)$$

**2.4. Fully Connected Layer.** After convolutional computation, the high-dimensional feature maps will be classified and predicted through a fully connected neural network. This layer is often used in many network architectures such as LeNet, AlexNet, and GoogLeNet. The equation is given as follows:

$$P = O_c \times (I_c + 1), \quad (4)$$

where  $I_c$  and  $O_c$  represent the number of input and output channels.

From equation (4), the number of parameters in the fully connected layer depends on the input dimensions. If dimension reduction is not performed, the number of input channels might be massive, and many parameters will be generated. According to Lin et al. [18], the fully connected layer is prone to overfitting which hampers the generalization ability of the overall network. Therefore, the later CNN architectures usually replace fully connected layers with global average pooling.

### 3. Efficient Light Convolutional Neural Network (ELNet)

An efficient light convolutional neural network (ELNet) is proposed to make the network architecture suitable for resource-constrained hardware. A schematic view of the network is depicted in Figure 2, where the red block is a depthwise separable convolution, the black dash line block represents an inception module, and the brown block is a

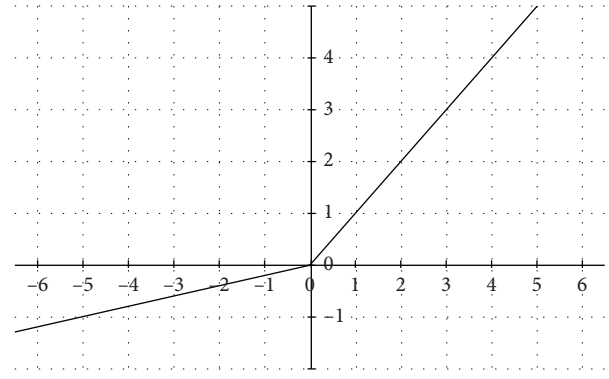


FIGURE 1: PReLU.

depthwise separable convolution combining with atrous convolution. The details of the architecture are described in Table 1.

In Table 1, Conv dw represents a depthwise separable convolution, and  $d$  means stride in the atrous convolution. The three convolutional modules used in ELNet are described as follows.

**3.1. Depthwise Separable Convolution.** Depthwise separable convolution separates the original convolution into two parts for the purpose of reducing operations as shown in Figure 3.

Compared with the conventional convolution method, one convolutional kernel will generate only one feature map according to its input dimensions. However, depthwise separable convolution performs multiple feature maps corresponding to each dimension, and then a  $1 \times 1$  convolutional layer is used to combine all the feature maps into one output. Although there is no difference between the output of the depthwise separable convolution and conventional convolution, the parameters of the depthwise separable convolution using one  $3 \times 3$  convolutional kernel are much less than those of the conventional convolution method. The calculations are listed as follows:

$$\begin{aligned} \text{conventional convolution} &: I_w \times I_h \times I_c \times k_w \times k_h \times k_c, \\ \text{depthwise separable convolution} &: I_w \times I_h \times I_c \times k_w \times k_h + I_c \times k_c \times I_w \times I_h, \end{aligned} \quad (5)$$

where  $I_w$ ,  $I_h$ , and  $I_c$  represent the width, height, and channel of the input, respectively,  $k_w$  and  $k_h$  are the width and height of the convolutional kernel, and  $k_c$  is the number of convolutional kernels in the convolutional layer.

**3.2. Atrous Convolution.** Atrous convolution [9], as shown in Figure 4, enlarges the FOV of filters by incorporating the larger context without growing parameters. The advantages of using atrous convolution are allowing the user to filter a larger context instead of using a bigger size of kernel and reducing the usage of pooling layers which brings less

operation consumption and accuracy improvement; besides, using less parameters can also avoid an overfitting problem.

**3.3. Inception Module.** Inception module uses various convolution kernels to extract features so that the feature maps are able to contain local features and global features. The schematic view of conventional convolutional layers and inception module are displayed in Figure 5 as comparison. Although both of the methods can map to the same size of FOV, local features in Figure 5(a) might be washed out at the end. On the contrary, the wash-out problem will not be



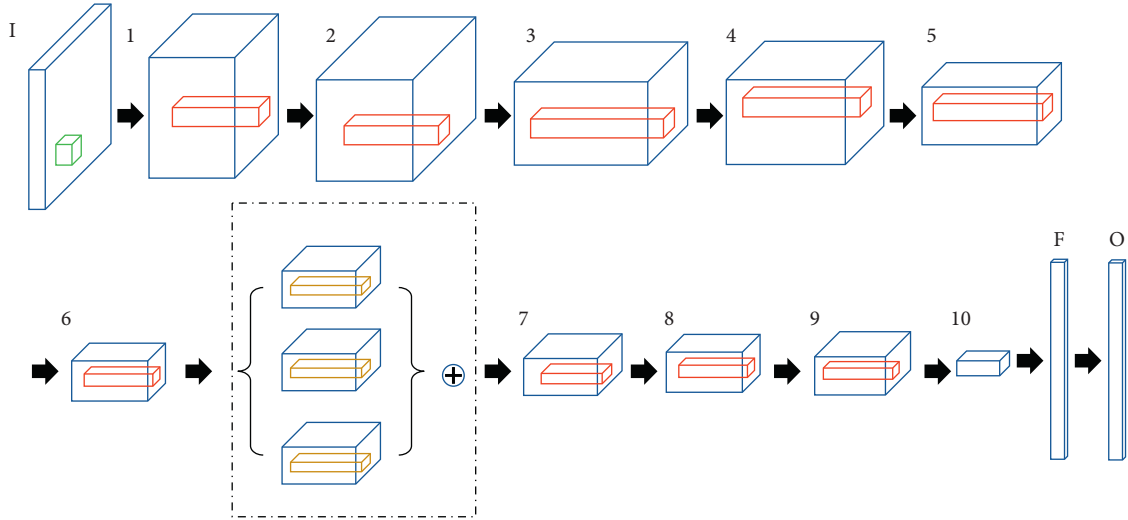


FIGURE 2: ELNet architecture.

TABLE 1: ELNet architecture.

	Type/stride	Filter shape ( $H \times W \times C \times N, d$ )	Input size
I	Conv/2	$3 \times 3 \times 3 \times 32$	$W \times H \times 3$
1	Conv dw/1	$3 \times 3 \times 32$	$W/2 \times H/2 \times 32$
	Conv/1	$1 \times 1 \times 32 \times 64$	$W/2 \times H/2 \times 32$
2	Conv dw/2	$3 \times 3 \times 64$	$W/2 \times H/2 \times 64$
	Conv/1	$1 \times 1 \times 64 \times 128$	$W/4 \times H/4 \times 64$
3	Conv dw/1	$3 \times 3 \times 128$	$W/4 \times H/4 \times 128$
	Conv/1	$1 \times 1 \times 128 \times 128$	$W/4 \times H/4 \times 128$
4	Conv dw/2	$3 \times 3 \times 128$	$W/4 \times H/4 \times 256$
	Conv/1	$1 \times 1 \times 128 \times 256$	$W/8 \times H/8 \times 256$
5	Conv dw/1	$3 \times 3 \times 256$	$W/8 \times H/8 \times 256$
	Conv/1	$1 \times 1 \times 256 \times 256$	$W/8 \times H/8 \times 512$
	Conv dw/2	$3 \times 3 \times 256$	$W/8 \times H/8 \times 256$
	Conv/1	$1 \times 1 \times 256 \times 512$	$W/16 \times H/16 \times 512$
6	(a) Atrous dw/1	$3 \times 3 \times 512, d = 1$	$W/16 \times H/16 \times 512$
	(b) Atrous dw/1	$3 \times 3 \times 512, d = 2$	
	(c) Atrous dw/1	$3 \times 3 \times 512, d = 3$	
	Add	$(a) + (b) + (c)$	$W/16 \times H/16 \times 512$
7	Conv/1	$1 \times 1 \times 512 \times 512$	$W/16 \times H/16 \times 512$
8	Conv dw/2	$3 \times 3 \times 512$	$W/16 \times H/16 \times 512$
	Conv/1	$1 \times 1 \times 512 \times 1024$	$W/32 \times H/32 \times 512$
9	Conv dw/1	$3 \times 3 \times 1024$	$W/32 \times H/32 \times 1024$
	Conv/1	$1 \times 1 \times 1024 \times 1024$	$W/32 \times H/32 \times 1024$
10	Average pooling	Global pooling	$W/32 \times H/32 \times 1024$
F	Fully connected	$1024 \times \text{Classes}$	$1 \times 1024$
O	Softmax	Classification answer	$1 \times \text{Class Numbers}$

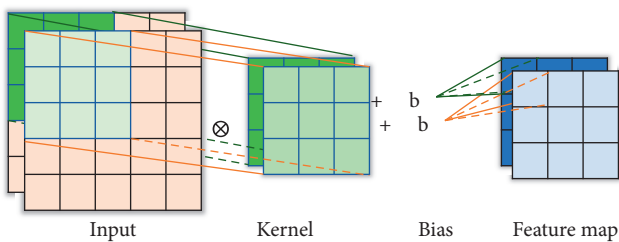


FIGURE 3: Depthwise separable convolution.

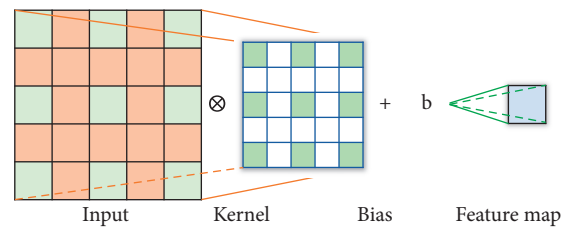


FIGURE 4: Atrous convolution.

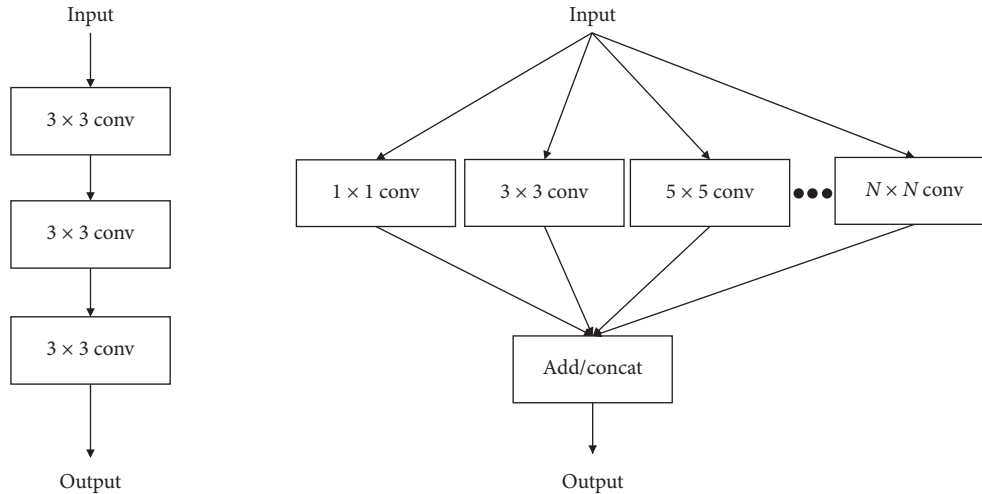


FIGURE 5: (a) Conventional convolutional layers. (b) Inception module.

considered when using the inception module (Figure 5(b)); however, fusing multiple feature maps is another question. In general, concatenation (Concat) and addition (Add) are two common methods; the former can retain characteristics of each convolution output but produce high-dimensional problems; in contrast, the latter does not have dimensional problems, yet relatively might lose the independence of each output.

#### 4. Results and Discussion

To deploy the systems on resource-constrained hardware for real-time data processing, large-scale datasets such as PASCAL VOC, ImageNet, and COCO are not considered. Thus, CIFAR-10 and CIFAR-100, two well-understood and widely used datasets, were provided to verify the performance of ELNet. The experimental results including parameters, FLOPs, and accuracy were compared, respectively, with the other state-of-the-art CNN architectures such as GoogLeNet [5], ResNet-50 [7], MobileNet [10], and All Convolutional Net (All-CNN-C) [19]. The hardware specifications and predefined parameters used in this study are listed in Tables 2 and 3.

**4.1. CIFAR-10 Dataset.** The CIFAR-10 dataset includes 60,000 colour images with the size of  $32 \times 32$  in a total of 10 classes. To fit into the proposed network, bilinear interpolation is used to resize the images into  $224 \times 224$  which provides more features than using the padding method. Table 4 shows the results in which the parameters and MFLOPs required in larger CNN models such as GoogLeNet and ResNet-50 models are very large. In other words, these models need longer training time and higher operations. To make the model suitable for general hardware equipment, models with less operations and lower complexity are more favourable. Therefore, the proposed model is also compared with MobileNet and All-CNN-C which are also called light models. According to the results, MobileNet uses less parameters and MFLOPs than others;

TABLE 2: Hardware specifications.

Hardware	Specification
GPU	NVidia GTX1080-Ti 11G
CPU	Intel Xeon E3-1225 v3 @ 3.2 GHz

TABLE 3: Predefined parameters.

Parameter	Value
Epoch	120
Optimizer	Nesterov's accelerated gradient
Learning rate	0.01
Learning rate decay	0.9
Learning rate decay frequency	40 (epochs/time)
Momentum	0.9
Batch size	100

yet, the accuracy is lower than that of ELNet. Even though All-CNN-C has the least parameter requirements, its MFLOPs are the highest which means the training time could be decreased by using better graphics processing units, but this increases the cost of hardware equipment. ELNet reaches a tradeoff between accuracy and parameters/MFLOPs which is closer to the purpose of this study than that of other methods.

**4.2. CIFAR-100 Dataset.** The CIFAR-100 dataset contains 100 classes which are more than in the CIFAR-10 dataset. Therefore, the accuracy shown in Table 5 is obviously relatively lower than the accuracy of classifying the CIFAR-10 dataset; yet, the accuracy of ELNet is still the highest.

To evaluate the effectiveness of three convolutional modules used in ELNet, Tables 6 and 7 show the results of classifying the CIFAR-100 dataset. Table 6 shows that using atrous convolution can not only widen the FOV which increases the accuracy from 67% to 69% but also reach the same accuracy (69%) as using a bigger kernel size. Additionally, the inception module has the ability to extract

TABLE 4: Experimental results using the CIFAR-10 dataset.

Model	Parameter ( $10^6$ )	MFLOPs	Accuracy (%)
GoogLeNet [5]	6.9	1,582	83.1
ResNet-50 [7]	25.6	3,857	88.1
MobileNet [10]	4.2	569	85.6
All-CNN-C [19]	1.37	13,965	90.9
ELNet (proposed)	2.1	257	92.3

TABLE 5: Experimental results using the CIFAR-100 dataset.

Model	Accuracy (%)
GoogLeNet [5]	56
ResNet-50 [7]	57.3
MobileNet [10]	65
All-CNN-C [19]	66.3
ELNet (proposed)	69

TABLE 6: The comparisons of using the atrous convolution.

Model	Parameter ( $10^6$ )	MFLOPs	Accuracy (%)
ELNet (no atrous convolution)	2.1	257	67
ELNet ( $7 \times 7$ kernel size)	2.2	262	69
ELNet	2.1	257	69

TABLE 7: The comparisons of using the inception module.

Model	Parameter ( $10^6$ )	MFLOPs	Accuracy (%)
ELNet (Add)	2.1	257	69
ELNet (Concat)	2.6	359	69.4
ELNet (no inception module)	2.1	257	68

features using different convolution kernel sizes. In order to keep the features, different fusion methods may display distinct results. From the experimental results (Table 7), concatenation shows better accuracy than the other two methods; however, it requires more parameters and MFLOPs; thus, the addition method might be the better choice for implementing the network in a resource-constrained environment.

Overall, the proposed ELNet showed better performance in comparison with either relatively larger CNN architectures (GoogLeNet and ResNet-50) or light CNN architectures (MobileNet and All-CNN-C). The accuracy of ELNet is acceptable if the environment of the deployed system is considered. Although the proposed ELNet reaches 92.3% and 69% in the CIFAR-10 and CIFAR-100 datasets, respectively, the accuracy can be improved by using more complex networks. The three convolution modules with depthwise separable convolution, atrous convolution, and inception modules can also be extended to these complex networks to lower the number of parameters and operations and preserve the accuracy of classification as well.

## 5. Conclusions

The contributions of this study listed in the following confirm that the ELNet can effectively reduce model complexity but maintain fine accuracy:

- (1) ELNet successfully combines three convolutional modules, depthwise separable convolution, atrous convolution, and inception module, for reducing the number of parameters and operations in the model
- (2) ELNet requires only 2.1 million training parameters and 2.57 mega FLOPs based on the input image size that is equal to  $224 \times 224$
- (3) The accuracy of ELNet reached 92.3% and 69% in CIFAR-10 and CIFAR-100 datasets, respectively

Therefore, the proposed ELNet can be applied on embedded systems for image classification applications. In addition, the architecture can integrate other methods such as parameter pruning, recursion, or other learning methodologies to optimize the network for further research.

## Data Availability

The CIFAR-10 and CIFAR-100 datasets are available to access from <https://www.cs.toronto.edu/~kriz/cifar.html>.

## Conflicts of Interest

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

## Acknowledgments

The authors would like to thank the support of the Intelligent Manufacturing Research Center (iMRC) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. This research was funded by the Ministry of Science and Technology of the Republic of China (Grant no. MOST 109-2221-E-167-027).

## References

- [1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, December 2012.
- [3] N. Srivastava, G. Hinton, A. Krizhevsky et al., "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, pp. 1929–1958, 2014.
- [4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.
- [5] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June 2015.

- [6] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, <http://arxiv.org/abs/1505.00387>.
- [7] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.
- [8] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, October 2016.
- [9] L. C. Chen, G. Papandreou, I. Kokkinos et al., "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2017.
- [10] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, <http://arxiv.org/abs/1704.04861>.
- [11] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," 2017, <http://arxiv.org/abs/1608.06049>.
- [12] F. N. Iandola, S. Han, M. W. Moskewicz et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, <http://arxiv.org/abs/1602.07360>.
- [13] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.
- [14] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, <http://arxiv.org/abs/1804.02767>.
- [15] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning*, Atlanta, GA, USA, June 2013.
- [16] K. He, X. Zhang, S. Ren et al., "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.
- [17] B. Xu, N. Wang, T. Chen et al., "Empirical evaluation of rectified activations in convolutional network," 2015, <http://arxiv.org/abs/1505.00853>.
- [18] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proceedings of the International Conference on Learning Representations*, Banff National Park, Canada, April 2014.
- [19] J. Springenberg, A. Dosovitskiy, T. Brox et al., "Striving for simplicity: the all convolutional net," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

## Research Article

# Multistep Prediction of Bus Arrival Time with the Recurrent Neural Network

Zhi-Ying Xie,<sup>1,2</sup> Yuan-Rong He,<sup>1,2</sup> Chih-Cheng Chen ,<sup>3,4</sup> Qing-Quan Li,<sup>5</sup>  
and Chia-Chun Wu <sup>6</sup>

<sup>1</sup>School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

<sup>2</sup>Digital Fujian Institute of Natural Disaster Monitoring Big Data, Xiamen, Fujian 361024, China

<sup>3</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

<sup>4</sup>Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

<sup>5</sup>Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen 518060, China

<sup>6</sup>Department of Industrial Engineering and Management, National Quemoy University, Kinmen 892, Taiwan

Correspondence should be addressed to Chih-Cheng Chen; [ccc@gm.cyut.edu.tw](mailto:ccc@gm.cyut.edu.tw) and Chia-Chun Wu; [ccwu0918@nqu.edu.tw](mailto:ccwu0918@nqu.edu.tw)

Received 30 October 2020; Revised 3 January 2021; Accepted 18 February 2021; Published 12 March 2021

Academic Editor: Bosheng Song

Copyright © 2021 Zhi-Ying Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate predictions of bus arrival times help passengers arrange their trips easily and flexibly and improve travel efficiency. Thus, it is important to manage and schedule the arrival times of buses for the efficient deployment of buses and to ease traffic congestion, which improves the service quality of the public transport system. However, due to many variables disturbing the scheduled transportation, accurate prediction is challenging. For accurate prediction of the arrival time of a bus, this research adopted a recurrent neural network (RNN). For the prediction, the variables affecting the bus arrival time were investigated from the data set containing the route, a driver, weather, and the schedule. Then, a stacked multilayer RNN model was created with the variables that were categorized into four groups. The RNN model with a separate multi-input and spatiotemporal sequence model was applied to the data of the arrival and leaving times of a bus from all of a Shandong Linyi bus route. The result of the model simulation revealed that the convolutional long short-term memory (ConvLSTM) model showed the highest accuracy among the tested models. The propagation of error and the number of prediction steps influenced the prediction accuracy.

## 1. Introduction

The rapid and continuous development of China has led to an increase in the number of vehicles. The National Bureau of Statistics of China announced that the number of privately owned vehicles reached 261.5 million in 2019 with 21.22 million vehicles increased in a year. 96 cities in China had more than one million registered vehicles [1]. The rapid increase of vehicles causes traffic congestion, parking problems, and environmental pollution. Public transportation affords a larger number of passengers and alleviates such problems. Mass transportation consumes less energy and emits less amount of pollutants than private transport. Therefore, urban planning puts a priority on public

transportation. New technologies such as bus rapid transit (BRT) and driverless bus have been developed significantly with huge investment to support the public transportation system. However, a trip by bus takes a relatively long time and is not punctual, which makes people avoid it. Encouraging people to use buses more often requires optimized bus routes and punctuality of bus operation [2, 3]. However, the absence of an accurate operation schedule often causes long waiting times and bus bunching on the same route. For the punctual operation of the public buses, the bus schedule needs to be optimized, which needs an accurate prediction of the arrival time of buses on a route accurately. This not only meets the demand of ordinary passengers who want to know the arrival times of a bus at boarding stations but also optimizes the



intelligent bus scheduling system and improves the operation efficiency of the bus company.

Several neural networks have been used to predict the arrival time of a bus: non-RNN network, RNN with the time series, and temporal and spatial RNN network. Several studies adopted non-RNN networks for predicting bus arrival and operation times using (1) MapReduce-based clustering with  $K$ -means [4], (2) a backpropagation (BP) neural network model [5], (3) a particle swarm algorithm [6], (4) a wide-depth recursive (WDR) learning model [7], and (5) RNN with the time series such as long short-term memory (LSTM) [8]. Models with LSTM processed the historical data of the global position system (GPS) and bus stop locations with the influence of different routes, drivers, weather conditions, time distribution [9], heterogeneous traffic flow, and real-time data [10–12]. The temporal and spatial RNN network with ConvLSTM or a spatiotemporal property model (STPM) was originally used to predict the precipitation [13]. However, it was also used for predicting bus arrival times based on the total operation time of a bus on a route, waiting and on-board times, transfer location wait times [14–16], and multilane short-term traffic flow [17] and for creating the multitime step deep neural network [18].

The bus is running on fixed lines with fixed stations. The spatial relationship between its stations determines the arrival times in the time series. Thus, this study used an RNN to predict the arrival time of a bus. A route of a bus has 30–40 bus stations in general. Arrival time prediction includes the time prediction of each station along the way from the starting to the finishing stop, the arrival times at subsequent stations, and the arrival time of the nearest vehicle to a station. This study first analyzed the bus arrival time. Based on the analysis, the input eigenvectors of a neural network were defined, and then, seven RNN models for predicting the arrival time from four categories were tested. Then, the proposed model was trained by the measured data of arrival and departure times of the buses in a route of Linyi, Shandong Province. Then, the multistep prediction of the arrival time was carried out.

This paper is organized as follows. Section 2 describes the theoretical background and introduces the recurrent neural network. Section 3 describes the pretreatment and analysis of data. Section 4 discusses the analysis result of the RNN model. Finally, Section 5 concludes this study.

## 2. Theoretical Background

A recurrent neural network (RNN) [19] has a feedback structure that processes sequential data for time-series prediction or classification. RNN is widely used in various applications, and new models using it have been suggested such as LSTM, GRU, and ConvLSTM. According to the data in this study, we divided the prediction into four categories and adopted a multistep prediction for bus arrival times. The time-series input data is essential for the prediction with optimal feature extraction and memory efficiency. The data is processed in an RNN with internal feedback and feed-forward connection, which retain and reflect the state or memory of a long context window [20]. The RNN suffers from a common disadvantage of the gradient disappearance

(gradient vanishing) and gradient explosion problem [21–23], which results in limited applications due to training problems. To solve the problems, Hochreiter et al. [24] proposed and continued improving LSTM for different applications [25, 26]. LSTM specializes in memorizing long sequences and effectively avoiding the problem of gradient disappearance. Hidden layers of LSTM use memory blocks that store the previous sequence information, while increasing the performance of three gates: input, output, and forget gates. These control the sequence information for memory. The gated recurrent unit (GRU) [27] is a modestly simplified LSTM. GRU combines the forget and input gate into an update gate and the cell and hidden state. A model with GRU is simpler and has less activation function and output computation than the standard LSTM model.

*2.1. Pure LSTM and Pure GRU Model.* Figure 1 shows the hidden units of LSTM which are replaced by memory blocks.

Calculating  $c_t$  and  $h_t$  requires the following equations:

$$\begin{aligned} i_t &= \sigma(W_i X_t + U_i h_{t-1} + b_i) \text{ (Input gate),} \\ f_t &= \sigma(W_f X_t + U_f h_{t-1} + b_f) \text{ (Forget gate),} \\ o_t &= \sigma(W_o X_t + U_o h_{t-1} + b_o) \text{ (Output gate),} \\ \tilde{c}_t &= \tanh(W_c X_t + U_c h_{t-1} + b_c) \text{ (New memory cell),} \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \text{ (Memory cell),} \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \quad (1)$$

In these equations,  $\odot$  Hadamard product is the multiplication of the corresponding elements in the operation matrix,  $W_i$ ,  $W_f$ ,  $W_o$ , and  $W_c$  are the weights of  $X_t$ ,  $U_i$ ,  $U_f$ ,  $U_o$ , and  $U_c$  are the weights of  $h_{t-1}$ ,  $b_i$ ,  $b_f$ ,  $b_o$ , and  $b_c$  are the bias conditions,  $\sigma$  is the sigmoid function, and  $\tanh$  is the hyperbolic tangent function.

Figure 2 shows the GRU. There is only one hidden state  $h_t$  in GRU. Through the linear transformation of the input tensor and hidden state, the weighted sum of the hidden state inflow is calculated with equations (2) and (3). The linear transformation for  $r_t$ ,  $h_{t-1}$ , and the input tensor is combined with the activation function of equation (4) to calculate the updated value of the hidden state. The mixed weight for calculation of the implicit state in the previous step is shown in equation (5). The final output  $h_t$  is the same as LSTM. Compared with LSTM, there is one less activation function calculation and output calculation as well as the final hidden state update, so the calculation is relatively simple.

$$z_t = \sigma(W_z X_t + U_z h_{t-1} + b_z) \text{ (Update gate),} \quad (2)$$

$$r_t = \sigma(W_r X_t + U_r h_{t-1} + b_r) \text{ (Reset gate),} \quad (3)$$

$$\tilde{h}_t = \tanh(W_h X_t + r_t \odot U_h h_{t-1} + b_h) \text{ (New memory cell),} \quad (4)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}. \quad (5)$$

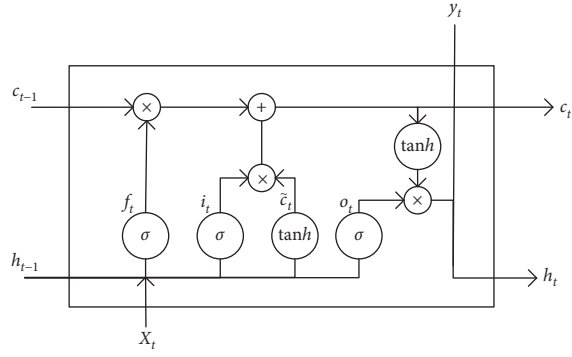


FIGURE 1: LSTM memory cell structure.

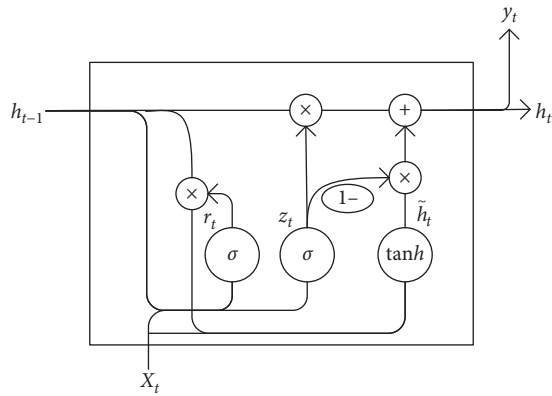


FIGURE 2: GRU unit structure.

Figure 3 illustrates a pure network model of LSTM and GRU. The input layer has the sequence of the arrival time series input, and the other two layers use a fully connected prediction network. Input and output data are 3D tensors with a shape  $[?, 41, 1]$  (“?” means that a dimension can have any length).

These models use a single layer of LSTM and GRU. In the input layer, variables are, such as route, direction, vehicle model, and driver, also regarded as a part of the time sequence.

**2.2. Multi-Input Model Separated by Time Series.** As the variable is not sensitive to any specific ordering, the RNN cannot process it alone. However, a BP network can process through a connection layer. Thus, the integration of RNN and BP was used for the prediction network (Figure 4).

The integrated network was in accordance with the characteristics of the input data. A two-part network used the time series-related input data such as route number, driver, departure time, and route length for LSTM processing. Through a connection layer, the prediction layer was processed. Since time series input data became shorter even with the addition of LSTM, the total trainable parameters were not significantly increased compared with pure LSTM.

**2.3. LSTM Stacking Model.** To achieve better accuracy of the prediction than a single layer, a multilayer LSTM was

employed. Stacking four LSTMs had hidden units in 256, 128, 64, and 32 layers, respectively. Figure 5(a) shows the diagram of the stacking models. There is also a two-way LSTM composition, in which the forward and backward connections also employ a reverse projection function, which is suitable in our case to verify arrival time predictions. Figure 5(b) shows the diagram of the bidirectional network models.

**2.4. Spatiotemporal Time-Series Model.** The bus operation is in a space-time domain although there are little changes in the spatial dimension for an operation in the fixed route. As ConvLSTM processes the data of time and space, it integrates a convolution of time and space into calculating each gate of LSTM. The following equations are used for the calculation:

$$i_t = \sigma(W_i * X_t + U_i * h_{t-1} + b_i) \quad (\text{Input gate}), \quad (6)$$

$$f_t = \sigma(W_f * X_t + U_f * h_{t-1} + b_f) \quad (\text{Forget gate}), \quad (7)$$

$$o_t = \sigma(W_o * X_t + U_o * h_{t-1} + b_o) \quad (\text{Output gate}), \quad (8)$$

$$\tilde{c}_t = \tanh(W_c * X_t + U_c * h_{t-1} + b_c) \quad (\text{New memory cell}), \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (\text{Memory cell}), \quad (10)$$

$$h_t = o_t \odot \tanh(c_t). \quad (11)$$

A ConvLSTM network with batch normalization (BN) consists of a specification and flattening layer and a prediction network. Figure 6 shows the diagram of the network that has a long training time and many parameters in more than five dimensions. This network is appropriate to process time-series data with spatial properties such as bus arrival times with high accuracy.

### 3. Pretreatment and Analysis of Data

**3.1. Data Characteristics.** A bus was equipped with a device that included a GPS and data communication module. The device transmitted data to a bus scheduling system. Table 1 shows the data structure of the reporting system.

The data of arrival and departure of a bus at a bus stop consists of route, speed, arrival and departure time, coordination, and driver's number. For obtaining the Lasso variable correlation [8], the bus number, number of bus stops, days in the week, distances between bus stops, arrival and departure times, and weather were included, too. The variables were grouped into two: dynamic and static variables [14]. The dynamic variables include driving times between bus stops, staying times, and weather, while the static variables include a route, direction, vehicle model, driver, arrival and departure times, days of the week, holidays, and working days. We selected variables related to the route and the arrival times at the previous stops as the input

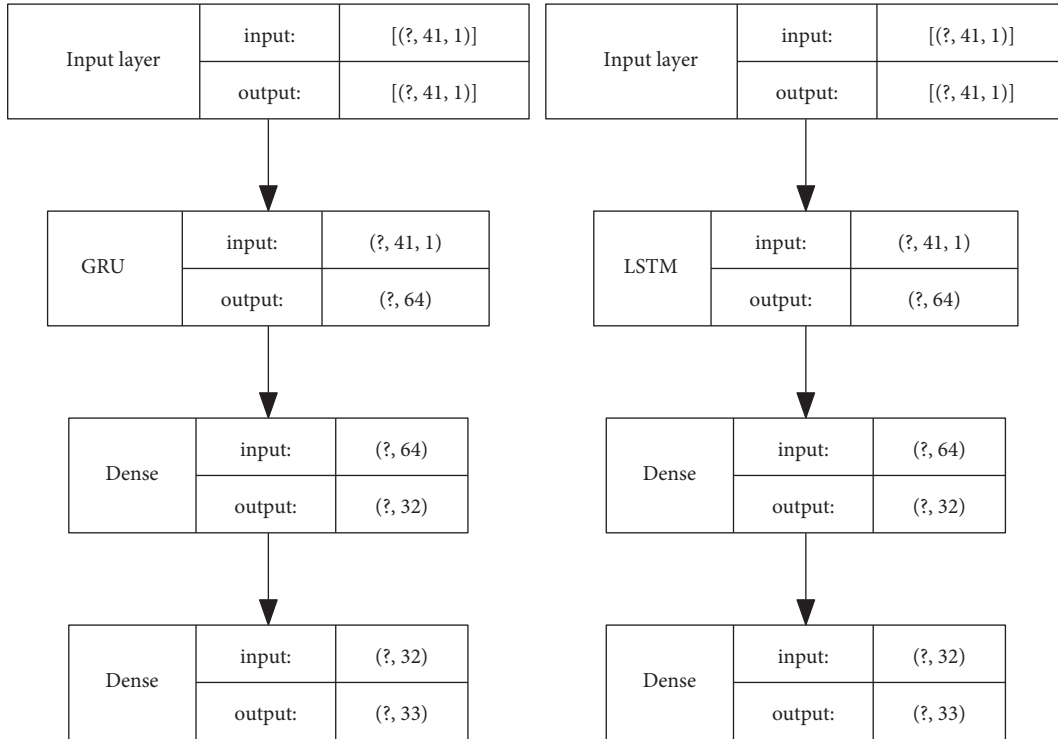


FIGURE 3: A pure LSTM and pure GRU network model.

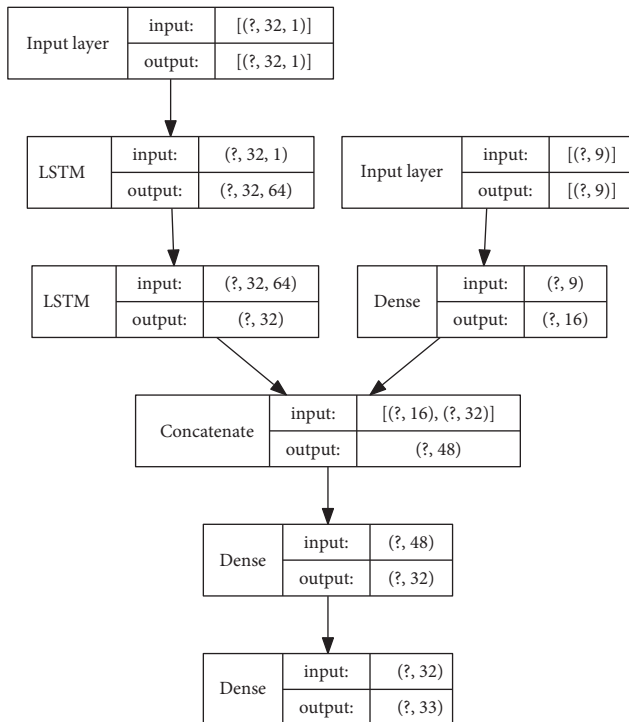


FIGURE 4: An LSTM-BP integrated network model.

features to predict the arrival time at a bus stop. Input and output features are as follows:

Input feature: [route\_no, direction, bus\_no, driver\_no, departure hours, departure minutes, days of the week, holidays, distance, and weather ( $x_{t-k}, \dots, x_t$ )]

Output prediction series: ( $-x_{t-k}, \dots, 0, x_{t+1}, \dots, x_{t+n}$ )

where  $x_t$  is the difference of the arrival time between the current station and the previous station.

### 3.2. Data Preprocessing

#### Step 1. Generating a Sample Dataset

According to the data in Section 3.1, an arrival time series was obtained from the data of arrival and departure times of a bus at a bus stop. For the convenience of calculation, the difference of the arrival times between two bus stops was calculated in seconds. Table 2 shows the example of the dataset. The existing sequence data are 120, 220, 250, and 260 which correspond to four bus stops A, B, C, and D. This means that 120 s is needed for a bus to drive from the starting location to A, 220 s from A to B, 250 s from B to C, and 260 s from C to D. When the bus arrives at C, the prediction of the arrival time to D is only needed. The input sequence is the sequence of all arrival times from the starting location to C, and the output sequence includes 260 s from C to D and the backward sequence from C to the starting location. The length of the input sequence is shorter than that of the output sequence as it only needs to predict the time to the finishing location of a bus. When predicting the time to the finishing location, it only needs to know the sequences before it. When the bus arrives at a bus stop

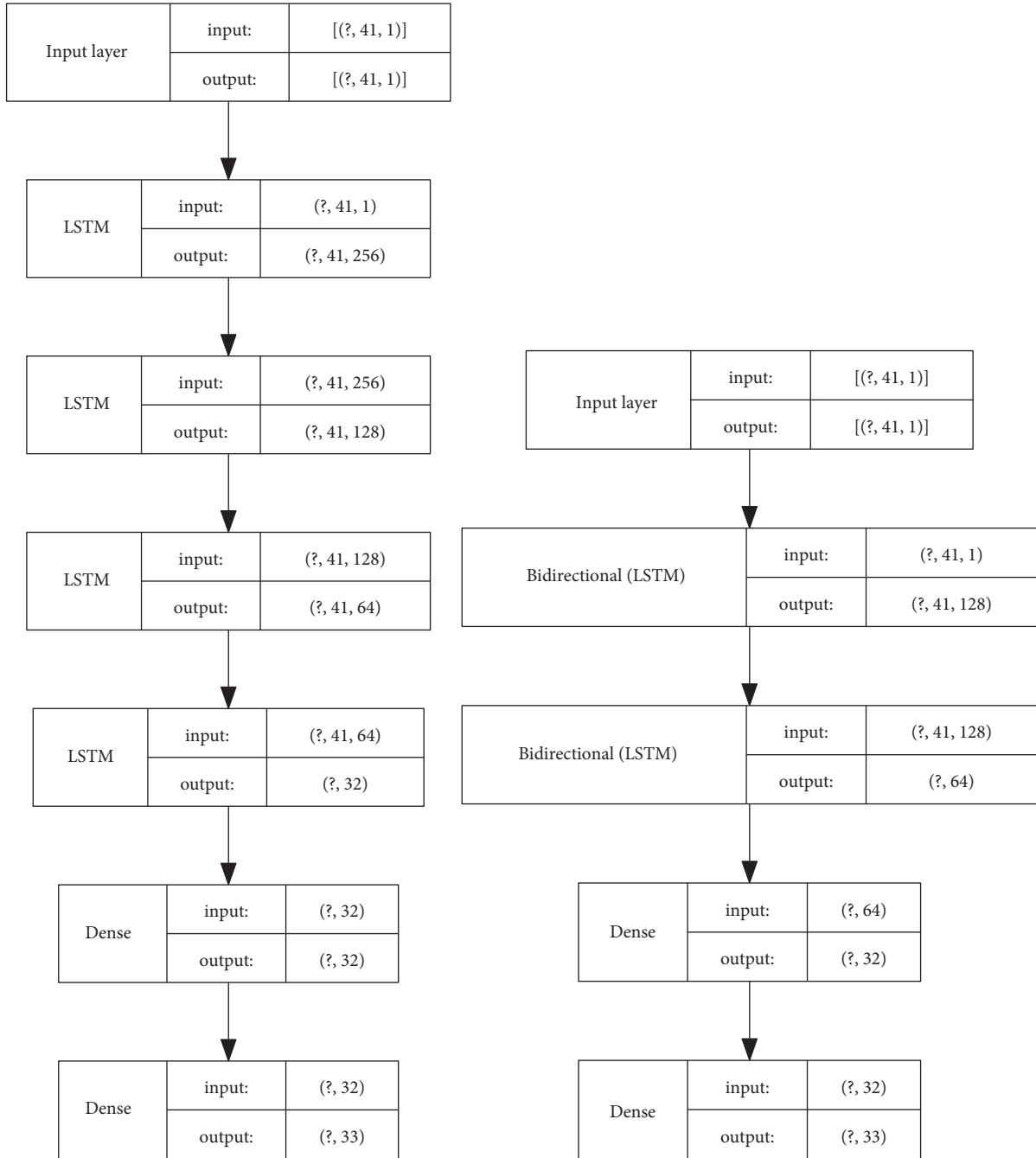


FIGURE 5: (a) A LSTM stacking model. (b) A bidirectional LSTM model.

between the starting and finishing location, for a consistent sequence length, the time to the previous bus stop is input as 0.

Figure 7 shows the time-series data of real arrival times. The blue and orange line is for the input and output sequence, respectively. The sequence has the predicted times of 0 at the current bus stop. An output sequence has negative numbers to maintain the correctness of the inverted time from the starting location to the bus stop.

### Step 2. Dataset Normalization

The variables had different dimensions and units which affected the results of data analysis. Thus, normalization was necessary to eliminate the differences. Standardizing with the Z-score and the minimum-maximum values were used so that the final values were ranged between 0-1. The equation for standardization is as follows:

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (12)$$

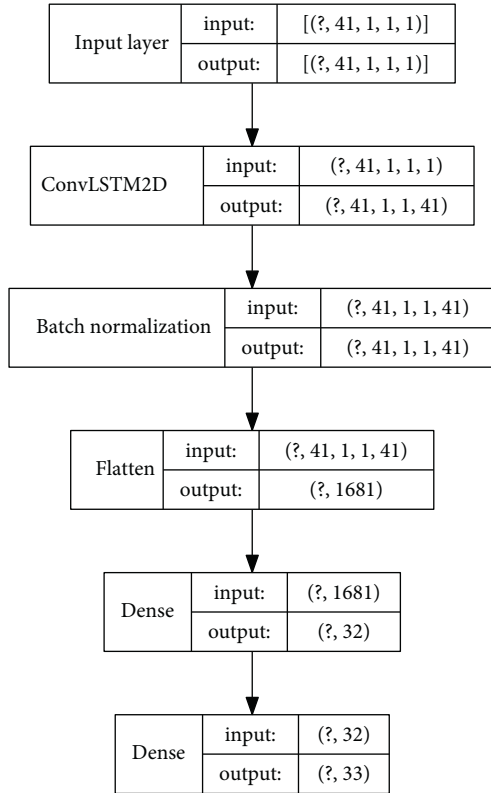


FIGURE 6: ConvLSTM network model.

TABLE 1: The data structure from a bus.

No.	Name	Types	Remarks
1	Route_no	Long	Route number
2	Route_name	String	Route name
3	Route_subno	Long	Run method no.
4	Up down	Int	Direction on route
5	Bus_no	Long	Vehicle number
6	Speed	Float	Speed
7	datetime_in	String	Entry time
8	driver_no	Long	Driver number
9	busstop_no	Long	Site number
10	busstop_name	String	Site name
11	busstop_lng	Double	Site longitude
12	busstop_lat	Double	Site latitude
13	busstop_serial	Int	Site serial number
14	busstop_type	Int	Site type
15	pack_datetime	String	Outbound time
16	inform_type	Int	Inform mode
17	netpack_type	Int	Entry mode

TABLE 2: Examples of the arrival time-series dataset format.

Input sequence	Output sequence
(120, 220, and 250)	(-120, -220, 0, and 260)
(120, 220, and 0)	(-120, 0, 250, and 260)
(120, 0, and 0)	(0, 220, 250, and 260)

Z-score standardization uses the mean and standard deviation of the data and is calculated as follows:

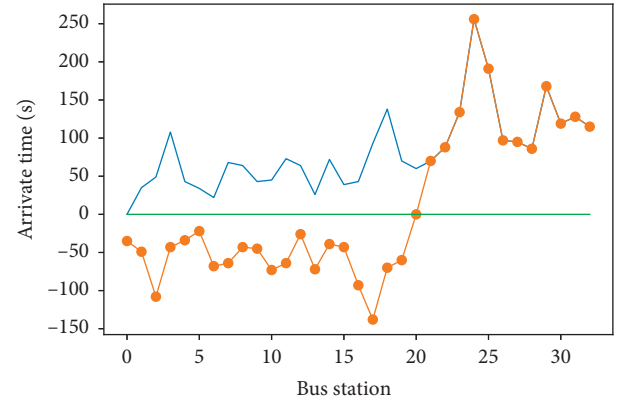


FIGURE 7: A diagram of the input and output sequence generated by the time-series data of arrival times.

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}, \quad (13)$$

where  $\mu$  is the mean and  $\sigma$  is the standard deviation of the data.

In this paper, after sorting the vehicle number and driver number on the route, the sorted sequence was used as the input data and normalized by equation (7). The route number, route direction, departure time (hh:mm), days of the week, holidays, distances from the starting location, weather, and other information were normalized, and their arrival time series were processed by equation (8).

#### 4. Analysis Result of the RNN Model

**4.1. Dataset.** The experiment was based on the data collected from March 28 to June 28, 2020, in Linyi, Shandong Province. The data was obtained from buses that ran on route no. 30 which had 36 bus stops (Figure 8). Tensorflow-GPU 2.0 was used for data processing and algorithm creation. The numbers of the dataset were 122,336 after pre-treatment, 78,303 in the training set, 19,590 in the verification set, and 24,443 in the test set.

**4.2. Training the RNN Prediction Model.** Seven network models were designed, trained, verified, and tested by using the preprocessed dataset. Pure LSTM and GRU were RNN models. LSTM-BP and GRU-BP were multiple input models with variable features separated from the time series. Bi-directional LSTM (LSTM-Bi) and LSTM-Stack were LSTM stack models. ConvLSTM was a spatiotemporal sequence model. Table 3 shows a model structure and a comparison of the parameters of the RNN network. Pure GRU had the smallest number of the parameter, while the stack model had the largest number.

The loss function selected the average absolute error (MAE) which was the difference between the prediction and real value. All network parameters were updated using the Adam optimization algorithm. The Adam algorithm performs first-order optimization. The first- and second-order optimizations were used for a dynamic design of independent adaptive learning rates for different parameters. The





FIGURE 8: Linyi city's 30 bus routes and site distribution.

TABLE 3: Seven RNN network model structures and their parameters.

Classification	Type of network	Type of layer	Output sequence	Number of parameters	Total number of parameters
Pure RNN	Pure GRU	GRU	(None, 41, 64)*	12864	15,009
		Dense	(None, 32)	1056	
	Pure LSTM	Dense	(None, 33)	1089	
		LSTM	(None, 64)	16896	
		Dense	(None, 32)	2080	
Multi-input hybrid model	GRU-BP	Dense	(None, 33)	1089	20,065
		Input layer	[(None, 32, 1)]	0	
		Input layer	[(None, 9)]	0	
		GRU	(None, 32, 64)	12864	
		Dense	(None, 16)	160	
	LSTM-BP	GRU	(None, 32)	9408	
		Concatenate	(None, 48)	0	
		Dense	(None, 32)	1568	
		Dense	(None, 33)	1089	
		Input layer	[(None, 32, 1)]	0	
		Input layer	[(None, 9)]	0	
LSTM-Bi	LSTM	(None, 32, 64)	16896	32,129	
	Dense	(None, 16)	160		
	LSTM	(None, 32)	12416		
	Concatenate	(None, 48)	0		
	Dense	(None, 32)	1568		
Stacking models	LSTM-Bi	Dense	(None, 33)	1089	525,281
		Bidirectional (LSTM (64))	(None, 41, 128)	33792	
	LSTM-Bi	Bidirectional (LSTM (32))	(None, 64)	41216	
		Dense	(None, 32)	2080	
	LTSM-Stack	Dense	(None, 33)	1089	
		LSTM	(None, 41, 256)	264192	
		LSTM	(None, 41, 128)	197120	
		LSTM	(None, 41, 64)	49408	
		LSTM	(None, 32)	12416	
	LTSM-Stack	Dense	(None, 32)	1056	
Dense		(None, 33)	1089		

TABLE 3: Continued.

Classification	Type of network	Type of layer	Output sequence	Number of parameters	Total number of parameters
Space-time model	ConvLSTM	ConvLSTM2D	(None, 41, 1, 1, 41)	62156	117,233
		BN	(None, 41, 1, 1, 41)	164	
		Flatten	(None, 1681)	0	
		Dense	(None, 32)	53824	
		Dense	(None, 33)	1089	

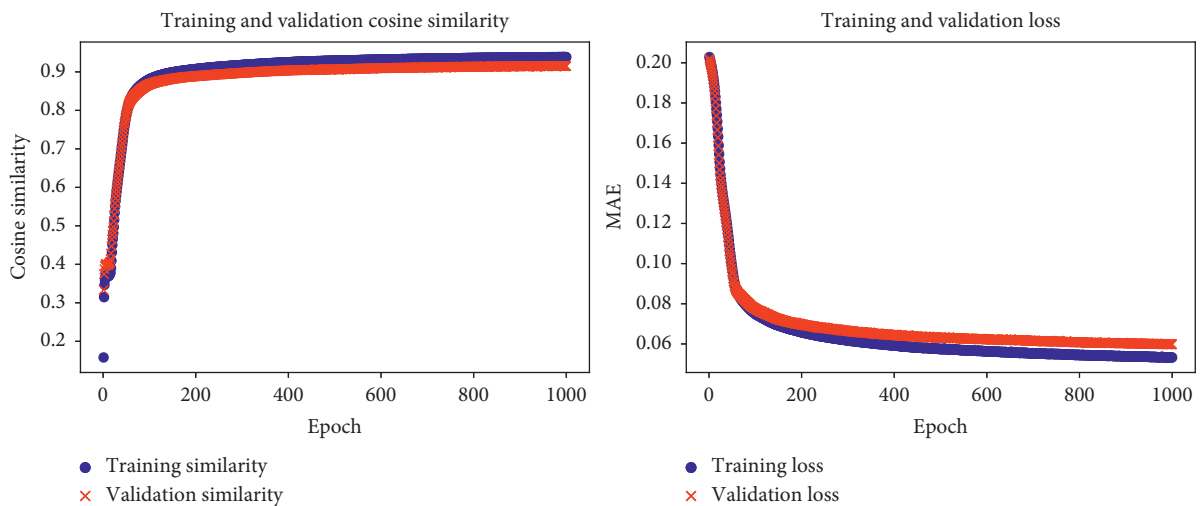


FIGURE 9: Loss function and accuracy diagram during training.

estimations were better than the traditional gradient descent method. Since the output was a time series, cosine similarity was used to determine the accuracy.

Figure 9 shows the training loss and accuracy when an epoch is 1000, a batch is 100, and a verification set is 20% of the training set. As the seven models show similar MAEs, only the training output of the ConvLSTM model was selected. In the process of training, the trend of training and validation data is consistent, and there is no fitting case. The super parameters of the selected model are suitable, too. As seen from Figure 10, the pure GRU had fewer training parameters and less training time than pure LSTM. The LSTM-Bi doubled the number of parameters and training time than the pure LSTM. The LSTM-Stack had five times more parameters than other models, but less training time than the ConvLSTM. The ConvLSTM had the longest training time, 5.8 times more parameters, and 12 times longer training time than the pure LSTM.

**4.3. Analysis of Results.** The test set was used to predict the training model, and the predicted arrival times are shown in Table 4.

The fitting degree of the real and the predicted value in Table 4 shows that the ConvLSTM provides the best prediction. The multi-input hybrid model, which separates the

parameters from the time series, not only increased the network complexity but also reduced the prediction accuracy. Table 5 shows the statistics of the prediction results by the seven models. MAE, RMSE, MAE, COS, number of training parameters, and time were used to quantitatively evaluate the seven network models. The prediction accuracy was improved from the pure LSTM to the ConvLSTM, as shown in Figure 11.

The results reveal the following:

- (1) The GRU was more efficient than the LSTM model with fewer parameters and considerable accuracy
- (2) The LSTM models except the ConvLSTM had more parameters and higher network accuracy than other models
- (3) The dataset property did not influence the results of the models but the complexity of the models
- (4) The ConvLSTM showed the highest accuracy as it processed the data of time and space, which indicated the need to include the space-related properties

In the process of arrival time-series prediction, the arrival times at subsequent bus stops were based on those at the previous bus stops. The ConvLSTM network model was selected to analyze the prediction accuracy through one- and two-step prediction and total time prediction.

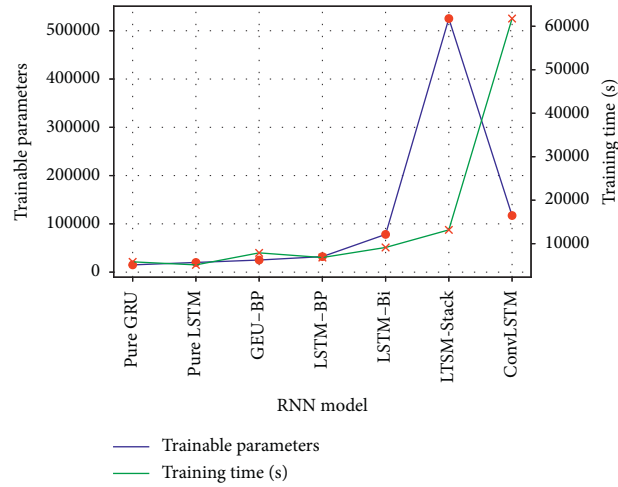


FIGURE 10: Comparison of training parameters and times of seven RNN models.

TABLE 4: List of arrival times predicted by the seven models.

Network classification	Network model	Predicted arrival time series
Pure RNN	Pure LSTM	
	Pure GRU	

TABLE 4: Continued.

Network classification	Network model	Predicted arrival time series
	LSTM-BP	
Multi-input hybrid model	GRU-BP	

TABLE 4: Continued.

Network classification	Network model	Predicted arrival time series
	LSTM-Bi	
Stacking models	LTSM-Stack	
Space-time model	ConvLSTM	

TABLE 5: Statistics of the prediction results of seven models.

Model	RMSE	MSE	MAE	COS	Trainable parameters	Training time (s)
Pure LSTM	37.7892	1428.029	23.1300	0.9492	20,065	5120
Pure GRU	37.7575	1425.6302	23.0065	0.9494	15,009	5820
LSTM-BP	37.6818	1419.9197	22.8906	0.9498	32,129	6812
GRU-BP	37.8738	1434.4282	22.8810	0.9496	25,089	7886
LSTM-Bi	37.4615	1403.3645	22.5548	0.9507	78,177	9126
LTSM-Stack	34.6279	1199.0963	18.9767	0.9585	525,281	13172
ConvLSTM	34.3812	1182.0734	17.0876	0.9595	117,233	61773

MAE: mean absolute error; RMSE: root mean square error; COS: cosign similarity.



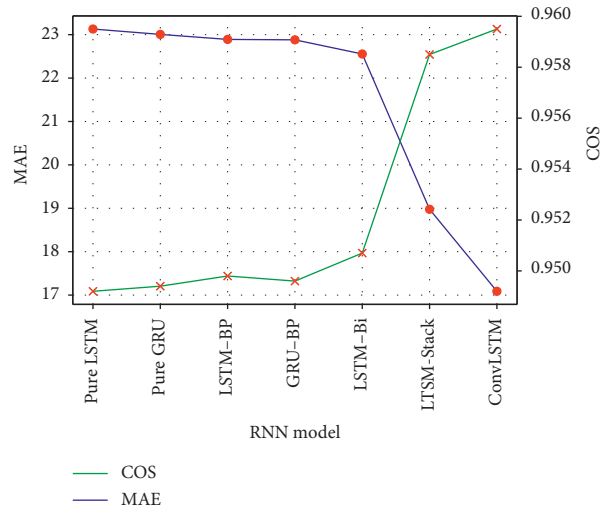


FIGURE 11: Prediction accuracies of the seven models.

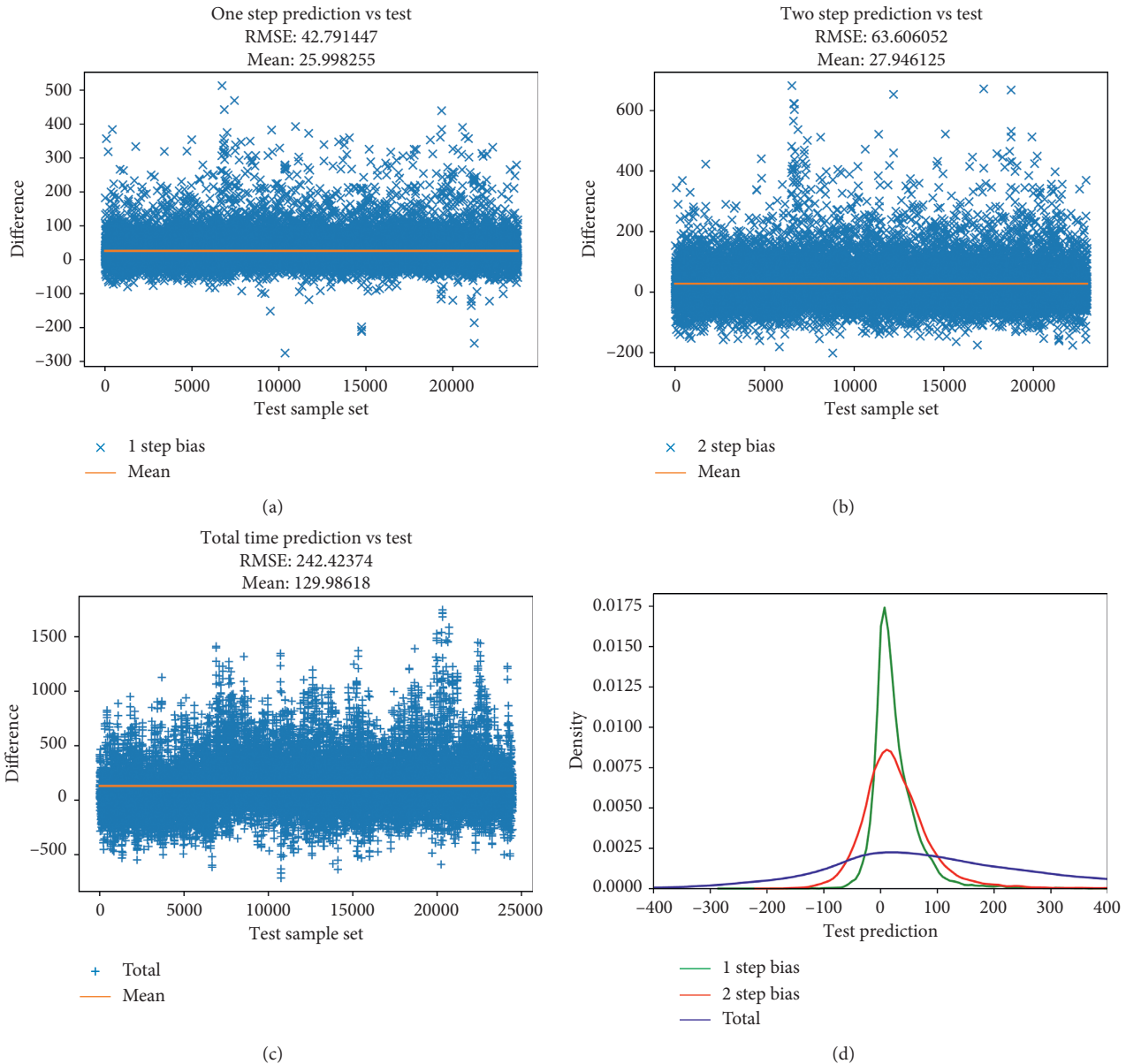


FIGURE 12: Multistep prediction comparison chart. (a) One step. (b) Two step. (c) Total. (d) Histogram.

Figure 12 shows the test sample set on the  $x$ -axis and the difference between the predicted and real values on the  $y$ -axis. The mean and RMSE were calculated from the mean values and mean square deviation of the differences. The one-step prediction had the highest accuracy, and the total time prediction (multistep prediction) showed the lowest accuracy. The regularity in the histogram of Figure 12 reveals that the one-step prediction has the smallest deviation and the highest error, which is related to the accumulation and propagation of errors in the prediction of the arrival times of the subsequent bus stops.

## 5. Conclusion

The public transport system is a complex system with a high degree of uncertainty. The system is understood as a multistep prediction problem in which uncertainty leads to poor prediction accuracy. This paper first analyzed the main variables affecting this uncertainty, and then, the variables such as route, direction, vehicle, driver, departure hour, departure minute, day of the week, holiday, distance from the starting location, and weather were selected. The arrival time series before the current bus stops was also selected. These variables fully reflected the impact on the arrival time-series prediction. Among RNN networks for time-series analysis, we processed the data by using seven different network models in four different types of networks.

We analyzed and compared the predictive power of the seven RNN models with the variables and parameters in the measured dataset. We noticed an improvement in prediction accuracy by adding variables in one- and two-step prediction models, but not in the multistep (total time prediction) model. The multistep model increased the network complexity only. The ConvLSTM showed the highest prediction accuracy with spatiotemporal data. The statistics of one-, two-, and multistep prediction showed that the accumulation and propagation of the sequence prediction error caused more steps and a large deviation of the predicted time. The accurate bus arrival time prediction encourages more people to use buses for transportation and allows operating companies to optimize bus schedules for increasing the efficiency of their operation. This also improves the traffic condition in cities.

Accurate bus arrival information also relieves the anxiety of users by decreasing waiting time and helps to provide passengers with an improved service. The accurate prediction of bus arrival times can be integrated into an intelligent bus scheduling system in a smart transportation system. Such a system improves the management of a public transport system, increases the economic benefits of the system, and ultimately brings social benefits.

## Data Availability

The nature of the data includes excel files, and the data can be accessed at <https://github.com/ricebow/multi-step-RNN>. There are no restrictions on data access. 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.

## Acknowledgments

This work was supported by the Fujian Province Natural Fund Project (Grant no. 2020J01263), Science and Technology Planning Foreign Cooperation Project of Longyan (Grant no. 2019LYF7003), Open Fund Project of Fujian University Engineering Research Center for Disaster Prevention and Mitigation of Southeast Coastal Engineering Structure of Putian University (Grant no. 2019005), and Open Foundation Project of Fujian Provincial Key Laboratory of Higher Education (Putian University) (Grant no. ST19004).

## References

- [1] National Bureau of Statistics, *Statistical Bulletin of the People's Republic of China on National Economic and Social Development*, National Bureau of Statistics, Beijing, China, 2020.
- [2] H. Lu, Z. Sun, and W. Qu, "Big data and its applications in urban intelligent transportation system," *Journal of Transportation Systems Engineering and Information Technology*, vol. 15, pp. 45–52, 2015.
- [3] D. Li, Y. Yao, and Z. Shao, "Big data in smart city," *Geomatics and Information Science of Wuhan University*, vol. 39, pp. 631–640, 2017.
- [4] F. Xie, J. Gu, S. Zhang et al., "Predicting model of bus arrival time based on Map reduce clustering and neural network," *Journal of Computer Application*, vol. 37, pp. 118–129, 2017.
- [5] L. Wang, Q. Su, and R. Zheng, "Bus arrival time prediction based on Elman's dynamic neural network," *Mechanical & Electrical Technology*, vol. 35, pp. 135–139, 2012.
- [6] Y. Ji, J. Lu, X. Chen et al., "Prediction model of bus arrival time based on particle swarm optimization and wavelet neural network," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, pp. 60–66, 2016.
- [7] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM, pp. 858–866, London, Kingdom, August 2018.
- [8] J. Lu, L. Sun, and Q. Shi, "Prediction of bus arrival time based on gated recurrent unit neural networks," *Journal of Nantong University (Natural Science Edition)*, vol. 19, pp. 43–49, 2020.
- [9] Q. Han, K. Liu, L. Zeng, G. He, L. Ye, and F. Li, "A bus arrival time prediction method based on position calibration and LSTM," *IEEE Access*, vol. 8, pp. 42372–42383, 2020.
- [10] A. A. Agafonov and A. S. Yumaganov, "Bus arrival time prediction using recurrent neural network with LSTM architecture," *Optical Memory and Neural Networks*, vol. 28, no. 3, pp. 222–230, 2019.
- [11] W. Xiangxue, X. Lunhui, and C. Kaixun, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, 2019.
- [12] Z. Huang, Q. Li, F. Li, and J. Xia, "A novel bus-dispatching model based on passenger flow and arrival time prediction," *IEEE Access*, vol. 7, pp. P106453–P106465, 2019.
- [13] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network," *A Machine*

- Learning Approach for Precipitation Nowcasting*, vol. 1, p. 9, 2015.
- [14] Y. Lai, L. Zhang, F. Yang, W. Lu, and T. Wang, "Bus arrival time prediction algorithm based on the spatio-temporal correlation attribute model," *Ruan Jian Xue Bao/Journal of Software*, vol. 31, no. 3, pp. 648–662, 2020.
  - [15] H. Liu, H. Xu, Y. Yan, Z. Cai, T. Sun, and W. Li, "Bus arrival time prediction based on LSTM and spatial-temporal feature vector," *IEEE Access*, vol. 8, pp. 11917–11929, 2020.
  - [16] P. He, G. Jiang, S.-K. Lam, and Y. Sun, "Learning heterogeneous traffic patterns for travel time prediction of bus journeys," *Information Sciences*, vol. 512, pp. 1394–1406, 2020.
  - [17] Y. Ma, Z. Zhang, and A. Ihler, "Multi-lane short-term traffic forecasting with convolutional LSTM network," *IEEE Access*, vol. 8, pp. 34629–34643, 2020.
  - [18] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.
  - [19] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.
  - [20] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, <https://arxiv.org/abs/1506.00019>.
  - [21] Lechevallier, Saporta - 2010 - in Proceedings of COMPSTAT'2010.pdf.
  - [22] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," *International Conference on Machine Learning. Omnipress*.vol. 116, 2004.
  - [23] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," 2015, <https://arxiv.org/abs/1503.00075>.
  - [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, p. 1735, 1997.
  - [25] J. Schmidhuber and F. Cummins, "Learning to forget: continual prediction with LSTM," in *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, Edinburgh, UK, September 1999.
  - [26] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 29, 2002.
  - [27] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, <https://arxiv.org/abs/1412.3555>.