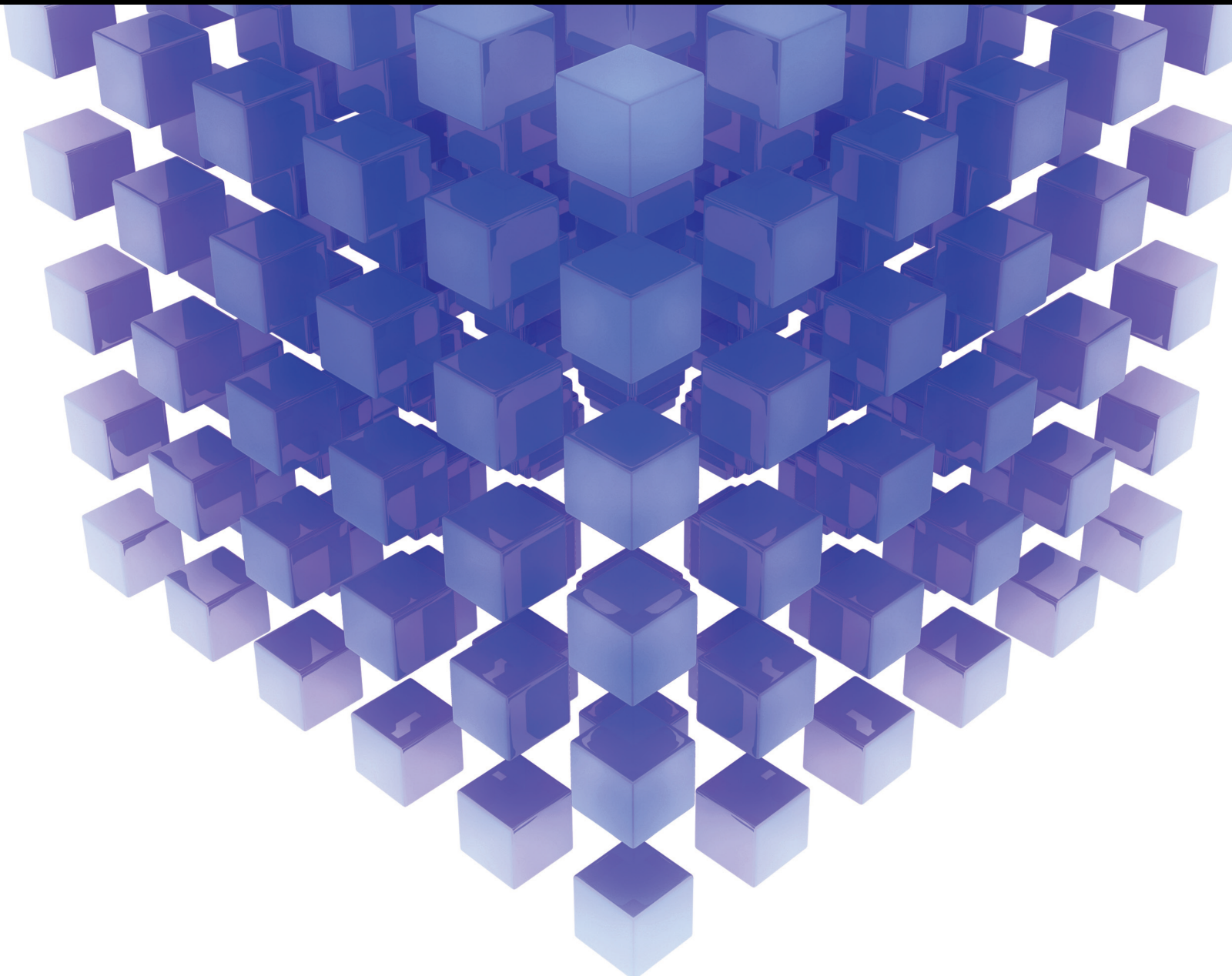


Mathematical Problems in Engineering

Machine Learning and Optimization Approaches in Smart Rail Systems

Lead Guest Editor: Li Zhu

Guest Editors: Cesar Briso-Rodriguez, Zhongbei Tian, and Shaofeng Lu





Machine Learning and Optimization Approaches in Smart Rail Systems

Mathematical Problems in Engineering

**Machine Learning and Optimization
Approaches in Smart Rail Systems**

Lead Guest Editor: Li Zhu


Guest Editors: Cesar Briso-Rodriguez, Zhongbei
Tian, and Shaofeng Lu



Copyright © 2023 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
Stylianos 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
Jyottheswara 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
Yuxing 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 Torricollo , 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

Analysis of the Abnormality of Traction Energy Consumption in Urban Rail Transit System

Xinjun Gao  and Xuetao Shi 







Research Article (11 pages), Article ID 1039287, Volume 2023 (2023)

RIS-Assisted Coverage Optimization for 5G-R Channel in Station Scenario Based on ML and RT

Bin Lu , Xinghai Guo , Xinyi Shan , and Fang Li 

Research Article (15 pages), Article ID 6899076, Volume 2023 (2023)

Design and Optimization in MEC-Based Intelligent Rail System by Integration of Distributed Multi-Hop Communication and Blockchain

Linlin Tian , Meng Li , Pengbo Si , Ruizhe Yang , Yang Sun , and Zhuwei Wang 

Research Article (11 pages), Article ID 8858263, Volume 2023 (2023)

Optimization of Metro Trains Operation Plans Based on Passenger Flow Data Analysis

Jun Yang , Yinghao Tang , Tan Ye , Xiao Han , and Mengjie Gong 

Research Article (11 pages), Article ID 7494127, Volume 2022 (2022)

Spatiotemporal Virtual Graph Convolution Network for Key Origin-Destination Flow Prediction in Metro System

Jun Yang , Xiao Han , Tan Ye , Yinghao Tang , Weidong Feng , Aili Wang , Huijun Zuo , and Qiang Zhang 

Research Article (11 pages), Article ID 5622913, Volume 2022 (2022)

Research Article

Analysis of the Abnormality of Traction Energy Consumption in Urban Rail Transit System

Xinjun Gao  and Xuetao Shi 

Signal and Communication Research Institute of China Academy of Railway Sciences Group Co.,Ltd, Beijing 100081, China

Correspondence should be addressed to Xuetao Shi; xuetaosh@163.com

Received 16 October 2022; Revised 13 December 2022; Accepted 7 April 2023; Published 22 June 2023

Academic Editor: Li Zhu

Copyright © 2023 Xinjun Gao and Xuetao Shi. 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.

Traction energy consumption (TEC) is a critical part of the total energy consumption in urban rail transit (URT) systems. Energy consumption patterns and abnormal analysis of TEC guarantee the energy-saving URT operation. With the rapid development of urbanization, the current energy consumption is becoming more and more prominent with some inherent drawbacks, such as complex original data, complicated statistical analysis, and abnormal energy consumption. This paper proposes a method for time accrual abnormal analysis of TEC. The system architecture of TEC typical values is presented, composed of three elements: research object, evaluation index, and time scale. The time series prediction algorithm calculates the typical values of the cumulative energy consumption index in each energy consumption mode. For the abnormality in TEC mode, the distance of the string vector is used as the similarity measure. Then, the similarity-based anomaly analysis method is used to judge the pattern abnormality. By comparing the advantages and disadvantages of engineering practice and theoretical research methods, we analyze the applicability of traditional anomaly detection algorithms to perform anomaly analysis of TEC in URT systems. The adopted time accrual abnormal analysis achieves a high fault detection rate, outperforming other models.

1. Introduction

The urban rail transit (URT) system is the main backbone of the passenger transportation system, with a large passenger capacity, short operation cycles, and low unit energy consumption. By 2022, a total of 541 cities in 79 countries have put URT systems into use, with a total mileage of more than 36,854 km. Although URT systems in China, represented by the subway system, have a short construction history, it has developed rapidly since the 21st century, as seen in Figure 1. The URT system has the advantage of energy saving in the condition of lower unit energy consumption. However, with the increasing operating mileage and passenger volume, the total energy consumption is rising. Primarily, traction energy consumption (TEC) has increased, accounting for 50% of the whole energy consumption systems. The onboard energy consumption recorder records a large amount of TEC

data in the form of time accumulation. Its energy consumption mode and abnormal analysis are significant to the URT energy-saving operation.

The URT system usually uses manual meter readings and statistics to obtain the TEC data, which is prone to transcription errors and has a large workload. In recent years, some trains have recorded the data during train operation by installing sensors and metering modules, thus accumulating a large amount of energy consumption-related data mainly in discrete time series. Currently, metro still uses a fixed period to count the accumulated TEC and operating mileage data, calculate the unit consumption index per 100 vehicle kilometers, and determine whether there is a TEC anomaly by combining the threshold. However, this period is too long to detect the TEC anomaly in time. At the same time, it is not practical for data to be recorded during train operation, and the threshold is used to judge the abnormality in a one-size-fits-all manner.

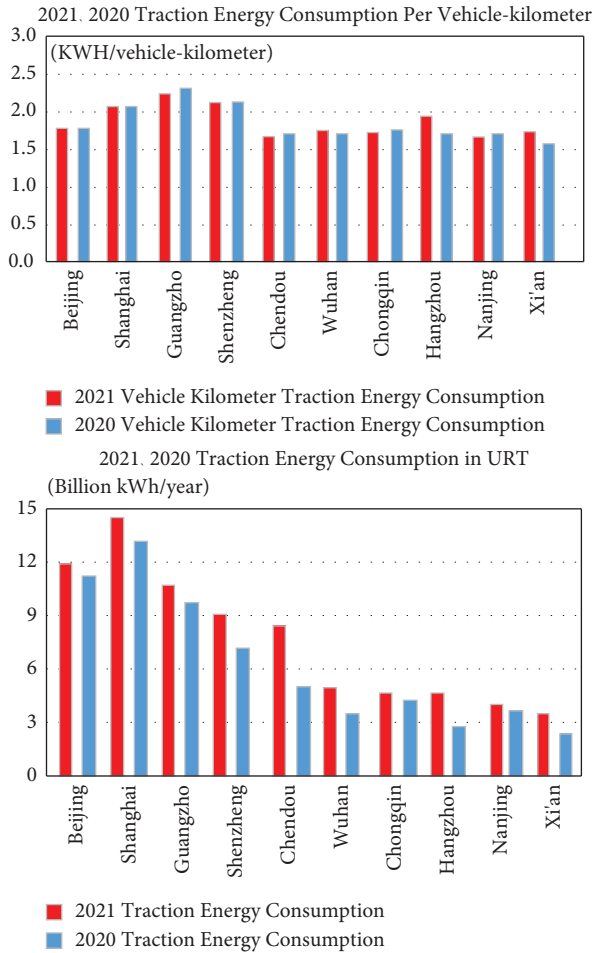


FIGURE 1: Development of urban rail transit in China.

There is massive data on TEC, but it fails to fully use it and clarify more targeted abnormal detection. Thus, a unified threshold is set for the abnormal analysis based on periodic statistics and empirical conclusions. By comparing and analyzing the advantages and disadvantages of engineering practice and theoretical research methods, this paper explores the applicability of traditional anomaly detection algorithms to anomaly TEC analysis. The main contributions of this work are as follows:

- (i) We propose typical TEC values comprising three elements, namely, research object, evaluation index, and time scale. The evaluation index determines the specific analysis framework under different time scales.
- (ii) Our method calculates typical values based on time series data, using symbolic approximate aggregation, and clustering algorithms to analyze the specific TEC patterns. The prophet algorithm predicts typical cumulative energy consumption index values for each energy consumption mode.
- (iii) We use a similarity-based anomaly analysis method to detect the TEC pattern abnormalities, using string vector distance as a similarity measure.

2. Related Work

The energy consumption in the URT system caused by rapid development has attracted the attention of many scholars at home and abroad. The energy consumption analysis methods rarely use the natural energy consumption data generated in the actual train operation for analysis and mining. Even fewer scholars use these data to conduct relevant studies on abnormal TEC analysis.

By analyzing and comparing the energy consumption data during the actual train operation, Lukaszewicz divided the influencing factors of TEC into three categories, namely, basic parameters, driving strategies, and external factors [1]. Xiaobin et al. analyzed the energy consumption component of the traction system and its influencing factors by collecting TEC data [2]. Bo and Hui analyzed the influence of TEC, such as train mass, line slope type, and running resistance through simulation methods [3]. Based on the measured data, REN et al. divided the factors affecting the TEC into three categories as follows: infrastructure conditions, transportation organization, and external environment, and quantitatively analyzed the train features and lines contained in each category through data mining methods [4].

The anomaly detection approach combines many fields, e.g., machine learning and statistics. It can be divided into density-, proximity-, and model-based anomaly detection algorithms according to the anomaly detection implementation. As a typical density-based anomaly detection approach, the local outlier factor (LOF) algorithm can simultaneously determine anomalies and quantitatively analyze the anomaly degree [5]. Xiaoxia et al. made a clustering analysis on the original energy consumption data to obtain different energy consumption characteristics. The decision tree method is used to detect energy consumption modes in classified datasets. The dynamically collected data can detect anomalies based on the LOF algorithm, which can analyze the anomalies of each sampling point [6].

In proximity-based anomaly detection, anomalies are defined as objects far away from most data. Its core lies in defining the proximity of data objects, e.g., Euclidean distance, Jaccard, and cosine similarity measure. Based on MapReduce architecture, Cao et al. proposed a distance-based outlier detection (DOD) method for a distributed database system with TB-level volume. This method can realize anomaly detection under massive data with less communication cost [7]. Bin and Yifei constructed robust mean and covariance matrix estimators and proposed an anomaly detection approach based on robust Mahalanobis distance to detect outliers caused by registration errors and measurement errors [8]. Proximity-based algorithm to calculate the proximity of a large number of data has large time and space complexity, high calculation cost, and poor applicability for datasets with sizeable regional density changes.

Model-based anomaly detection requires statistical models to describe normal data and detect abnormal data. Hong proposed a new test statistic based on the sample quantile for extreme value distribution, suitable for simple data elimination tasks [9]. Habib et al. used clustering

algorithms and normalization to detect anomalies in the sensor data [10]. Peng et al. filtered noise data and used clustering algorithms to detect node anomalies in wireless sensor networks [11]. Tang Shulu et al. used density peak clustering to detect abnormal targets in low-dimensional space for hyperspectral image data [12].

3. Architecture of Specific TEC in URT System

This chapter proposes the system architecture of TEC typical values, discussing their significance and constructing a standard value system with three elements, namely, index, time scale, and specific TEC patterns and energy consumption indexes. An anomaly analysis framework from mode to point is proposed based on the standard value system.

3.1. Composition of the TEC Typical Values. The data analysis of TEC prioritizes prediction and evaluation, with little focus on anomaly detection. Anomaly detection in other fields often employs machine learning algorithms, but it lacks robust explanations and practical applicability. The improved TEC evaluation scheme expands on the existing methods, including line unit consumption, typical values, and a comprehensive evaluation index. The scheme comprises three elements as follows: research object, evaluation index, and time scale, allowing for evaluation of lines, trains, and power consumption units on hourly, daily, weekly, monthly, and yearly scales [13]. Its architecture is depicted in Figure 2.

The upgraded TEC assessment plan uses historical data as a standard for identifying TEC anomalies during regular URT operation. The energy consumption data is collected, preprocessed, and compared to the established TEC plan. Analysts identify the cause of any abnormality and guide engineers to address the issue, as shown in Figure 3.

4. TEC Anomaly Analysis Method

Line level: the accumulated TEC data collected and uploaded during all operations on a certain line and the mileage data recorded by the transportation management system (TMS) were summarized, the incorrect data were eliminated, and then the TEC data for each statistical period were calculated.

Train level: firstly, discrete time series are constructed based on the historical cumulative TEC data generated by train operation. Specific TEC patterns were obtained by mode analysis and verified by combining common operation diagrams and training sets. The first step of data analysis is to determine whether there is an abnormal TEC mode by comparing its similarity with each typical TEC pattern. Then, the accumulated energy consumption values were compared at each point of the peculiar energy consumption mode with the standard weight of the energy consumption index point by the end to determine the abnormal time point.

Energy consumption unit level: TEC is divided into traction unit energy consumption and auxiliary energy consumption. Also, the same analysis process as that of the

train is adopted for traction unit energy consumption, i.e., the analysis of abnormalities from the typical mode to the specific values of the cumulative energy consumption index. For the auxiliary energy consumption, the accumulated energy consumption index per unit of time is solved, combined with the usage of additional equipment to determine the abnormality.

4.1. Anomaly Detection Algorithm

4.1.1. Data Dimensionality Reduction. The anomaly detection algorithm is as seen in Algorithm 1. First, we need to reduce the accumulated TEC dimension. Given a time series of lengths $Q = q_1$ and q_2, \dots, q_m . We turn it into a data sequence of length w . $Q' = q'_1, q'_2, \dots, q'_w$, where $w < m$. Then, the compression ratio for dimensionality reduction of time series data is k , and q'_i satisfies the following equation:

$$k = \frac{m}{w},$$

$$q'_i = \frac{1}{k} \sum_{j=k(i-1)+1}^{k*i} q_j, i = 1, 2, \dots, w. \quad (1)$$

In order to reduce the dimensionality further, the Piecewise Aggregate Approximation (PAA) is usually applied prior to the symbolic aggregate approximation (SAX). SAX is used to transform a sequence of rational numbers (i.e., a time series) into a sequence of letters (i.e., a string). An illustration of a time series of 128 points converted into the word of 8 letters. Besides, we use the 4 symbol alphabets a, b, c, and d as in Figure 4. The cut lines for this alphabet are shown as the thin blue lines on the plot given below.

4.1.2. Z-Normalize Data. Before transforming time series with PAA, we Z-normalize data. Time series subsequences tend to have a high Gaussian distribution. The standardization step is based on the Z-score method, where the original dataset is transformed to satisfy the Gaussian distribution of $N(0,1)$, $\mu = 0$, and $\sigma = 1$, and the standardization formula is as follows:

$$Q' = \frac{q'_i - \mu}{\sigma}. \quad (2)$$

The normalized time series has a Gaussian distribution, which is discretized using a sequence of breakpoints denoted as B. The breakpoints partition the distribution into equal probability intervals, and sequence values are approximated using the breakpoint list and PAA. The $\mu - 1$ values of the breakpoint list correspond to the standard normal distribution random variables, as shown in Table 1. The probability values of the Gaussian distribution corresponding to the adjacent breakpoints are equal.

4.1.3. PAA Follows the Standard Procedure. To detect anomalous patterns in feature data, we convert the time series to PAA representation and then to symbols. We use a pattern discovery algorithm combined with a time series

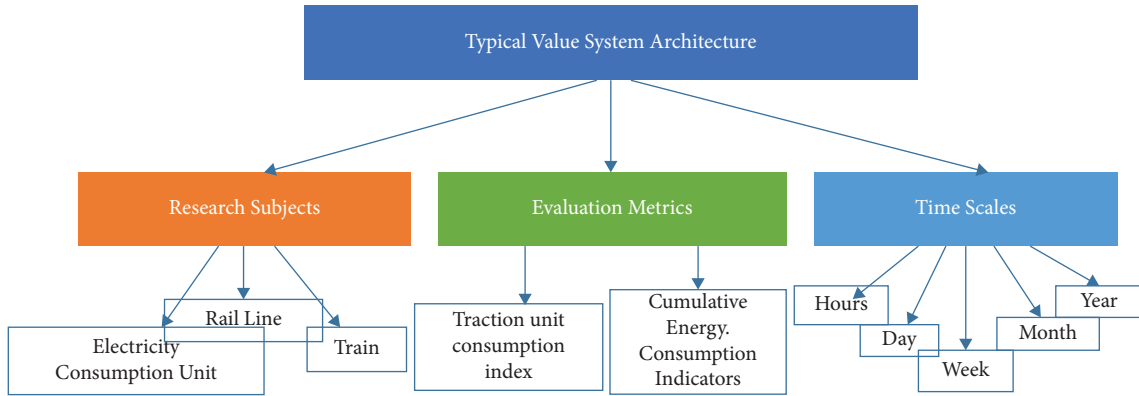


FIGURE 2: The structure of TEC typical values.

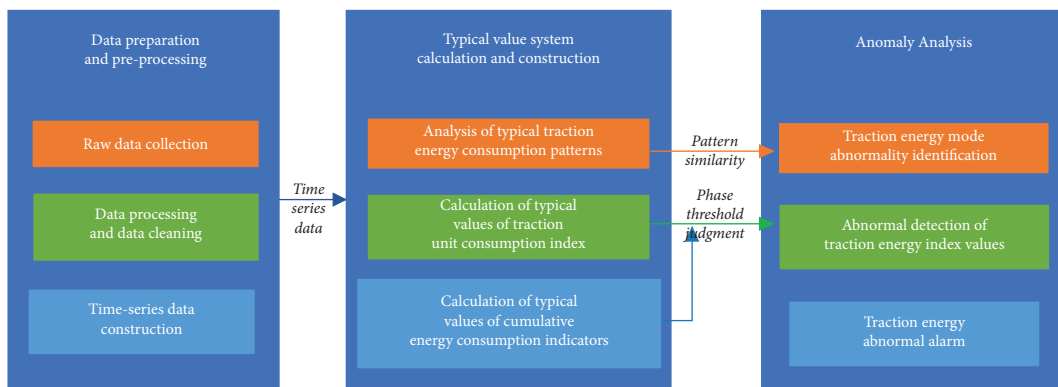


FIGURE 3: The framework of outlier analysis.

Require: T:24-dimensional original TEC time series data, n :the length of time series discord.

Ensure: The length of Discord and strings of length ω .

```

(1) function Z-normalization (ts)
(2)   ts.mean ← mean (ts)
(3)   ts.dev ← sd (ts)
(4)   (ts - ts.mean)/ts.dev
(5)   ts_znorm = Z-NORMALIZATION (ts)
(6) function PAA (ts, paa_size)
(7)   len == paa_size
(8)   if len == paa_size then
(9)     ts
(10)  else if len%%paa_size == 0 then
(11)    colMeans (matrix (ts, nrow = len %% paa_size, byrow = F))
(12)  else
(13)    res = rep.int (0, paa_size)
(14)  end if
(15)  return s_paa = paa (ts_znorm, paa_size)
(16) end function
(17) Use the 4 symbols alphabet a,b,c,d
(18) SAX transform of ts into string through 9-points PAA: "baabcbbc":
(19) ts_to_string (dat_paa_9, cuts_for_asize (9))
(20) discords = find_discords_hotsax (dd)
  
```

ALGORITHM 1: Discretization of the PAA representation of time series into SAX.

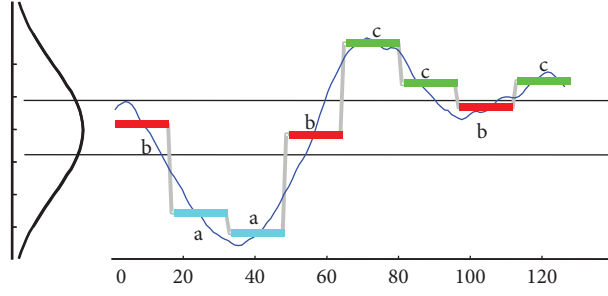


FIGURE 4: Time series processed by PAA.

TABLE 1: The breakpoints.

β_i	μ							
	3	4	5	6	7	8	9	
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	
β_4			0.84	0.43	0.18	0	0.14	
β_5				0.97	0.57	0.32	0.14	
β_6					1.07	0.67	0.43	
β_7						1.15	0.76	
β_8							1.22	

distance metric based on the nonlinear statistical feature representation. However, relying only on the mean can result in the lost information, as two series with different patterns can have the same mean and variance. Therefore, we also use morphological features like slope and angle to accurately represent a time series. The slope values for each segment of the compressed subsequence can be calculated using the following equations:

$$\bar{q}_i = \frac{k \sum_{j=j_0}^{k+i} j q_j - (\sum_{j=j_0}^{k+i} j) (\sum_{j=j_0}^{k+i} q_j)}{k \sum_{j=j_0}^{k+i} j^2 - (\sum_{j=j_0}^{k+i} j)^2}, \quad (3)$$

$$j_0 = k(i-1) + 1. \quad (4)$$

5. Experiments

This section uses the actual operating energy consumption data collected from a train set of the Beijing URT Operating Company. The train consisted of 6 motor vehicles without cabs and 2 trailer vehicles with cabs, and data collection involved recording second-level cumulative traction unit energy consumption for each motor vehicle and secondary cumulative auxiliary energy consumption for each trailer. Hourly cumulative TEC data for the train's daily traction energy mode analysis was obtained by processing the original data file. The resulting 24 data points form a 24-dimensional original TEC time series data $Q = q^1, q^2, \dots, q^{24}$, with each dimension representing the cumulative TEC of each period.

The data are divided into 24 dimensions, and a morphological feature-based symbolic representation method is used to identify three TEC patterns in the time series data.

The algorithm involves transforming the parameter time series into characters with actual semantics by first converting the original time series into the PAA representation and then converting the PAA data into a string. The algorithm is implemented in Python and is available on PyPi for installation using pip.

5.1. Characteristics of TEC Data. This study utilizes data from the energy consumption metering devices installed on several lines and train groups in the Beijing Subway. The device records instantaneous voltage and current values and accumulated energy consumption, consisting of a voltage sensor, current sensor, and metering module per vehicle [14]. Each vehicle is equipped with a set of multitrain energy consumption measurement devices, as depicted in Figure 5.

Each metering device wirelessly uploads data to the database terminal, which can be analyzed to retrieve instantaneous voltage, current, and accumulated energy consumption values. The motor train data file records the cumulative energy consumption and regenerative energy of the traction unit, while the trailer data file records the cumulative energy consumption of all auxiliary equipment powered by the additional inverter. Tables 2 and 3 present the data for the motor train and trailer, respectively.

The original data file records regenerative energy as negative due to electric braking, and trains frequently switch between traction and braking, resulting in fluctuating instantaneous voltage and current values. Analyzing energy consumption based on voltage and current values alone is challenging, so this study uses the accumulated energy consumption as the research object. Hourly cumulative energy consumption values are obtained from the original data file, and discrete univariate time series data is

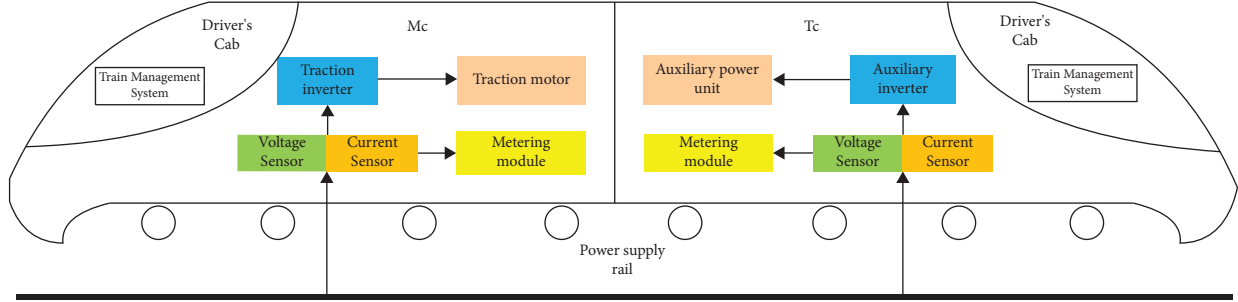


FIGURE 5: The structure of metering device.

TABLE 2: Dataset of motor train.

Time	Voltage (V)	Electric current (A)	Cumulative energy consumption (kW·h)	Regenerative energy (kW·h)
2022/5/21 11:34:01	813.3	380.4	248.1	-11.0
2022/5/21 11:34:02	838.9	268.4	248.2	-11.0
2022/5/21 11:34:03	844.9	219.3	248.3	-11.0
2022/5/21 11:34:04	857.7	152.0	248.3	-11.0
2022/5/21 11:34:05	866.2	93.9	248.3	-11.0
2022/5/21 11:34:06	873.0	56.1	248.4	-11.0

TABLE 3: Dataset of trailer train.

Time	Voltage (V)	Electric current (A)	Cumulative energy consumption (kW·h)
2022/5/21 11:34:01	886.8	22.5	109.5
2022/5/21 11:34:02	886.4	23.4	109.5
2022/5/21 11:34:03	887.0	23.8	109.5
2022/5/21 11:34:04	886.4	23.7	109.5
2022/5/21 11:34:05	886.6	24.0	109.6
2022/5/21 11:34:06	887.0	22.9	109.6

constructed based on these values. The time series data consists of cumulative energy consumption values from a train on the Beijing Subway between August 5th and August 11th, 2021. The time series curve of energy consumption is depicted in Figure 6.

The TEC level is impacted by the complex energy flow process and changeable train operating conditions. In engineering practice, the TEC index's general value for each line is determined based on the historical statistical data, and the fluctuation interval is set as the threshold for rough abnormal judgment. Figure 7 shows the average traction unit consumption of the Beijing metro based on the field investigation and historical data statistics.

TEC time series data display periodic and seasonal characteristics, with energy consumption values affected by the total load rate and auxiliary equipment opening. Fluctuations in the time series curve are complex and typically contain multiple peaks. The trend of the time series represents changes in train TEC levels over time, with each point's energy consumption value related to the adjacent periods. The original dataset used in this study is the data file uploaded by the energy consumption metering device, with high data quality and few errors or missing data despite occasional failures in collection, calculation, storage,

transmission, and analysis links. Time series subsequences tend to have a high Gaussian distribution in Figure 8.

5.2. Similarity Measure. The similarity measure, as a measure of how close two things are, is used to measure the anomalies in a single time series, as shown in Figure 9. The closer two things are, the more similar they are, while the farther away two things are, the less similar they are. Dist is a function that takes sequences Q ($Q = q_1 \dots q_m$) and C ($C = c_1 \dots c_m$) as parameters and returns a non-negative value R , which is considered as the similar distance between the two and must be symmetric. Their Euclidean distance is defined as the first equation given below. In the second equation, PAA distance lower-bounds the Euclidean Distance.

$$D(Q, C) \equiv \sqrt{\sum_{i=1}^m (q_i - c_i)^2}, \quad (5)$$

$$D_{R(\bar{Q}, \bar{C})} \equiv \sqrt{k} \sqrt{\sum_{i=1}^w (\bar{q}_i - \bar{c}_i)^2}. \quad (6)$$

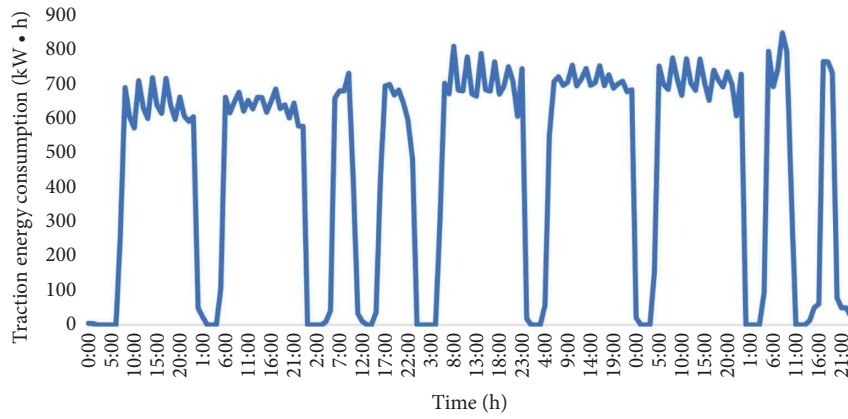


FIGURE 6: The time series curve of energy consumption.

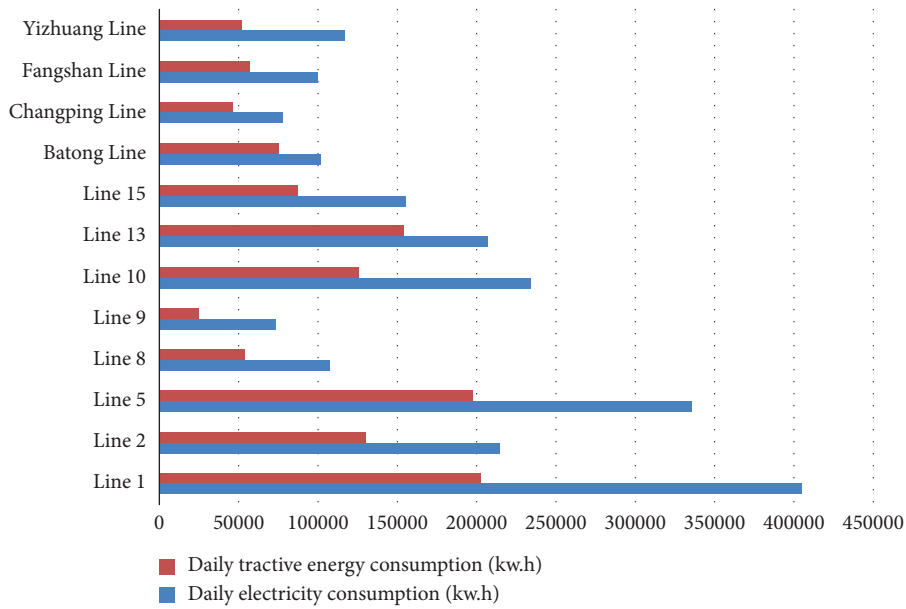


FIGURE 7: The average traction unit consumption of each line in Beijing subway.

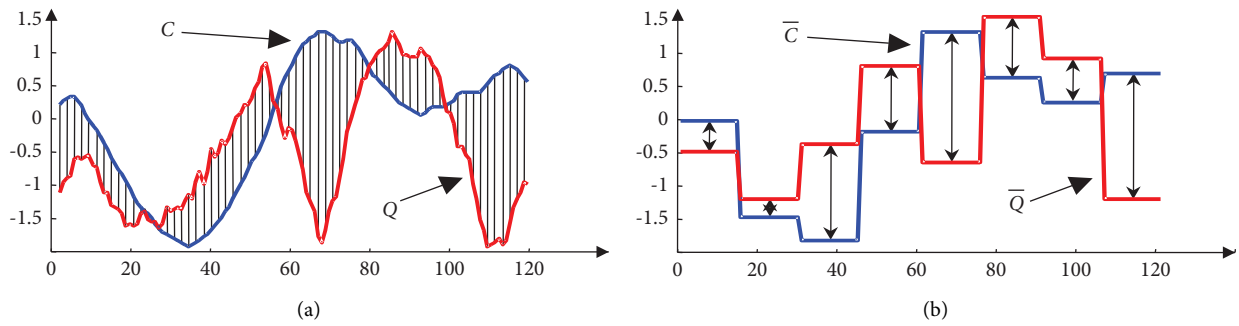


FIGURE 8: (a) PAA distance lower-bounds the Euclidean distance. (b) Euclidean distance.

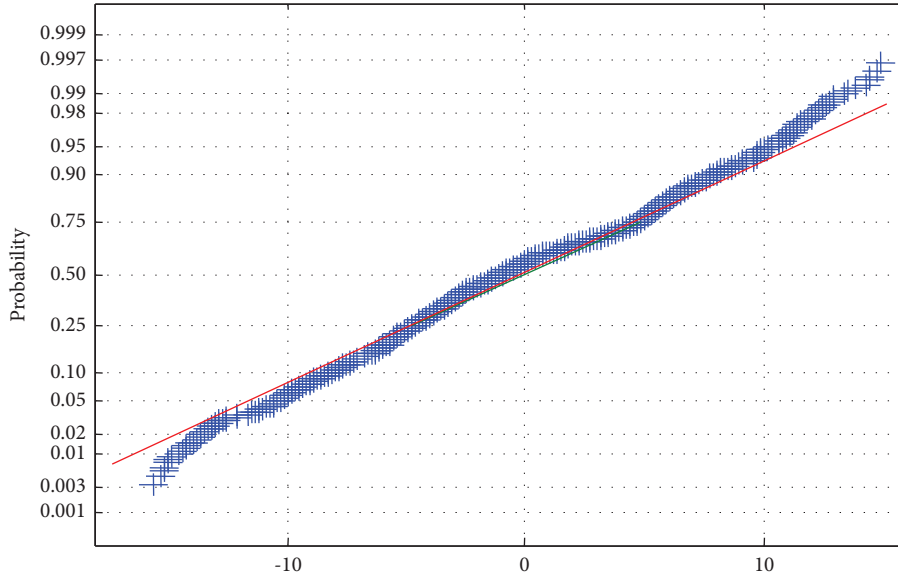


FIGURE 9: A normal probability plot of the distribution of values from subsequences of length 128.

Equation (7) defines a function that computes the minimum distance between the string representations of the original time series O and C . This function can be efficiently implemented using table lookup. Additionally, time series subsequences exhibit a Gaussian distribution, which is a characteristic tendency.

$$\text{MTNDIST}(\hat{Q}, \hat{C}) \equiv \sqrt{k} \sqrt{\sum_{i=1}^w (\text{dist}(\hat{q}_i, \hat{c}_i))^2}, \quad (7)$$

where the dist function is implemented by using the lookup table for the particular set of the breakpoints (alphabet size), as shown in the table below, and where the singular values for each cell (q, c) is computed as follows:

$$\text{cell}(q, c) = \begin{cases} 0, & \text{if } |q - c| \leq 1, \\ \beta_{\max(q,c)-1} - \beta_{\min(q,c)-1}, & \text{otherwise.} \end{cases} \quad (8)$$

To convert a time series of an arbitrary length to SAX, we need to define the alphabet cuts. Saxpy retrieves cuts for a normal alphabet (we use size 3 here) via `cuts_for_asize` function: `from saxpy import cuts_for_asize`
`Cuts_for_asize(3)`.

First, we use the “`ts_to_string`” function to convert a time series into letters using SAX. However, before applying this function, we must z-normalize the input time series using a normal alphabet to obtain a string, such as `abcba`. Next, to investigate the structure of the input time series and identify any anomalous (i.e., discords) or recurrent (i.e., motifs) patterns, we used the “time series to SAX conversion via sliding window” approach. This approach is commonly employed, and Saxpy implements this workflow. The result is represented as a data structure of resulting words and their respective positions on time series as follows:

```
defaultdict(list,
'aac': [4, 10, 11, 30, 35],
```

```
'abc': [12, 14, 36, 44],
'acb': [5, 16, 21, 37, 43],
'acc': [13, 52, 53],
'bac': [3, 19, 34, 45, 51],
'bba': [31],
'bbb': [15, 18, 20, 22, 25, 26, 27, 28, 29, 41, 42, 46],
'bbc': [2],
'bca': [6, 17, 32, 38, 47, 48],
'caa': [8, 23, 24, 40],
'cab': [9, 50],
'cba': [7, 39, 49],
'cbb': [33],
'cca': [0, 1])
```

Anomalies in TEC patterns are defined as operating conditions that deviate significantly from the specific TEC patterns. In time series data mining, retrieval, clustering, classification, summary, and anomaly detection are usually performed based on series similarity, including temporal similarity, shape similarity, and change similarity. Similarity measures based on Minkowski distance, cosine similarity, correlation, and mutual information are often used to measure the similarity of two time series. Euclidean distance model is simple, intuitive, easy to understand, and fast, and it is often used to measure the similarity of discrete time series.

The Euclidean distance between the two time series can be expressed as the square root of the sum of the squared differences of each pair of corresponding points. The distance metric defined by the PAA approximation can be viewed as the square root of the sum of the squared differences between each pair of corresponding PAA coefficients multiplied by the square root of the compression rate, as shown in Figure 10. The distance between two SAX representations of a time series requires finding the distance

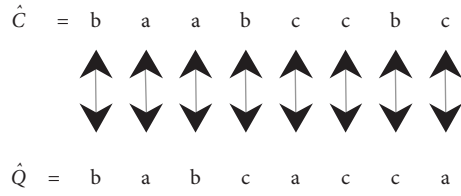


FIGURE 10: The symbols between two time series after PAA.

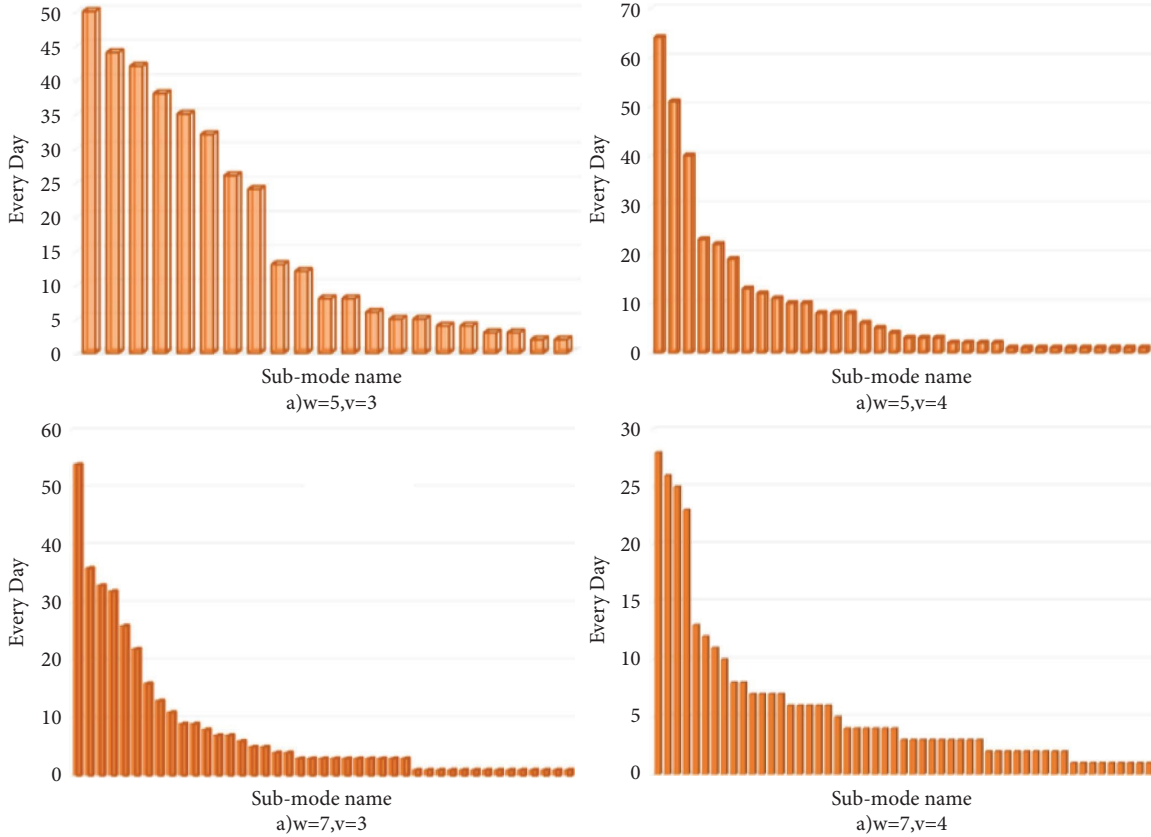


FIGURE 11: Number of patterns and days under different values.

between each pair of symbols, squaring them, summing them, taking the square root, and finally multiplying by the square root of the compression rate. By rigorous proof, we get it in the following equation:

$$\sqrt{\sum_{i=1}^n (q_i - c_i)^2} \geq \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^w (\bar{q}_i - \bar{c}_i)^2} n(\bar{Q} - \bar{C})^2 \geq n(\text{dist}(\hat{Q}, \hat{C}))^2. \quad (9)$$

5.3. *Experimental Results.* Higher values of ω and μ result in more detailed energy consumption levels and more complex TEC submodes, which can have a significant impact on

the subsequent analysis using ML algorithms. The TEC pattern analysis aims to investigate the TEC level's variation over time within a day, using daily TEC time series data as the research object. Low energy consumption levels are represented by the letter a , high energy consumption levels by c , and medium energy consumption levels by b . For example, on August 11th, 2021, when $\omega = 7$ and $\mu = 3$, the original time series is represented as the string $baabccbc$ after processing and conversion. The data of the subject train's 354 days of operation in 2021 are processed to form 42 string vectors representing various TEC variation patterns, as shown in Figure 11. For the submodel conclusions, refer to Table 4.

TABLE 4: Subpatterns of PAA processing.

PAA	Days
Aaabbbccc	63
Abbbbcca	43
Aaabccccc	38
Aacbbccc	36
Abcbbccc	31
Ccabbbcac	27
Acabbbccc	22
Accbbccc	19
Caabbbccc	16
Aaabcbccc	14
Aabbabccc	13
Acbbbccc	10
Bbabbbcaa	9
Accaabccc	9
Aabbabccc	8
Abbaabccc	7
Aaacbccc	6
Bbaaabcaa	6
Abaaabccb	5
Baababccc	5
Ccaabccc	4
Abbaabccc	4
Aaccabccc	4
Abbaabcaa	4
Baccabccc	4
Aaaaabccc	3
Abbcabccc	3
Caaacbccc	3
Baabccbc	2
Ccabccbc	2
Baabccac	2
Ccbccbc	1
Acabccbc	1
Bacbbabc	1
Aaabccbc	1
Acabccbc	1
Baccbcbc	1
Acabccccc	1
Ccbaacbc	1
Caababbc	1
Cbabccab	1
Cbbaacbc	1

6. Conclusion

Based on the improved TEC evaluation scheme and anomaly analysis framework, we propose an anomaly analysis method of TEC for urban rail lines, trains, and traction units. The value anomaly detection method based on the improved TEC evaluation scheme combines the advantages of mathematical statistics, prediction algorithms, and manual experience in setting thresholds. It has a good adaptability to the characteristics of TEC data, analysis needs, and practical

applications. In the numerical simulation experiments, the effectiveness of the new method for TEC analysis is verified by comparing the feature identification and anomaly detection results. Meanwhile, compared with the traditional way, the new approach is able to find and detect the anomalous patterns better and has stronger robustness.

Data Availability

No datasets are available to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by the Beijing Natural Science Foundation (L221016).

References

- [1] P. Lukaszewicz, *Energy consumption and running times for trains issn 1103-470x; isrn kth/fkt/d-01/25-se*, SAGE, Tyne, UK, PhD, 2001.
- [2] Z. Xiaobin, B. Yun, and Z. Shanshan, "Metro train energy consumption characteristics based on empirical data analysis," *Journal of Transportation Systems Engineering and Information Technology*, vol. 21, no. 6, pp. 264–271, 2021.
- [3] L. Bo and L. Hui, "Research on the Beijing subway energy management platform," *Information Security and Technology*, vol. 7, no. 1, 2016, (in Chinese).
- [4] J. H. Ren, Q. Zhang, and F. Liu, "Analysis of factors affecting traction energy consumption of electric multiple unit trains based on data mining," *Journal of Cleaner Production*, vol. 262, no. 1, pp. 47–58, 2020.
- [5] J. Hang, L. Tun, G. Hansu, D. Xianghua, and G. Ning, "A dynamic real-time abnormal detection method for university building energy consumption," *Computer Engineering*, vol. 34, no. 4, p. 7, 2017.
- [6] Q. Xiaoxia, X. Dan, and W. Bo, "Data mining method for real-time monitoring of energy consumption," *Journal of Chongqing University*, vol. 35, no. 7, pp. 133–137, 2012.
- [7] L. Cao, Y. Yan, C. Kuhlman, Q. Wang, and E. A. Rundensteiner, "Multi-tactic distance-based outlier detection," in *Proceedings of the IEEE International Conference on Data Engineering*, San Diego, CA, USA, April 2017.
- [8] W. Bin and C. Yifei, "Multivariate outlier detection based on robust mahalanobis distance," *Statistics and Decision Making*, vol. 3, 2005.
- [9] Z. Hong, "Test of multiple outliers in type i extreme distribution samples," *Journal of UEST of China*, vol. 18, 2008.
- [10] U. Habib, G. Zucker, M. Blochle, F. Judex, and J. Haase, "Outliers detection method using clustering in buildings data," in *Proceedings of the Conference of the IEEE Industrial Electronics Society*, Yokohama, Japan, November 2015.

- [11] Z. Peng, F. Xin, and Z. Jianguo, "Clustering anomaly detection algorithm based on spatial correlation in wireless sensor networks," *Application Research of Computers*, vol. 30, no. 5, p. 4, 2013.
- [12] T. Shulu, Z. Chunhui, and C. Ying, *Improved Detection of Hyperspectral Anomalies Based on Density Peak Background Purification*, Engineering Journal of Heilongjiang University, Harbin, China, 2021.
- [13] L. Zhu, Y. Li, F. R. Yu, B. Ning, and X. Wang, "Cross-layer defense methods for jamming-resistant cbtc systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2020.
- [14] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled cbtc systems against mitm attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–13, 2020.

Research Article

RIS-Assisted Coverage Optimization for 5G-R Channel in Station Scenario Based on ML and RT

Bin Lu ¹, Xinghai Guo ², Xinyi Shan ², and Fang Li ³

¹China Telecom Research Institute, Guangzhou 510630, China

²School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

³Guangzhou Maritime University, Guangzhou 510700, China

Correspondence should be addressed to Fang Li; fanglee.whut@gmail.com

Received 19 October 2022; Revised 15 March 2023; Accepted 26 April 2023; Published 6 June 2023

Academic Editor: Ardashir Mohammadzadeh

Copyright © 2023 Bin Lu 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.

As one of the most important radio communication scenarios in vertical industry, the high-speed railway (HSR) station is facing the challenge of coverage optimization due to its complex structure. Regarding the wireless network planning and optimization of HSR stations as a part of the customized network, this paper makes an analysis on the 5G-R channel in the HSR station scenario. Channel characteristics, including path loss, power ratio (PR), and angular spread (AS), are extracted on the basis of ray tracing (RT). Multipath components can also be distinguished based on RT. In order to achieve a better performance, the reconfigurable intelligent surface (RIS) technology is adopted to the network deployment. Moreover, machine learning (ML) is used to locate the best direction of the beam. The analysis results show that the received power in RIS-assisted channels is significantly promoted. Our research can provide a deep understanding to the 5G-R channel in station scenario and a well reference for the design and optimization of the customized network.

1. Introduction

From 2019, global operators began to deploy 5G networks and 5G customized networks commercially. The 5G customized network is the best practice private network for cloud network integration and the network foundation for accelerating the digital transformation of thousands of businesses.

High-speed railway (HSR) station is an important part of the railway transportation system as well as one of the important scenarios of the 5G customized network. The construction of “smart railway station” needs the solid support of an efficient and stable wireless communication network, which can provide a high speed data transmission and satisfactory passenger service. However, the current global system for mobile communications railway (GSM-R) system suffers insufficient bearing capacity and serious band interference problems [1]. As a development of the mobile communication technology, the fifth-generation communication (5G) can meet the requirements such as high

reliability and low latency. Thus, the 5G-railway (5G-R) communication system will be an evolutionary option for the next-generation railway communication [2].

In the existing research, most of the efforts made on the HSR scenarios focus on urban scenario [3], rural scenario [4], tunnels [5], and crossing bridges [6]. Due to the different structural features from the aforementioned scenarios, HSR station scenarios generally include metallic surfaces of train bodies, railway equipment, and other devices. This will make different channel characteristics distinguished from other scenarios [7]. Therefore, the existing statistical models cannot be applied to predict the coverage inside the station directly.

Generally, lots of methods have been studied for channel modeling. Ray tracing (RT) is one of the most popular deterministic modeling methods, which can be used to describe the wave propagation in this scenario [8]. The RT method can abstract the electromagnetic waves with various propagation mechanisms into the light-like rays [9]. Since the reflection paths can be extracted through RT simulations,

it is technically practical to use the reconfigurable intelligent surface (RIS) to change channel characteristics to optimize the coverage [10], which can meet challenges of the coverage optimization caused by the high frequency band and large path loss in the 5G-R system [11].

As a passive device, RIS can improve the channel without introducing other interference, so it has attracted extensive attention in the research [12, 13]. RIS is a two-dimensional structure with a large number of reflective elements, which can induce an adjustable independent shift on the incident signal [14, 15]. Thus, RIS is suitable for enhancing the coverage in key regions at a low cost [16]. A large number of studies have been conducted using RIS for coverage optimization. In [17], the cell coverage is maximized by optimizing the RIS orientation and horizontal distance. The RIS-assisted channel model is presented in [18], and the principle of phase-shift adjustment at RIS beamforming is applied to enhance the coverage in desired regions [19]. The RIS can be programmed to discrete adjustment of the reflection phase [20], which provides an opportunity for the introduction of various artificial intelligence methods.

In the existing studies, there are two main ideas for the optimization of RIS. The first one is to calculate the RIS-assisted channel model and optimize the required parameters [17, 21, 22]. The calculation of this case is accurate, but it is difficult to apply to large and complex systems. The second idea is to use machine learning methods, including deep learning (DL) and reinforcement learning (RL), to solve the required parameters [23, 24]. Moreover, in relevant studies, the authors in [25] propose that the communication system can be abstracted as a fuzzy system. The artificial neural networks (ANNs) can be built to optimize the parameters of the complex communication system [26].

In this paper, the 5G-R wireless network planning and optimization for HSR stations are conducted, which are assisted by the RIS technology and ML algorithm for the best direction orientation of the beam. The extraction of 5G-R channel characteristics is achieved by the RT technology. The main contributions of this paper are as follows:

- (1) High precision reproduction of wireless signal propagation in HSR station scenario and channel characteristics extraction based on RT platform are conducted. Focusing on the complex structure, we build the physical model of the HSR station. Considering the rich multipath components of the radio propagation in the HSR station, we analyze the distribution and sources of electromagnetic waves using RT platform.
- (2) RIS-assisted wireless network planning and optimization for HSR station scenario are designed. The deployment of the wireless network is mainly based on traversal iteration with high cost and low efficiency. We thus adopt the RIS technology to obtain a better design, which can improve network coverage with a low cost.
- (3) The best beam orientation of RIS based on the ML algorithm is realized. In order to obtain the best

radiation direction, we use the ML algorithm to locate the best direction of the beam. Thus, with the combination of RIS and ML, wireless coverage can be efficiently and accurately planned and optimized for the HSR station scenario.

The rest of the paper is organized as follows: Section 2 introduces the RT configuration. Section 3 analyzes channel parameters and proposes the advice. Finally, we draw the conclusion in Section 4.

2. Ray-Tracing Simulations

A high-performance computing cloud-based platform named CloudRT is applied to simulate the wave propagation inside the HSR station [9]. The CloudRT architecture can be described as the structure in Figure 1. A typical HSR station scenario is reconstructed as shown in Figure 2(a). The model consists of ground, railway, platform, awning, columns, and electric traction racks. Surfaces of these scatterers are presented in the figure. The material's electromagnetic (EM) parameters at 2.1 GHz band of different parts are listed in Table 1. As one of popular deterministic channel modeling methodologies, ray tracing can provide various channel parameters, including channel transfer function (CTF), center frequency, number of multipaths, multipath components (MPCs) properties, received signal level (RSL), and physical environment information for each TX-RX link. The CTF can be transformed to channel impulse response (CIR) by inverse Fourier transform, from which the power delay profile (PDP), delay spread, power ratio, Doppler spread, etc., can be extracted. The small-scale channel characteristics in power, delay, and frequency domains can thus be characterized. Moreover, for the space domain, the multipath characteristics, containing comprehensive data of the MPCs, including type (i.e., LOS or NLOS path), bounces, delay, transmission distance, electric field strength, transmission loss, angle of arrival (AOA), elevation angle of arrival (EOA), angle of departure (AOD), and elevation angle of departure (EOD) can be obtained as well. For the large-scale channel characteristics, the path loss and shadow fading can be obtained from the RSL and physical environment information which consist of snapshot index, time, TX position, and RX position. Consequently, we can depict the radio channel for the HSR station scenario from multiple dimensions by exploiting RT simulation.

As shown in Figure 2(b), the simulation is carried out on 8 railway lines marked by the red box. The RX is deployed on the top of a HSR train. Simulation is performed every 10 m of movement as the train moves along the selected railway lines. The TX is deployed on the side of the station. The TX and RX adopt directional antenna and omnidirectional antenna, respectively, and their antenna patterns are exhibited in Figure 3.

In order to ensure accurate simulation results as well as efficient computation, a presimulation is conducted to determine the maximum reflection order. The contribution of each order of reflection is calculated via RT simulations with reflected rays up to the 2nd order in Figure 4.

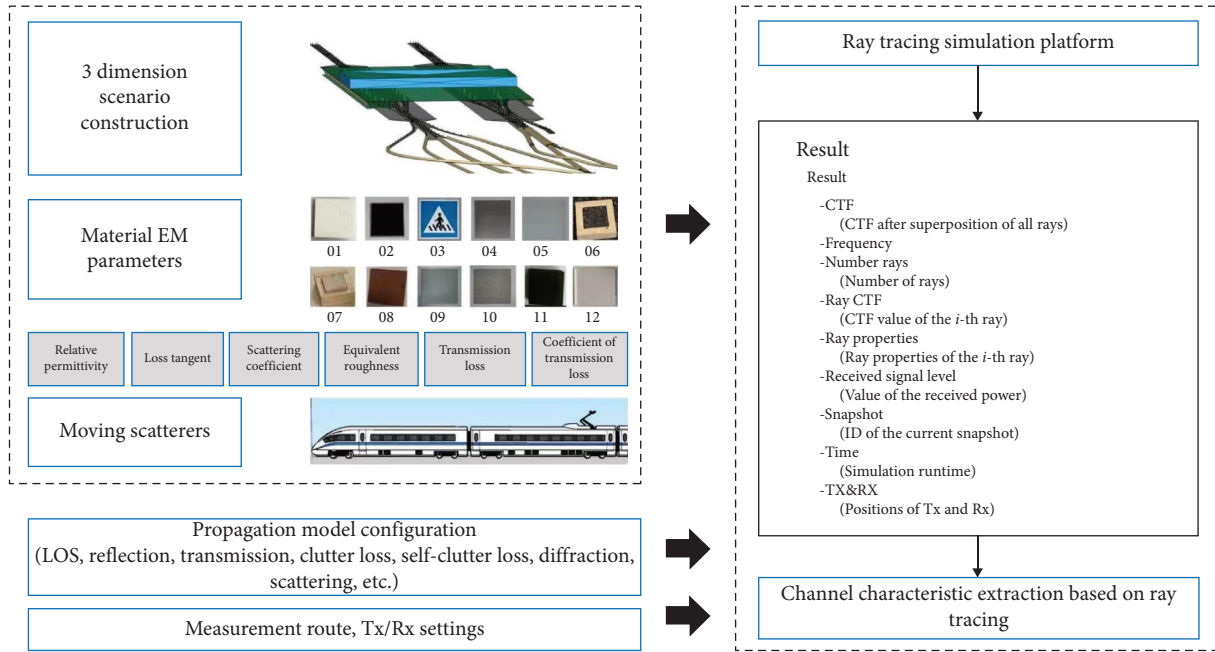


FIGURE 1: CloudRT architecture.

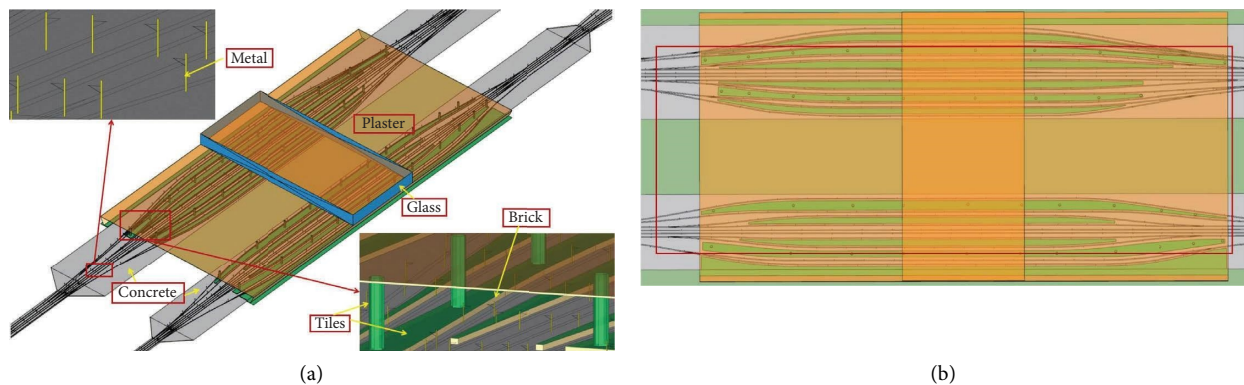


FIGURE 2: (a) Model of HSR station scenario. (b) Simulation region.

TABLE 1: EM properties of materials at 2.1 GHz.

Material	ϵ_r	$\tan \delta$
Metal	1.000	10^6
Glass	6.270	0.140
Brick	3.750	0.091
Concrete	5.310	0.097
Tiles	6.275	0.030
Plaster	2.940	0.058

It can be seen from the power percentage of every order that there is no essential difference between the 1st order and the 2nd order of reflection. Thus, the maximum simulation reflection order is determined as the 1st order. The detailed configuration is listed in Table 2. Simulations are conducted after completing the relevant configurations. One of the snapshots of RT simulation is exhibited in Figure 5, where

the yellow sphere and green sphere represent the location of TX and RX, respectively.

3. Channel Characterization of the Original Channel

In this section, the scenario is divided into the LOS region and the NLOS region. The channel characteristics, including path loss model and angular spread, are analyzed based on the simulation results. Moreover, the coverage inside the station is presented.

3.1. Path Loss Modeling. The simulated and fitted curves of path loss with distance are shown in Figure 6. It can be seen from the figures that the path loss models of both LOS and NLOS regions of the HSR station obey the A-B model as follows:

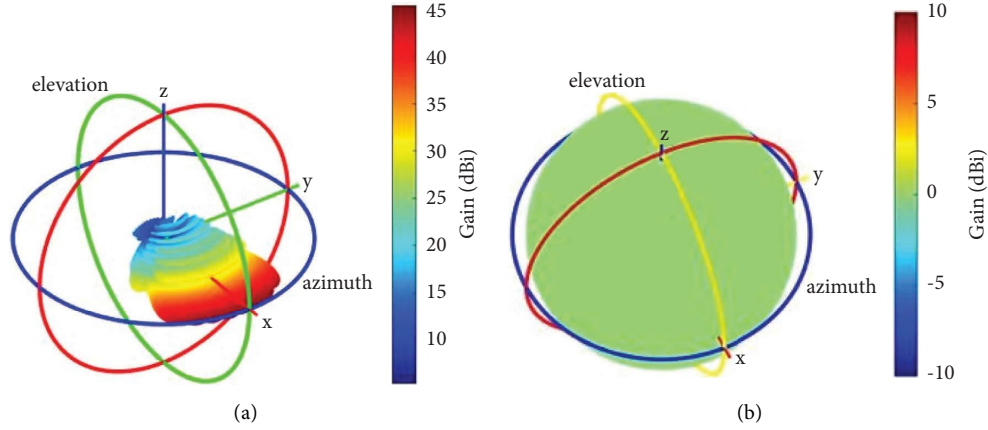


FIGURE 3: Antenna gain patterns of (a) TX antenna and (b) RX antenna.

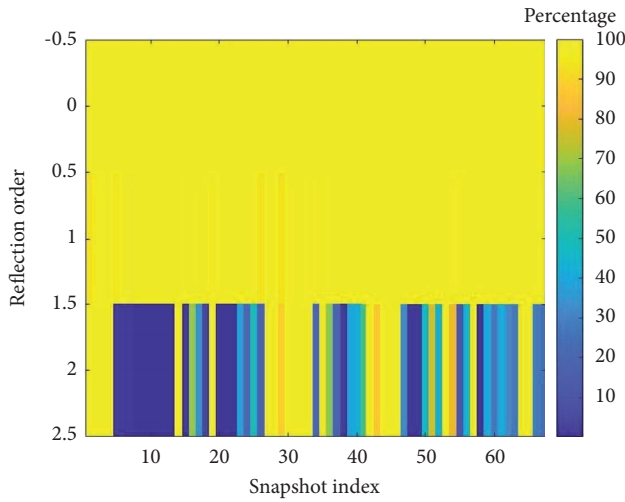


FIGURE 4: Power distribution of every order reflection.

TABLE 2: Configuration of RT simulations.

Item	Configuration
Center frequency	2.1 GHz
Bandwidth	10 MHz
Subcarrier interval	15 kHz
TX antenna type	Vertical polarized directional
RX antenna type	Vertical polarized omnidirectional
TX antenna gain	0 dBi
RX antenna gain	0 dBi
TX height	45 m above the ground
TX power	43 dBm
RX height	0.49 m above the train
Train height	4.5 m

$$PL = A \times 10 \times \log_{10} d + B + X_{\sigma}, \quad (1)$$

where PL denotes the path loss in dB. d is the distance between TX and RX in meters. A and B represent the slope and intercept of the model, respectively. X_{σ} is a Gaussian random variable with a mean of 0 and standard deviation of σ . The parameters in equation (1) are estimated based on the

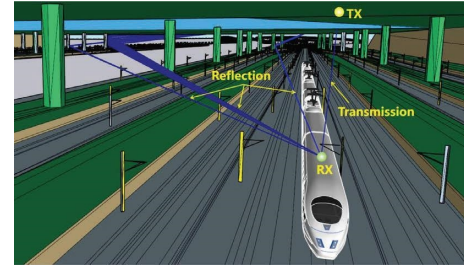


FIGURE 5: One snapshot of RT simulations.

simulation results according to the nonlinear least squares principle as listed in Table 3. The results show that the path loss exponents are 6.77 and 3.99 in the LOS and NLOS regions, respectively.

3.2. Angular Spread. The angular spread is an important characteristic to indicate the channel selective fading in the spatial domain [27], which is one of the references of antenna deployment type in space [28]. Azimuth angular spread of arrival (ASA), azimuth angular spread of departure (ASD), elevation angular spread of arrival (ESA), and elevation angular spread of departure (ESD) are calculated through the same approach of 3GPP protocol [29] as follows:

$$\sigma_{AS} = \sqrt{\frac{\sum_{n=1}^N (\theta_{n,\mu}^2 P_n)}{\sum_{n=1}^N P_n}}, \quad (2)$$

where σ_{AS} represents the angular spread. P_n represents the power of the n -th multipath. N denotes the number of multipaths received by receivers. The values of N and P_n are stored in the RT simulation results. $\theta_{n,\mu}$ is defined by

$$\sigma_{n,\mu} = \text{mod}(\theta_n - \mu_{\theta} + \pi, 2\pi) - \pi, \quad (3)$$

where θ_n represents the azimuth/elevation angle of arrival/departure (AoA, AoD, EoA, and EoD) of the n -th multipath, which can be obtained in the dataset of simulation results. μ_n is calculated by

TABLE 3: Parameters for the fitted model.

Area	A	B	σ
LOS	6.77	-71.94	7.93
NLOS	3.99	-2.96	7.27

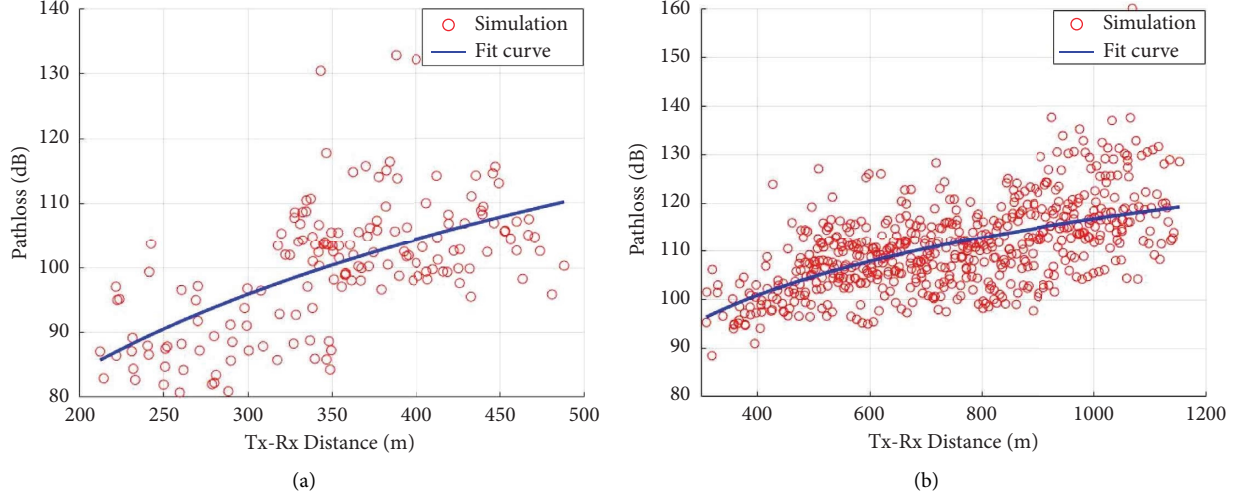


FIGURE 6: Path loss in (a) LOS region and (b) NLOS region.

$$\mu_{\theta} = \frac{\sum_{n=1}^N \theta_n P_n}{\sum_{n=1}^N P_n}. \quad (4)$$

Figures 7 and 8 show the cumulative density functions (CDFs) of the angular spread in LOS and NLOS regions, respectively. The mean values and standard deviations of the four angular spreads in both regions are shown in Table 4, in which μ and σ represent the mean value and the standard deviation, respectively.

From the table, it can be concluded that the overall ASS of the channel in the LOS region are higher than those in the NLOS region. This is due to the fact that the LOS region usually distributes in the open region of the scenario, where multipaths can come from all directions. Thus, the angles between multipaths and the main path are larger. The low height of the scatter in this scenario makes the arrival and departure angles in elevation plane smaller. Consequently, the ASS in the elevation plane are lower than those in the azimuth plane.

3.3. Power Ratio. The power ratio (PR) is defined as the ratio between the power of the strongest path and the sum of the power of the rest paths. This value is generally described as Rician K -factor in the LOS region. The PR [30] is calculated by

$$\text{PR}[\text{dB}] = 10 \log_{10} \left(\frac{P_{\text{strongest}}}{P_{\text{rest}}} \right), \quad (5)$$

where PR denotes the power ratio in dB. $P_{\text{strongest}}$ and P_{rest} represent the power of the strongest path and the sum of rest paths in mW, respectively. The CDFs of PR in both LOS and

NLOS regions are illustrated in Figure 9. It can be found that the mean values of the PR in the LOS and NLOS regions are 5.31 dB and 6.94 dB, respectively.

3.4. SS-RSRP Coverage. The SS-RSRP is defined as the linear average over the power contributions (in mW) of the resource elements (REs) that carry secondary synchronization signals (SS). The SS-RSRP can be calculated by subtracting the path loss from the transmitting power per RE. For 5G-R system, the transmitting power per RE can be calculated by

$$P_r = \frac{P_T \times N_c}{N_{\text{RB}} \times 12}, \quad (6)$$

where P_r denotes the transmitting power per RE. P_T is the transmitting power per channel. N_c is the number of channels. N_{RB} means the number of resource blocks, which equals 51 in this configuration. According to equation (6), the transmitting power per RE is 31.55 dBm. The SS-RSRP of each receiving point in this scenario is then derived and illustrated in Figure 10. According to our investigation in China Mobile Communication Corporation (CMCC), the SS-RSRP threshold is set as -95 dBm. In coverage planning, no more than 5% of the receiving region can be below this threshold. The receiving points with SS-RSRP below and above the threshold are defined as the weak and strong regions as shown in Figure 10. In this situation, a total of 40 receiving points have SS-RSRP below -95 dBm, failing to meet the standard.

From the result shown in Figure 10, it can be found that the coverage in the NLOS region is not very satisfactory. However, Figure 8 shows that the angle spreads are relatively

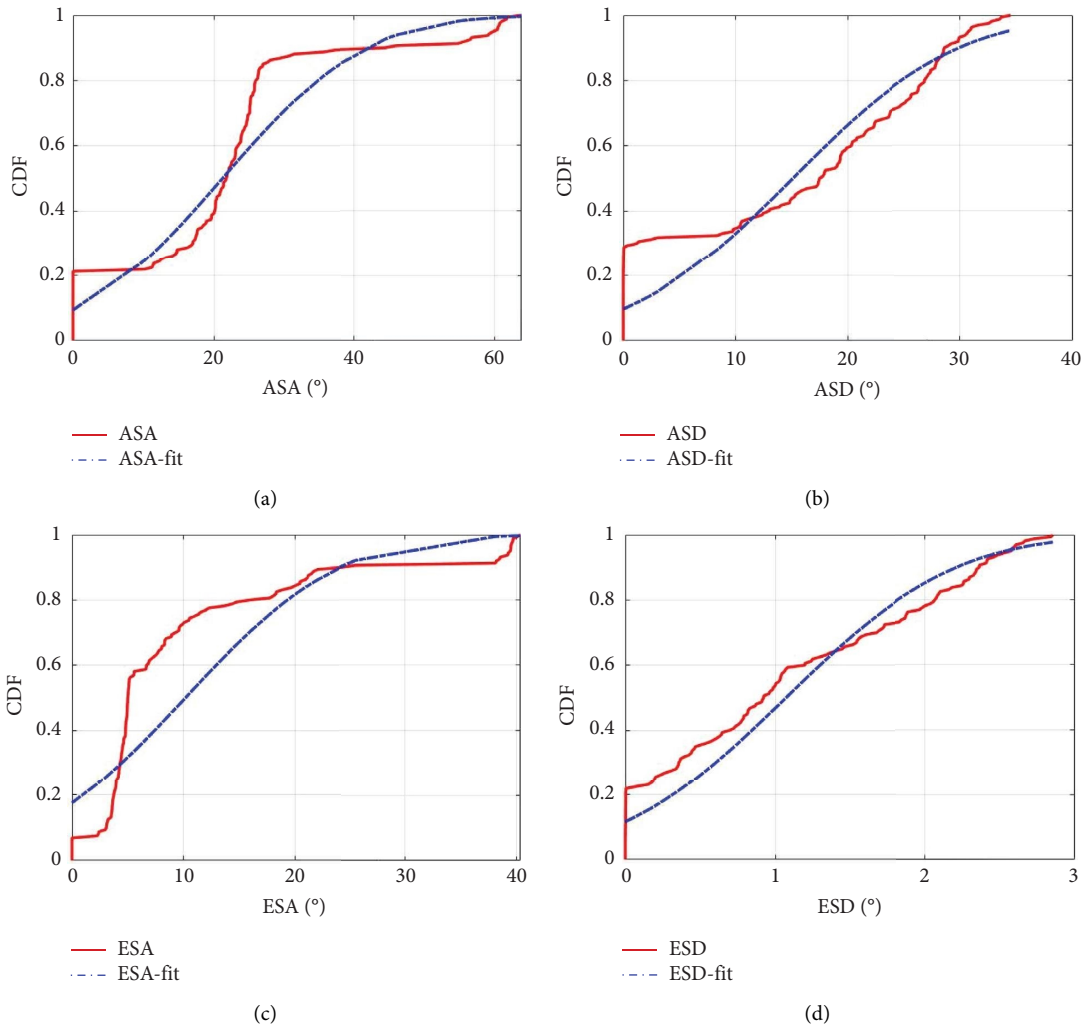


FIGURE 7: Angular spread in the LOS region. (a) ASA. (b) ASD. (c) ESA. (d) ESD.

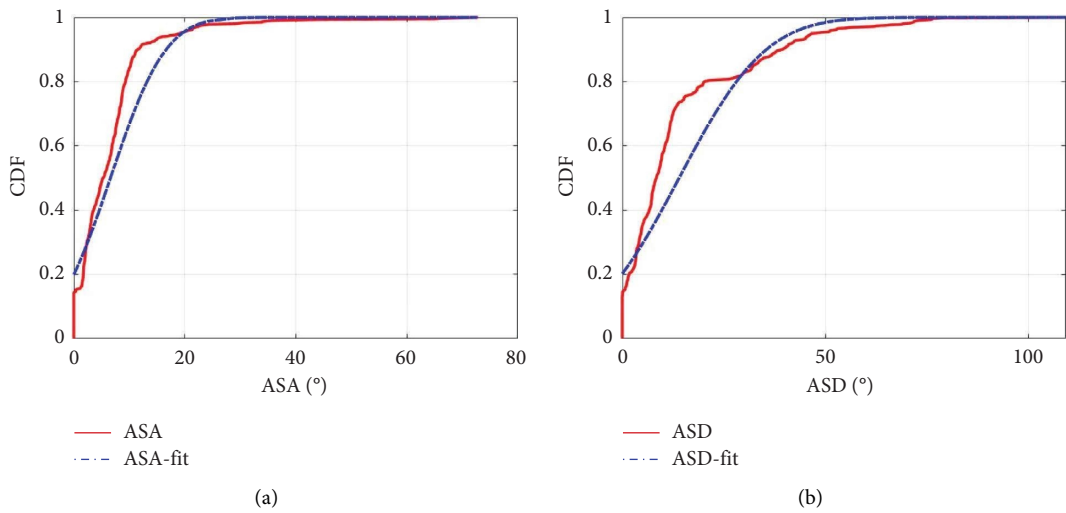


FIGURE 8: Continued.

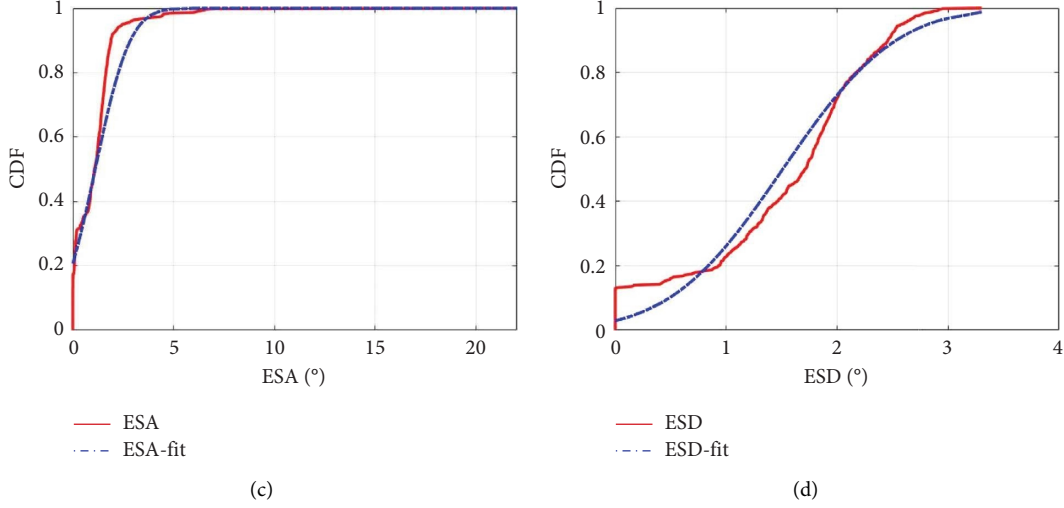


FIGURE 8: Angular spread in the NLOS region. (a) ASA. (b) ASD. (c) ESA. (d) ESD.

TABLE 4: EM properties of materials at 2.1 GHz.

	LOS	NLOS
μ_{ASA}	21.17°	6.60°
σ_{ASA}	16.06°	7.83°
μ_{ASD}	15.08°	13.99°
σ_{ASD}	11.61°	16.78°
μ_{ESA}	10.18°	1.11°
σ_{ESA}	10.89°	1.36°
μ_{ESD}	1.07°	1.51°
σ_{ESD}	0.89°	0.80°

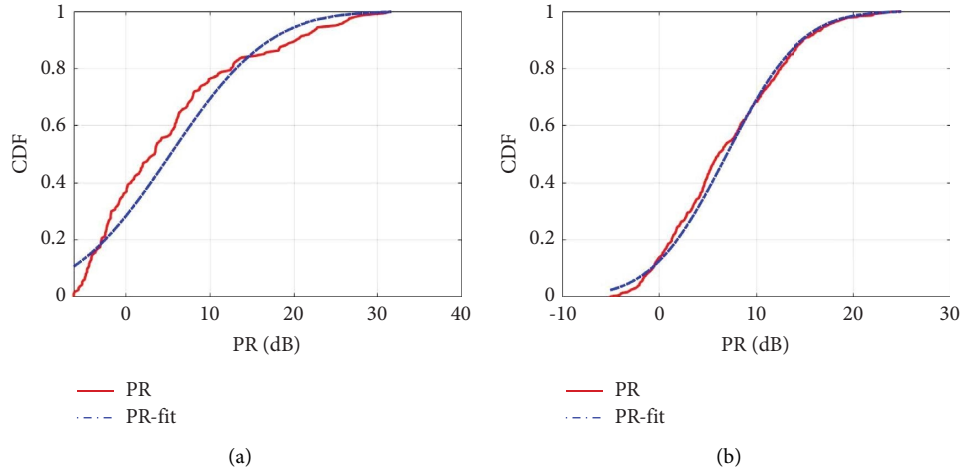


FIGURE 9: PR in the LOS and NLOS regions (a) LOS region. (b) NLOS region.

concentrated. Thus, the RIS can be deployed to enhance the coverage.

4. Model of RIS-Assisted Channel

4.1. RIS Beamforming. The geometric structure of the RIS is defined as a $M \times N$ array, as shown in Figure 11.

In the structure, the geometric center of the RIS is set to the origin of the Cartesian coordinate system. The RIS is placed on the x - y plane, with each row parallel to the x axis and each column parallel to the y axis. The location of the unit can be represented as

$$L_{m,n} = \left(\left(n - \frac{N+1}{2} \right) dx, \left(\frac{M+1}{2} - m \right) dy, 0 \right), \quad (7)$$

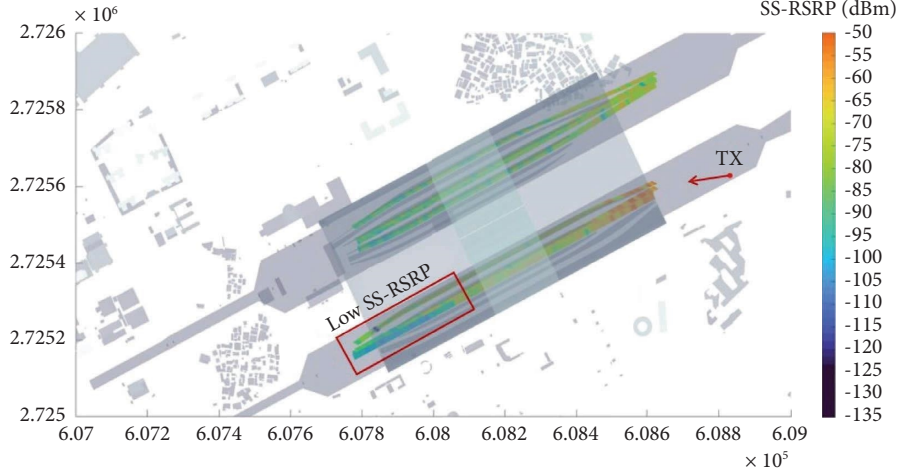


FIGURE 10: Coverage of the original channel.

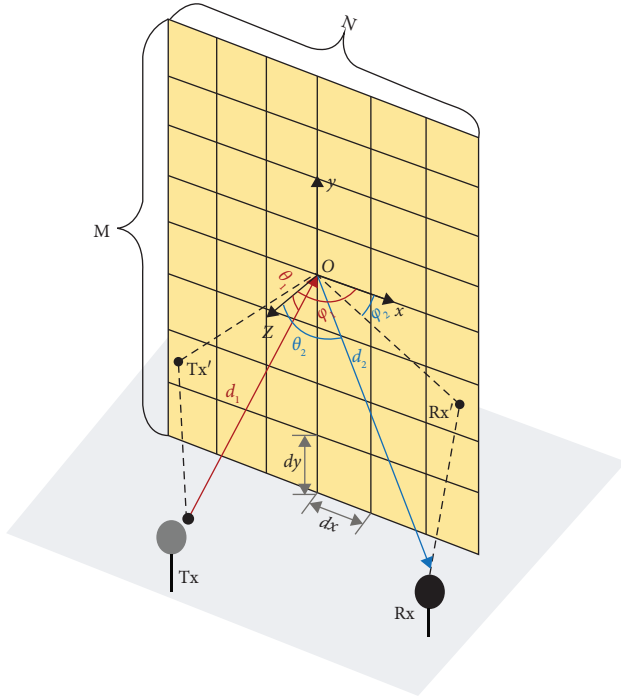


FIGURE 11: RIS schematic.

where dx and dy are the width and length of the unit, respectively.

We use the spherical coordinate system (d, θ, ϕ) to express the positions of TX and RX. (d_1, θ_1, ϕ_1) and (d_2, θ_2, ϕ_2) represent the locations of TX and RX, respectively, viewed from the geometric center of the RIS plane. θ is the zenith angle, which means the angle between the ray and the plane xOz . ϕ represents the azimuth angle, i.e., the angle between the projection of the ray on the plane xOz and the x axis.

To simulate the RIS gain in the radio channel, this paper will employ the beamforming pattern generation method for multiple-input multiple-output (MIMO) system based on the 3GPP 37.840 protocol. The detailed generation process is as follows:

$$20 \times \log_{10} P_E(\theta, \phi) = G_{E, \text{Max}} - \min(-A_{E, H}(\phi) + A_{E, V}(\theta), A_m), \quad (8)$$

$$A_{E, H}(\phi) = -\min\left[12\left(\frac{\phi}{\phi_{3\text{dB}}}\right)^2, A_m\right],$$

$$A_{E, V}(\theta) = -\min\left[12\left(\frac{\phi - 90}{\phi_{3\text{dB}}}\right)^2, \text{SLA}_v\right],$$

where $P_E(\theta, \phi)$ is the amplitude value of radiation unit pattern. $G_{E, \text{Max}}$ denotes the maximum directional gain of the radiation element. $A_m = 30$ dB is the front-to-back ratio of the beam. $A_{E, H}(\phi)$ and $A_{E, V}(\theta)$ are the horizontal and vertical radiation patterns of the radiation unit. $\phi_{3\text{dB}}$ and $\theta_{3\text{dB}}$ are the horizontal and vertical 3 dB beam width. SLA_v is the sidelobe limit.

$$v_{n,m} = \exp\left(i \cdot 2\pi \left((m-1) \cdot \frac{dy}{\lambda} \cdot \sin(\theta_{\text{tilt}}) + (n-1) \cdot \frac{dx}{\lambda} \cdot \sin(\theta_{\text{tilt}}) \cdot \sin(\phi_{\text{escan}}) \right)\right), \quad (9)$$

where $v_{n,m}$ represents the unit spacing of the $m \times n$ antennas array. λ is the wavelength. $\theta_{\text{tilt}} \in [-90^\circ, 90^\circ]$ and $\phi_{\text{escan}} \in [0^\circ, 180^\circ]$ denote the electric dip angle and the horizontal steering angle, respectively.

$$w_{n,m} = \frac{1}{\sqrt{MN}} \exp\left(i \cdot 2\pi \left((m-1) \cdot \frac{dy}{\lambda} \cdot \sin(\theta_{\text{etilt}}) + (n-1) \cdot \frac{dx}{\lambda} \cdot \sin(\theta_{\text{etilt}}) \cdot \sin(\phi_{\text{escan}}) \right)\right), \quad (10)$$

where $w_{n,m}$ represents the unit weight of the $M \times N$ antennas array. $\sqrt{MN}/1$ is the normalization coefficient.

Based on the equations mentioned above and the 3GPP 37.840 protocol, we can set up the corresponding variable parameters given in Table 5 and obtain the beam pattern as shown in Figure 12.

4.2. PSO Algorithm. According to the 2.1 GHz center frequency of the 5G-R communication, the wave length equals 0.14 m. Therefore, the size of the 16×16 RIS is 1.25 square meters. The aforementioned beam pattern generation methodology indicates that the elevation angle and azimuth angle of the pattern correspond to a combination of weight coefficients for each of element in the $M \times N$ array. Although the active RIS can adjust the weight coefficients adaptively in terms of the channel estimation outcomes from base station, relatively fixed weight coefficients for each deployment position will be more practical in industry. We thus propose to determine the beam pattern direction based on RT space domain outcomes. After that, the beam pattern direction will be treated as RIS pattern direction in the following radio coverage enhancement. Figure 13 exhibits one of surfaces which have multiple reflection and scattering phenomena in the station scenario. The surface with 1.2 square meters is on the wall of the station. Multiple reflection paths and scattering paths with different powers influence the RSLs of the corresponding RX positions on the railway lines. Taking the mentioned surface above as an example, the particle swarm optimization (PSO) algorithm will find an optimal RIS pattern direction to replace the reflection paths and scattering paths. As one of common machine learning algorithms, the PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. The optimal RIS pattern direction based on PSO outcomes will increase the RSLs of the RX positions which are affected by the reflection paths and scattering paths.

The RIS 3-dB beamwidth in azimuth plane and elevation plane equals to 15° and 12° , respectively. Therefore, we set an ellipse in the space domain to express the longitudinal section of the RIS pattern (see Figure 13).

The angular information of the RIS pattern direction is in the viewpoint of the surface deployed by the RIS. The objective function of the PSO algorithm aims to select an angle of the RIS pattern direction to obtain maximum multipaths energy contained in the ellipse, as given as follows:

$$\operatorname{argmax} \sum_{k=0}^{K_T} 10 \cdot \log_{10} [P_{\text{RIS}}(Az_k, El_k)], \quad (11)$$

TABLE 5: Parameters of beam pattern.

Parameters	Value
$M \times N$	16×16
dx, dy	$\lambda/2$
Frequency	2.1 GHz
$\phi_{3\text{dB}}$	4.2°
$\theta_{3\text{dB}}$	4.2°

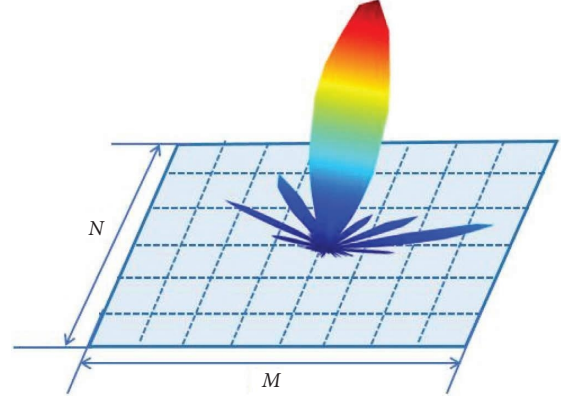


FIGURE 12: RIS beamforming.

where $P_{\text{RIS}}(Az_k, El_k)$ represents the power of the RIS pattern. K_T denotes total numbers of the scattering and reflection paths on the RIS. Az_k and El_k express the azimuth angle and elevation angle of the k -th path, respectively. Afterwards, the multipaths energy will be enhanced by the RIS gain. The angular granularities for the circular center adjustment in azimuth plane and elevation plane are infinitesimal in the mathematical aspect. Many candidate circular centers will appear. Owing to many candidates of the RIS pattern direction, the PSO can acquire the optimal RIS pattern direction in accordance with the algorithm structure diagram shown in Figure 14.

Here, the PSO algorithm exploits 100 swarms. The minimum and maximum azimuth angles are 0° and 180° , respectively. On the other hand, the minimum elevation angle equals to -90° , while the maximum elevation angle equals to 90° . The $\text{rand}(\cdot)$ is in Figure 14. w denotes stochastic variable which meets uniform distribution on $[0, 1]$. The Az_p represents the azimuth angle of the p -th particle. The El_p represents the elevation angle of the p -th particle. Moreover, we define the Az'_p and El'_p as the swarm's best known outcomes of the last iteration. The D_{ang} expresses the difference between two adjacent iteration outcomes as

$$D_{\text{ang}} = \sqrt{\left(Az'_{\text{gbest}} - Az_{\text{gbest}}\right)^2 + \left(El'_{\text{gbest}} - El_{\text{gbest}}\right)^2}. \quad (12)$$

The $THR_c = 3$ in Figure 14 represents the threshold of convergence determination. The factor A_{cc} will accumulate one if the difference of two adjacent iteration outcomes D_{ang} is lower than 0.5. Otherwise, the factor A_{cc} will turn to zero. To limit the PSO iteration, the maximum iteration $\max_{\text{iter}} = 50$ is configured. Moreover, the inertia weight is 0.8. Both of the acceleration constants, c_1 and c_2 , equal 1.5. Table 6 lists

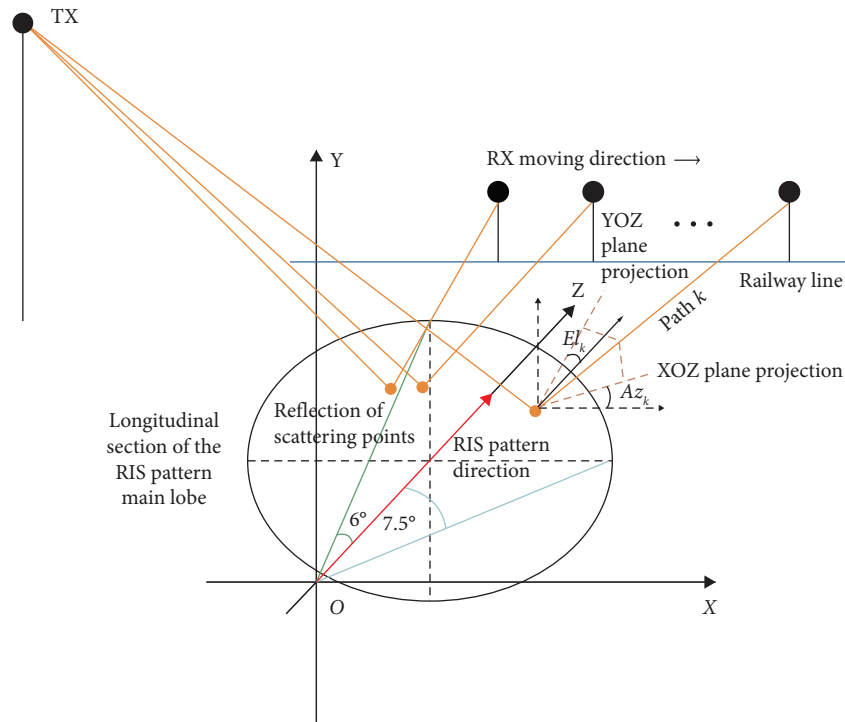


FIGURE 13: Longitudinal section of the RIS pattern.

the detailed PSO parameters configuration. On the basis of the PSO algorithm, Figure 15 shows the iteration outcomes.

Figure 15 illustrates the PSO outcomes which have completed convergence after 16 iterations. Since the station scenario contains many regions deploying the RIS, we employ the PSO algorithm to determine the RIS pattern direction based on space domain channel properties extraction. Consequently, a relatively fixed weight coefficient for each deployment position of the RIS will be obtained.

4.3. Comparison and Analysis. The following content will exhibit the results on the radio coverage enhancement for the station scenario. Figure 16 presents the SS-RSRP of the RIS-assisted channel. The result shows that the RIS will bring a good coverage improvement.

Figure 17 exhibits the path loss comparison between the RIS-assisted channel and the original channel. After deploying the RIS, the path loss has a significant decrease in the region with multipaths enhanced by RIS.

On the other hand, the paper also considers other optimization methods in practice. According to our investigation, manual antenna direction adjustment is a common method for radio coverage optimization coverage in practical engineering application. The antenna rotation direction is manually adjusted in terms of the weak coverage region of the scenario.

In the original channel, the TX directional antenna points to the center of the scenario. As shown in the Figure 10, the weak coverage points that the SS-RSRP fails to meet the -95 dBm and is mainly concentrated at the red rectangle region. Thus, the antenna will be rotated to face the

red rectangle region. Figure 18 exhibits that the antenna is rotated 12° counterclockwise and raised 1° vertically to obtain a better radio coverage. The corresponding coverage heatmap is shown in Figure 18.

In this case, there are 37 weak coverage points, which represents a 7.5% reduction in weak coverage.

Manual adjustment of the antenna direction is difficult to acquire an optimal radio coverage optimization outcome. The ergodic method of antenna direction will provide an idea to find the relatively appropriate azimuth angles and elevation angles of the antenna. On the basis of RT simulation, the ergodic method will obtain radio coverages in each preset angle. For the azimuth angle, the antenna is rotated horizontally from 60° counterclockwise to 30° clockwise, with a step of 2° . For the elevation angle, the antenna is raised from 1° to 5° vertically with the step of 1° . After that, the ergodic method calculates the coverage outcomes from 225 cases in total and obtains the optimal angle by comparing the weak coverage ratio.

Figure 19 shows the optimum angle (azimuth angle: 16° clockwise; elevation angle: 1°) based on the ergodic method. The antenna direction adjustment makes 27 weak coverage points. The weak coverage ratio is reduced by 32.5%. The optimal coverage heatmap of the ergodic method is shown in the Figure 19. Figure 19 displays that the SS-RSRP is significantly enhanced in the red rectangle regions.

Table 7 lists the number of weak coverage points of different methods. In the table, the reference method, method 1, method 2, and method 3 represent the original channel, "RIS + AI" method, manual adjustment, and ergodic method, respectively. According to the Table 7, the

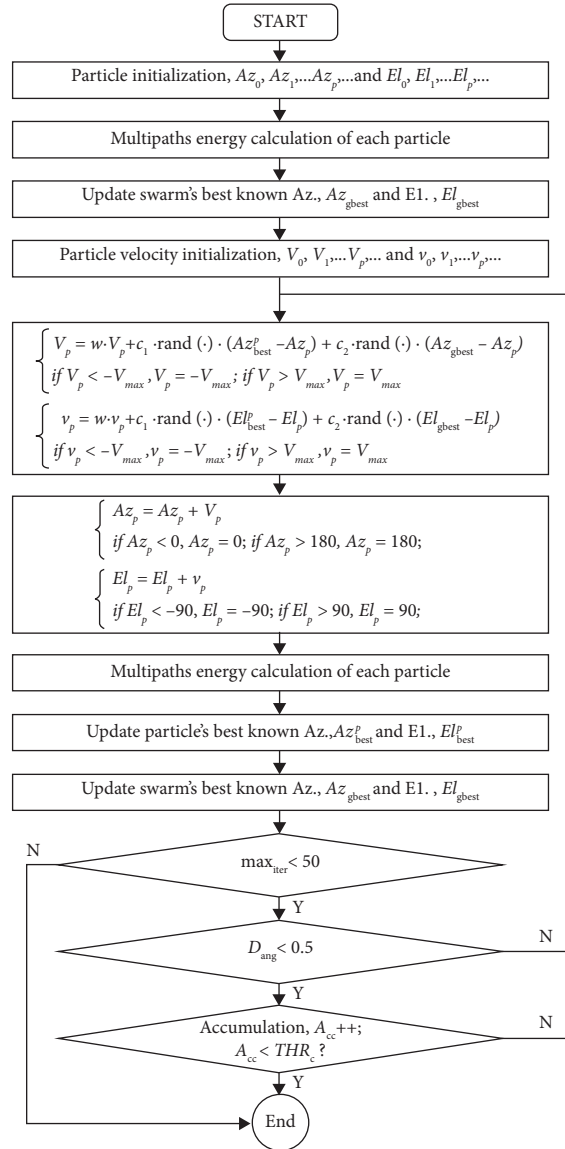


FIGURE 14: The diagram on the optimal RIS pattern direction selection based on the PSO.

TABLE 6: PSO parameters configuration.

Parameter name	Configuration data
Swarm	100
Inertia weight, w	0.8
Acceleration constant, c_1	1.5
Acceleration constant, c_2	1.5
Maximum velocity, V_{\max}	5°
Minimum velocity, V_{\min}	-5°
Convergence threshold, THR_c	3
Maximum iteration, \max_{iter}	50

“RIS + AI” method has the least amount of weak coverage points. In the station scenario, some of receiving points with weak coverage have few multipath components with low

power. RIS will enhance power of the multipaths for these regions. Consequently, the received power of the original weak coverage region will be improved.

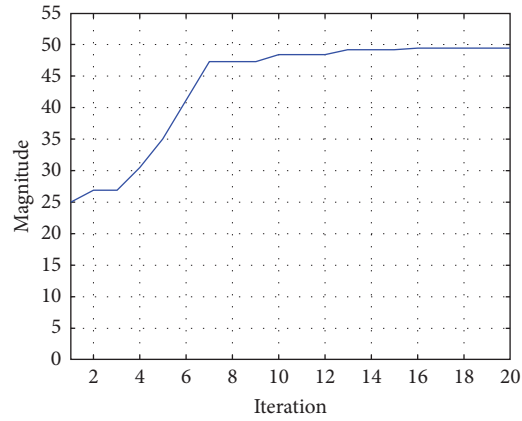


FIGURE 15: The PSO convergence process.

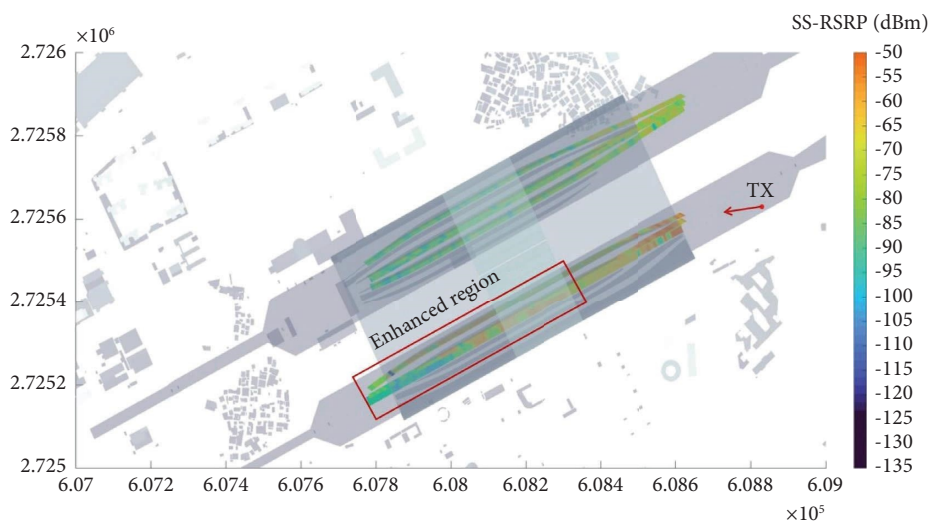


FIGURE 16: Coverage of the RIS-assisted channel.

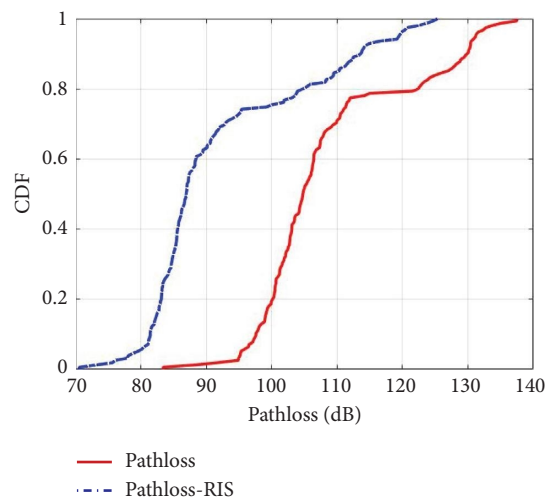


FIGURE 17: Comparison of RIS-enhanced and original path loss.

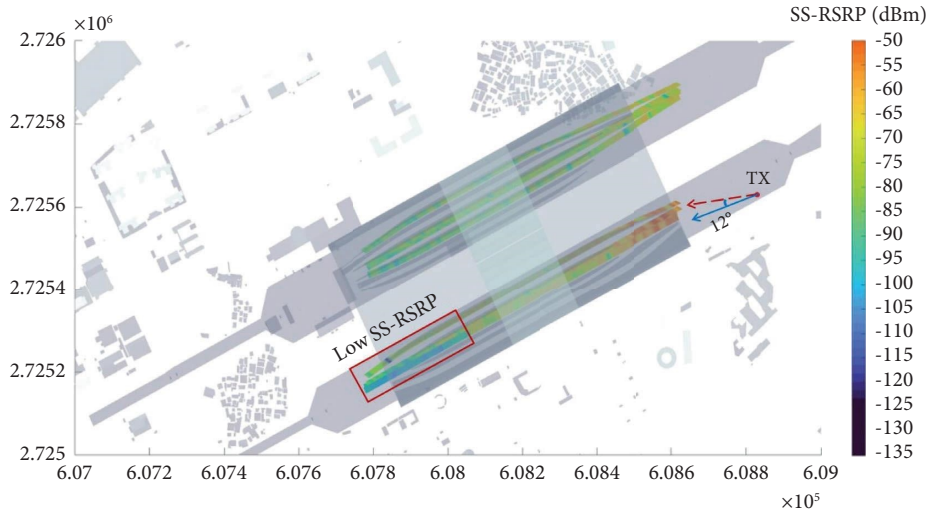


FIGURE 18: Coverage of manual adjustment channel.

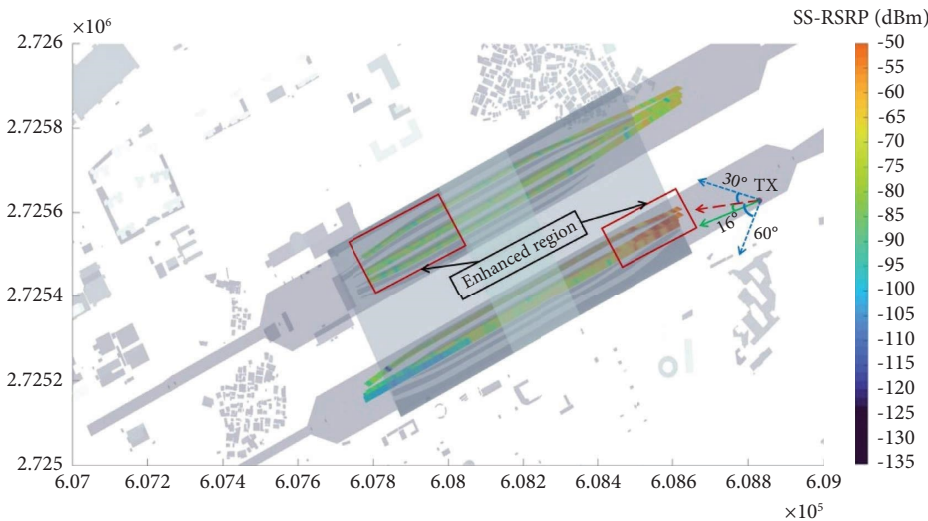


FIGURE 19: Coverage of ergodic channel.

TABLE 7: Number of weak coverage points.

Method	Reference method	Method 1	Method 2	Method 3
Number	40	25	37	27

5. Conclusion

In this paper, the HSR station channels at 5G-R band are characterized in terms of path loss, power ratio, and angular spread through RT simulation. The power of the main paths in both the LOS and NLOS regions of the station scenario is compared, which shows a weak power in the NLOS region. Due to the structural characteristic of the scenario, the angular spread of the channel in the horizontal plane is larger than that in the elevation plane. Moreover, there are 5.3% of the receiving points that fail to meet the threshold, which makes it necessary to optimize the network.

Compared to most of the previous work, the “RIS + AI” method is innovatively adopted to enhance the coverage. The channel characteristics in the space domain and PSO algorithm is combined to determine the RIS pointing. Upon comparison, the receiving point path loss of the RIS-assisted channel is improved on average by 7.51 dB. The research results of this paper will provide a well reference for the deployment of segmentation scenarios and large-scale promotion of 5G customized networks. In the future research, the fuzzy systems can be considered to describe the networks with more TXs and RISs, and the neural network algorithms can be adopted to solve the optimization.

Data Availability

The datasets generated and analyzed during this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The study was supported by the Natural Science Foundation of Hubei Province (2022CFB285) (Project: Research on characteristics of fast time-varying channel of Internet of Vehicles for B5G) and partly supported by 2022 Scientific Research Program of Hubei Provincial Department of Education (no. B2022391).

References

- [1] Z. Zhong, K. Guan, W. Chen, and B. Ai, "Challenges and perspective of new generation of railway mobile communications," *ZTE Communications*, vol. 27, no. 04, pp. 44–50, 2021.
- [2] C. Li, J. Xie, and W. Gao, "Heterogeneous network access technologies based on 5G-R services for high-speed railway," *ZTE Communications*, vol. 27, no. 04, pp. 18–23, 2021.
- [3] Y. Zhao, X. Wang, G. Wang, R. He, Y. Zou, and Z. Zhao, "Channel estimation and throughput evaluation for 5g wireless communication systems in various scenarios on high speed railways," *China Communications*, vol. 15, no. 4, pp. 86–97, 2018.
- [4] D. He, B. Ai, M. Schmieder et al., "Influence analysis of typical objects in rural railway environments at 28 ghz," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2066–2076, 2019.
- [5] Q. Tang, K. Guan, Z. Ma, D. Luo, and H. Xu, "Research on wireless signal coverage in urban tunnels based on high-performance ray tracing," *Journal of Beijing Jiaotong University*, vol. 65, no. 4, pp. 1912–1924, 2021.
- [6] D. He, B. Ai, K. Guan et al., "Influence of typical railway objects in a mmwave propagation channel," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 2880–2892, 2018.
- [7] K. Guan, B. Peng, D. He et al., "Measurement, simulation, and characterization of train-to-infrastructure inside-station channel at the terahertz band," *IEEE Transactions on Terahertz Science and Technology*, vol. 9, no. 3, pp. 291–306, 2019.
- [8] Y. Zhao and J. Guo, "An improved SBR ray-tracing channel simulation method," *Journal of Beijing Jiaotong University*, vol. 45, no. 05, pp. 1–7, 2021.
- [9] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5g and beyond wireless communications: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 10–27, 2019.
- [10] Y. Sun, K. An, Y. Zhu, C. Li, and Y. Li, "Intelligent reflecting surface assisted anti-jamming approach for wireless communications," *Chinese Journal of Radio Science*, vol. 36, no. 06, pp. 877–886, 2021.
- [11] T. Li, Y. Xu, H. Tong, and K. Pang, "Low-band information and historical data aided non-uniform millimeter wave beam selection algorithm in 5g-r high-speed railway communication scene," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2809–2823, 2022.
- [12] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.
- [13] G. Sun, R. He, B. Ai et al., "A 3d wideband channel model for ris-assisted mimo communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 8016–8029, 2022.
- [14] Y.-C. Liang, R. Long, Q. Zhang, J. Chen, H. V. Cheng, and H. Guo, "Large intelligent surface/antennas (lisa): making reflective radios smart," *Journal of Communications and Information Networks*, vol. 4, no. 2, pp. 40–50, 2019.
- [15] Y. Liu, S. Zhang, F. Gao, J. Tang, and O. A. Dobre, "Cascaded channel estimation for ris assisted mmwave mimo transmissions," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 2065–2069, 2021.
- [16] J. An, C. Xu, L. Gan, and L. Hanzo, "Low-complexity channel estimation and passive beamforming for ris-assisted mimo systems relying on discrete phase shifts," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1245–1260, 2022.
- [17] S. Zeng, H. Zhang, B. Di, Z. Han, and L. Song, "Reconfigurable intelligent surface (RIS) assisted wireless coverage extension: RIS orientation and location optimization," *IEEE Communications Letters*, vol. 25, no. 1, pp. 269–273, 2021.
- [18] M. Nemati, J. Park, and J. Choi, "Ris-assisted coverage enhancement in millimeter-wave cellular networks," *IEEE Access*, vol. 8, pp. 188171–188185, 2020.
- [19] E. Shtaiwi, H. Zhang, S. Vishwanath, M. Yousef, A. Abdelhadi, and Z. Han, "Channel estimation approach for ris assisted mimo systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 452–465, 2021.
- [20] Q. Feng, Y. Lin, M. Shan, and L. Li, "Bessel beam design with coding metasurface," *Chinese Journal of Radio Science*, vol. 36, no. 06, pp. 867–876, 2021.
- [21] A. Papazafeiropoulos, Z. Abdullah, P. Kourtessis, S. Kisseleff, and I. Krikidis, "Coverage probability of star-ris-assisted massive mimo systems with correlation and phase errors," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1738–1742, 2022.
- [22] C. Wu, Y. Liu, X. Mu, X. Gu, and O. A. Dobre, "Coverage characterization of star-ris networks: noma and oma," *IEEE Communications Letters*, vol. 25, no. 9, pp. 3036–3040, 2021.
- [23] M. Xu, S. Zhang, J. Ma, and O. A. Dobre, "Deep learning-based time-varying channel estimation for ris assisted communication," *IEEE Communications Letters*, vol. 26, no. 1, pp. 94–98, 2022.
- [24] R. Zhong, Y. Liu, X. Mu, Y. Chen, and L. Song, "Ai empowered ris-assisted noma networks: deep learning or reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 182–196, 2022.
- [25] M. Ardashir, M. H. Sabzalian, and W. Zhang, "An interval type-3 fuzzy system and a new online fractional-order learning algorithm: theory and practice," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 09, pp. 44–50, 2020.
- [26] H. Wang, K. A. Alattas, A. Mohammadzadeh, M. H. Sabzalian, A. A. Aly, and A. Mosavi, "Comprehensive review of load forecasting with emphasis on intelligent computing approaches," *Energy Reports*, vol. 8, p. 13, 2022.
- [27] Y. Wang, X. Chen, X. Liu et al., "Improvement of diversity and capacity of mimo system using scatterer array," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 1, pp. 789–794, 2022.

- [28] X. Chen, M. Zhao, H. Huang et al., "Simultaneous decoupling and decorrelation scheme of mimo arrays," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 2164–2169, 2022.
- [29] K. Guan, B. Peng, D. He et al., "Channel characterization for intra-wagon communication at 60 and 300 GHz bands," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5193–5207, 2019.
- [30] D. H. L. C. Z. Z. Xinghai Guo and K. E. Guan, "Multi-dimensional channel characteristics analysis for avionics compartment," *ZTE Communications*, vol. 27, no. 04, pp. 44–50, 2021.

Research Article

Design and Optimization in MEC-Based Intelligent Rail System by Integration of Distributed Multi-Hop Communication and Blockchain

Linlin Tian ¹, Meng Li ^{1,2}, Pengbo Si ^{1,2}, Ruizhe Yang ^{1,2}, Yang Sun ¹
and Zhuwei Wang ¹

¹Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

²Beijing Laboratory of Advanced Information Networks, Beijing 100124, China

Correspondence should be addressed to Meng Li; limeng720@bjut.edu.cn

Received 15 August 2022; Revised 17 October 2022; Accepted 24 November 2022; Published 3 February 2023

Academic Editor: Li Zhu

Copyright © 2023 Linlin Tian 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.

Mobile edge computing technology has emerged as a novel computing paradigm that makes use of resources close to the devices of the smart rail system. Nevertheless, it is difficult to support data offloading to the stations directly from different trains due to the limited coverage of the stations equipped with MEC servers. Therefore, multi-hop ad hoc network is considered and introduced in this case. In this paper, an improved architecture is proposed for the MEC-based smart rail system by blockchain and multi-hop data communication. The requesting trains can offload the tasks to MEC servers by multi-hop transmission between trains, even when requesting trains are not covered by servers. Furthermore, we utilize the blockchain technology for the authenticity and anti-falsification of information during multi-hop transmission. Then, the offloading routing path and offloading strategy are co-optimized to minimize both delay and cost of the system. The proposed majorization problem is formulated as a Markov decision process (MDP) and solved by deep reinforcement learning (DRL). In comparison to other existing schemes, simulation results demonstrate that the proposed scheme can greatly improve system performance.

1. Introduction

As the smart rail system continues to grow, it is urgent to realize the dynamic aggregation, deep mining, and effective utilization of various application data by building high-performance ubiquitous computing power. Cloud computing was employed to resolve the issue, which is because of the constrained processing power of the trains [1]. However, it is obvious that cloud computing architecture cannot meet the real-time requirements for information processing in the smart rail system, on account of the rapid mobility of trains [2, 3]. Fortunately, mobile edge computing (MEC), as an emerging technology, solves the issue mentioned above effectively. Meanwhile, the MEC technology performs computing tasks on edge servers close to the device rather than on the cloud, which meets the sensitive delay

requirements. At the same time, it brings high-quality services to users [4, 5].

However, the coverage of the stations equipped with MEC servers is limited, so it is impractical to only consider that the trains are within the range of the MEC servers. Multi-hop ad hoc network has no fixed topology. Train nodes can spontaneously create wireless network for communication between trains to exchange information and data. Each train is not only a transceiver but also a router [6]. Therefore, we consider integrating the multi-hop ad hoc network and MEC technology. The requesting trains can offload the tasks to MEC servers by multi-hop transmission between trains, which enables the servers to be utilized in a wider range while meeting the low-latency requirement.

Although the combination of multi-hop ad hoc network and MEC in the smart rail system can bring great

advantages, since a large amount of information related to traffic and driving is involved, how to effectively guarantee the security and reliability during data multi-hop transmission is worth considering. Fortunately, due to the distributed, immutable, and safety nature of blockchain, blockchain is applicable to prevent the information related to traffic and driving from being leaked or manipulated for the MEC-enabled smart rail system with multi-hop connection [7–11].

However, there are still significant obstacles to overcome before multi-hop ad hoc and blockchain can be effectively applied in the MEC-enabled smart rail system. For instance, how to properly select the routing path and the offloading decision with the high-speed movement of trains is deemed as the crucial problem. In addition, how to balance the delay and cost caused by the process of data delivery, offloading, and consensus in the MEC-enabled smart rail system also needs to be considered.

In this paper, to deal with the mentioned issues, we propose an improved optimization framework for the MEC-enabled smart rail system by multi-hop data communication and blockchain. Then, the offloading routing path, offloading strategy, and block size are co-optimized to minimize both delay and cost of the system during communication and computation process. Furthermore, by specifying the state space, action space, and reward function, a discrete Markov decision process (MDP) is formulated to characterize the dynamic jointly proposed problem. Additionally, we utilize dueling deep Q-learning network (DQN) for obtaining the optimal strategy.

The rest of this paper is structured as follows. Section 2 mainly proposes the system model. Then, we formulate the collaborative majorization problem in Section 3. In Section 4, the formulated problem is solved by dueling DQN algorithm. The experiment results are presented and discussed in Section 5. The last part summarizes the conclusion of this paper and the future directions.

2. System Model

In this section, we depict the system model, which consists of the network model, multi-hop routing path model, communication model, computation model, and blockchain model.

2.1. Network Model. In Figure 1, the architecture of a high-speed railway and a train station equipped with MEC servers which are managed by various suppliers is shown. The available computing resource and price of each MEC server are different in a real-time environment. We denote the set of these MEC servers as $R = \{1, \dots, r, \dots, R\}$. There are several high-speed trains running on the tracks. We denote $V = \{1, \dots, v, \dots, V\}$ as the set of all trains. Multi-hop ad hoc network is utilized to assist requesting trains in computation task offloading. Trains can act as relaying nodes to realize information interaction with other trains by spontaneously creating wireless network.

There may be malicious relaying nodes when offloading computation tasks by relaying. Therefore, the trust-based blockchain system is utilized to ensure the authenticity and anti-falsification of data information during the relaying and offloading process. The last-hop relaying train sends the data consensus requirement and transaction information to the blockchain system for transaction verification after receiving the relaying task. Through the consensus mechanism, the requesting train node and the other relaying train nodes in the routing path check the information data. In the blockchain system, all trains are regarded as blockchain nodes. These nodes can play either a normal or a consensus node role. Normal trains are in charge of transferring and accepting ledger information, while consensus trains are in charge of creating new blocks and carrying out the consensus process. Each relaying train in the routing path is regarded as a candidate for consensus nodes, and we consider the trust value of each relaying train when voting for consensus.

In this work, the requesting trains can maximize the use of multi-hop ad hoc network to offload tasks to the MEC servers, even if the trains are not in the communication range of servers. A consensus process is initiated when the last relaying train receives the offloading task, and the security of information is guaranteed when all consensus nodes reach a consensus successfully. For each task, there are two important elements to be considered: latency and cost. In terms of latency, considering the link quality along the whole routing path and the processing capability of servers, the total expectable latency of one successful end-to-end transfer and calculation is evaluated. In terms of cost, the total expected cost is assessed, including the data relaying cost of each relaying train in the whole path and computing cost of different MEC servers. As a result, we can select the optimal routing path and offloading decision which has the minimal latency and cost.

2.2. Multi-Hop Routing Path Model. Firstly, to determine the performance of each pair of trains in multi-hop routing path, we depict a link model, which considers channel fading and mobility of trains. Then, based on the link quality obtained above, the routing metric about link correlation is utilized to select the optimal multi-hop routing path.

2.2.1. Link Quality. We utilize the Nakagami distribution model to represent the fading of radio wave propagation [12]. Thus, the successful delivery probability between sender train v_i and receiver train v_j in spite of channel fading can be obtained by

$$p_{ij}^f(t) = 1 - F_d(r_{rt}; m, \varphi), \quad (1)$$

where $F_d(r_{rt}; m, \varphi)$ is the cumulative distribution function of the receiving signal power, r_{rt} is the reception threshold of a signal, and φ is the average signal strength. The fading parameter m is related to distance $D(v_i, v_j, t)$ between train v_i and train v_j at current time t as follows:

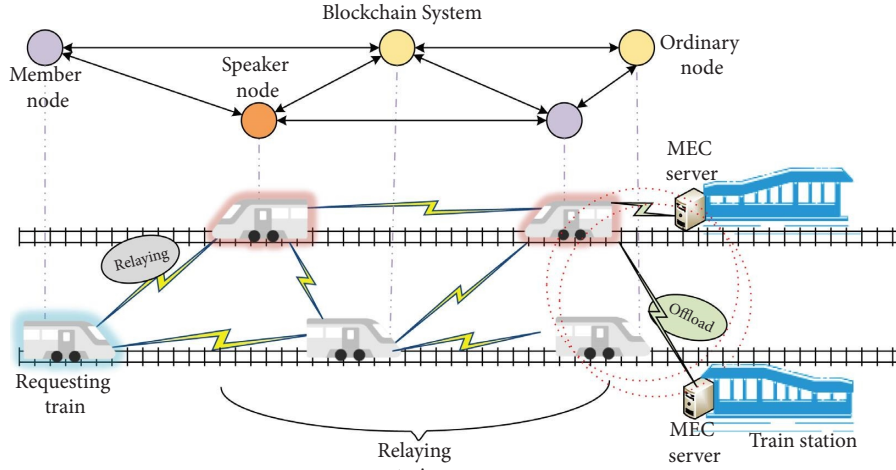


FIGURE 1: Structure of the proposed network scenario.

$$m(D(V_i, V_j, t)) = \begin{cases} 1, D(V_i, V_j, t) \geq 150m, \\ 1.5, 50m \leq D(V_i, V_j, t) < 150m, \\ 3, D(V_i, V_j, t) < 50m. \end{cases} \quad (2)$$

Under a real urban environment, the train does not always run at a constant speed, and its speed changes with acceleration or deceleration. In this case, the movement of the trains can be abstracted by a Wiener process [13]. Assuming that the trains only have two directions toward each

end, one direction is designated as positive. Therefore, the velocity variation of train v_i during interval $[t_0, t]$ can be defined as follows:

$$\Delta v_i^{t-t_0} = v_i^t - v_i^{t_0} = \mu_i(t - t_0) + \delta_i \sqrt{t - t_0}, \quad (3)$$

where $v_i^{t_0}$ and v_i^t represent the velocity of train v_i at time t_0 and t . The drift parameter μ_i denotes the acceleration or deceleration of train v_i , and the parameter δ_i follows the Gaussian distribution. Therefore, the relative distance variation of train v_i and v_j in period $[t_0, t]$ can be obtained as

$$\Delta D(V_i, V_j, t) = (v_i^{t_0} - v_j^{t_0} + \mu_i - \mu_j + \Delta v_i^{t-t_0} - \Delta v_j^{t-t_0}) * (t - t_0), \quad (4)$$

and $D(v_i, v_j, t)$ is the distance between two trains at current time t under different circumstances, which are presented as follows.

Circumstance 1: two trains are moving in one direction:

$$D(V_i, V_j, t) = \begin{cases} \Delta D(V_i, V_j, t) + d_{ij}, \text{ if } v_i > v_j, \text{ train } V_i \text{ is in front of train } V_j, \\ \Delta D(V_i, V_j, t) - d_{ij}, \text{ if } v_i > v_j, \text{ train } V_i \text{ is behind train } V_j. \end{cases} \quad (5)$$

Circumstance 2: two trains are running in the opposite direction:

$$D(V_i, V_j, t) = \begin{cases} \Delta D(V_i, V_j, t) - d_{ij}, \text{ two trains are driving towards each other,} \\ \Delta D(V_i, V_j, t) + d_{ij}, \text{ two trains are driving away from each other,} \end{cases} \quad (6)$$

where d_{ij} denotes the distance between two trains at time t_0 . Thus, we can predict link availability and obtain the probability of link availability on link L_{ij} as

$$P_{ij}^a(t) = P(D(V_i, V_j, t)) < R, \quad (7)$$

where R is the communication range of trains. Therefore, according to equations (1) and (7), we can

obtain the link quality of link L_{ij} , which represents the probability of successful transmission of the packets and is calculated by

$$p_{ij}(t) = p_{ij}^f(t) \times p_{ij}^a(t). \quad (8)$$

2.2.2. Routing Path Quality. Once data packet loss occurs on one link, this packet should be retransmitted from the source

$$Fr_N(t) = (1 - p_{s1}(t)) + 2 \cdot p_{s1}(t) \cdot (1 - p_{12}(t)) + \dots + N \cdot p_{s1}(t) \cdot p_{12}(t) \cdot \dots \cdot (1 - p_{(N-1)N}(t)), \quad (9)$$

where $p_{s1}(t)$ is the link quality between the source train and the first relaying train and $p_{(N-1)N}(t)$ represents link quality between the relaying train $N - 1$ and relaying train N in the routing path.

Therefore, we define the expected times of data transfer in an N -hop routing path as $Fc_N(t)$ when one packet is successfully transferred from source to destination and present it as follows:

$$Fc_N(t) = \frac{Fr_N(t) + N \cdot p_{s1}(t) \cdot \prod_{i=2}^N p_{(i-1)i}(t)}{p_{s1}(t) \cdot \prod_{i=2}^N p_{(i-1)i}(t)}, \quad (10)$$

where $p_{s1}(t) \cdot \prod_{i=2}^N p_{(i-1)i}(t)$ is the aggregation of the link quality of all links in an N -hop routing path.

2.3. Communication Model. In this section, we describe the communication process of the system, including the representation of the communication latency and relaying cost.

2.3.1. Communication Delay. The communication delay is mainly caused by three parts, including the requesting train offloading the task to the last-hop relaying train through the multi-hop ad hoc network, the last-hop relaying train initiating a consensus after receiving the task, and the last-hop relaying train offloading the task to MEC server.

Firstly, the requesting train offloads the task to the last-hop relaying train through the multi-hop ad hoc network. The set of relaying trains in the routing path for the task $I_{MV}(t)$ is denoted as $N = \{1, \dots, n, \dots, N\}$ (except the requesting train) and $N \in V$. The data transmission rate per hop communication between relaying train v_i and v_j is obtained as follows:

$$r_{ij}(t) = W \cdot \log_2 \left(1 + \frac{P_t}{D(V_i, V_j, t)^h \cdot P_n} \right), \quad (11)$$

where W represents the channel bandwidth, P_t is the transmit power, h indicates the path-loss exponent, and P_n is the background noise power. Thus, the delivery delay of task $I_{MV}(t)$ in V2V N -hop connections can be expressed as

to the destination, on account of the retransmission mechanism in the transport layer. As a result, there are various retransmission times and consumption of network resources for packets in different multi-hop paths. This phenomenon is referred to as a link correlation. Thus, the expected retransmission probability $Fr_N(t)$ of the N -hop path can be calculated by

$$T_{\text{tran},V2V}(t) = Fc_N(t) \times \left[\frac{B_{MV}(t)}{r_{s1}(t)} + \sum_{n=2}^N \frac{B_{MV}(t)}{r_{(n-1)n}(t)} \right], \quad (12)$$

where $B_{MV}(t)$ denotes input data size required by task $I_{MV}(t)$, $r_{s1}(t)$ is the transmission rate between the source train (i.e., requesting train) and the first relaying train, and $r_{(n-1)n}(t)$ is the transmission rate between the relaying trains v_{n-1} and v_n .

Secondly, the last-hop relaying train sends the data to the blockchain system for transaction verification after receiving the relaying task, so as to guarantee that the data are true without tampering. The delay generated by consensus process is defined as $T_{bc}(t)$, which will be described in detail in the blockchain model.

Finally, the last-hop relaying train offloads the task to the MEC server managed by different suppliers through wireless communication. The Shannon–Hartley theory is used to estimate the uplink rate for data transmission from the last-hop relaying trains to MEC server via LTE cellular network, and it can be calculated as

$$R_{r(r \in \mathcal{R})}(t) = B \cdot \log_2 \left(1 + \frac{p_{V2I}(t) \times g_{i,r}^c(i \in V)}{\Psi^2 + \sum_{j \neq i, j \in V} p_{V2I}(t) \times g_{j,r}^c} \right), \quad (13)$$

where B represents the channel bandwidth, Ψ^2 represents the background noise power, $p_{V2I}(t)$ denotes the transmission power of the train (all trains have the same transmitting power), and $g_{i,r}^c(i \in V)$ is the channel gain between the train user v_i and MEC server r .

Therefore, the transmission delay caused in this process is calculated as

$$T_{\text{tran},V2I}(t) = \frac{B_{MV}(t)}{R_r(t)}. \quad (14)$$

As mentioned above, we can obtain the total delay of communication process through

$$T_{\text{tran}}(t) = T_{\text{tran},V2V}(t) + T_{bc}(t) + T_{\text{tran},V2I}(t). \quad (15)$$

2.3.2. Communication Cost. We assume that each train has its own relaying price [14]. Corresponding to the relaying trains in the routing path for the task $I_{MV}(t)$, the relaying

price (per unit data volume) sequence is $\{Pr_1(t), Pr_2(t), \dots, Pr_n(t), \dots, Pr_N(t)\}$. Therefore, the total train relaying cost can be obtained by

$$Pr_{\text{sum}}(t) = \sum_{n=1}^N Pr_n(t) \times B_{MV}(t). \quad (16)$$

2.4. Computation Model. Assume that each MEC server operated by different suppliers has its corresponding processing capacity and price for computing offloading task. The computing capacity and price (per unit task complexity)

of MEC server are $F_{r(r \in \mathcal{R})}(t)$ and $Pc_{r(r \in \mathcal{R})}(t)$, respectively. Then, the calculation delay and cost are presented as

$$T_{\text{cal}}(t) = \frac{C_{MV}(t)}{F_r(t)}, r \in \mathcal{R}, \quad (17)$$

and

$$Pc_{\text{sum}}(t) = Pc_r(t) \times C_{MV}(t), r \in \mathcal{R}, \quad (18)$$

where $C_{MV}(t)$ denotes the required CPU cycles for task $I_{MV}(t)$.

Thus, the total delay and cost, including offloading delivery process and calculation process, are represented by

$$T_{\text{sum}}(t) = T_{\text{tran}}(t) + T_{\text{cal}}(t) = T_{\text{tran},V2V}(t) + T_{bc}(t) + T_{\text{tran},V2I}(t) + T_{\text{cal}}(t), \quad (19)$$

and

$$P_{\text{sum}}(t) = Pr_{\text{sum}}(t) + Pc_{\text{sum}}(t). \quad (20)$$

2.5. Blockchain Model. In this paper, the delegated Byzantine fault tolerance (dBFT) consensus mechanism is adopted in our blockchain system to increase the efficiency of a consensus process without tampering [15]. Moreover, each relaying train in the routing path is regarded as a candidate for consensus nodes, and we consider the trust value of candidates to determine the nodes participating in the next round of consensus, which improves the throughput of blockchain, reduces the CPU cycles of transaction confirmation, and then effectively reduces the consensus latency [16]. The higher the trust value of the relaying node is, the more likely it is to be selected as a consensus node. The set of

the selected consensus nodes is denoted by $K = \{1, 2, \dots, k, \dots, K\}$ and $K \in N$. The dBFT consensus protocol can be adopted in the proposed system model to dynamically adapt to the change of the number of train nodes [13].

2.5.1. Calculation of Trust Value. Generally, the trust value is determined by its direct trust value and indirect trust value [17, 18]. The trust value of the train node $v (v \in V)$ is defined as D_v^{trust} and $D_v^{\text{trust}} \in [0, 1]$. Similar to [19], the threshold of trust value is set as 0.5. One node is trustworthy to be a candidate for consensus only if its trust value is higher than 0.5.

We utilize subjective logic to compute direct trust value of the blockchain nodes, which can be obtained as follows:

$$D_{i(i \in V, i \neq v)}^{\text{direct}} \rightarrow v = \begin{cases} 0.5 + (NH_{i \rightarrow v} - 0.5) \times NC_{i \rightarrow v}, & \text{if } NH_{i \rightarrow v} \geq 0.5, \\ NH_{i \rightarrow v} \times NC_{i \rightarrow v}, & \text{otherwise,} \end{cases} \quad (21)$$

where $NH_{i \rightarrow v}(t)$ is the node honesty (NH) and represents the uncertainty during offloading due to unstable and noisy communication channels between the relaying trains [18]. $NC_{i \rightarrow v}(t)$ is the remaining node capacity (NC) of the trains to complete task.

For the computation of indirect trust value, the number of times one node has been voted for consensus in the past is taken into account. The blockchain system regularly updates and records the selection of consensus nodes. Thus, the indirect trust value of one blockchain node can be defined as

$$D_v^{\text{indirect}} = \frac{VN_v}{VN_{\text{all}}}, \quad (22)$$

where VN_{all} is the total number of consensus processes and VN_v is the number of times the relaying train v has been voted for consensus.

Therefore, the trust value of one candidate for consensus is represented by

$$D_v^{\text{trust}} = w_{\text{direct}} \cdot \frac{\sum_{i=1}^{v-1} D_{i \rightarrow v}^{\text{direct}} + \sum_{i=v+1}^V D_{i \rightarrow v}^{\text{direct}}}{V} + w_{\text{indirect}} \cdot D_v^{\text{indirect}}, \quad (23)$$

where w_{direct} and w_{indirect} represent the weight of direct and indirect trust values, respectively. Meanwhile, w_{direct} and $w_{\text{indirect}} \in [0, 1]$ and $w_{\text{direct}} + w_{\text{indirect}} = 1$.

2.5.2. Consensus Process. The specific dBFT consensus process is depicted in Figure 2. We assume that generating/validating one signature and message authentication code (MAC) requires α and β CPU cycles, respectively.

At first, the last-hop relaying train sends consensus requirement and transaction information to blockchain system upon receiving the offloading task. Then, the speaker of the consensus process in this round is assigned by blockchain. The assigned speaker packages the hash of the transactions as a prepare request message to launch a proposal and broadcast it to initiate a new consensus. During this phase, one signature and $K - 1$ MACs are generated by the speaker. Thus, the computation cycles for the speaker node in this process are represented as

$$c_1(t) = \alpha + (K - 1) \cdot \beta. \quad (24)$$

Secondly, each member collects all the transactions information of the prepare request message. If the transactions are verified successfully, the members add the transactions to the consensus module and broadcast the prepare response messages to all consensus nodes. During this phase, members need to validate the signatures and MACs of the proposal and contained transactions and then generate one signature and $K - 1$ MACs for forming prepare response messages. Therefore, the computation cycles for the member nodes are calculated by

$$c_2(t) = \alpha + \beta + \frac{S_{bc}(t)}{L} \cdot (\alpha + \beta) + \alpha + (K - 1) \cdot \beta, \quad (25)$$

where $S_{bc}(t)$ is the total transaction batch size at time slot t and L represents the average size of transactions.

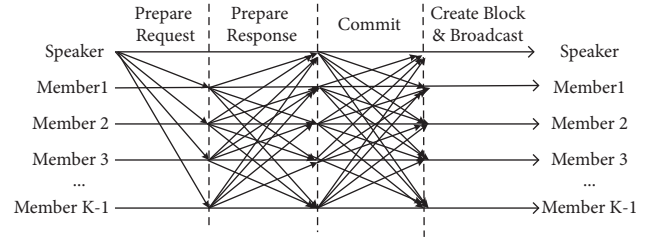


FIGURE 2: Consensus process of dBFT.

Then, if at least $K - f$ prepare response messages are received before the timeout, each consensus node verifies whether the messages are correct at first. Once the verification is successful, the commit messages are broadcast to other consensus nodes. During this phase, the consensus nodes verify $K - f$ signatures and MACs and then generate one signature and $K - 1$ MACs for forming commit messages. Thus, for each consensus node, the consumed CPU cycles can be represented as

$$c_3(t) = (K - f) \cdot (\alpha + \beta) + \alpha + (K - 1) \cdot \beta. \quad (26)$$

Finally, if the consensus nodes have collected more than $K - f$ commit messages and verified successfully, the consensus process is regarded as completed. At the same time, one block is produced and broadcast to blockchain system. During this phase, $K - f$ signatures and MACs should be verified by one consensus node. Thus, the computation cycles for each consensus node in this process are represented as

$$c_4(t) = (K - f) \cdot (\alpha + \beta). \quad (27)$$

In terms of above analysis, the delay of consensus process is represented by

$$T_{bc}(t) = \underbrace{\frac{c_1(t)}{f_{sp}(t)}}_{\text{Prepare Request}} + \underbrace{\max_{k \in K(k \neq sp)} \frac{c_2(t)}{f_k(t)}}_{\text{Prepare Request}} + \underbrace{\max_{k \in K} \frac{c_3(t)}{f_k(t)}}_{\text{Commit}} + \underbrace{\max_{k \in K} \frac{c_4(t)}{f_k(t)}}_{\text{CreateBlock\&Broadcast}} + T_i(t) + 3T_b(t), \quad (28)$$

where $f_{sp}(t)$ denotes the computing capacity of the speaker, $f_k(t)$ represents the computing capacity of the consensus node k , $T_i(t)$ is the block generation interval, and $T_b(t)$ is the broadcast delay between nodes.

3. Problem Formulation

In this section, it is necessary to jointly optimize routing path selection, offloading decision, and block size selection in a real-time environment so as to decrease delay and cost of the proposed network. The majorization issue is characterized as

a MDP by identifying the state space, action space, and reward function.

3.1. State Space. During each time slot t , we define the state space as a union of the link quality between each pair of all trains $P_L(t) = \{P_{ij}(t)\}_{i, j = 1, 2, \dots, V}$, relaying price of all trains $\text{Pr}(t) = \{\text{Pr}_1(t), \text{Pr}_2(t), \dots, \text{Pr}_V(t)\}$, computing resource of MEC servers $F(t) = \{F_1(t), F_2(t), \dots, F_R(t)\}$, and computing price of MEC servers $Pc(t) = \{Pc_1(t), Pc_2(t), \dots, Pc_R(t)\}$, which is represented as

$$S(t) = \{P_L(t), Pr(t), F(t), Pc(t)\}. \quad (29)$$

3.2. *Action Space.* The action space involves the routing path selection, the offloading decision, and block size selection. Formally, the action space $A(t)$ is denoted as

$$A(t) = \{N(t), a_o(t), S_{bc}(t)\}, \quad (30)$$

where (t) is the set of relaying trains arranged in consecutive routing order in the multi-hop routing path. $a_o(t) = \{0, 1, \dots, R-1\}$ indicates the offloading decision,

and $a_0(t) = 0$ represents that the task is executed on the MEC server managed by the first supplier, while $a_0(t) = R-1$ indicates that the task is executed on the MEC server managed by the R -th supplier. $S_{bc}(t) \in \{1, 2, \dots, S_{\max}\}$ represents different level for block size and S_{\max} is the maximum block size.

3.3. *Reward Function.* We define the reward function to improve system performance and then devise immediate reward as

$$r(t) = \begin{cases} w_1 \cdot \frac{1}{T_{sum}(t)} + w_2 \cdot \frac{1}{P_{sum}(t)}, & \text{if C1 - C3 are satisfied,} \\ w_1 \cdot \frac{1}{T_{sum}(t)} + w_2 \cdot \frac{1}{P_{sum}(t)} - \vartheta, & \text{otherwise,} \end{cases} \quad (31)$$

$$s.t. \text{ C1: } T_{sum}(t) \leq \tau,$$

$$\text{C2: } T_{bc}(t) \leq \varepsilon \times T_i(t),$$

$$\text{C3: } S_{bc}(t) \leq S_{\max},$$

where w_1 and $w_2 \in [0, 1]$ are the weights of the latency and the cost, respectively, and $w_1 + w_2 = 1$. ϑ is the penalty value.

In this problem, C1 indicates the time limitation for completely offloading tasks, where τ is the maximum tolerable delay. C2 denotes the latency limitation of completing a block, where $\varepsilon > 1$. The maximum size of the all transactions in a single consensus process is indicated by C3.

4. Problem Solution

In this paper, due to high dynamic characteristics of the proposed system, we adopt the dueling DQN algorithm to solve the proposed joint optimization issue. Dueling DQN is widely considered as a significant improvement to conventional DQN. Different from the natural DQN, dueling DQN divides the Q -network into two parts, action advantage function with independent of state $A(s_t, a_t; \omega, \xi)$ and state-value function $V(s_t; \omega, \theta)$, which are calculated

separately [20, 21]. It is easy to find which action has better feedback by learning $A(s_t, a_t; \omega, \xi)$. Finally, we can obtain the output of the dueling DQN network by merging two fully connected layers, which is denoted as

$$Q^\pi(s_t, a_t; \omega, \theta, \xi) = V^\pi(s_t; \omega, \theta) + A^\pi(s_t, a_t; \omega, \xi), \quad (32)$$

where ω is the convolution layer parameter, θ represents the parameter of the specific connected layer of the state-value function, and ξ denotes the parameter of the specific connected layer of the action advantage function. However, there is an unidentifiable problem in equation (32), which means that the respective roles of state-value function and action advantage function in the final Q value cannot be identified. To address that problem, dueling DQN sets expected value of the action advantage function to be zero at the selected action and implements the forward mapping of the last module of the network, which is written as

$$Q^\pi(s_t, a_t; \omega, \theta, \xi) = V^\pi(s_t; \omega, \theta) + A^\pi(s_t, a_t; \omega, \xi) - \frac{1}{|A|} \times \sum_{a'} A^\pi(s_t, a'_t; \omega, \xi). \quad (33)$$

The separation of environmental state value and action advantage in dueling DQN solves the problem of repeated calculation of the same state value, enhancing the capability of estimating the environmental state with a clear optimization objective [22]. Therefore, we adopt dueling DQN in our

proposed network to decrease computational complexity and training time.

Finally, the training process is formally described in Algorithm 1.


```

(1) Initialization:
    Initialize the experience memory  $D$  and the mini-batch size  $B$ ;
    Initialize evaluated network with the weight and bias set  $\omega$ ;
    Initialize target network with the weight and bias set  $\omega^-$ ;
    Initialize the greedy coefficient  $\epsilon$ ;
(2) for  $I = 1, \dots, I_{\max}$  do
(3)   Reset the state of trains and MEC servers with a random initial observation  $s_{ini}$ , and  $s(t) = s_{ini}$ ;
(4)   for  $H = 1, \dots, H_{\max}$  do
(5)     Randomly choose a probability  $p$ ;
(6)     if  $p < \epsilon$  then
(7)       Randomly choose an action  $a(t) \neq a^*(t)$  based on  $\epsilon$ -greedy policy;
(8)     else
(9)        $a(t) = a^*(t) = \operatorname{argmax}_{a \in A} Q(s', a'; \omega', \theta', \xi)$ ;
(10)    end if
(11)    Execute action  $a(t)$  and obtain the reward  $r(t)$ , and proceed to the next observation  $s(t+1)$ ;
(12)    Store the experience  $(s(t), a(t), r(t), s(t+1))$  into experience replay memory;
(13)    Randomly sample a mini-batch of  $(s(i), a(i), r(i), s(i+1))$  from experience replay memory  $D$ ;
(14)    Obtain two parts of evaluated network, including  $V(s(t))$  and  $A(a(t))$ , and merge them as  $Q(s(t), a(t); \omega, \theta, \xi)$  through
    equation (33);
(15)    Obtain target Q value in target network by  $Q_{\text{target}}(s) = re(t) + \gamma \max_{a \in A} Q(s', a'; \omega', \theta', \xi)$ ;
(16)    Train evaluated network to minimize loss function  $L(\omega)$  by  $L(\omega) = E[(Q_{\text{target}}(s) - Q(s, a'; \omega', \theta', \xi))^2]$ ;
(17)    Every several training steps, modify target Q-network according to evaluated Q-network;
(18)     $s(t) \rightarrow s(t+1)$ ;
(19)  end for
(20) end for

```

ALGORITHM 1: Performance optimization framework for MEC-enabled smart rail system by multi-hop data transmission and blockchain based on dueling DQN.

5. Simulation Results and Discussion

In this part, we depict the effectiveness of the proposed scheme through simulation experiments. Firstly, the simulation environment and parameters are presented. Then, we analyze and discuss the results and the performance of the proposed framework.

5.1. Simulation Parameters. In the simulation experiments, we consider the network scenario with five trains running on the track, as well as two MEC servers managed by different suppliers. Furthermore, we summarize other significant simulation parameters in Table 1.

In order to assess how well the proposed framework performs, we consider five comparison schemes as follows. (1) The routing path is picked at random in the proposed method without path selection. (2) An approach without offloading selection: the MEC servers conduct the computing tasks at random. (3) Block sizes for created blocks are fixed in the proposed approach. (4) A technique based on natural DQN solves the problem as it is formulated. (5) PBFT-based scheme: all blockchain nodes participate in the consensus process.

5.2. Performance Comparison of Convergence. The convergence of the proposed optimization framework under various learning rates is shown in Figure 3. As can be seen in this figure, the learning rate (10^{-1}) performs better than other schemes. It is because the large learning rate (10) might fall into local

optimum and fail to obtain the globally optimal solution of the proposed problem. Moreover, the small learning rate (10^{-7}) likely led to the slow convergence rate and took longer to find the optimal value. Hence, in this paper, the learning rate is selected carefully and set to be 10^{-1} .

As shown in Figure 4, we examine the convergence performance under the different algorithms. It can be observed that dueling DQN reaches higher system reward and performs more stably than the scheme with natural DQN. The reason is that the chosen routing path, the selected MEC server, and the selected block size can hardly affect the changes of state in our scenario. Thus, our proposed scheme with dueling DQN has more advantage to the decision of the agent in this case.

Figure 5 depicts the comparison of system reward with training steps under our proposed dBFT-based scheme and PBFT-based scheme. We can see that dBFT-based scheme gets higher total reward. It is because all nodes need to participate in the consensus process under the PBFT-based scheme. On the contrary, there is only a part of the trusted nodes participating for consensus under the dBFT-based scheme. The dBFT algorithm reduces the computation cycles and improves the efficiency of the consensus process. Therefore, our proposed dBFT-based scheme is more suitable for the smart rail system with high-speed mobility.

5.3. Performance Comparison of Different Aspects. Figure 6 presents the relationship between total latency and task data size under different schemes. One observation is

TABLE 1: Experiment parameters.

Parameters	Value
The number of nodes V	5
CPU cycles required to generate and verify a signature α	2 Mcycles [23]
CPU cycles required to generate and verify a MAC β	1 Mcycles [23]
Average transaction size L	20 kB
Average computing resources of MEC servers $F_R(t)$	$4 \times 10^{10} - 6 \times 10^{10}$ CPU cycles/s [2]
Size of offloading task $B_{MV}(t)$	500 kB
Computation density of application	200 CPU cycles/bit [2]
Unit price for data relaying $Pr(t)$	500-1000
Unit price for data computation $Pc(t)$	2000-3000
Weights of latency w_1	0.6
Weights of cost w_2	0.4
Learning rate	10^{-1}



FIGURE 3: Total reward with various learning rates.

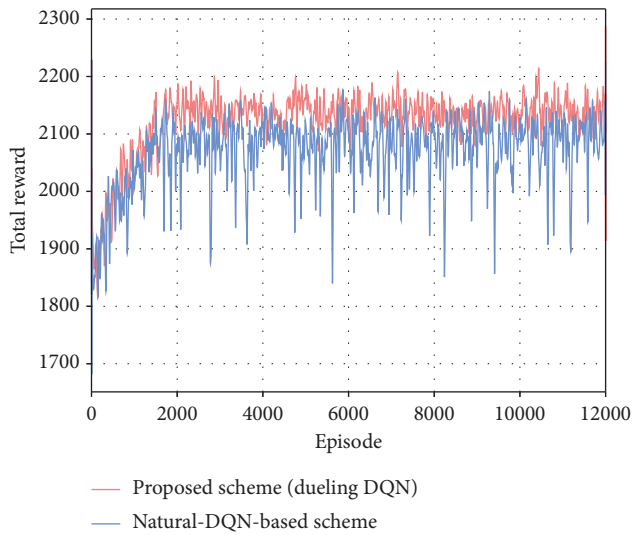


FIGURE 4: Total reward with diverse learning algorithms.

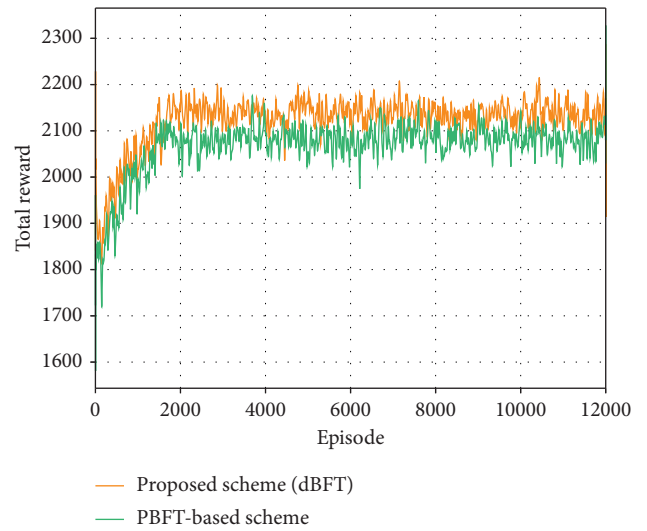


FIGURE 5: Total reward under different consensus mechanisms.

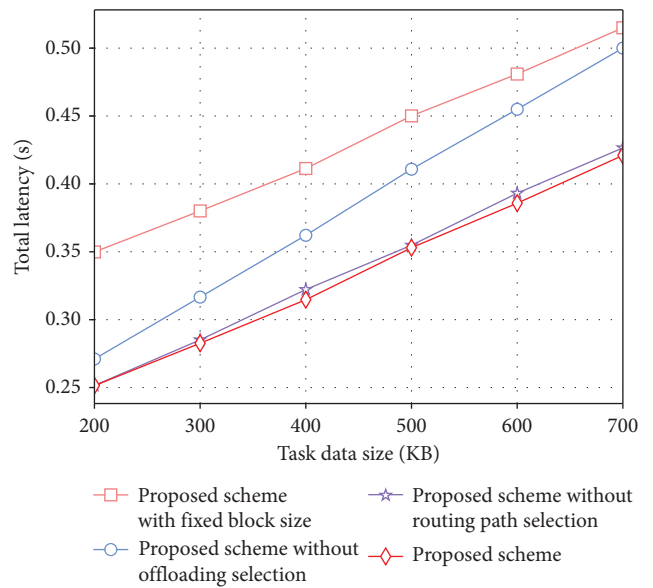


FIGURE 6: Total latency versus task data size under various schemes.

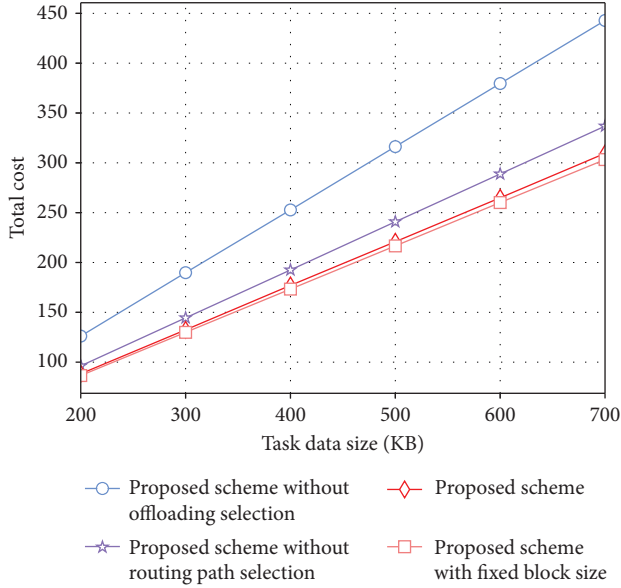


FIGURE 7: Total cost versus task data size under various schemes.

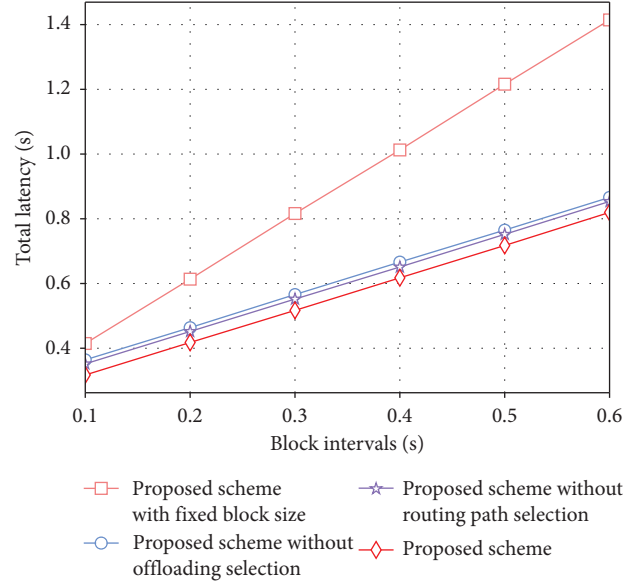


FIGURE 9: Total latency versus block intervals under various schemes.

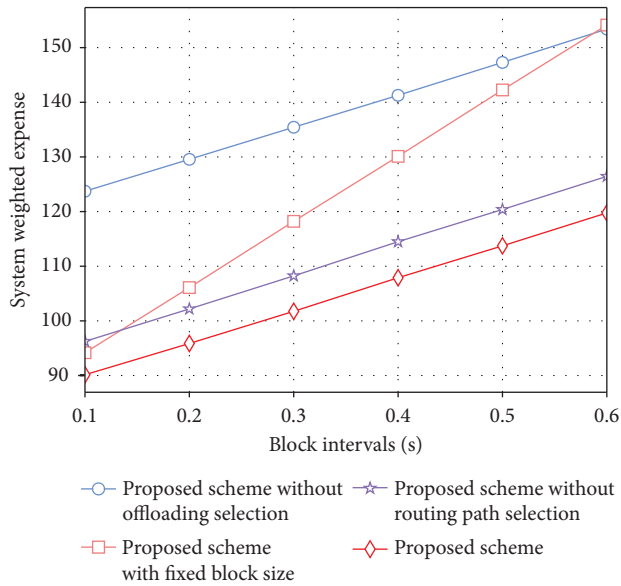


FIGURE 8: System weighted expense versus block intervals under various schemes.

that total latency under all schemes increases with the increase of task data size. The reason is that the increase in task size takes longer for end-to-end delivery and offloading computation. Moreover, we can see from this figure that the total latency under our proposed scheme is consistently lower than that of others. It is because our proposed scheme simultaneously optimizes the routing path selection, offloading strategy, and block size selection. On the contrary, previous baselines just optimize a portion of these items.

Figure 7 illustrates the comparison of total cost with the size of the task data under various schemes. We can see that as the task data size increases, so does total cost of all schemes. Furthermore, our proposed scheme is superior to

the schemes without routing path selection and offloading selection. Nevertheless, the scheme without fixed block size performs better than the proposed scheme. The reason is that the state of link quality fluctuates according to the high-speed movement of trains.

As shown in Figures 8 and 9, we examine the system weighted expense and the total latency under different block intervals. It can be observed that all the schemes gain a higher system weighted expense and total latency with the increase of the block intervals. The overall system latency and cost make up the system weighted expense. It is because the block generation interval rises, which makes the delay of blockchain higher. Additionally, system weighted expense and total latency of the previous baselines are visibly higher than those of our proposed scheme. Therefore, with joint consideration of the adaptive routing path selection, the optimal offloading decision, and the appropriate block size selection, our proposed scheme acts the best compared with other schemes.

6. Conclusions

In this paper, an improved optimization framework for the MEC-enabled smart rail system is proposed. In order to enable the MEC servers to be utilized in a wider range while meeting the low-latency requirement, multi-hop ad hoc network was applied to our proposed network model. Moreover, the blockchain technology based on the dBFT consensus mechanism was considered and introduced to effectively guarantee the security and reliability during multi-hop data transmission. Then, in order to reduce system latency and cost, the routing path selection, offloading strategy, and block size selection are co-optimized. We described the proposed dynamic majorization problem as a MDP and adopted dueling DQN to solve it. Simulation results demonstrated that the performance of the proposed

scheme is better than existing baseline schemes. Furthermore, in the future, other routing mechanisms and cloud-edge collaborative architecture would be considered in our multi-hop ad hoc network for the smart rail system.

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 in part by the National Natural Science Foundation of China under grant no. 61901011, in part by the Beijing Natural Science Foundation under grant nos. L211002, 4222002, and L202016, and in part by the Foundation of Beijing Municipal Commission of Education under grant nos. KM202110005021 and KM202010005017.

References

- [1] H. Tu, "Research on the application of cloud computing technology in urban rail transit," in *Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 828–831, Dalian, China, August 2020.
- [2] Y. X. Liu, *Research on End-Edge-Cloud Collaborative Computing for Intelligent Railway*, Beijing Jiaotong University, Beijing, China, 2021.
- [3] K. Zhang, Y. M. Mao, S. P. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.
- [4] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.
- [5] H. C. Zhang, *Research on Computing Offloading Method Based on Mobile Edge Computing in Internet of Vehicles*, Zhejiang University of Science and Technology, Hangzhou, Zhejiang, China, 2020.
- [6] J. Balen, G. Martinovic, K. Paridel, and Y. Berbers, "PVCN: assisting multi-hop communication in vehicular networks using parked trains," in *Proceedings of the 2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 119–122, St. Petersburg, Russia, October 2012.
- [7] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain-empowered CBTC system," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.
- [8] J. F. Xie, H. Tang, T. Huang et al., "A survey of blockchain technology applied to smart cities: research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2794–2830, 2019.
- [9] J. Kang, R. Yu, X. Huang et al., "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.
- [10] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2019.
- [11] L. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, "Cross-layer defense methods for jamming-resistant CBTC systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2021.
- [12] X. M. Zhang, K. H. Chen, X. L. Cao, and D. K. Sung, "A street-centric routing protocol based on microtopology in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5680–5694, 2016.
- [13] X. M. Zhang, X. L. Cao, L. Yan, and D. K. Sung, "A street-centric opportunistic routing protocol based on link correlation for urban VANETs," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1586–1599, 2016.
- [14] Z. Z. Deng, Z. Cai, and M. G. Liang, "A multi-hop VANETs-assisted offloading strategy in vehicular mobile edge computing," *IEEE Access*, vol. 8, pp. 53062–53071, 2020.
- [15] A. Boualouache, H. Sedjelmaci, and T. Engel, "Consortium blockchain for cooperative location privacy preservation in 5G-enabled vehicular fog computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7087–7102, 2021.
- [16] Z. H. Xue, *Improvement and Research of Blockchain Consensus Algorithm Based on Byzantine Fault Tolerant*, Lanzhou Jiaotong University, Lanzhou, Gansu, China, 2021.
- [17] G. J. Han, J. F. Jiang, L. Shu, and M. Guizani, "An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic sensor network," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2447–2459, 2015.
- [18] J. W. Kang, Z. H. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.
- [19] J. Feng, Q. Q. Pei, F. R. Yu, X. Chu, and B. Shang, "Computation offloading and resource allocation for wireless powered mobile edge computing with latency constraint," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1320–1323, 2019.
- [20] Z. H. Xuan, G. Y. Wei, and Z. W. Ni, "Power allocation in multi-agent networks via dueling DQN approach," in *Proceedings of the 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, pp. 959–963, Nanjing, China, October 2021.
- [21] W. L. Jiang, C. Bao, G. Q. Xu, and Y. Wang, "Research on Autonomous Obstacle Avoidance and Target Tracking of UAV Based on Improved Dueling DQN Algorithm," in *Proceedings of the 2021 China Automation Congress (CAC)*, pp. 5110–5115, Beijing, China, October 2021.
- [22] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial internet of things: a dueling deep Q-learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627–4639, 2019.
- [23] F. X. Guo, F. R. Yu, H. L. Zhang, H. Ji, M. Liu, and V. C. M. Leung, "Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1689–1703, 2020.

Research Article

Optimization of Metro Trains Operation Plans Based on Passenger Flow Data Analysis

Jun Yang ^{1,2}, Yinghao Tang ¹, Tan Ye ¹, Xiao Han ¹ and Mengjie Gong ¹

¹Big Data and Internet of Things Research Center, China University of Mining and Technology-Beijing, Beijing 100083, China

²Key Laboratory of Intelligent Mining and Robotics, Ministry of Emergency Management, Beijing, China

Correspondence should be addressed to Yinghao Tang; mytangyh@163.com

Received 19 June 2022; Accepted 6 September 2022; Published 20 September 2022

Academic Editor: Li Zhu

Copyright © 2022 Jun Yang 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.

Metro intelligent system produces massive passenger flow and traffic data every day, among which route, station, and operation data are important for optimizing the train operation scheme. We collect passenger flow information of Shenzhen metro, analyze the passenger flow pattern and its distribution characteristics based on the data warehouse of the Hadoop platform, and optimize the train operation scheme in this paper. Using dynamic passenger flow data, an optimization model with train departure and dwell time as decision variables and passenger waiting time, passenger ride time, train full load ratio, and train operation balance as objectives is developed. An improved parallel genetic algorithm (GA) incorporating a simulated annealing algorithm (SAA) and an optimal individual retention strategy is used to find the optimal result. To verify the usefulness of the method, simulation experiments are conducted on the optimization model and method using the real passenger flow and train operation data of Shenzhen metro, and the simulation results are compared with the original plan.

1. Introduction

The metro system is characterized by large capacity, fast speed, high frequency, and punctuality. It has become one of the best schemes to alleviate urban traffic congestion [1]. Metro system produces a large number of passenger flow data [2] such as passenger origin-destination (OD) information and train operation data. Using big data to analyze passenger flow data can improve rail transit train transportation efficiency [3] and passenger satisfaction.

The intelligent construction of the metro is an important means to relieve the pressure of urban traffic, and train schedule optimization is one of the important ones [4]. In the metro system, passenger origin-destination (OD) information is very important. It can be used for the optimization of the metro train operation plan. The train operation plans are developed from historical traffic data. It determines the train's departure time at each station, its dwelling time at the station, and its arrival time at the station. It needs to meet some operational constraints such as train

full load factor and travel time. Through the analysis of OD data and passenger flow data, we can optimize the train operation scheme to improve passenger satisfaction [5] and reduce the operation cost of the metro.

Lots of research have been performed on metro schedule optimization by many scholars. In terms of optimization models and optimization objectives. Wang et al. [6] proposed a mixed integer programming model based on time-varying demand, which minimizes the passenger waiting time and the number of passengers unable to transfer, using train capacity as a constraint. Zhang et al. [7] developed two nonlinear nonconvex programming models considering the variation of train frequency, train running time, and stopping time, and under the constraints of train operation and passengers getting on and getting off process, the train timetable with the minimum full passenger travel time is designed. Qu et al. [8] proposed a two-step optimization model to change the metro schedule, in which the train departure interval is used as a decision term to reduce the waiting time of people in the first-step model. In the second

step model, the total energy consumption of all trains is minimized by taking the train leave and arrival times at various stations as the decision terms. Wu et al. [9] proposed a multi-objective train schedule optimization method with the objectives of minimizing total energy consumption, average waiting time, and average maximum load deviation and demonstrated through a case study that the method can be used to reduce the total energy consumption, the maximum load deviation and the waiting time of passengers. Xie et al. [10] designed a synchronized metro schedule and stopping timetable optimization model for passengers and energy saving and demonstrated experimentally that it is very effective in reducing train energy consumption, running time, and delay probability. In terms of optimization methods, Wihartiko et al. [11] used an improved integer programming model of the genetic algorithm to solve the bus schedule problem in chromosome design, initial population recovery technique, chromosome reconstruction, and generation-specific chromosome extinction, respectively. Shang et al. [12] established a total passenger travel time model to minimize the total passenger travel time and proposed a spatial branching delimitation algorithm to solve the model. Wang et al. [13] proposed a linear weighted compromise algorithm and a heuristic algorithm to find the best solution for the bi-objective integer programming model with the train stopping time control. Guo et al. [14] proposed a mixed integer nonlinear programming model for generating optimal train schedules and maximizing interchange synchronization events, and then a hybrid optimization algorithm (PSO-SA) combining particle swarm optimization and simulated annealing is designed, and its superiority is proved by comparing with many algorithms. Tang et al. [15] combines the genetic algorithm and the simulated annealing algorithm to find the best result of an optimization model considering multiple constraints. Liu et al. [16] developed a mathematical model of it considering headway time distance and dwell time. Then an improved artificial bee colony algorithm is designed to solve this problem. Tang et al. [17] developed a bi-objective optimization model considering the minimization of full passenger waiting time and departure time and designed an improved nondominated ranking genetic algorithm (NSGA-II) for fast search of Pareto optimal solutions by using a specific coding scheme. Huang et al. [18] proposed a two-step model for matching metro passenger relationships and reducing the full waiting time of passengers, respectively, and designed a hybrid MCMC-GASA (Markov chain Monte Carlo genetic algorithm simulated annealing) approach to solve the problem.

A review of the literature shows that there has been extensive discussion and research by many experts in the area of the subway train schedule optimization problem, and in previous studies, it was common to assume a constant passenger flow model at a particular moment in time and then to optimize the train travel plan for that particular moment in time. The reality is that passenger flows vary dynamically with time distribution [19], and in previous train schedule optimization, the passenger flow distribution

is often first assumed to be normal or some other distribution pattern. However, modeling passenger flow patterns in complex scenarios by such approximate estimation models is inaccurate, which may lead to the inapplicability of the optimization model to the normalized environment. With the rapid development of big data technology, big data analysis methods provide new methods and techniques for train schedule optimization in the metro. We collect historical passenger ticket card data from the metro AFC, clean the data through a Hadoop big data platform, and then calculate the passenger arrival rate at each station and the passenger disembarkation rate between stations distributed over time. A multi-objective train schedule optimization model that takes into account train movements and passenger demand is proposed. Then a parallel genetic algorithm (GA) incorporating a modified simulated annealing algorithm is designed and the optimal subindividual retention strategy is added to get the best result. We use the measured data of Shenzhen metro to evaluate the proposed model and a solution method, and the result shows that the method is effective and accurate.

Other parts of this article are as follows: in Section 2, we describe the methodology for AFC data acquisition and processing. In Section 3, we develop a multi-objective optimization model considering metro operations and passenger travel demand. In Section 4, we propose a parallel improved genetic algorithm incorporating simulated annealing algorithm to solve the multi-objective optimization function. Section 5 brings in the multi-objective optimization model based on real historical passenger flow data of the Shenzhen metro and solves the optimal solution. Finally, Section 6 gives the conclusion of this paper.

2. Data Acquisition and Processing

2.1. Description of Data. The raw data we capture is the ticket card information from the metro automatic fare collection (AFC) system. When a passenger through the gate to ride the subway, the passenger information is saved in the AFC system and a corresponding travel data set is generated. The data set includes start station address, start line, start station time, destination station address, destination line, and destination time. Shenzhen metro generates approximately 5.9 million records per day, each record containing more than 60 attributes. To facilitate data statistics in the future, the source data is cleaned and transformed, and only the fields we can use are retained, as shown in Table 1.

The start station ID (indicated by $s_station$) is the station number where the passenger enters the station. Start line ID (denoted by s_line) is the line where the passenger enters the station. Inbound time (denoted by s_time) is the time when the passenger entered the station. Destination station ID (denoted by $d_station$) is the station where the passenger left the station. The destination line ID (denoted by d_line) is the line on which the passenger exits the station. The exit time (represented by d_time) is the time the passenger left the metro station. Thus, a passenger's ride record can be expressed as

TABLE 1: Example of passenger travel data.

s_time	s_station	s_line	d_time	d_station	d_line
20-08-19 18:47:09	268030	268	20-08-19 19:10:05	268033	268
20-08-19 20:11:09	241019	241	20-08-19 20:31:29	241011	241

$$x = (s_time, s_station, s_line, d_time, d_station, d_line). \quad (1)$$

2.2. Data Processing. In recent years, big data analysis technology has been developing, and accordingly, big data platforms are becoming more and more advanced and perfect [20]. The core features of big data platforms are scalable distributed storage and efficient parallel data processing and computing capabilities. In this paper, we set up a multinode Hadoop platform and add the corresponding ecological components, such as Hive and HBase, and then

$$C_i^j = (C | d_time \in t_2, s_line = lineid, d_line = lineid, i \in s_station, j \in d_station). \quad (4)$$

The passengers' arrival rate at j stations can be calculated by dividing $C_{in_station}^j$ by t_1 .

$$\lambda_{t_1}^j = \frac{C_{in_station}^j}{t_1} \quad (j \in in_station). \quad (5)$$

The proportion of passengers leaving stations i can be calculated by dividing $C_{in_station}^j$ by C_i^j .

$$\rho_i^j = \frac{C_i^j}{C_{in_station}^j} \quad (i \in in_station, j \in out_station). \quad (6)$$

3. Multi-Objective Optimization Model

To improve the operational efficiency of the metro, we develop a passenger flow data-driven dynamic optimization model of the metro train operation plan in this section based on the passenger flow and travel data preprocessed by the Hadoop platform described in the previous part. The optimization model considers both metro operation and passenger experience, including train operation stability and train loading efficiency, and passenger experience including passenger ride and waiting time and the number of passengers on the train. We use a metro line consisting of k metro stations and l trains [21] as the target of our study, specifying the starting station as station 1 and the ending station as station k . To quantify the various parameters to describe the mathematical model, to better match the actual situation of metro operations as well as to simplify the overall optimization model, the following assumptions are required in this paper to build the model in terms of both passengers and metro trains.

complete data processing and model building in this big data platform.

To reduce data interference and computational effort, we take the raw data stored in HDFS for data cleaning and then use Hive to store the data. Calculations are performed using Hive to get the passenger arrival and disembarkation rates.

Calculate the number of passengers who take the metro at station j in the same line in the period t_1 .

$$C_{in_station}^j = (C | s_time \in t_1, s_line = lineid, j \in s_station). \quad (2)$$

Count the number of passengers who leave stations j in the same line during period t_2 .

$$C_{out_station}^j = (C | d_time \in t_2, d_line = lineid, j \in d_station). \quad (3)$$

Calculate the number of passengers who take the metro from station i and get off at station j in the period t_2 .

- (1) Only one train can stop at the same station in the same direction of subway operation at the same time, and there will be no overtaking when parallel trains are running on the subway line.
- (2) When the train enters the metro station, all passengers line up to get off and get on following the principle of "first off, then on, first to arrive, first to serve."
- (3) The maximum capacity of each train is a fixed value. When the number of passengers waiting on the platform exceeds the capacity of the train, the remaining passengers need to continue to wait on the platform and wait for a train to arrive.

Assumption (1) is generally applicable to most urban transportation systems to ensure that trains operate in sequence. Assumption (2) is in line with the mainstream passenger queuing principle, and assumption (3) can improve the running stability of the train and the comfort of passengers.

3.1. Model of Train Operation. Describing the operation of a train is generally performed by train exit time, interstation running time, entry time, and dwell time [22]. Given a train l and a subway station k , the travel interval between train l and its preceding train $l-1$ can be expressed as the difference between the exit times of the two trains at station k :

$$h_{(l,k)} = d_{(l,k)} - d_{(l-1,k)}, \quad (7)$$

where $d_{(l,k)}$ is the moment of departure of train l from station k and $d_{(l-1,k)}$ is the moment of departure of train $l-1$

from station k . $d_{(l,k)}$ can be represented by the moment $a_{(l,k)}$ when train l arrives at station k and the stop time $s_{(l,k)}$ at station k .

$$d_{(l,k)} = a_{(l,k)} + s_{(l,k)}. \quad (8)$$

The time $a_{(l,k)}$ at which the train arrives at station k can be described as the total of the train's departure time $d_{(l,k-1)}$ from the last station and traveling time $r_{(l,k-1)}$ between the two stations.

$$a_{(l,k)} = d_{(l,k-1)} + r_{(l,k-1)}. \quad (9)$$

The running time is usually a preset fixed value because the distance between stations is certain and the train runs in autopilot mode between the two stations.

The stopping time $s_{(l,k)}$ of train l at station k can be expressed by this equation:

$$s_{(l,k)} = s_{min} + a \cdot \frac{U_{(l,k)}}{2N_{door}} + b \cdot \frac{D_{(l,k)}}{N_{door}}, \quad (10)$$

where s_{min} is the minimum stopping time of the train, a and b are two parameters that denote the time required for a passenger to board and alight respectively, which can be obtained analytically, N_{door} is the number of trains opening their doors at stations, for the convenience of calculation, we assume that the passengers who are going to get on the train will consciously form two lines, and the passengers who are going to get off the train will form one line in the train, $U_{(l,k)}$ and $D_{(l,k)}$ denote the number of passengers getting on and getting off train l at station k , respectively, these two parameters can be estimated from the historical data.

In addition, to improve safe train operation, two adjacent trains need to satisfy the minimum headway time constraint, i.e., the difference between the arrival time of train l at station k and the departure time of the previous train $l-1$ from station k should be greater than a constant, which can be described as $d_{(l,k)} - d_{(l-1,k)} \geq Hmin$.

3.2. Model of Passenger Demand. The number of passengers in a train l when the train leaves the station k is $P_{(l,k)}$. It can be represented by the number of passengers $P_{(l,k-1)}$ in train l when it leaves station $k-1$, the number of passengers $D_{(l,k)}$ who get off from station k and the number of passengers $U_{(l,k)}$ who get on board at station k :

$$P_{(l,k)} = P_{(l,k-1)} - D_{(l,k)} + U_{(l,k)}. \quad (11)$$

There is a maximum amount of passengers that a train can carry when it is running. As a result, passengers may become stranded at stations during peak traffic. The number of passengers boarding the train at the station k is $U_{(l,k)}$. It can be expressed by the number of passengers $P_{(l,k)}^{remain}$ remaining in the train at station k and the number of passengers $W_{(l,k)}^{wait}$ waiting at station k :

$$U_{(l,k)} = \min(P_{(l,k)}^{remain}, W_{(l,k)}^{wait}), \quad (12)$$

where the number of remaining passengers in train l at station k is $P_{(l,k)}^{remain}$. It can be represented by the maximum

number of passengers on board as $Q_{(l,max)}$, the number of passengers on board as $P_{(l,k-1)}$, and the number of passengers off the train as $D_{(l,k)}$:

$$P_{(l,k)}^{remain} = Q_{(l,max)} - (P_{(l,k-1)} - D_{(l,k)}). \quad (13)$$

The number of passengers waiting for train l at station k is $W_{(l,k)}^{wait}$. It can be expressed by the number of passengers $W_{(l-1,k)}^{remain}$ stranded at station k by the previous train $l-1$ and the number of passengers $\lambda_k (d_{(l,k)} - d_{(l-1,k)})$ arriving in the travel interval between adjacent train l and train $l-1$, where λ_k is the passenger arrival rate in the interval between two adjacent trains $(d_{(l,k)} - d_{(l-1,k)})$ [23].

$$W_{(l,k)}^{wait} = W_{(l-1,k)}^{remain} + \lambda_k (d_{(l,k)} - d_{(l-1,k)}). \quad (14)$$

The number of passengers $W_{(l,k)}^{remain}$ stranded by train l at station k can be described as

$$W_{(l,k)}^{remain} = W_{(l,k)}^{wait} - U_{(l,k)}. \quad (15)$$

The number of passengers on train l who get off at station k is $D_{(l,k)}$. It can be represented by the number of passengers who boarded at the previous stations as $\sum_{i=1}^{k-1} U_{(l,i)}$, and the passenger boarding and alighting ratio O-D matrix as $E_{(i,k)}$:

$$D_{(l,k)} = \sum_{i=1}^{k-1} U_{(l,i)} E_{(i,k)}. \quad (16)$$

Big data analysis techniques can be used to statistically analyze historical passenger flow data to determine the proportion of passengers boarding and disembarking at each stop.

3.3. Multi-Objective Optimization Function. The optimization of train schedules based on dynamic and uneven passenger flows mainly includes train operation optimization and passenger satisfaction optimization. The train operation optimization mainly includes reducing the deviation of the actual train capacity from the desired capacity and ensuring the balance of train operation. Passenger satisfaction optimization consists of reducing the waiting time in the station and the travel time between stations.

The waiting time J_1 of passengers at the platform is a sum of the waiting time of passengers who are stranded after the departure of the previous train and the waiting time of new arrivals in the interval between the operation of two trains. It can be expressed as

$$J_1 = \sum_{l=2}^N \sum_{k=1}^M \left(\frac{1}{2} \lambda_k |d_{(l,k)} - d_{(l-1,k)}| + W_{(l-1,k)}^{remain} |d_{(l,k)} - d_{(l-1,k)}| \right). \quad (17)$$

Passenger travel time is the sum of the time passengers who are on board when the train is running and the time passengers who wait on board when the train stops at each station and can be expressed as

$$J_2 = \sum_{l=1}^N \sum_{k=1}^M \left((P_{(l,k)} - D_{(l,k+1)}) s_{(l,k)} + P_{(l,k)} r_{(l,k)} \right). \quad (18)$$

The train running balance J_3 can be expressed as the difference between the stopping times of two adjacent trains running between stations at each station, and can be expressed as

$$J_3 = \sum_{l=2}^N \sum_{k=1}^M (|r_{(l,k)} - r_{(l-1,k)}| + |s_{(l,k)} - s_{(l-1,k)}|). \quad (19)$$

The difference J_4 between the actual capacity of the train and the desired capacity of the train can be expressed as follows:

$$J_4 = \sum_{l=2}^N \sum_{k=1}^M |P_{(l,k)} - P_{\max}|. \quad (20)$$

Considering the above elements to be optimized, the multi-objective optimization function can be described as

$$\begin{aligned} \min J &= a \cdot J_1 + b \cdot J_2 + c \cdot J_3 + d \cdot J_4, \\ \text{s.t. } &\begin{cases} d_{(l,k)} - d_{l-1,k} \geq d_{(\min,k)}, \\ s_{(l,k)} \leq s_{\max}, \end{cases} \end{aligned} \quad (21)$$

where a, b, c, d denote the weights of each objective, which are set differently according to different optimization needs. It is vital to increase the values of a and b suitably during peak passenger periods in order to carry passengers rapidly and decrease waiting and journey times. The stability of train operation should be improved and the operating cost should be decreased during the low-peak time of passenger flow, thus the values of c and d need to be suitably increased. The weights can be set in a balanced manner, taking into account the stability of train operation and the length of time passengers must wait, during the stable period of passenger flow. In conclusion, when choosing the weights for each optimization target, it is important to take into account both the passenger flow and the optimization requirements. The best weights should be chosen after conducting numerous tests.

4. Solution Method

To find the best solution for the multi-objective optimization model proposed in the previous section, we designed an improved parallelized genetic algorithm and completed the algorithm implementation in Hadoop big data platform.

4.1. Improved Genetic Algorithm. Genetic algorithm is a computing model that models natural selection and biological evolution, and it is a way of searching for optimal solutions by simulating the natural evolutionary process. GA provides a number of benefits, including the capacity to handle continuous and discrete variables, the adaptability of constraint definition, the capacity to handle huge search spaces, and the capacity to provide numerous optimal or good solutions [24]. The simulated annealing algorithm is derived from the solid annealing principle and has shown to be quite successful in locating the global optimum for a variety of NP-hard combinatorial problems [25]. Starting from a certain initial temperature, the probabilistic abrupt

change property of SA can help the objective function to obtain the global optimal solution in the desired time as the temperature decreases [26]. Given the benefits of these two methods, Gandomkar et al. [27] presented a hybrid algorithm that combines GA and SAA to optimize the distributed generation resource allocation problem.

The advantage of the genetic algorithm is that it can quickly search out the whole solution in the solution space, excellent global search ability, overcoming the fast descent trap problem of other algorithms; suitable for distributed computing, natural parallelism speeds up the convergence speed. Relatively, genetic algorithm local search ability is insufficient, a simple genetic algorithm is time-consuming and less efficient for search in late evolution. SAA has a relatively powerful local search ability [28], but it cannot make the optimization search process the most promising area. Therefore, we improved the genetic algorithm and designed an adaptive genetic algorithm incorporating a simulated annealing algorithm with an optimal individual replacement strategy as follows:

- (a) Encoding: The code consists of the train's departure moment at the origin station and the stopping time at each station, using a real number code whose values are generated within the departure interval and stopping constraints. An individual in the initial population can be represented as $(h_1, h_2, \dots, h_l, t_{11}, t_{12}, \dots, t_{1k}, t_{2k}, \dots, t_{l1}, t_{l2}, \dots, t_{lk})$, where l is the number of trains, k is the number of stations, h_i denotes the interval between the departure of the train i and the preceding train from the first station, and t_{ij} denotes the stopping time of the train i at the station j .
- (b) Selection: The genetic algorithm uses the roulette wheel selection method, but the probabilistic selection is random, to retain the good individuals, we use the best individual replacement strategy, i.e., we replace the individuals with low fitness values with those with high fitness values, thus increasing the fitness of the offspring. The specific selection method is as follows:
 - (1) Find the individual x_b with the highest fitness by calculating the fitness of each individual in the current population, assuming that the number of individuals in the population is N .
 - (2) Calculate the probability $p(x_i)$ that an individual is selected and the cumulative probability $q(x_i)$.

$$p(x_i) = \frac{\text{Fit}(x_i)}{\sum_{i=1}^N \text{Fit}(x_i)}, \quad (22)$$

$$q(x_i) = \sum_{j=1}^i p(x_j).$$

- (3) Randomly generate N numbers between $[0, 1]$ in the array m as the selection probability. If the cumulative probability $q(x_i)$ is greater than the

Input: < key, value >, where the key is individual in one population, and the value is fitness in one population.

Output: < key', value' >, where key' is the best individual in the iterative process, and value' is the best fitness value individual to key'.

Algorithm Procedure:

- (1) Identify the number of iterations as M.
- (2) Initiate integer $i = 0$.
- (3) While($i < M$):
 - Compute individual fitness;
 - Compute individual cumulative probability;
 - Select;
 - Crossover;
 - Mutation;
 - $i++$;
 - End while
- (4) Compute individual fitness;
- (5) Find the best chromosome and fitness;
- (6) Output the chromosome and fitness;

ALGORITHM 1: Mapper.

Input: < key, value >pair, where the key is the best individual in each population, and the value is the best fitness in each population.

Output: < key', value' >pair, where key' is the ideal individual in all populations, and value' is the best fitness in all optimal individuals of the population.

Algorithm Procedure:

- (1) Identify the number of population N;
- (2) For $i = 1$ to N:
 - Find maximum fitness;
 - End For
- (3) Output the chromosome and fitness;

ALGORITHM 2: Reduce.

element $m[i]$ in the array, the individual x_i is selected, if it is less than $m[i]$, the next individual x_{i+1} is compared until an individual is selected.

- (4) Repeat step 3 until N individuals are selected.
 - (5) Find the individual x_w with the lowest fitness by recalculating the fitness of the N freshly created individuals.
 - (6) Replace the worst individual x_w with the previously selected best individual x_b to form the next generation population.
- (c) Crossover: Two individuals are selected for simulated binary crossover operation based on the set crossover probability, and then the child fitness value $\text{Fit}(c)$ and the parent fitness value $\text{Fit}(p)$ are calculated for the simulated annealing operation. Let T_0 denotes the initial temperature, α is a positive number less than 1 and generally takes values between 0.8 and 0.99 the temperature calculation formula is

$$T(n+1) = \alpha T(n). \quad (23)$$

The new state is accepted at annealing with a probability according to the Metropolis criterion.

$$P = \begin{cases} 1, \text{Fit}(c) > \text{Fit}(p), \\ \exp\left(\frac{\text{Fit}(p) - \text{Fit}(c)}{T}\right), \text{Fit}(c) \leq \text{Fit}(p). \end{cases} \quad (24)$$

- (d) Mutation: Regular polynomial variation encoded in real numbers for chromosomes that have completed the crossover operation according to a set probability of mutation.

4.2. Parallel Genetic Algorithm. Based on the improved genetic algorithm proposed earlier, we have proposed an improved parallel genetic algorithm. The specific algorithm is described as follows:

In Algorithm 1, Step (3) is the regular genetic operation, including selecting individuals with high fitness from the population and eliminating individuals with low fitness, crossing chromosomes with a certain probability, and mutating chromosomes with a certain probability. Step (4) is to compute the individual fitness after the iteration. Step (5) is to choose the optimal chromosome and fitness. Steps (6) is to output the

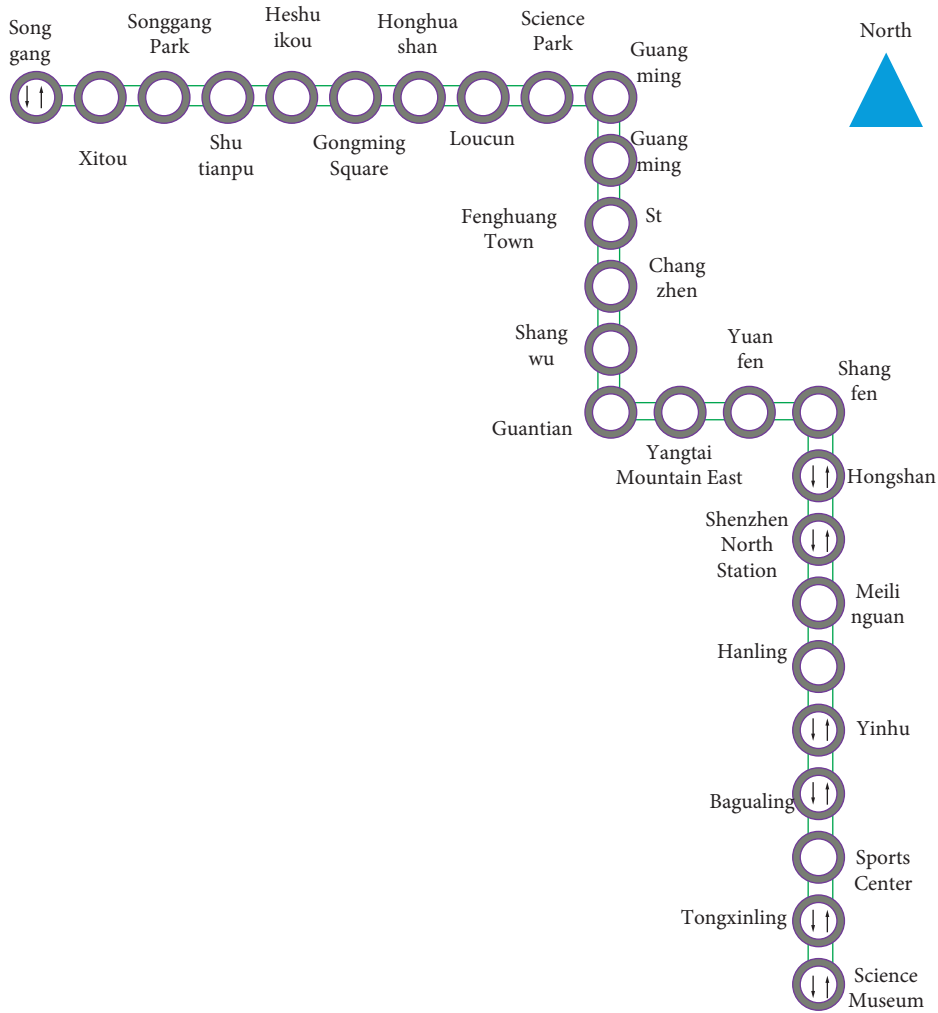


FIGURE 1: Station distribution of Shenzhen metro line 6.

TABLE 2: Stopping time of each station.

Station name	Stop Time(s)	Station name	Stop Time(s)
Songgang	40	Guantian	40
Xitou	40	Yangtai mountain east	40
Songgang park	40	Yuanfen	40
Shutianpu	40	Shangfen	40
Heshuikou	40	Hongshan	40
Gongming square	40	ShenzhenNorth station	45
Honghuashan	40	Meilinguan	40
Loucun	40	Hanling	40
Science park	40	Yinhu	40
Guangming	40	Bagualing	40
Guangming st	40	Sports center	40
Fenghuang town	45	Tongxinling	45
Changzhen	45	Science museum	40
Shangwu	40		

TABLE 3: Running time between stations.

Station section	Time(s)	Station section	Time(s)
Songgang-xitou	72	Shangwu-Guantian	104
Xitou-Songgang park	87	Guantian-Yangtai mountain east	246
Songgang park-shutianpu	136	Yangtai mountain east-Yuanfen	119
Shutianpu-Heshuikou	104	Yuanfen-Shangfen	132
Heshuikou-Gongming square	89	Shangfen-Hongshan	152
Gongming square-Honghuashan	120	Hongshan-ShenzhenNorth station	101
Honghuashan-Loucun	118	ShenzhenNorth station-meilinguan	191
Loucun-science park	86	Meilinguan-Hanling	160
Science park-Guangming	116	Hanling-Yinhu	146
Guangming-Guangming st	84	Yinhu-Bagualing	81
Guangming st-Fenghuang town	120	Bagualing-Sports center	68
Fenghuang town-changzhen	162	Sports center-tongxinling	70
Changzhen-Shangwu	238	Tongxinling-Science museum	82

TABLE 4: Detailed parameters of the train.

Type	Number of doors	Rated passenger capacity	Maximum passenger capacity
A	10	1860	2738

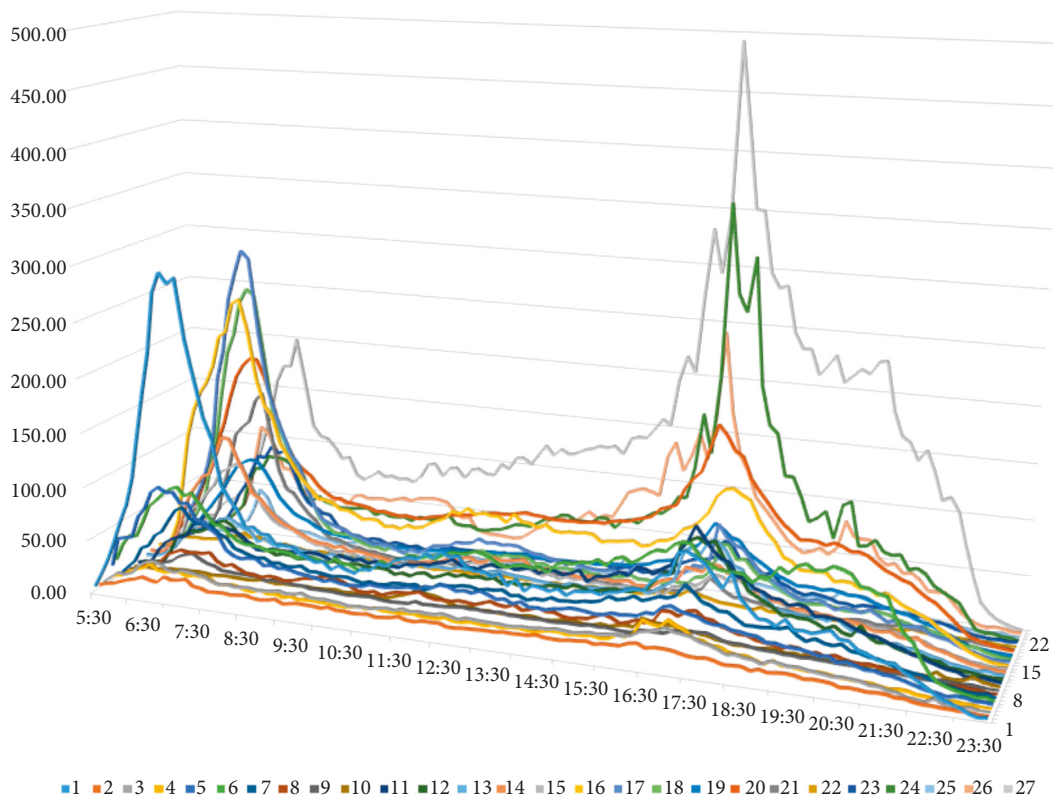


FIGURE 2: The passenger arrival rate of Shenzhen metro line 6.

intermediate $\langle \text{key}, \text{value} \rangle$ pair, and $\langle \text{key}', \text{value}' \rangle$ pair. In Algorithm 2, Step (2) is to find the optimal chromosome and fitness in each population's optimal solution. Step (3) is to output the final chromosome $\langle \text{key}, \text{value} \rangle$ pairs and fitness value $\langle \text{key}', \text{value}' \rangle$ pairs to the sequence file on HDFS.

5. Numerical Results

With the intention of verifying the performance of our designed optimization method in the multi-objective optimization model of the metro schedules, we collected the AFC data of Shenzhen metro line 6. The dataset

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0.01	0.05	0.03	0.08	0.15	0.04	0.02	0.01	0	0.06	0.03	0.02	0.06	0.04	0.06	0.03	0.04	0.07	0.11	0.01	0	0	0.01	0.01	0.01	0.02
2	0	0.05	0.03	0.08	0.16	0.04	0.03	0.01	0.01	0.07	0.03	0.02	0.04	0.03	0.11	0.04	0.02	0.07	0.11	0	0	0.01	0.01	0	0	0.01
3	0	0	0.03	0.09	0.15	0.05	0.03	0.01	0	0.06	0.03	0.03	0.05	0.04	0.1	0.04	0.03	0.07	0.12	0	0	0	0.01	0.01	0.01	0.03
4	0	0	0	0.05	0.12	0.1	0.05	0.02	0.01	0.09	0.03	0.03	0.04	0.04	0.08	0.04	0.03	0.08	0.12	0.01	0	0.01	0.01	0.01	0.01	0.01
5	0	0	0	0	0.14	0.11	0.06	0.02	0.01	0.1	0.06	0.03	0.06	0.03	0.07	0.03	0.03	0.08	0.13	0.01	0	0.01	0	0	0	0.02
6	0	0	0	0	0	0.09	0.06	0.03	0.01	0.2	0.06	0.06	0.05	0.03	0.07	0.02	0.03	0.09	0.14	0.01	0	0	0.01	0.01	0	0.02
7	0	0	0	0	0	0	0.04	0.01	0.01	0.21	0.1	0.05	0.08	0.03	0.08	0.03	0.05	0.09	0.15	0.01	0	0	0.01	0.01	0.01	0.03
8	0	0	0	0	0	0	0	0.01	0.01	0.2	0.07	0.07	0.06	0.05	0.12	0.04	0.04	0.1	0.16	0.02	0	0.01	0	0	0.01	0.02
9	0	0	0	0	0	0	0	0	0.04	0.19	0.08	0.05	0.07	0.04	0.15	0.02	0.02	0.11	0.17	0.01	0	0	0.01	0	0	0.02
10	0	0	0	0	0	0	0	0	0	0.21	0.06	0.07	0.06	0.05	0.13	0.04	0.05	0.12	0.18	0	0.01	0.01	0	0.01	0	0.01
11	0	0	0	0	0	0	0	0	0	0	0.1	0.07	0.1	0.08	0.12	0.04	0.06	0.11	0.19	0.02	0	0.01	0.02	0.01	0.02	0.05
12	0	0	0	0	0	0	0	0	0	0	0	0.03	0.12	0.05	0.14	0.07	0.15	0.11	0.19	0.02	0.01	0.01	0.02	0.02	0.02	0.04
13	0	0	0	0	0	0	0	0	0	0	0	0	0.08	0.07	0.17	0.08	0.11	0.14	0.22	0.03	0	0.01	0.02	0.01	0.02	0.03
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0.19	0.09	0.11	0.14	0.24	0.02	0.01	0.02	0.04	0.01	0.02	0.06
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0.09	0.13	0.14	0.24	0.02	0.02	0.02	0.05	0.01	0.05	0.07
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.15	0.11	0.23	0.07	0.02	0.03	0.06	0.02	0.04	0.17
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.21	0.05	0.16	0.07	0.03	0.02	0.12	0.02	0.08	0.24
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.24	0.41	0.04	0.01	0.02	0.07	0.02	0.07	0.11
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.49	0.07	0.02	0.03	0.1	0.03	0.08	0.18	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.15	0.05	0.07	0.21	0.07	0.17	0.28	
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0.04	0.21	0.09	0.21	0.4	
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0.2	0.14	0.35	0.25	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.24	0.11	0.19	0.47	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.28	0.52
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.22	0.78	
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

FIGURE 3: Proportion of passengers getting off between stations.

TABLE 5: Initial information of the genetic algorithm.

Parameters	Value
Population size	50
Crossover rate	0.95
Variation rate	0.05
Genetic generation	500

contains a total of 15 million passenger trips and the data file size is over 25 GB. Shenzhen metro line 6 has a total of 27 stations, the distribution of which is shown in the map below (Figure 1).

The existing train schedules have fixed stopping times at each station as shown in the following Table 2.

Since the subway trains are in automatic mode, the train runs between two adjacent stations for a fixed period of time. This is shown in the following table (Table 3).

The train is a 6-part A-type train and the other information about the train are listed as follows. (Table 4).

The passenger arrival rate with time distribution is obtained using the historical passenger flow data statistics with the Hadoop big data platform for the study period. The following figure shows the distribution of passenger arrival rate at each station of Shenzhen metro line 6 over time in a day (Figure 2).

We decided to focus on two hours of the morning peak period to perform more precise schedule optimization research. In Figure 3, the statistical exit ratios between stations are displayed, where the final station is on the horizontal axis and the starting station is on the numerical axis. The data in the figure is 0, which means that few or no passengers get off from the station during the period.

A total of 17 trains are scheduled to depart during this period with a departure interval of 435 s. Using the departure

TABLE 6: Departure interval of trains at the departure station.

Number	Headways	Number	Headways
Train 1	400	Train 10	395
Train 2	468	Train 11	376
Train 3	394	Train 12	366
Train 4	421	Train 13	409
Train 5	411	Train 14	384
Train 6	399	Train 15	395
Train 7	386	Train 16	435
Train 8	401	Train 17	435
Train 9	420		

interval of trains at the first station and stopping time at each station as the decision variables, the improved genetic algorithm introduced above is used to find the best result. The input information for setting up the genetic algorithm is listed below (Table 5).

The waiting time and travel time of passengers are the first optimization objectives, and the train operation balance is the secondary optimization objective. Therefore, the weights of the optimization function are set as $a=0.4$, $b=0.3$, $c=0.2$, $d=0.1$, respectively. The optimized train schedule does not increase the number of departures, and the departure interval of each train at the departure station is shown in the table below. Table 6.

The results of the comparison between the original train timetable and the optimized timetable are shown in Figure 4, where the horizontal axis is the arrival and departure time of trains at various stations and the vertical axis of each station of line 6 (Table 7).

The experimental results show that the optimized metro schedule reduces passenger waiting time by 21.42%, reduces passenger travel time by 22.56% and increases train full capacity by 2.65% compared to the existing schedule. It can be seen that the optimized metro timetable driven by

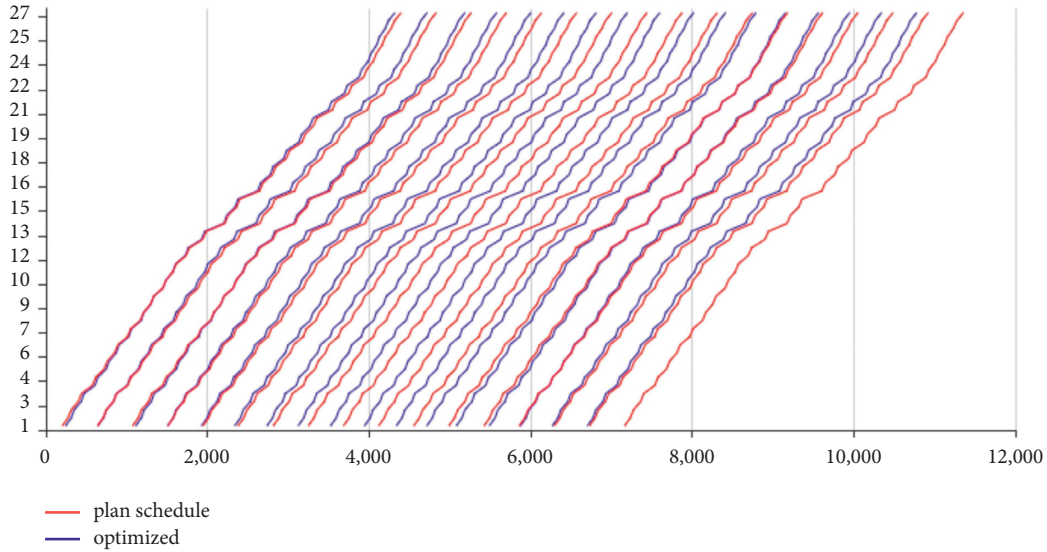


FIGURE 4: Comparison of original and optimized train schedules.

TABLE 7: Comparison of train schedule parameters before and after optimization.

Timetable	Original timetable	Optimized timetable
Passengers' total waiting time	32613	25625.1(-21.42%)
Total passengers' travel time	80235	62131.4(-22.56%)
Difference between actual train capacity and desired train capacity	321503	330032(+2.65%)

passenger flow data improves passenger satisfaction and train operation efficiency more than the existing planned schedule.

6. Conclusion

By analyzing and mining past passenger flow data, which the metro system creates in large quantities, it is possible to significantly increase operational efficiency and passenger pleasure. In this paper, we built a Hadoop big data platform to process and analyze the enormous historical passenger flow data of the Shenzhen metro, then we built a data warehouse to calculate the passenger inbound rate and the station-to-station disembarkation ratio of each station that changes at any time of the day through the Hive component. A multi-objective model considering both trains and passengers is proposed to optimize the train timetable. We have designed a parallel genetic algorithm incorporating simulated annealing algorithm improvements, using the best individual replacement strategy to retain the best individuals to get the best solution. Results of experiments using actual data from Shenzhen metro line 6 show that an improved train timetable can decrease passengers' waiting and transit times while also enhancing the balance of train operations and transportation effectiveness.

In future studies, we will further develop the proposed model with AFC data for multiple line interchanges. We will consider train operations for train turnarounds and turn-

backs for the study, and another task to be performed is to analyze the passenger travel characteristics on holidays and weekends to optimize various nonworking day train schedules based on it.

Data Availability

Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Beijing Municipal Natural Science Foundation (Grant No. L201015); the National Key R&D Program of China (Grant No. 2020YFC0833104); and The Green, Intelligent and Safe Mining of Coal Resources (Grant No. 52121003).

References

- [1] E. B. Setyawan and D. Diah Damayanti, "Integrated railway timetable scheduling optimization model and rescheduling recovery optimization model: a systematic literature review," in *Proceedings of the 5th International Conference on*

- Industrial Engineering and Applications (ICIEA)*, pp. 226–230, Singapore, April 2018.
- [2] J. Feng, L. Liu, Q. Pei, and K. Li, “Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks,” *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.
 - [3] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, “Joint security and train control design in blockchain-empowered CBTC system,” *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.
 - [4] J. Yang, Y. Zheng, K. Yan et al., “SPDNet: a real-time passenger detection method based on attention mechanism in subway station scenes,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7978644, 13 pages, 2021.
 - [5] L. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, “Cross-layer defense methods for jamming-resistant CBTC systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2021.
 - [6] Y. Wang, D. Li, and Z. Cao, “Integrated timetable synchronization optimization with capacity constraint under time-dependent demand for a rail transit network,” *Computers & Industrial Engineering*, vol. 142, Article ID 106374, 2020.
 - [7] T. Zhang, D. Li, and Y. Qiao, “Comprehensive optimization of urban rail transit timetable by minimizing total travel times under time-dependent passenger demand and congested conditions,” *Applied Mathematical Modelling*, vol. 58, pp. 421–446, 2018.
 - [8] Y. Qu, H. Wang, J. Wu, X. Yang, H. Yin, and L. Zhou, “Robust optimization of train timetable and energy efficiency in urban rail transit: a two-stage approach,” *Computers & Industrial Engineering*, vol. 146, Article ID 106594, 2020.
 - [9] X. Wu, H. Dong, and C. K. Tse, “Multi-objective timetabling optimization for a two-way metro line under dynamic passenger demand,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4853–4863, 2021.
 - [10] J. Xie, J. Zhang, K. Sun, S. Ni, and D. Chen, “Passenger and energy-saving oriented train timetable and stop plan synchronization optimization model,” *Transportation Research Part D: Transport and Environment*, vol. 98, Article ID 102975, 2021.
 - [11] F. D. Wihartiko, A. Buono, and B. P. Silalahi, “Integer programming model for optimizing bus timetable using genetic algorithm,” *IOP Conference Series: Materials Science and Engineering*, vol. 166, Article ID 012016, 2017.
 - [12] P. Shang, R. Li, and L. Yang, “Optimization of urban single-line metro timetable for total passenger travel time under dynamic passenger demand,” *Procedia Engineering*, vol. 137, pp. 151–160, 2016.
 - [13] H. Wang, X. Yang, J. Wu, H. Sun, and Z. Gao, “Metro timetable optimisation for minimising carbon emission and passenger time: a bi-objective integer programming approach,” *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 673–681, 2018.
 - [14] X. Guo, H. Sun, J. Wu, J. Jin, J. Zhou, and Z. Gao, “Multi-period-based timetable optimization for metro transit networks,” *Transportation Research Part B: Methodological*, vol. 96, pp. 46–67, 2017.
 - [15] J. Tang, Y. Yang, and Y. Qi, “A hybrid algorithm for Urban transit schedule optimization,” *Physica A: Statistical Mechanics and Its Applications*, vol. 512, pp. 745–755, 2018.
 - [16] H. Liu, M. Zhou, X. Guo, Z. Zhang, B. Ning, and T. Tang, “Timetable optimization for regenerative energy utilization in subway systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3247–3257, 2019.
 - [17] J. Tang, Y. Yang, W. Hao, F. Liu, and Y. Wang, “A data-driven timetable optimization of urban bus line based on multi-objective genetic algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2417–2429, 2021.
 - [18] J. Huang, T. Zhang, and R. Wei, “Urban railway transit timetable optimisation based on passenger-and-trains matching – a case study of beijing metro line,” *PRO*, vol. 33, no. 5, pp. 671–687, 2021.
 - [19] J. Yang, X. Dong, and S. Jin, “Metro passenger flow prediction model using attention-based neural network,” *IEEE Access*, vol. 8, Article ID 30953, 2020.
 - [20] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, “Big data analytics in intelligent transportation systems: a survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
 - [21] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, “A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.
 - [22] Y. Wang, L. Zhu, Q. Lin, and L. Zhang, “Leveraging big data analytics for train schedule optimization in urban rail transit systems,” in *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1928–1932, Maui, HI, USA, November 2018.
 - [23] W. Xu, P. Zhao, and L. Ning, “A passenger-oriented model for train rescheduling on an urban rail transit line considering train capacity constraint,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 1010745, 9 pages, 2017.
 - [24] S. D. Dao, K. Abhary, and R. Marian, “A bibliometric analysis of Genetic Algorithms throughout the history,” *Computers & Industrial Engineering*, vol. 110, pp. 395–403, 2017.
 - [25] Z. G. Wang, Y. S. Wong, and M. Rahman, “Development of a parallel optimization method based on genetic simulated annealing algorithm,” *Parallel Computing*, vol. 31, no. 8–9, pp. 839–857, 2005.
 - [26] Z. Xin, L. Yu, L. Lianhui, Z. Jun, L. Yan, and H. Xiangdong, “A combination test suite generation method based on adaptive simulated annealing genetic algorithm for software product line testing,” in *Proceedings of the 2nd Asia-Pacific Computer Science and Application Conference (CSAC 2017)*, pp. 641–648, 2017.
 - [27] M. Gandomkar, M. Vakilian, and M. Ehsan, “A combination of genetic algorithm and simulated annealing for optimal DG allocation in distribution networks,” in *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 645–648, Saskatoon, SK, Canada, May 2005.
 - [28] H. Liu, Z. Lin, Y. Xu, Y. Chen, and X. Pu, “Coverage uniformity with improved genetic simulated annealing algorithm for indoor Visible Light Communications,” *Optics Communications*, vol. 439, pp. 156–163, 2019.

Research Article

Spatiotemporal Virtual Graph Convolution Network for Key Origin-Destination Flow Prediction in Metro System

Jun Yang ^{1,2}, Xiao Han ¹, Tan Ye ¹, Yinghao Tang ¹, Weidong Feng ³, Aili Wang ³,
Huijun Zuo ⁴ and Qiang Zhang ⁴

¹Big Data and Internet of Things Research Center, China University of Mining and Technology, Beijing, China

²Key Laboratory of Intelligent Mining and Robotics, Ministry of Emergency Management, Beijing, China

³SinoRail (Beijing) Network Technology Research Institute Co. Ltd, Beijing, China

⁴Unit 61741 of PLA, Beijing 100094, China

Correspondence should be addressed to Xiao Han; 874326845@qq.com

Received 5 July 2022; Accepted 22 August 2022; Published 19 September 2022

Academic Editor: Li Zhu

Copyright © 2022 Jun Yang 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.

Short-term Origin-Destination (OD) flow prediction plays a major part in the realization of Smart Metro. It can help traffic managers implement dynamic control strategies to improve operation safety. Also, it can assist passengers in making reasonable travel plans to improve the passenger experience. However, there are problems that the dimension of OD short-term traffic prediction is much higher than the base number of metro stations and the OD matrix is sparse. To resolve the above two problems, a threshold-based method is proposed to extract key OD pairs first. OD passenger flow contains the attribute information of the Origin-Destination station and exhibits similar time evolution characteristics, so the spatial and temporal correlation needs to be considered in the prediction. Pearson correlation matrix is used to build a virtual graph and model the virtual connection between OD pairs. A spatiotemporal virtual graph convolutional network (ST-VGCN), which combines the advantages of a graph neural network and gated recurrent neural network, is proposed to identify spatial associations and temporal patterns simultaneously. The proposed method is evaluated on 39 days of real-world data from Shenzhen Metro, which outperforms other benchmarks. The research in this work can contribute to the development of short-term OD flow forecasts and help to provide ideas for the research on real-time operation and management of rail transit. Furthermore, it can help to establish passenger flow prediction and early warning mechanisms to quickly evacuate a large number of passengers in case of emergency.

1. Introduction

In the wake of developments in intelligent cities, the analysis and mining of big traffic data have attracted widespread attention in the field of intelligent transportation. As an important means of transportation to alleviate road traffic congestion in big cities, urban rail transit has gradually entered the era of network operation. At the same time, a large amount of passenger flow data is generated, which makes the research on passenger flow prediction more and more important [1]. The most basic thing is modern technology has greatly guaranteed the safety of train operation [2], ensuring the growing demand for safe travel of residents. Shenzhen Metro, for example, the average travel scale of

which in a single day is up to 5 million person-times, carries more than one-third of the traffic flow of the entire city and occupies an important position in the urban public transportation system. However, the ever-increasing travel demand has brought greater and greater pressure to the operation of urban rail transit. Especially during the morning and evening rush hours, the demand for commuting is very large, so the operating efficiency of the metro directly affects the overall commuting efficiency of the city. In addition to the research on the operation safety of train control systems [3], more and more researchers have paid attention to the passenger flow data generated by metro operations and conducted multilevel analysis [4] on it to ensure the operation safety. Because of the enormous scale of

the current metro network system, passenger travel shows great complexity. Most of the existing research pays more attention to the prediction of inbound and outbound passenger flow, daily ridership, hourly ridership, and sectional passenger flow. OD passenger flow reflects the direction of flow in and out of the station, which is a visual display of passenger travel needs and contains precious information.

Because of the good connectivity between stations in the metro network, the travel demand of passengers is usually expressed by an abstract OD matrix. The elements in the matrix represent the OD passenger flow. As important carriers of the metro passenger flow, they represent the travel times between station O and station D within a specific time interval. OD matrix can well describe the source and destination of the ridership between any two stations, which is the foundation for travel behavior characteristic analysis. The existing information platform establishes a channel of communication between the metro system and passengers to provide passengers with metro network status prediction information and help passengers plan their trips reasonably. At the same time, OD passenger flow prediction combined with real-time detection of passengers in densely crowded areas of metro stations [5] can provide better services for ensuring operational safety. However, the prediction results may differ from the actual situation, especially during peak periods. This inaccuracy disrupts travel plans and creates a sense of mistrust for the information while affecting metro managers to make judgments. Therefore, how to accurately predict the OD passenger flow of the whole network is an important issue for the refined and comprehensive prediction of the rail transit network. By predicting the fine-grained passenger flow and its dynamic changes in the network state, metro managers can timely adjust the operation plan and carry out reasonable passenger flow organization and control [6], which is of great practical significance to ensure traffic safety. Among them, passenger flow organization and control includes, but is not limited to, adjusting schedules, adding services, disseminating information to passengers in response to the condition of demand surge, and incentivizing passengers to delay or change travel time.

The research of metro OD passenger flow based on AFC data mainly includes three aspects, OD matrix, the main passenger flow direction of the origin station, and key OD pairs. Since the OD matrix is sparse and highly skewed, combined with the analysis of the main direction of the origin station, it can be known that the key OD pairs reflect the travel routes of large demand, which largely determines the metro operation status of the network. Meanwhile, they are the main object of metro vehicle allocation which means higher requirements for ensuring efficient and orderly trains [7]. At the same time, the OD pair implicitly includes the information of station O and station D and is regular in time variation like time series. For instance, most of the flow in the morning rush is commuters whose commuting routes are relatively fixed. OD pairs with O as residential areas and D as business areas show a similar regularity. That is, OD pairs show some dependence on the consecutive time intervals and the similar spatial properties of the Origin-

Destination station. Therefore, how to determine the key OD pairs and capture the spatiotemporal dependence between them at the level of metro network becomes a key problem.

In this work, we propose a method to extract and predict flow of key OD pairs in metro network. According to the historical data of the AFC system, we extract the OD matrix and then obtain the key OD pairs through filtering. We introduce a spatiotemporal virtual graph convolutional network (ST-VGCN) to model the spatiotemporal dependencies between key OD pairs. The virtual topology is used to establish a similarity map. The graph convolution combined with GRU is used to extract spatiotemporal features and predict the key OD passenger flow. We carry out an experiment on the real-world dataset from Shenzhen Metro to prove the advantage of ST-VGCN. The main contributions of this study are as follows:

- (1) We obtain OD matrix by processing AFC historical data and extract the key OD pairs that have a large demand for metro travel by setting reasonable thresholds
- (2) We propose a spatiotemporal virtual graph convolutional network (ST-VGCN) model, in which we establish a similarity graph based on the Pearson coefficient as a virtual topology, to capture the spatial and temporal dependencies between key OD pairs
- (3) We have verified our method through a case study of Shenzhen Metro, and the validity of the model was confirmed by comparative tests

The rest of this paper is organized as follows: In Section 2, we review related work on OD traffic forecast. In Section 3, we describe our main work on the key OD flow prediction. In Section 4, we evaluate our method based on real data and present our results and analysis. In Section 5, we summarize our paper and discuss several possible directions for future work.

2. Related Work

Most of the researches concentrate on the prediction of road traffic flow and traffic speed [8]. More and more researchers focus on urban rail transit forecasting, among which the research on OD prediction is relatively less. Considering the similarity between OD sequence data and inbound/outbound data of urban rail transit stations, we will introduce relevant studies on traffic prediction in related work and further summarize the research on OD passenger flow prediction.

Early researchers used methods based on statistical learning, such as Historical Average [1] (HA), Autoregressive Integrated Moving Average [9] (ARIMA), Vector Autoregressive model [10] (VAR), etc. Later, traditional machine learning and neural network methods were applied to traffic flow prediction. Wu et al. [11] studied the travel time prediction problem using support vector regression (SVR). Zhu et al. [8] came up with a linear conditional Gaussian Bayesian network model for short-term traffic flow forecast, which took into account spatiotemporal

characteristics and velocity information. Yang and Hou [12] used a hybrid model based on wavelet analysis and least squares support vector machine to complete short-term rail transit passenger flow prediction. Jiao et al. [13] proposed an improved Kalman filter model based on Bayesian combination and nonparametric regression, in which real-time passenger traffic is deviated from historical data to mitigate the volatility of original data.

Because of the good performance of deep learning models in other fields, many scholars try to use them in traffic prediction. Zhang et al. [14] presented deep-ST, which is the first model that uses convolutional neural network to mine spatial dependence between grids. On this basis, ST-ReSNet [15] adopted the framework of residual convolution network and considered the time series of three different trends to mine the spatiotemporal relationship; then, external factors such as weekdays and weather were integrated to improve the accuracy of traffic flow prediction. Liu et al. [16] used Long Short-Term Memory (LSTM) and full connection layer(FC) to predict passenger flow inbound and outbound of metro stations. Ma et al. [17] converted the data of metro passengers (i.e., Destination station) in Beijing into images by using Convolutional Neural Networks(CNN) and a bi-directional LSTM layer. Yang et al. [18] proposed a novel attention mechanism-based end-to-end neural network to predict the inbound and outbound passenger flow, which improved the prediction effectiveness.

All the above models can effectively capture complex time features, and the proposed Graph Convolutional Neural Network (GCN) solves the problem of Network spatial correlation effectively. Traditional neural networks can only effectively extract some features from standard grid data and cannot deal with the complex and nonlinear traffic data well. In this case, we need to consider using graph convolutional neural network. Chen et al. [19] constructed topology, similarity (based on dynamic time-warping distance), and correlation graph to represent the dependency between passengers at different transfer stations, and then used a variant of graph neural network to conduct demand forecasting. Defferrard et al. [20] applied a three-dimensional convolution operation to seamlessly capture irregular spatiotemporal dependence on metro network. Guo et al. [21] proposed the ASTGCN model, which added an attention mechanism to take into account the dynamic influence of different time periods and different places on adjacent time periods and adjacent places. Yan et al. [22] proposed a spatiotemporal graph convolution model (STGCN) combining graph convolution and CNN based on spatial domain. Zhao et al. [23] put forward a special passenger flow forecasting prediction model based on temporal graph convolutional network (T-GCN), which combines a graph convolution network (GCN) and gated recursive network (GRU). In this model, GCN learns complex nonlinear structures to capture spatial topology, while the gated loop unit understands dynamic changes of traffic and captures time-dependent data.

There are many studies on the estimation and prediction of short-term traffic origin and destination, especially in the area of taxi and car-hailing travel [24]. The research on

passenger flow OD prediction starts from residents travel, road traffic, and urban public transportation network, which is called travel distribution prediction or traffic distribution prediction.

One of the important differences between OD prediction of traffic and public transport is the high dimension of data. A network with N sites consists of $N \times N$ OD pairs, and it is not extensible to exploit a model with a traditional method for each OD pair. Matrix/tensor decomposition is an effective method to solve the high-dimensional problem of OD matrix prediction, and many researches [25] have explored this based on it. Deep learning is also one of the mainstream OD prediction methods. Because the data is high dimensional, Toqué et al. [26] applied LSTM networks to selected high-traffic OD pairs with heavy traffic. Wang et al. [27] improved OD flow prediction network of GCN + LSTM by multi-task learning. Shen et al. [28] mixed CNN with gravity model to predict the OD matrix of the metro system. The effect of deep learning model is usually affected by noise in sparse Metro OD matrix. To reduce the influence of noise, Zhang et al. [29] developed an index called OD attraction degree (ODAD) to cover up non-important OD pairs, indicating that shielding OD pairs close to zero can improve the prediction of LSTM. Meanwhile, a Channel-wise Attentive Split-CNN (CAS-CNN) model [30] is developed for metro OD matrix prediction. Gong et al. [31] proposed a real-time delayed data collection problem and discussed how to address it. Peyman et al. [32] considered the issue of delayed data availability, which is a challenge in the prediction of dynamic Origin-Destination (OD) demand.

From the traditional method to the current artificial intelligence method, it is all to better capture the regularity of the prediction content. Most of the predictions for inflow and outflow are at the station level, while the prediction of OD flow often involves stations in the entire network. The traditional method mainly makes predictions based on statistical laws, but it cannot show the influence of various factors, and the prediction accuracy is low. The GRU method is often used for simple time series prediction, which is very effective, but it lacks the capture of spatial information in the problem of rail transit OD passenger flow prediction. Using the knowledge of graph theory and convolution, GCN can well capture the physical space information brought by the metro network and the virtual space information brought by the OD passenger flow. Masking unimportant OD pairs was proposed in previous methods, which has a strong inspiration for us. Therefore, in this work, we study how to extract important OD pairs for research. At the same time, according to the analysis of the properties of key OD passenger flow, GRU combined with GCN is selected to capture spatiotemporal information to achieve good results.

3. Methodology

3.1. Key OD Pairs Prediction Problem. The Metro AFC system records the original travel data of passengers, including card number, entry station number and time, transaction type (inbound or outbound), and other data. In

this way, we know the entry and exit stations and the corresponding time. We summarize them at a fixed time interval and calculate the traffic demand from O to D in the statistical period based on O , regardless of whether the journey is completed or not. We finally form an $N \times N$ OD matrix, where N represents the total number of metro stations in the dataset. We extract the key OD pair set as

$$K_{OD} = \{k_1, k_2, \dots, k_n\}. \quad (1)$$

In our work, a virtual graph between key OD pairs denoted as $G = (V, E)$ is established, which is built to represent the connection relationship between OD pairs. V and E denote the set of nodes and the set of edges, respectively. In our work, V is equivalent to the set of key OD pairs used to encode the characteristics of nodes, which refers to the traffic time series of each OD pair. The adjacency matrix A that only contains elements of 0 and 1 represents the virtual connection relationship between OD pairs. The corresponding element is 1 if there is an association relationship between nodes, otherwise, it is 0. The data collected by the key OD pair within t consecutive moments is expressed as $X = [x^1, \dots, x^t]$, where $x^t = [x(k_1), x(k_2), \dots, x(k_n)]^T$ represents the data collected by each key OD pair at the time t . The feature matrix $X^{n \times p}$ represents the attribute characteristics of the node, n represents the number of key OD pairs, and p represents the historical time sequence length. The purpose of our paper is to predict the passenger flow of the future m time intervals according to the historical passenger flow data of key OD pairs in previous n time intervals, which can be expressed as the following learning function:

$$[X^{T-n}, \dots, X^{T-1}, X^T; G] \xrightarrow{f} [X^{T+1}, \dots, X^{T+m}], \quad (2)$$

where X^T represents the values of all OD pairs at time interval T and f is a mapping function.

3.2. Key OD Pairs Extraction. The OD matrix describes the number of travels between each OD pair in the system during the time interval. The amount of stations in the metro system is N , and the size of the generated OD pairs is $N \times N$, which means a huge number. The heat map formed by the OD matrix at the 10-minute granularity is shown in Figure 1(a). As shown by the color bar on the right side of Figure 1(a), the brighter the color indicates the greater the amount of OD between the two stations. It can be seen that the graph has a large black part, that is, most of the elements in the OD matrix are 0, which means that there is no passenger travel demand between most of the stations. Therefore, not all OD pairs are the ones we need to pay attention to. The passenger flow of some noncritical OD pairs is scarce, and the travel demand is very random. The contribution of this kind of OD passenger flow is very small, implying that relatively few key OD pairs account for the vast majority of the overall OD passenger flows. So we use the historical passenger flow data set $H^s = \{h^1, h^2, \dots, h^s\}$ of each OD pair to filter OD pairs, where s is the length of the selected historical data, h^i represents the OD value of the period. In order to achieve the extraction of key OD pairs, we set three thresholds, including

the proportion z of nonzero value, the randomness judgment value r and the proportion p of value that is greater than the randomness judgment value. The key OD pairs meet the following conditions:

$$\frac{|H^{s'}|}{|H^s|} * 100\% \leq z \text{ and } \frac{|H^{s''}|}{|H^s|} * 100\% > p, \quad (3)$$

where $H^{s'} = \{h^i \mid h^i \in H^s \text{ and } h^i = 0, i = 1, 2, \dots, s\}$ means a subset of key OD pairs set in which elements value is zero and $H^{s''} = \{h^i \mid h^i \in H^s \text{ and } h^i > r, i = 1, 2, \dots, s\}$ a subset in which elements value is greater than randomness judgment value r .

We selected 14 OD pairs and showed them in Figure 1(b). The horizontal axis represents time. From this picture, we can see that the critical and noncritical OD pairs exhibit large differences. And different OD pairs show different time-varying laws. The color bar of OD14 is all black, indicating that there is no OD demand between the two stations. OD11–OD13 have some faint colors, indicating that the passenger travel demand between such noncritical OD pair is very random. The remaining OD pairs show different color distributions, representing different passenger flow characteristics.

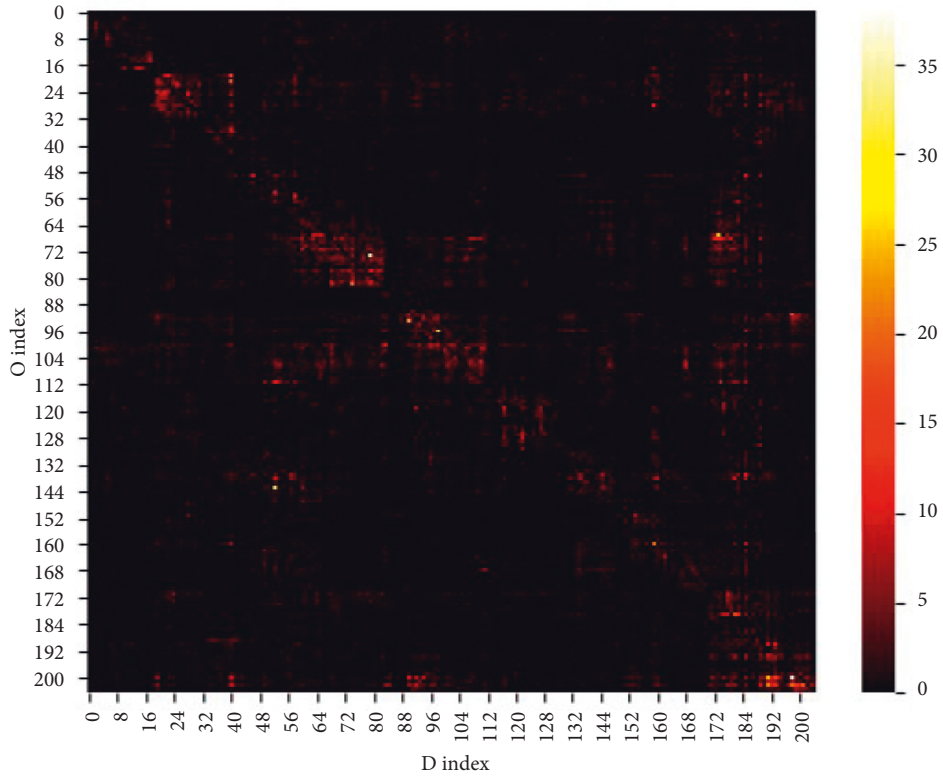
3.3. Virtual Graph Construction. Graph neural network performs graph convolution based on the relationships between nodes. We conduct research and prediction based on key OD pairs, so it loses the dependency information that comes with the physical topology of the real stations. However, the OD pairs combine the spatial properties of the origin-destination station by themselves. For example, if station O is a residential area and station D is an office area, this kind of OD pair has similar properties. The OD pairs show correlation, that is, OD pairs may have similar traffic distribution characteristics because of similar functionality. So virtual connection edges can be established to generate an adjacency matrix.

To measure the degree of correlation between two variables, researchers usually use the Pearson correlation coefficient. In the quantification of the correlation between time series, it can be used to measure how two continuous signals change together with time, and the correlation coefficient shows their relationship. The Pearson correlation coefficient can be calculated by the following formula:

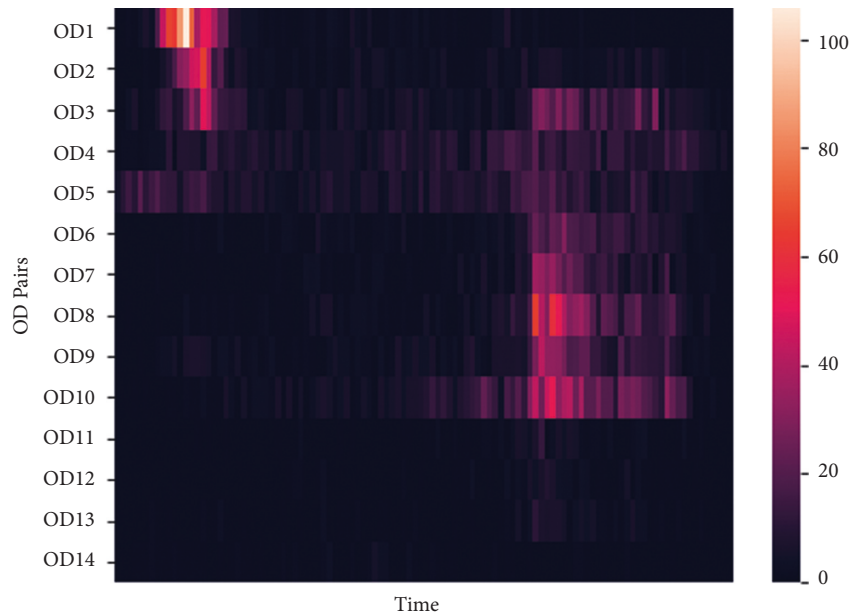
$$\rho_{X,Y} = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}}, \quad (4)$$

where X and Y represent two variables and \bar{X} and \bar{Y} represent the mean of the variables.

According to the above calculation formula, we obtain the correlation matrix R that describes the relation between key OD pairs. R_{ij} represents the Pearson correlation coefficient between observed historical passenger flow data series of the i_{th} and j_{th} OD pair. Classical GCNs encode adjacency between nodes to represent arbitrarily structured graphs. A binary-encoded adjacency matrix A is usually used to represent the connectivity between nodes. $A_{ij} = 1$ if nodes i and j are directly connected in the graph, otherwise $A_{ij} = 0$. According to the meaning of the correlation coefficient, if



(a)



(b)

FIGURE 1: Visual display of raw data. (a) Heat map formed by an OD matrix. (b) Heat map of some OD pairs over time.

the absolute value of the relative coefficient is larger, the correlation between variables is stronger. We set a threshold c to determine whether to establish a virtual connection between OD pairs. The formula is as follows:

$$A_{ij} = \begin{cases} 1, & R_{ij} \geq c, \\ 0, & R_{ij} < c. \end{cases} \quad (5)$$

3.4. Spatiotemporal Virtual-Graph Convolution Network. After completing the virtual graph construction, we use the combined model of GCN and GRU to model the spatiotemporal dependencies of key OD pairs. The prediction framework of this paper is shown in Figure 2. The main structure of the model is divided into three parts, which are the input layer, feature extraction layer, and output layer.

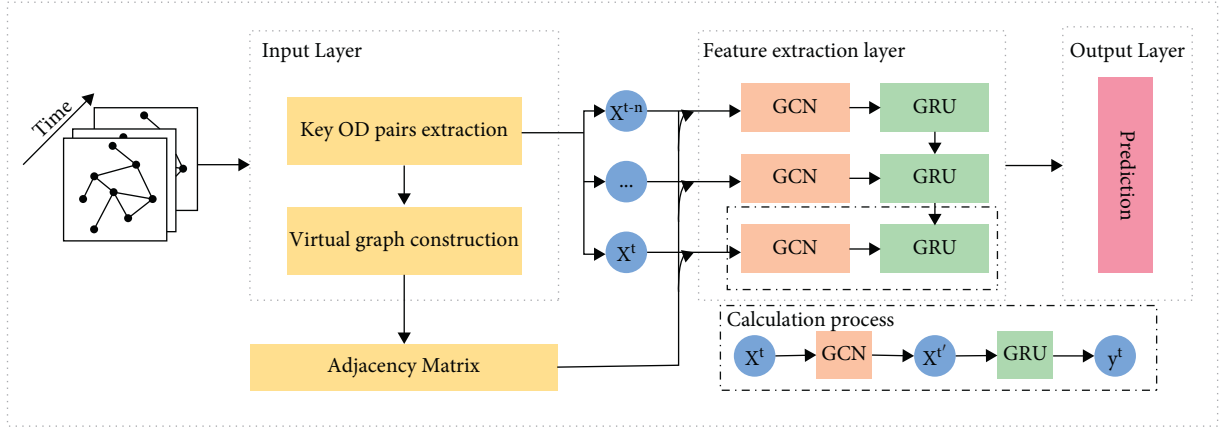


FIGURE 2: The structure of the ST-VGCN model.

The input layer receives raw historical OD matrix data. After Key OD pairs extraction, feature matrix X that represents multiple time steps historical flow of all key OD pairs is generated. And through virtual graph construction, adjacency matrix A is generated. The feature extraction layer is composed of a graph convolution module and a GRU module, and the inputs of it are X and A . Firstly, the graph convolution module receives the data from the input layer and learns both node features and structural information end-to-end through graph convolution operation to obtain rich node information and aggregate spatial features. Then the result with spatial features of each node is sent to the GRU module to capture the time series features. The final output layer gets the prediction result.

There are two main categories of GCN methods: spectral-based and spatial-based. In our work, we use a graph convolutional network based on the spectral method. Given a feature matrix X representing node features and an adjacency matrix A representing structural features, the graph convolution operation computes the information of nodes using the information of related nodes. The core calculation formula is as follows:

$$f(X, A) = \sigma(\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2} X W). \quad (6)$$

Here, $\hat{A} = A + I_N$ represents the sum of the adjacency matrix A and the identity matrix I_N ; \hat{D} represents the degree matrix of \hat{A} ; $\hat{D}^{-1/2} \hat{A} \hat{D}^{-1/2}$ represents the \hat{A} is normalized; W represents the weight matrix; and σ denotes the activation function.

GRU is one of the most widely used recurrent neural networks for processing series data. It can be regarded as a combination of reset gate and update gate. It is used to model the sequence information that has undergone graph convolution operations to capture its temporal features and complete the prediction task. After the original sequence $X = [x^1, \dots, x^t]$ passing through the graph convolution layer, new sequence data containing spatial information is obtained as $X' = [x^{1'}, \dots, x^{t'}]$. We input the new sequence data into the GRU network. The feature extraction layer improves the basic GRU structure in combination with graph convolution operation. The result is shown in Figure 3.

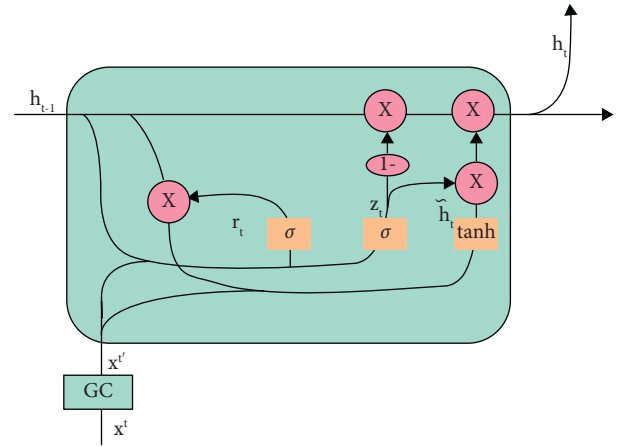


FIGURE 3: Basic GRU structure in combination with graph convolution operation.

This process can be described by the following equations:

$$\begin{aligned} z_t &= \sigma(W_z[f(X^t, A), h_{t-1}] + b_z), \\ r_t &= \sigma(W_r[f(X^t, A), h_{t-1}] + b_r), \\ \tilde{h}_t &= \tanh(W_h[f(X^t, A), (r_t * h_{t-1})] + b_h), \\ h_t &= z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t. \end{aligned} \quad (7)$$

In the formula, h_{t-1} is the hidden state at time $t - 1$; X^t is the flow information of all key OD pairs at time t ; r_t is the reset gate in the GRU model, which is about how the new input information is integrated with the previous memory; the z_t update gate indicates the amount of previous memory saved to the current time step; \tilde{h}_t is the memory content stored at time t ; h_t is the output state at time t . GRU takes the hidden state at time $t - 1$ and the current key OD pair flow information that has undergone graph convolution operations as input, and obtains the flow state at time t . The key OD flow is predicted in order to make the forecast result as close as possible to the actual traffic demand. Therefore, we need to select the loss function to estimate the degree of inconsistency between the predicted value and the real value of the model. Our ultimate goal is to minimize losses during

training process. The loss function chosen in this paper is shown as follows:

$$\text{loss} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (8)$$

where y_t and \hat{y}_t are the true OD flow and predicted values and n represents the length of observation window.

4. Experiments

4.1. Data Description. Because of few public benchmarks for metro passenger forecasts, we construct the dataset MetroSZ2020 with 39 consecutive days of metro smart card data which records card number, origin number, destination number, and the entry and exit time of each metro trip from Shenzhen, China. Because some data in the Shenzhen metro network cannot be obtained, after data cleaning, the data of 205 stations are available. As shown in Table 1, MetroSZ2020 covers 205 metro stations in Shenzhen from August 23rd to September 30th, 2020. We select 6:00–24:00 as the metro operating period. We count the OD matrix every 10 minutes, which means 108 matrixes per day and contains a total of 42025 OD pairs per matrix.

Three thresholds for z , r , and p are set as 80%, 10 and 10% respectively. We test different threshold combinations in experiments and obtain different numbers of key OD pairs. However, considering the hardware resource problem and the model training situation, the above threshold combination is finally selected. By analyzing the OD matrix heat map similar to Figure 1(a), it can be seen that it is meaningful to set the nonzero value ratio z to at least 80%. The randomness judgment value r needs to be changed according to the change of time granularity. The time granularity selected in this paper is 10 min. For the OD matrix of 205 stations, setting r to 10 can already indicate that the OD passenger flow has a certain regularity. For a station with regular passenger flow, it takes a sustained period of time to express the regularity, so setting p to 10% is a reasonable choice. The key OD pairs that meet less than 80% of the historical time series data is zero and more than 10% of the data value is greater than 10 are extracted. A total of 490 key OD pairs of data are used as input and we scale the data to (0, 1], divide by the maximum value into the data. When proceeding with virtual graph construction, we set the threshold c as 80%. The data is divided into training data and validating data according to the ratio, which is 0.8 in our research.

4.2. Model Configurations. Our experiment is completed in the Pytorch environment on a workstation equipped with an Intel(R) Core(TM) i7-6800k processor whose cache is 15M and working frequency is up to 3.40 GHz, 16 GB memory space, and NVIDIA GeForce GTX 1080 Ti graphics card. We train the model using the Adam optimizer. To obtain the best experimental results, we manually adjust the determined parameters including the number of hidden units and the number of training epoch which may greatly affect the prediction precision. As shown in Figure 4, the horizontal axis indicates the different parameter choices and the vertical axis

TABLE 1: The description of the dataset.

Description	MetroSZ2020
Date	2020/8/23–2020/9/30
Time	6:00–24:00
Station number	205
Matrix dimension	205*205
Key OD pairs number	490
Time interval	10 min

indicates the variation of the different metrics. The red dots indicate the performance of each metric under the selected parameters. First, we test the training epoch in the set [500, 1,000, 1,500, 2,000, 3,000, 3,500, 4,000, 4,500, 5,000] and analyze the variation of the model performance. Figure 4(a) shows the results of metrics for different training epochs, and Figure 4(b) shows the variation of metrics for different hidden units. With the increase of training epoch value, the variation of evaluation metrics stabilizes, and there is a turning point at 3,000. So we fix the training epoch at 3,000 and select the number of hidden cells from the set [32, 64, 100, 128, 256]. As shown in Figure 4(b), the model becomes stable when hidden units reach 128. Therefore, the training epoch is affirmed as 3000 and the number of hidden cells as 128. Besides, the time step is affirmed as 12, the batch size as 64, and the learning rate as 0.001.

4.3. Evaluation Metrics. In this work, we choose root mean square error (RMSE), mean absolute error (MAE), and linear regression coefficient of determination (R^2) as evaluation metrics. In real experiments, each evaluation metric represents a different meaning. RMSE and MAE represent the error between the predicted value and the actual value. R^2 reflects the fitting effect of the model, which measures the ability of prediction results to represent actual data by calculating correlation coefficients. As we all know, the smaller the error, the closer the predicted value is to the real value, and the higher the fitting degree, the better the prediction effect of the model. The specific calculation formulae are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_t - \hat{Y}_t)^2},$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_t - \hat{Y}_t|, \quad (9)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_t - \hat{Y}_t)^2}{\sum_{i=1}^n (Y_t - \bar{Y}_t)^2},$$

where Y_t represents the real OD information, \hat{Y}_t represents the predicted OD information, \bar{Y}_t represents the mean of the predicted OD information, and n is the number of nodes.

4.4. Results and Discussion. In this section, we select the following five baseline methods for comparison with the model in this paper, including three traditional time series

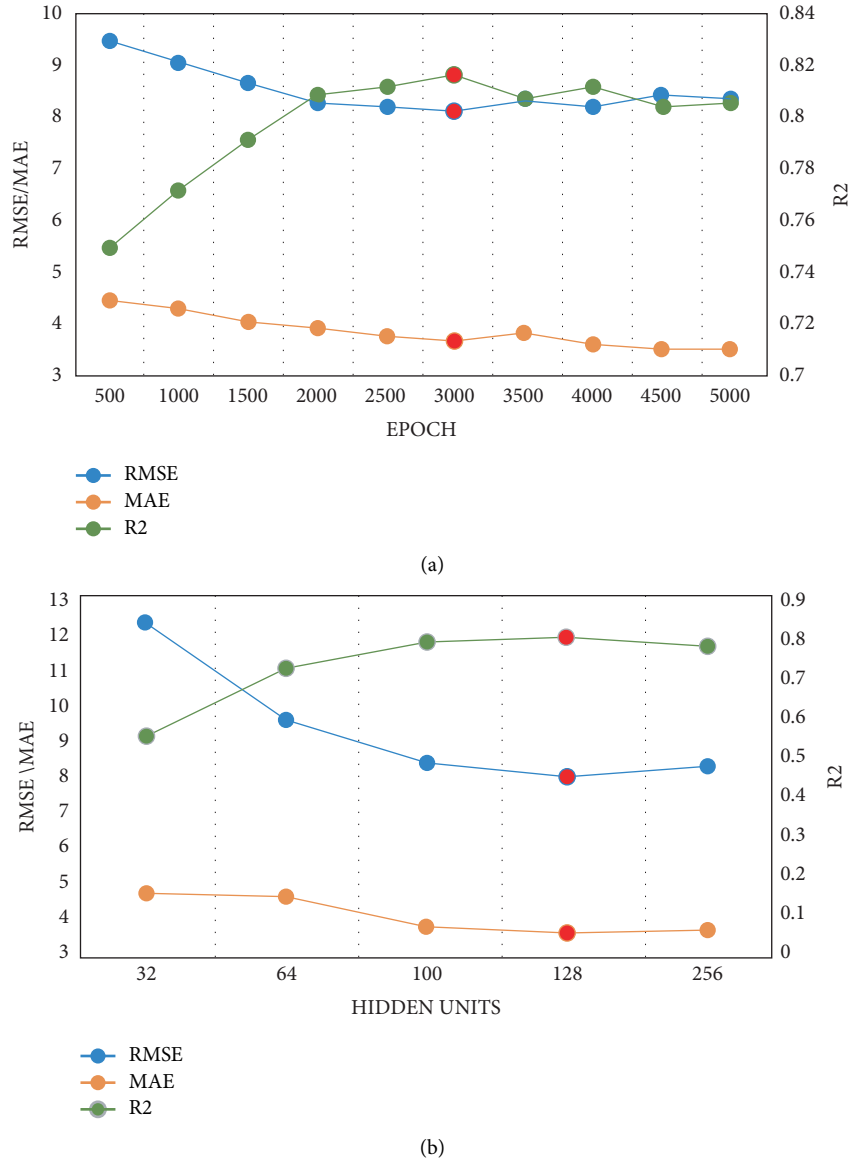


FIGURE 4: Parameter tuning results. (a) Variation of predicted performance under different epochs. (b) Variation of predicted performance under different hidden units.

models as well as two generalized deep learning models, namely (1) Historical Averaging model (HA), (2) Autoregressive Integrated Moving Average model (ARIMA), (3) Support Vector Regression model (SVR), (4) Graph Convolutional Network model (GCN), and (5) Gated Recurrent Unit model (GRU). To verify the effectiveness of the model in our work for the key OD flow prediction problem, we compare the evaluation metrics with some comparative experiments, as shown in Table 2.

We explore the performance of ST-VGCN for key OD pair flow prediction at 10-minute time granularity. We can find that ST-VGCN performs much better than HA model on all performance metrics, and its RMSE is reduced by approximately 26.15% and 0.6% compared to ARIMA and SVR models, respectively. GCN and GRU are both deep learning methods, but GCN only focuses on spatial

TABLE 2: Comparison of performances of different models.

Models	RMSE	MAE	R^2
HA	15.134	5.787	0.360
ARIMA	11.321	6.065	0.105
SVR	8.416	3.748	0.802
GCN	13.619	5.191	0.482
GRU	12.071	5.015	0.593
ST-VGCN	8.360	3.492	0.805

relations while GRU only focuses on temporal relations. ST-VGCN model takes both into account and its RMSE is reduced by about 38.62% and 30.74% compared to GCN and GRU models, respectively. Other indicators also perform significant improvements. Through the above analysis and comparison, we verify the improvement of the

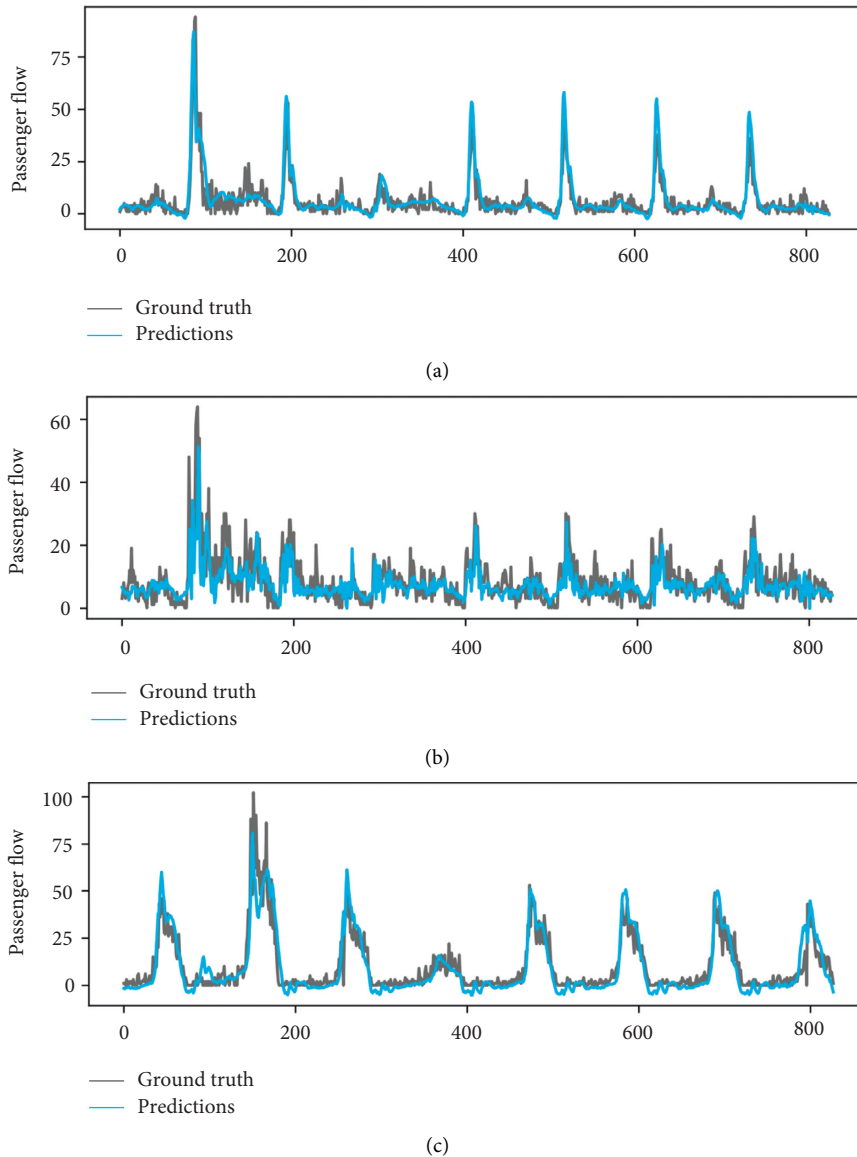


FIGURE 5: The real and predicted time series of three selected OD pairs.

proposed ST-VGCN model in three indicators. And we believe that the model in our work is effective. The comparison results validate the superiority of the proposed ST-VGCN model. For the 10-minute granularity of key OD pairs flow, the prediction result is visualized. As shown in Figure 5, three different passenger travel patterns can be seen. Figures 4(b) and 5(c) have explicit spike moments, and the spike moment in 5(c) is later than that in 5(a). After analysis, it is concluded that the nature of the OD pair of stations represented in Figure 5(a) is that station O is a residential area and station D is an office area. Figure 5(c) represents that station O is an office area and station D is a residential area, which is in line with the residents' commuting pattern. Figure 5(b) maintains a relatively flat trend, where station D is an airport station, so it maintains a certain level of passenger travel demand throughout the day.

As can be seen, our model can well model different passenger flow demands. Accurate predictions can provide effective reference information for passenger travel. At the same time, the modeling and analysis of the key OD pairs flow can assist the emergency response [33] when an unexpected event occurs, and reduce the continuous impact caused by the occurrence of emergencies.

5. Conclusions

Our work mainly defines and studies the key OD pair flow prediction problem and proposes a complete set of procedures for accomplishing the prediction. The key steps are to obtain key OD pairs, model the correlation between them, and use the ST-VGCN model to complete the prediction. The ST-VGCN model achieved the best prediction results on a real-world dataset when compared with five preexisting

models. Through experiments and analysis, we have obtained the following conclusions:

- (1) OD flow has the problem that the data is too sparse and its dimension is too high, yet most OD pairs have a small contribution to the overall passenger flow. So it is necessary to extract key OD pairs for research by setting a threshold filter.
- (2) The OD pairs exhibit certain correlations with each other due to the same nature of the origin-destination station and time variation pattern. It makes sense to consider the spatiotemporal characteristics of OD passenger flow when predicting.
- (3) The proposed ST-VGCN model can combine temporal and spatial information to improve the ability of prediction. By establishing a virtual graph, GCN is used to capture spatial properties. GRU is used to capture time series information.

Overall, research on key OD pair flow prediction can provide important insights for metro operation and management. In the future, the impact of multiple sources of data needs to be considered which can further improve the prediction ability of the model. It is also an open question worthy to be explored how to determine the threshold value for filtering key OD pairs more reasonably. Also, the problem of not being able to obtain real-time OD information due to trip duration is not explored in this study, and we will consider the problem of how to obtain real-time information in the next work.

Data Availability

Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data are not available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Beijing Municipal Natural Science Foundation (Grant No. L201015); the National Key R&D Program of China (Grant No. 2020YFC0833104); and the Green, Intelligent and Safe Mining of Coal Resources (Grant No. 52121003).

References

- [1] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 261–266.
- [2] Li. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, "Cross-layer defense methods for jamming-resistant cbtc systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2021.
- [3] Y. Li, Li. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled cbtc systems against mitm attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.
- [4] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 755–764, 2013.
- [5] J. Yang, Y. Zheng, K. P. Yan et al., "SPDNet: a real-time passenger detection method based on attention mechanism in subway station scenes," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7978644, 13 pages, 2021.
- [6] G. Filmon, "Habtemichael and Mecit Cetin. Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.
- [7] Li. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain-empowered cbtc system," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.
- [8] Z. Zhu, Bo. Peng, C. Xiong, and L. Zhang, "Short-term traffic flow prediction with linear conditional Gaussian bayesian network," *Journal of Advanced Transportation*, vol. 50, no. 6, pp. 1111–1123, 2016.
- [9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.
- [10] E. Zivot and J. Wang, "Vector Autoregressive Models for Multivariate Time Series," *Modeling Financial Time Series with S-Plus®*, Springer, New York, NY, USA, pp. 385–429, 2003.
- [11] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.
- [12] J. Yang and Z. Hou, "A wavelet analysis based ls-svm rail transit passenger flow prediction method," *China Railway Science*, vol. 34, no. 3, pp. 122–127, 2013.
- [13] P. Jiao, R. Li, T. Sun, Z. Hou, and A. Ibrahim, "Three revised kalman filtering models for short-term rail transit passenger flow prediction," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9717582, 10 pages, 2016.
- [14] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPACIAL '16, New York, NY, USA, 2016.
- [15] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI Press, San Francisco, CA, USA, February 2017.
- [16] Y. Liu, Z. Liu, and R. Jia, "Deeppf: a deep learning based architecture for metro passenger flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 18–34, 2019.
- [17] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2278–2288, 2019.
- [18] J. Yang, X. Dong, and S. Jin, "Metro passenger flow prediction model using attention-based neural network," *IEEE Access*, vol. 8, Article ID 30953, 2020.

- [19] J. Chen, L. Liu, H. Wu, J. Zhen, G. Li, and L. Lin, "Physical-virtual collaboration graph network for station-level metro ridership prediction," 2020, <https://arxiv.org/abs/2001.04889>.
- [20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering," 2016, <https://arxiv.org/abs/1606.09375>.
- [21] S. Guo, Y. Lin, F. Ning, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence*, AAAI Press, Honolulu, HI, USA, February 2019.
- [22] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, <https://arxiv.org/abs/1801.07455>.
- [23] L. Zhao, Y. Song, M. Deng, and H. Li, "Temporal graph convolutional network for urban traffic flow prediction method," 2018, <https://arxiv.org/abs/1811.05320>.
- [24] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial-temporal network for taxi origin-destination demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3875–3887, 2019.
- [25] J. Liu, F. Zheng, H. J. van Zuylen, and J. Li, "A dynamic od prediction approach for urban networks based on automatic number plate recognition data," *Transportation Research Procedia*, vol. 47, pp. 601–608, 2020.
- [26] F. Toqué, E. Côme, M. Khalil El Mahrsi, and L. Oukhellou, "Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1071–1076, IEEE, Rio de Janeiro, Brazil, November 2016.
- [27] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1227–1235, Anchorage AK USA, August 2019.
- [28] L. Shen, Z. Shao, Y. Yu, and X. M. Chen, "Hybrid approach combining modified gravity model and deep learning for short-term forecasting of metro transit passenger flows," *Transportation Research Record*, vol. 2675, no. 1, pp. 25–38, 2021.
- [29] J. Zhang, F. Chen, Z. Wang, and H. Liu, "Short-term origin-destination forecasting in urban rail transit based on attraction degree," *IEEE Access*, vol. 7, Article ID 133452, 2019.
- [30] J. Zhang, H. Che, F. Chen, W. Mae, and Z. He, "Short-term Prediction of Urban Rail Transit Origin-Destination Flow: A Channel-wise Attentive Split-Convolutional Neural Network Method," *Transportation Research Part C: Emerging Technologies*, vol. 124, 2020.
- [31] Y. Gong, Z. Li, J. Zhang, W. Liu, and Yu. Zheng, "Online spatio-temporal crowd flow distribution prediction for complex metro system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 865–880, 2022.
- [32] N. Peyman, H. N. Koutsopoulos, and J. Zhao, "Dynamic origin-destination prediction in urban rail systems: a multi-resolution spatio-temporal deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5106–5115, 2022.
- [33] J. Yang and Z. Hou, "A grey Markov based on large passenger flow real-time prediction model," *Journal of Beijing Jiaotong University*, vol. 37, no. 02, pp. 119–123, 2013.