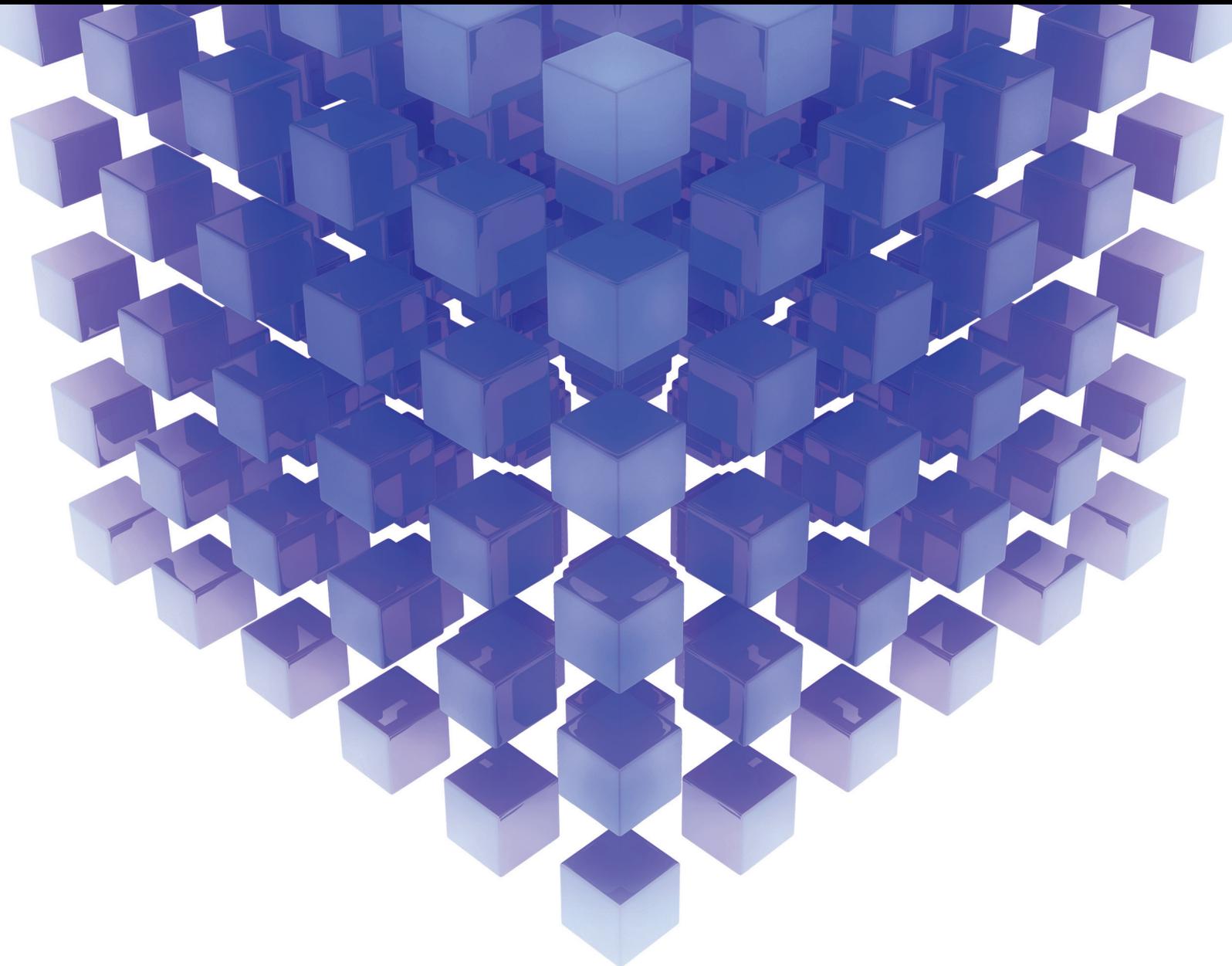


Mathematical Problems in Engineering

Computational Intelligence in Data-Driven Modelling and Its Engineering Applications

Lead Guest Editor: Qian Zhang

Guest Editors: Sarah Spurgeon, Li Xu, and Dingli Yu





Computational Intelligence in Data-Driven Modelling and Its Engineering Applications

Mathematical Problems in Engineering

Computational Intelligence in Data-Driven Modelling and Its Engineering Applications

Lead Guest Editor: Qian Zhang

Guest Editors: Sarah Spurgeon, Li Xu, and Dingli Yu



Copyright © 2018 Hindawi. 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.

Editorial Board

- Mohamed Abd El Aziz, Egypt
AITOUCHE Abdelouhab, France
Leonardo Acho, Spain
José A. Acosta, Spain
Daniela Addressi, Italy
Paolo Adesso, Italy
Claudia Adduce, Italy
Ramesh Agarwal, USA
Francesco Aggogeri, Italy
Juan C. Agüero, Australia
R Aguilar-López, Mexico
Tarek Ahmed-Ali, France
Elias Aifantis, USA
Muhammad N. Akram, Norway
Guido Ala, Italy
Andrea Alaimo, Italy
Reza Alam, USA
Nicholas Alexander, UK
Salvatore Alfonzetti, Italy
Mohammad D. Aliyu, Canada
Juan A. Almendral, Spain
José Domingo Álvarez, Spain
Cláudio Alves, Portugal
Juan P. Amezcua-Sanchez, Mexico
Lionel Amodeo, France
Sebastian Anita, Romania
Renata Archetti, Italy
Felice Arena, Italy
Sabri Arik, Turkey
Francesco Aristodemo, Italy
Fausto Arpino, Italy
Alessandro Arsie, USA
Edoardo Artioli, Italy
Fumihiko Ashida, Japan
Farhad Aslani, Australia
Mohsen Asle Zaem, USA
Romain Aubry, USA
Matteo Aureli, USA
Richard I. Avery, USA
Viktor Avrutin, Germany
Francesco Aymerich, Italy
Sajad Azizi, Belgium
Michele Baccocchi, Italy
Seungik Baek, USA
Adil Bagirov, Australia
Khaled Bahlali, France
Laurent Bako, France
Pedro Balaguer, Spain
Stefan Balint, Romania
Ines Tejado Balsera, Spain
Alfonso Banos, Spain
Jerzy Baranowski, Poland
Roberto Baratti, Italy
Andrzej Bartoszewicz, Poland
David Bassir, France
Chiara Bedon, Italy
Azeddine Beghdadi, France
Denis Benasciutti, Italy
Ivano Benedetti, Italy
Rosa M. Benito, Spain
Elena Benvenuti, Italy
Giovanni Berselli, Italy
Giorgio Besagni, Italy
Michele Betti, Italy
Jean-Charles Beugnot, France
Pietro Bia, Italy
Carlo Bianca, France
Simone Bianco, Italy
Vincenzo Bianco, Italy
Vittorio Bianco, Italy
Gennaro N. Bifulco, Italy
David Bigaud, France
Antonio Bilotta, Italy
Paul Bogdan, USA
Guido Bolognesi, UK
Rodolfo Bontempo, Italy
Alberto Borboni, Italy
Paolo Boscariol, Italy
Daniela Boso, Italy
Guillermo Botella-Juan, Spain
Boulaïd Boulkroune, Belgium
Fabio Bovenga, Italy
Francesco Braghin, Italy
Ricardo Branco, Portugal
Maurizio Brocchini, Italy
Julien Bruchon, France
Matteo Bruggi, Italy
Michele Brun, Italy
Vasilis Burganos, Greece
Tito Busani, USA
Raquel Caballero-Águila, Spain
Filippo Cacace, Italy
Pierfrancesco Cacciola, UK
Salvatore Caddemi, Italy
Roberto Caldelli, Italy
Alberto Campagnolo, Italy
Eric Campos-Canton, Mexico
Marko Canadija, Croatia
Salvatore Cannella, Italy
Francesco Cannizzaro, Italy
Javier Cara, Spain
Ana Carpio, Spain
Caterina Casavola, Italy
Sara Casciati, Italy
Federica Caselli, Italy
Carmen Castillo, Spain
Inmaculada T. Castro, Spain
Miguel Castro, Portugal
Giuseppe Catalanotti, UK
Nicola Caterino, Italy
Alberto Cavallo, Italy
Gabriele Cazzulani, Italy
Luis Cea, Spain
Miguel Cerrolaza, Venezuela
M. Chadli, France
Gregory Chagnon, France
Ludovic Chamoin, France
Ching-Ter Chang, Taiwan
Qing Chang, USA
Michael J. Chappell, UK
Kacem Chehdi, France
Peter N. Cheimets, USA
Xinkai Chen, Japan
Luca Chiapponi, Italy
Francisco Chicano, Spain
Nicholas Chileshe, Australia
Adrian Chmielewski, Poland
Ioannis T. Christou, Greece
Hung-Yuan Chung, Taiwan
Simone Cinquemani, Italy
Roberto G. Citarella, Italy
Joaquim Ciurana, Spain

John D. Clayton, USA
Francesco Clementi, Italy
Piero Colajanni, Italy
Giuseppina Colicchio, Italy
Vassilios Constantoudis, Greece
Enrico Conte, Italy
Francesco Conte, Italy
Alessandro Contento, USA
Mario Cools, Belgium
José A. Correia, Portugal
Jean-Pierre Corriou, France
J.-C. Cortés, Spain
Carlo Cosentino, Italy
Paolo Crippa, Italy
Andrea Crivellini, Italy
Frederico R. B. Cruz, Brazil
Erik Cuevas, Mexico
Maria C. Cunha, Portugal
Peter Dabnichki, Australia
Luca D'Acierno, Italy
Weizhong Dai, USA
Andrea Dall'Asta, Italy
Purushothaman Damodaran, USA
Farhang Daneshmand, Canada
Giuseppe D'Aniello, Italy
Sergey N. Dashkovskiy, Germany
Fabio De Angelis, Italy
Samuele De Bartolo, Italy
Abílio De Jesus, Portugal
Pietro De Lellis, Italy
Alessandro De Luca, Italy
Stefano de Miranda, Italy
Filippo de Monte, Italy
M. do Rosário de Pinho, Portugal
Michael Defoort, France
Alessandro Della Corte, Italy
Xavier Delorme, France
Laurent Dewasme, Belgium
Angelo Di Egidio, Italy
Roberta Di Pace, Italy
Ramón I. Diego, Spain
Yannis Dimakopoulos, Greece
Zhengtao Ding, UK
M. Djemai, France
Alexandre B. Dolgui, France
Georgios Dounias, Greece
Florent Duchaine, France
George S. Dulikravich, USA
Bogdan Dumitrescu, Romania
Horst Ecker, Austria
Saeed Eftekhar Azam, USA
Ahmed El Hajjaji, France
Antonio Elipe, Spain
Fouad Erchiqui, Canada
Anders Eriksson, Sweden
R. Emre Erkmen, Australia
Gilberto Espinosa-Paredes, Italy
Leandro F. F. Miguel, Brazil
Andrea L. Facci, Italy
Giacomo Falcucci, Italy
Giovanni Falsone, Italy
Hua Fan, China
Nicholas Fantuzzi, Italy
Yann Favennec, France
Fiorenzo A. Fazzolari, UK
Giuseppe Fedele, Italy
Roberto Fedele, Italy
Arturo J. Fernández, Spain
Jesus M. Fernandez Oro, Spain
Massimiliano Ferraioli, Italy
Massimiliano Ferrara, Italy
Francesco Ferrise, Italy
Eric Feulvarch, France
Barak Fishbain, Israel
S. Douwe Flapper, Netherlands
Thierry Floquet, France
Eric Florentin, France
Alessandro Formisano, Italy
Francesco Franco, Italy
Elisa Francomano, Italy
Tomonari Furukawa, USA
Juan C. G. Prada, Spain
Mohamed Gadala, Canada
Matteo Gaeta, Italy
Mauro Gaggero, Italy
Zoran Gajic, USA
Erez Gal, Israel
Jaime Gallardo-Alvarado, Mexico
Ugo Galvanetto, Italy
Akemi Gálvez, Spain
Rita Gamberini, Italy
Maria L. Gandarias, Spain
Arman Ganji, Canada
Zhiwei Gao, UK
Zhong-Ke Gao, China
Giovanni Garcea, Italy
Luis Rodolfo Garcia Carrillo, USA
Jose M. Garcia-Aznar, Spain
Akhil Garg, China
Alessandro Gasparetto, Italy
Oleg V. Gendelman, Israel
Stylios Georgantzinos, Greece
Fotios Georgiades, UK
Mergen H. Ghayesh, Australia
Georgios I. Giannopoulos, Greece
Agathoklis Giaralis, UK
Pablo Gil, Spain
Anna M. Gil-Lafuente, Spain
Ivan Giorgio, Italy
Gaetano Giunta, Luxembourg
Alessio Gizzi, Italy
Jefferson L.M.A. Gomes, UK
Emilio Gómez-Déniz, Spain
Antonio M. Gonçalves de Lima, Brazil
David González, Spain
Chris Goodrich, USA
Rama S. R. Gorla, USA
Kannan Govindan, Denmark
Antoine Grall, France
George A. Gravvanis, Greece
Fabrizio Greco, Italy
David Greiner, Spain
Simonetta Grilli, Italy
Jason Gu, Canada
Federico Guarracino, Italy
Michele Guida, Italy
José L. Guzmán, Spain
Quang Phuc Ha, Australia
Petr Hájek, Czech Republic
Zhen-Lai Han, China
Thomas Hanne, Switzerland
Mohammad A. Hariri-Ardebili, USA
Xiao-Qiao He, China
Luca Heltai, Italy
Nicolae Herisanu, Romania
Alfredo G. Hernández-Díaz, Spain
M.I. Herreros, Spain
Eckhard Hitzer, Japan
Paul Honeine, France
Jaromir Horacek, Czech Republic
Muneo Hori, Japan

András Horváth, Italy
 S. Hassan Hosseinnia, Netherlands
 Mengqi Hu, USA
 Gordon Huang, Canada
 Sajid Hussain, Canada
 Asier Ibeas, Spain
 Orest V. Iftime, Netherlands
 Przemyslaw Ignaciuk, Poland
 Giacomo Innocenti, Italy
 Emilio Insfran Pelozo, Spain
 Alessio Ishizaka, UK
 Nazrul Islam, USA
 Benoit Jung, France
 Benjamin Ivorra, Spain
 Payman Jalali, Finland
 Mahdi Jalili, Australia
 Łukasz Jankowski, Poland
 Samuel N. Jator, USA
 J. C. Jauregui-Correa, Mexico
 Reza Jazar, Australia
 Khalide Jbilou, France
 Piotr Jędrzejowicz, Poland
 Isabel S. Jesus, Portugal
 Linni Jian, China
 Bin Jiang, China
 Zhongping Jiang, USA
 Emilio Jiménez Macías, Spain
 Ningde Jin, China
 Xiaoliang Jin, USA
 Liang Jing, Canada
 Dylan F. Jones, UK
 Palle E. Jorgensen, USA
 Vyacheslav Kalashnikov, Mexico
 Tamas Kalmar-Nagy, Hungary
 Tomasz Kapitaniak, Poland
 Julius Kaplunov, UK
 Haranath Kar, India
 K. Karamanos, Belgium
 Krzysztof Kecik, Poland
 Jean-Pierre Kenne, Canada
 Chaudry M. Khalique, South Africa
 Do Wan Kim, Republic of Korea
 Nam-Il Kim, Republic of Korea
 Jan Koci, Czech Republic
 Ioannis Kostavelis, Greece
 Sotiris B. Kotsiantis, Greece
 Manfred Krafczyk, Germany
 Frederic Kratz, France
 Petr Krysl, USA
 Krzysztof S. Kulpa, Poland
 Shailesh I. Kundalwal, India
 Jurgen Kurths, Germany
 Cedrick A. K. Kwuimy, USA
 Kyandoghere Kyamakya, Austria
 Davide La Torre, Italy
 Risto Lahdelma, Finland
 Hak-Keung Lam, UK
 Giovanni Lancioni, Italy
 Jimmy Lauber, France
 Antonino Laudani, Italy
 Hervé Laurent, France
 Aimé Lay-Ekuakille, Italy
 Nicolas J. Leconte, France
 Dimitri Lefebvre, France
 Eric Lefevre, France
 Marek Lefik, Poland
 Yaguo Lei, China
 Kauko Leiviskä, Finland
 Thibault Lemaire, France
 Roman Lewandowski, Poland
 Chen-Feng Li, China
 Jian Li, USA
 En-Qiang Lin, USA
 Zhiyun Lin, China
 Peide Liu, China
 Peter Liu, Taiwan
 Wanquan Liu, Australia
 Bonifacio Llamazares, Spain
 Alessandro Lo Schiavo, Italy
 Jean Jacques Loiseau, France
 Francesco Lolli, Italy
 Paolo Lonetti, Italy
 Sandro Longo, Italy
 António M. Lopes, Portugal
 Sebastian López, Spain
 Pablo Lopez-Crespo, Spain
 Luis M. López-Ochoa, Spain
 Ezequiel López-Rubio, Spain
 Vassilios C. Loukopoulos, Greece
 Jose A. Lozano-Galant, Spain
 Helen Lu, Australia
 Gabriel Luque, Spain
 Valentin Lychagin, Norway
 Antonio Madeo, Italy
 José María Maestre, Spain
 Alessandro Magnani, Italy
 Fazal M. Mahomed, South Africa
 Noureddine Manamanni, France
 Paolo Manfredi, Italy
 Didier Maquin, France
 Giuseppe Carlo Marano, Italy
 Damijan Markovic, France
 Francesco Marotti de Sciarra, Italy
 Rui Cunha Marques, Portugal
 Luis Martínez, Spain
 Rodrigo Martinez-Bejar, Spain
 Guiomar Martín-Herrán, Spain
 Denizar Cruz Martins, Brazil
 Benoit Marx, France
 Elio Masciari, Italy
 Franck Massa, France
 Paolo Massioni, France
 Alessandro Mauro, Italy
 Fabio Mazza, Italy
 Laura Mazzola, Italy
 Driss Mehdi, France
 Roderick Melnik, Canada
 Pasquale Memmolo, Italy
 Xiangyu Meng, USA
 Jose Merodio, Spain
 Alessio Merola, Italy
 Mahmoud Mesbah, Iran
 Luciano Mescia, Italy
 Laurent Mevel, France
 Mariusz Michta, Poland
 Aki Mikkola, Finland
 Giovanni Minafò, Italy
 Hiroyuki Mino, Japan
 Pablo Mira, Spain
 Dimitrios Mitsotakis, New Zealand
 Vito Mocella, Italy
 Sara Montagna, Italy
 Roberto Montanini, Italy
 Francisco J. Montáns, Spain
 Luiz H. A. Monteiro, Brazil
 Gisele Mophou, France
 Rafael Morales, Spain
 Marco Morandini, Italy
 Javier Moreno-Valenzuela, Mexico
 Simone Morganti, Italy
 Caroline Mota, Brazil

Aziz Moukrim, France
Dimitris Mourtzis, Greece
Emiliano Mucchi, Italy
Josefa Mula, Spain
Jose J. Muñoz, Spain
Giuseppe Muscolino, Italy
Marco Mussetta, Italy
Hakim Naceur, France
Alessandro Naddeo, Italy
Hassane Naji, France
M. Nakano-Miyatake, Mexico
Keivan Navaie, UK
AMA Neves, Portugal
Luís C. Neves, UK
Dong Ngoduy, New Zealand
Nhon Nguyen-Thanh, Singapore
Tatsushi Nishi, Japan
Xesús Nogueira, Spain
Ben T. Nohara, Japan
Mohammed Nouari, France
Mustapha Nourelfath, Canada
Włodzimierz Ogryczak, Poland
Roger Ohayon, France
Krzysztof Okarma, Poland
Mitsuhiro Okayasu, Japan
Alberto Olivares, Spain
Enrique Onieva, Spain
Calogero Orlando, Italy
A. Ortega-Moñux, Spain
Sergio Ortobelli, Italy
Naohisa Otsuka, Japan
Erika Ottaviano, Italy
Pawel Packo, Poland
Arturo Pagano, Italy
Alkis S. Paipetis, Greece
Roberto Palma, Spain
Alessandro Palmeri, UK
Pasquale Palumbo, Italy
Jürgen Pannek, Germany
Elena Panteley, France
Achille Paolone, Italy
George A. Papakostas, Greece
Xosé M. Pardo, Spain
Vicente Parra-Vega, Mexico
Manuel Pastor, Spain
Petr Páta, Czech Republic
Pubudu N. Pathirana, Australia

Surajit Kumar Paul, India
Sitek Paweł, Poland
Luis Payá, Spain
Alexander Paz, USA
Igor Pažanin, Croatia
Libor Pekař, Czech Republic
Francesco Pellicano, Italy
Marcello Pellicciari, Italy
Haipeng Peng, China
Mingshu Peng, China
Zhengbiao Peng, Australia
Zhi-ke Peng, China
Marzio Pennisi, Italy
Maria Patrizia Pera, Italy
Matjaz Perc, Slovenia
A. M. Bastos Pereira, Portugal
Ricardo Perera, Spain
Francesco Pesavento, Italy
Ivo Petras, Slovakia
Francesco Petrini, Italy
Lukasz Pieczonka, Poland
Dario Piga, Switzerland
Paulo M. Pimenta, Brazil
Antonina Pirrotta, Italy
Marco Pizzarelli, Italy
Vicent Pla, Spain
Javier Plaza, Spain
Kemal Polat, Turkey
Dragan Poljak, Croatia
Jorge Pomares, Spain
Sébastien Poncet, Canada
Volodymyr Ponomaryov, Mexico
Jean-Christophe Ponsart, France
Mauro Pontani, Italy
Cornelio Posadas-Castillo, Mexico
Francesc Pozo, Spain
Christopher Pretty, New Zealand
Luca Pugi, Italy
Krzysztof Puszynski, Poland
Giuseppe Quaranta, Italy
Vitomir Racic, Italy
Jose Ragot, France
Carlo Rainieri, Italy
K. Ramamani Rajagopal, USA
Ali Ramazani, USA
Higinio Ramos, Spain
Alain Rassineux, France

S.S. Ravindran, USA
Alessandro Reali, Italy
Jose A. Reinoso, Spain
Oscar Reinoso, Spain
Fabrizio Renno, Italy
Nidhal Rezg, France
Ricardo Riaza, Spain
Francesco Riganti-Fulginei, Italy
Gerasimos Rigatos, Greece
Francesco Ripamonti, Italy
Jorge Rivera, Mexico
Eugenio Roanes-Lozano, Spain
Bruno G. M. Robert, France
Ana Maria A. C. Rocha, Portugal
José Rodellar, Spain
Luigi Rodino, Italy
Rosana Rodríguez López, Spain
Ignacio Rojas, Spain
Alessandra Romolo, Italy
Debasish Roy, India
Gianluigi Rozza, Italy
Rubén Ruiz, Spain
Antonio Ruiz-Cortes, Spain
Ivan D. Rukhlenko, Australia
Mazen Saad, France
Kishin Sadarangani, Spain
Andrés Sáez, Spain
Mehrddad Saif, Canada
John S. Sakellariou, Greece
Salvatore Salamone, USA
Vicente Salas, Spain
Jose Vicente Salcedo, Spain
Nunzio Salerno, Italy
Miguel A. Salido, Spain
Roque J. Saltarén, Spain
Alessandro Salvini, Italy
Sylwester Samborski, Poland
Ramon Sancibrian, Spain
Giuseppe Sanfilippo, Italy
Miguel A. F. Sanjuan, Spain
Vittorio Sansalone, France
José A. Sanz-Herrera, Spain
Nickolas S. Sapidis, Greece
Evangelos J. Sapountzakis, Greece
Luis Saucedo-Mora, Spain
Marcelo A. Savi, Brazil
Andrey V. Savkin, Australia

Roberta Sburlati, Italy
Gustavo Scaglia, Argentina
Thomas Schuster, Germany
Oliver Schütze, Mexico
Lotfi Senhadji, France
Junwon Seo, USA
Joan Serra-Sagrasta, Spain
Gerardo Severino, Italy
Ruben Sevilla, UK
Stefano Sfarra, Italy
Mohamed Shaat, Egypt
Mostafa S. Shadloo, France
Leonid Shaikhet, Israel
Hassan M. Shanechi, USA
Bo Shen, Germany
Suzanne M. Shontz, USA
Babak Shotorban, USA
Zhan Shu, UK
Nuno Simões, Portugal
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
Jussi Sopanen, Finland
Marco Spadini, Italy
Bernardo Spagnolo, Italy
Paolo Spagnolo, Italy
Ruben Specogna, Italy
Vasilios Spitas, Greece
Sri Sridharan, USA
Ivanka Stamova, USA
Rafał Stanisławski, Poland
Florin Stoican, Romania
Salvatore Strano, Italy
Yakov Strelniker, Israel
Guang-Yong Sun, China
Ning Sun, China
Sergey A. Suslov, Australia
Thomas Svensson, Sweden
Andrzej Swierniak, Poland
Andras Szekrenyes, Hungary
Kumar K. Tamma, USA
Yang Tang, Germany

Hafez Tari, USA
Alessandro Tasora, Italy
Sergio Teggi, Italy
Ana C. Teodoro, Portugal
Tai Thai, Australia
Alexander Timokha, Norway
Gisella Tomasini, Italy
Francesco Tornabene, Italy
Antonio Tornambe, Italy
Javier Martinez Torres, Spain
Mariano Torrisi, Italy
George Tsiatas, Greece
Antonios Tsourdos, UK
Federica Tubino, Italy
Nerio Tullini, Italy
Andrea Tundis, Italy
Emilio Turco, Italy
Vladimir Turetsky, Israel
Mustafa Tutar, Spain
Ilhan Tuzcu, USA
Efstratios Tzirtzilakis, Greece
Filippo Ubertini, Italy
Francesco Ubertini, Italy
Mohammad Uddin, Australia
Hassan Ugail, UK
Giuseppe Vairo, Italy
Eusebio Valero, Spain
Pandian Vasant, Malaysia
Marcello Vasta, Italy
Carlos-Renato Vázquez, Mexico
M. E. Vázquez-Méndez, Spain
Josep Vehi, Spain
Martin Velasco Villa, Mexico
K. C. Veluvolu, Republic of Korea
Fons J. Verbeek, Netherlands
Franck J. Vernerey, USA
Georgios Veronis, USA
Vincenzo Vespri, Italy
Renato Vidoni, Italy
V. Vijayaraghavan, Australia
Anna Vila, Spain
Rafael J. Villanueva, Spain
Francisco R. Villatoro, Spain
Uchechukwu E. Vincent, UK
Gareth A. Vio, Australia
Francesca Vipiana, Italy
Stanislav Vítek, Czech Republic

Thuc P. Vo, UK
Jan Vorel, Czech Republic
Michael Vynnycky, Sweden
Hao Wang, USA
Liliang Wang, UK
Shuming Wang, China
Yongqi Wang, Germany
Roman Wan-Wendner, Austria
Jarosław Wąs, Poland
P.H. Wen, UK
Waldemar T. Wójcik, Poland
Changzhi Wu, Australia
Desheng D. Wu, Sweden
Hong-Yu Wu, USA
Yuqiang Wu, China
Michalis Xenos, Greece
Guangming Xie, China
Xue-Jun Xie, China
Gen Q. Xu, China
Hang Xu, China
Joseph J. Yame, France
Xinggong Yan, UK
Mijia Yang, USA
Yongheng Yang, Denmark
Luis J. Yebra, Spain
Peng-Yeng Yin, Taiwan
Qin Yuming, China
Elena Zaitseva, Slovakia
Arkadiusz Zak, Poland
Daniel Zaldivar, Mexico
Francesco Zammori, Italy
Vittorio Zampoli, Italy
Rafal Zdunek, Poland
Ibrahim Zeid, USA
Huaguang Zhang, China
Kai Zhang, China
Qingling Zhang, China
Xian-Ming Zhang, Australia
Xuping Zhang, Denmark
Zhao Zhang, China
Yifan Zhao, UK
Jian G. Zhou, UK
Quanxin Zhu, China
Mustapha Zidi, France
Gaetano Zizzo, Italy
Zhixiang Zou, Germany

Contents

Computational Intelligence in Data-Driven Modelling and Its Engineering Applications

Qian Zhang , Sarah Spurgeon, Li Xu , and Dingli Yu
Editorial (2 pages), Article ID 2576239, Volume 2018 (2018)

A Semisupervised Framework for Automatic Image Annotation Based on Graph Embedding and Multiview Nonnegative Matrix Factorization

Hongwei Ge , Zehang Yan, Jing Dou, Zhen Wang , and ZhiQiang Wang
Research Article (11 pages), Article ID 5987906, Volume 2018 (2018)

Wind Power Prediction Based on Nonlinear Partial Least Square

Qian Wang , Yang Lei , and Hui Cao 
Research Article (9 pages), Article ID 6829274, Volume 2018 (2018)

Multisource Data-Driven Modeling Method for Estimation of Intercity Trip Distribution

Yilin Li, Haiquan Wang , Jiejie Zhao, and Bowen Du 
Research Article (11 pages), Article ID 8948676, Volume 2018 (2018)

An Improved Method for Cross-Project Defect Prediction by Simplifying Training Data

Peng He , Yao He, Lvjun Yu, and Bing Li 
Research Article (18 pages), Article ID 2650415, Volume 2018 (2018)

J Wave Autodetection Using Analytic Time-Frequency Flexible Wavelet Transformation Applied on ECG Signals

Deng-ao Li , Jie Zhou , Jumin Zhao , and Xinyan Liu
Research Article (11 pages), Article ID 6791405, Volume 2018 (2018)

Predicting Traffic Dynamics with Driver Response Model for Proactive Variable Speed Limit Control Algorithm

Huixuan Ye, Lili Tu, and Jie Fang 
Research Article (10 pages), Article ID 6181756, Volume 2018 (2018)

Statistical Process Monitoring with Biogeography-Based Optimization Independent Component Analysis

Xiangshun Li , Di Wei, Cheng Lei, Zhiang Li, and Wenlin Wang
Research Article (14 pages), Article ID 1729612, Volume 2018 (2018)

Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets

Xingyu Zhou , Zhisong Pan , Guyu Hu , Siqi Tang, and Cheng Zhao
Research Article (11 pages), Article ID 4907423, Volume 2018 (2018)

Adaptive Learning Based Tracking Control of Marine Vessels with Prescribed Performance

Zhao Xu , Shuzhi Sam Ge, Changhua Hu , and Jinwen Hu
Research Article (9 pages), Article ID 2595721, Volume 2018 (2018)

The Application of Group Theory in Communication Operation Pipeline System

Jianqing Zhang, Feng Xiong , and Jinsheng Kang 
Research Article (10 pages), Article ID 9507823, Volume 2018 (2018)

Modeling and Simulation of Gas Emission Based on Recursive Modified Elman Neural Network

Lin Wei , Yongqing Wu, Hua Fu, and Yuping Yin

Research Article (10 pages), Article ID 9013839, Volume 2018 (2018)

A Systematic Optimization Design Method for Complex Mechatronic Products Design and Development

Jie Jiang, Guofu Ding, Jian Zhang , Yisheng Zou, and Shengfeng Qin 

Research Article (14 pages), Article ID 3159637, Volume 2018 (2018)

Editorial

Computational Intelligence in Data-Driven Modelling and Its Engineering Applications

Qian Zhang ¹, **Sarah Spurgeon**,² **Li Xu** ³, and **Dingli Yu**¹

¹Liverpool John Moores University, Liverpool, UK

²University College London, London, UK

³Zhejiang University, Hangzhou, Zhejiang, China

Correspondence should be addressed to Qian Zhang; q.zhang@ljmu.ac.uk

Received 6 September 2018; Accepted 6 September 2018; Published 30 September 2018

Copyright © 2018 Qian Zhang 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.

Modern engineering systems show increasing complexity due to their high nonlinearity and large disturbances and uncertainties introduced into the systems. In many circumstances, the conventional mathematical models that can accurately describe these complex systems and can be exploited in real-life applications, such as differential equations or statistical models, do not exist. However, with the fast development of advanced sensing, measurement, and data collection technologies, large amounts of data that represent input-output relationships of the systems become available. This makes data-driven modelling (DDM) possible and practical.

Data-driven modelling aims at information extraction from data and is normally used to elicit numerical predictive models with good generalisation ability, which can be viewed as regression problems in mathematics. It analyses the data that characterise a system to find relationships among system state variables (input, internal, and output variables) without taking into account explicit knowledge about physical behaviours. Many paradigms utilised in DDM have been established based on statistics and/or computational intelligence. For instance, artificial neural networks (ANNs) and fuzzy rule-based systems (FRBSs) serve as fundamental model frameworks, which are alternatives to statistical inference methods. Evolutionary algorithms (EAs), swarm intelligence (SI), and machine learning (ML) methods provide learning and optimisation abilities for calibrating and improving the intelligent or statistical models. In recent years, DDM has found widespread applications, ranging across machinery manufacturing, materials processing, power and energy systems, transport, and so forth.

This special issue intends to bring together the state-of-the-art research, application, and review of DDM techniques. It aims at not only stimulating deep insights on computational intelligence approaches in DDM but also promoting their potential applications in complex engineering problems.

This special issue has received 61 manuscripts and only 12 high-quality papers have been accepted and published (20% acceptance rate). The accepted papers involve a variety of data-driven modelling and data analytics techniques and contribute to a wide range of application areas, including transportation, environment, telecommunication, automatic control, product design, and finance. A brief introduction for each contribution is provided in the following paragraphs.

L. Wei et al. proposed a new architecture for Elman neural network (ENN), which was named recursive modified Elman neural network (RMENN), to enhance the model's approximation ability. The stability of the model was proven and its learning algorithm was also provided. The advised network has been applied to predict the gas emission relating to Kailuan mining group, China. It was shown that RMENN outperformed ENN in prediction accuracy and convergence speed.

Q. Wang et al. combined the partial least square (PLS) approach with the back-propagation neural network (BPNN) and the radial basis function neural network (RBFNN) to predict short-term wind power. The proposed method was validated using the data collected from a wind farm of Northwest China. The experimental results showed that the new method performed better than other conventional methods, such as the single PLS, the single RBFNN, or support vector machine (SVM).

An adaptive control method was designed by Z. Xu et al. to manipulate marine vessels with unknown dynamics and unknown external disturbances. In the method, an RBFNN was used to approximate the dominant dynamics and disturbances. The simulation demonstrated that the proposed controller outperformed the original controller in performance and made the tracking error stay in the predefined bounds.

A paradigm for mechatronic products design was introduced by J. Jiang et al. The work involved design space reduction, data-driven system modelling, and optimal design of parameters. In modelling, multiple surrogate models were developed using BPNN. The proposed strategy was used into the design of high-speed trains considering seven performance indices and was validated to be effective through simulation experiments.

X. Zhou et al. applied a deep learning network, the generative adversarial network (GAN) to predict the price change for stock market. The advised model, which was named GAN-FD, was validated using the high-frequency data from Chinese stock market. It was compared with other modelling strategies, including Long Short-Term Memory (LSTM), ANN, and SVM, and showed better performance.

X. Li et al. proposed a new version of independent component analysis (ICA) for data-driven fault detection. It utilised a novel biogeography-based optimisation (BBO) method to replace the normal Newton iteration method. The new ICA, named BBO-IAC, was validated using an actuator diagnosis problem. It showed that BBO-IAC could detect faults with better efficiency than other methods, such as FastICA and PSO-ICA.

A cross-project defect prediction paradigm with an improved training data selection mechanism was proposed by P. He et al. The method was tested using 15 datasets from 14 open-source projects. The experimental results showed that the proposed method could design more effective defect predictors than some other similar approaches.

A data-driven approach was designed by Y. Li et al. to approximate the intercity trip distribution. In this work, a Poisson model was used to estimate the correlation between trip distribution and city features and k-means clustering was employed in feature analysis and interpretation. The approach was implemented to the case of analysing Chinese highway trips, which involved 17 cities of Shandong Province.

H. Ye et al. proposed a dynamic driver response model for variable speed limit (VSL) control using linear and nonlinear regression methods. The elicited model was then utilised to improve the VSL control algorithm using an optimisation technique. A simulation study was carried out based on the field data of a freeway corridor allocated in Whitemud Drive, Edmonton, Alberta, Canada, and showed the new model and control algorithm worked effectively.

A semi-supervised framework for automatic image annotation with multi-label images was suggested by H. Ge et al. The method employed graph embedding and multi-view nonnegative matrix methods for feature fusion and dimension deduction. It then employed a k-nearest neighbour (KNN) based method for image annotation. The framework

was validated using experiments and showed good performance in accuracy and efficiency.

J. Zhang et al. applied group theory in telecom operation systems. The model of pipeline entity group was constructed for information transmission. The equations of network traffic were elicited according to the matrix of pipeline network and flux conservation principle. An optimisation method was then suggested to improve the general information network system.

D. Li et al. designed an approach to detect J wave from electrocardiogram (ECG) signals, which was named analytic time-frequency flexible wavelet transformation (ATFFWT). Fuzzy entropy (FE) was then employed to capture hidden but useful information and least squares-support vector machine (LS-SVM) was employed for classification. Some experiments demonstrated that the approach outperformed other similar methods.

Conflicts of Interest

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

Acknowledgments

The guest editors sincerely thank all the authors for their quality contributions to this special issue and thank all the reviewers for their valuable comments and suggestions. The lead guest editor would also like to express the deepest gratitude to other co-editors for their great support and cooperation throughout the development of the special issue.

*Qian Zhang
Sarah Spurgeon
Li Xu
Dingli Yu*

Research Article

A Semisupervised Framework for Automatic Image Annotation Based on Graph Embedding and Multiview Nonnegative Matrix Factorization

Hongwei Ge ¹, Zehang Yan,¹ Jing Dou,¹ Zhen Wang ,² and ZhiQiang Wang¹

¹School of Computer Science and Technology, Dalian University of Technology, Dalian 116023, China

²School of Mathematical Science, Dalian University of Technology, Dalian 116023, China

Correspondence should be addressed to Hongwei Ge; hwge@dlut.edu.cn

Received 25 February 2018; Accepted 4 June 2018; Published 27 June 2018

Academic Editor: Qian Zhang

Copyright © 2018 Hongwei Ge 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.

Automatic image annotation is for more accurate image retrieval and classification by assigning labels to images. This paper proposes a semisupervised framework based on graph embedding and multiview nonnegative matrix factorization (GENMF) for automatic image annotation with multilabel images. First, we construct a graph embedding term in the multiview NMF based on the association diagrams between labels for semantic constraints. Then, the multiview features are fused and dimensions are reduced based on multiview NMF algorithm. Finally, image annotation is achieved by using the new features through a KNN-based approach. Experiments validate that the proposed algorithm has achieved competitive performance in terms of accuracy and efficiency.

1. Introduction

The advent of Internet age brings the explosive growth of image resources. Although managing and retrieving images by semantic tags is a common and effective way, there are still a large number of untagged or not fully tagged images. However, it is not easy to carry out manual annotation regarding the cost of human resources and the semantic nuances of annotation under the background of various cultures, religions, and languages. Moreover, the cognition bias caused by subjectivity could induce semantic discrepancies as well. Thus, how to design an efficient automatic image annotation algorithm to provide accurate labels for untagged images has been an urgent problem.

Automatic image annotation (AIA) refers to the process that computers automatically provide one or more semantic tags that can reflect the content of a specific image through algorithms. It is a mapping from images to semantic concepts, namely, the process of understanding images. Image annotation is based on image feature representations, and features utilized in different tasks have different representation abilities [1–3]. For example, global color and texture features

have been successfully used in retrieving similar images [4], while local structure features perform well in tasks of object classification and matching [5, 6]. In general, features that depict images from different views can provide complementary information. Thus a rational fusion of multiview features contributes to more comprehensive depiction for images, which can be beneficial to image searching, classification, or other related tasks.

Many multiview learning algorithms have been proposed for operating some tasks such as classification, retrieval, and clustering based on multiview features. According to the levels of feature fusion, multiview learning methods can be grouped into two categories [7]: feature-level fusion such as MKL [8], SVM-2K [9], and CCA [10] and classifier-level fusion such as hierarchical SVM [11]. Some experimental studies show that classifier-level fusion outperforms simple feature concatenation, whereas sophisticated feature-level fusion usually performs better than classifier-level fusion [11, 12].

Recently, many image annotation algorithms use a variety of underlying features to improve annotation performance [8–10]. On one hand, the multiview features improve the

accuracy, but on the other hand the strategies decrease the efficiency and applicability of the algorithms because of the increase of feature dimensions. Moreover, many existing multiview learning algorithms are unsupervised; that is, they do not make use of the label information in the training set. Such fused features may not effectively contain the semantic relationship between samples. This paper proposes a semisupervised learning framework based on graph embedding and multiview NMF (GENMF). In GENMF, feature fusion and dimension reduction are firstly performed by the proposed graph embedded multiview NMF algorithm, and then the new obtained features are used to annotate images through KNN-based approach.

2. Related Works

Existing image annotation algorithms can be roughly divided into two categories [13]: model-based learning methods and database-based retrieval methods. Model-based methods explore the relationship between high-level semantic concepts and low-level visual features to discover a mapping function through machine learning or knowledge models for image annotation. Unlike model-based methods, database-based methods do not need to set up the mapping function based on the training set but directly provide a sequence of candidate labels according to the already annotated images in the database.

There are three kinds of model-based learning methods for image annotation: classification based methods, possibility based methods, and topic model-based methods. Classification based methods [14–16] treat tags as specific class labels and explore the mapping relations between low-level visual features and labels through machine learning methods. The essence of this kind of methods is transforming image annotation to image classification. Different classifiers are used to establish mapping functions between low-level features (from images or regions) and semantic concepts. Labels with the high confidence from the classifiers are annotated to images. Different from classification based methods, possibility based methods [17, 18] do not use classifiers to build the mapping functions but explore the relationship between the underlying features of the image and the semantic labels based on unsupervised probability and statistics models. They utilize the relations to calculate the joint probability of images and labels or the conditional probability of labels given an image and then estimate the possible labels through statistical inference. Topic model-based methods [19, 20] use latent topics to associate low-level visual features with high-level semantic concepts to implement image annotation.

The model-based methods have three difficulties in practical applications. First, the learning models trained on the datasets with finite image types and semantic labels can hardly reflect the characteristics of feature distributions in the real world, which leads to unsatisfactory annotation performance when facing new features and semantic labels. Second, the limited size of training sets may result in overfitting and low generalization ability of the models. Third, low-level features may often fail to express high-level semantic

information because they belong to different feature spaces. Thus, it is also hard to establish a mapping model between image features and semantic concepts because of the semantic gap.

The essence of retrieval based method is directly providing a list of candidate labels for the images to be tagged based on the existing datasets with complete and valid label information. Most common retrieval methods are based on KNN [21–23]: they retrieve k images with the highest similarity to the input image from the database, and the labels of the k images are sorted based on the statistical relationship or weighted statistical relationship to generate the candidate labels of the input images. The other category is graph-based methods [24–27] that utilize image feature distance to establish relevant graphs of samples. Based on the assumption that neighboring images in the relevant graph have similar labels (label smoothness), the similarity between nodes and the global structural characteristics of the relevant graph are used to propagate and enrich the node information including labels and classes. This kind of semisupervised learning methods is suitable for not fully tagged datasets existing on the Internet.

Traditional graph-based methods usually label images by aggregating multiple features into one feature and building a relation graph based on this feature. In [25], it is pointed out that traditional methods cannot effectively capture the unique information for each feature and proposes to utilize different features to establish relation subgraphs and then link these subgraphs to form a supergraph. Based on the supergraph, label propagation is achieved through the graph-based method. In [26], different feature graphs are built based on different features of the images and then the relationship between images is constructed through the graph-based method based on different feature graphs. Furthermore, the relationship between images and different features can be also constructed. Finally, the two relationships, namely, the relation between images and the relation between images and different features, can be fused by a designed objective function to obtain good candidates for the labels.

In [27], a graph learning KNN (GLKNN) is proposed by combining KNN-based method and graph-based method. GLKNN first uses graph-based method to propagate the labels of the K nearest neighbors to the new image and obtain one sequence of candidate labels, then GLKNN employs the naive-Bayes nearest neighbor algorithm to establish the relationship between labels and image features for obtaining another sequence of candidate labels. Finally, the two candidate label sequences are linearly combined as the final predicted labels. In [28], graph embedding discriminant analysis is applied to classify marine fish species by constructing intraclass similarity graph and interclass penalty graph. Although the algorithm improves the performance of classification and clustering by utilizing class labels to build graph embedded term, the traditional graph embedding algorithm is not suitable for multilabel problems with multilabel images because there is no intraclass and interclass relationship. In [21, 22], different models based on metric methods are proposed to enhance the representation ability of features and further improve the performance of image

annotation. However, the metric based feature processing only linearly embeds the original features and does not reduce the feature dimension. In [13], multiple features are fused by concatenation, which ignores the manifold characters of different features and high feature dimension results in low efficiency of the algorithm.

For reducing the dimensions of each feature for annotation, an extended local sensitive discriminant analysis algorithm is proposed by constructing relevant and irrelevant graphs in [29]. Generally, feature dimension reduction methods based on NMF decomposition are for single-view features. References [30, 31] extend this method to multiview features by simply concatenating multiple vectors into one feature vector before further dimension reduction. However, this concatenation way can cause vector dimension disaster. Besides, multiview features are descriptions from different views for images so that simple connection does not make good sense. Then a multiview NMF model based on shared coefficient matrix is developed for capturing the latent feature patterns in multiview features [32], where different view features have their own basis matrices and share a coefficient matrix. The proposed model is used for solving classification and clustering problems and is not suitable for multilabel problems with multilabel images.

Based on the above reviews, this paper proposes a semisupervised learning model based on multiview NMF and graph embedding. A novel multiview NMF algorithm based on graph embedding is developed to fuse the multiview features and reduce the dimension of the fused features by designing appropriate graph embedded regularization terms. Then, the image annotation is performed by using the new features through a KNN-based algorithm.

3. The Proposed Methods

In this section, we elaborate the proposed semisupervised framework for automatic image annotation. First, the graph embedding terms for multilabel problems are constructed through semantic similarity matrix. Second, an objective function is established by adding graph embedded semantic constraints. Third, the update rules for optimizing are derived in detail. Finally, the overall framework of the algorithm is presented.

3.1. Graph Embedding for Multilabel Problem. The traditional graph embedding model is introduced for classification problems, in which each sample has only one label, so that the Laplacian matrices L and L^P can be given according to whether they belong to the same category or not. However, for multilabel problems, a sample usually contains multiple category labels. Therefore, traditional graph embedding methods cannot be directly applied to multilabel problems. In this paper, we give a relation matrix according to whether samples are related or not. By setting appropriate thresholds, the relevant matrix and the irrelevant matrix can be obtained, and they can be used to calculate Laplacian matrices L and L^P , respectively.

Let $\{x_i, y_i\}$ denote the i -th sample and $Y \in \mathbb{R}^{n_1 * m}$ denote label matrix, where n_1 is the number of samples in the training

set, m is the number of labels, y_i represents the i -th row of Y , and $y_{.i}$ represents the i -th column. The semantic similarity between sample i and sample j can be formulated as $y_i C y_j$, where C is a priori label relation matrix similar to that in [33].

$$C_{ij} = \cos(y_{.i}, y_{.j}) = \frac{\langle y_{.i}, y_{.j} \rangle}{\|y_{.i}\| \|y_{.j}\|} \quad (1)$$

$y_{.i} \in \mathbb{R}^{m \times 1}$ denotes the sample vector and $\|y_{.i}\|$ denotes the L2-norm of $y_{.i}$. Then, the semantic similarity matrix of samples can be obtained by the following formula:

$$(W^s)_{ij} = y_i C y_j \quad (2)$$

Given thresholds T_u and T_l ($T_u \geq T_l$), samples with similarity greater than T_u are relevant, and samples with similarity less than T_l are irrelevant. Therefore, the relevant matrix W and the irrelevant matrix W^P are constructed as follows:

$$W_{ij} = \begin{cases} W_{ij}^s, & W_{ij}^s > T_u \\ 0, & W_{ij}^s \leq T_u \end{cases} \quad (3)$$

$$W_{ij}^P = \begin{cases} 1, & W_{ij}^s \leq T_l \\ 0, & W_{ij}^s > T_l \end{cases} \quad (4)$$

The corresponding Laplacian matrices are formulated as follows:

$$L = D - W \quad (5)$$

$$L^P = D^P - W^P \quad (6)$$

where $D_{jj} = \sum_l W_{jl}$ and $D_{jj}^P = \sum_l W_{jl}^P$.

Having the relevant and irrelevant matrices, the following two constraint items C_1 and C_2 are incorporated to make feature representations in the new feature space consist with semantic concepts:

$$C_1 = \sum_{i,j=1}^{n_1} \|v_i - v_j\|^2 W_{ij} = \sum_{i=1}^{n_1} v_i^T v_i D_{ii} - \sum_{i,j=1}^{n_1} v_i^T v_j W_{ij} \quad (7)$$

$$= Tr(V^T D V) - Tr(V^T W V) = Tr(V^T L V)$$

$$C_2 = \sum_{i,j=1}^{n_1} \|v_i - v_j\|^2 W_{ij}^P = \sum_{i=1}^{n_1} v_i^T v_i D_{ii}^P - \sum_{i,j=1}^{n_1} v_i^T v_j W_{ij}^P \quad (8)$$

$$= Tr(V^T D^P V) - Tr(V^T W^P V) = Tr(V^T L^P V)$$

where n_1 denotes the number of samples in the training set and v_i and v_j represent the visual feature vectors of sample i and sample j , respectively.

3.2. An Automatic Image Annotation Model Based on Multi-view Feature NMF and Graph Embedding. Let $X = \{X^{(v)}\}_{v=1}^M$ denote the data matrix, where $X^{(v)} \in \mathbb{R}^{D^{(v)} * N}$ is the feature matrix corresponding to the v -th view, $D^{(v)}$ is the dimension

of feature vectors, M is the number of views, and N is the number of samples. The objective function can be formulated as

$$O_1 = \sum_{v=1}^M \|X^{(v)} - U^{(v)}V^T\|^2 \quad (9)$$

$$\text{s.t. } u_{ij}^{(v)} \geq 0, \quad v_{ij} \geq 0$$

where $U^{(v)} \in \mathbb{R}^{D^{(v)} \times K}$ and $V \in \mathbb{R}^{N \times K}$ are nonnegative matrices and K denotes the dimension of the new low-dimensional feature.

Furthermore, graph embedding regularization terms (7) and (8) are combined with the above loss function, then

$$O_1 = \sum_{v=1}^M \|X^{(v)} - U^{(v)}V^T\|^2 + \text{Tr}\left(\left((V^l)^T \check{L}V^l\right)\right) \quad (10)$$

$$\text{s.t. } u_{ij} \geq 0, \quad v_{ij} \geq 0$$

where $\check{L} = (\alpha L - \beta L^P)$ and α and β are two equilibrium coefficients. Equation (10) consists of two terms, where the first is the error term, and the second is the constraint term that makes semantic constrains on V by using graph embedding regularization. It implies that the semantic related sample features are closer and vice versa. It is worth noting that the model is semisupervised since that V^l refers to data with labels, and the graph embedding term is used to constrain V^l .

3.3. Update Rules Derivation. The established model is semisupervised, and only part of the data has label information. The objective function can be rewritten in the form of block matrix. The following subsection will give the derivation of update rules.

The update rule of formula (10) is derived as follows:

$$\begin{aligned} O_1 &= \sum_{v=1}^M \text{Tr}\left(\left(X^{(v)} - U^{(v)}V^T\right)\left(X^{(v)} - U^{(v)}V^T\right)^T\right) \\ &+ \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) \\ &= \sum_{v=1}^M \left[\text{Tr}\left(X^{(v)} X^{(v)T}\right) - 2 \text{Tr}\left(X^{(v)} V U^{(v)T}\right) \right. \\ &\quad \left. + \text{Tr}\left(U^{(v)} V^T V U^{(v)T}\right) \right] + \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) \\ &\quad - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) \end{aligned} \quad (11)$$

Let $\psi_{ij}^{(v)}$ and φ_{ij} be the Lagrange multipliers of constraint conditions $u_{ij}^{(v)} \geq 0$ and $v_{ij} \geq 0$, respectively, $\Psi^{(v)} = [\psi_{ij}^{(v)}]$, $\Phi = [\varphi_{ij}]$. Then the Lagrange function can be written as

$$L = \sum_{v=1}^M \left[\text{Tr}\left(X^{(v)} X^{(v)T}\right) - 2 \text{Tr}\left(X^{(v)} V U^{(v)T}\right) \right.$$

$$\begin{aligned} &\left. + \text{Tr}\left(U^{(v)} V^T V U^{(v)T}\right) + \text{Tr}\left(\Psi^{(v)} U^{(v)T}\right) \right] \\ &+ \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) \\ &+ \text{Tr}\left(\Phi V^T\right) \end{aligned} \quad (12)$$

The partial derivative of L with respect to $U^{(v)}$ is as follows:

$$\frac{\partial L}{\partial U^{(v)}} = -2X^{(v)}V + 2U^{(v)}V^T V + \Psi^{(v)} \quad (13)$$

where $X^{(v)} = [X^{(v)l}, X^{(v)u}]$, $V = [(V^l)^T, (V^u)^T]^T$, and $\Phi = [(\Phi^l)^T, (\Phi^u)^T]^T$, the symbol l means labelled and the symbol u means unlabelled. Thus, $X^{(v)l}$ and V^l refer to the data with labels. Then (12) can be rewritten as

$$\begin{aligned} L &= \sum_{v=1}^M \left[\text{Tr}\left(X^{(v)} X^{(v)T}\right) - 2 \text{Tr}\left(\left[X^{(v)l}, X^{(v)u}\right] \right. \right. \\ &\quad \left. \left. \cdot \left[\left(V^l\right)^T, \left(V^u\right)^T\right]^T U^{(v)T}\right) \right. \\ &\quad \left. + \text{Tr}\left(U^{(v)} \left[\left(V^l\right)^T, \left(V^u\right)^T\right] \left[\left(V^l\right)^T, \left(V^u\right)^T\right]^T \right. \right. \\ &\quad \left. \left. \cdot U^{(v)T}\right) + \text{Tr}\left(\Psi^{(v)} U^{(v)T}\right) \right] + \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) \\ &\quad - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) + \text{Tr}\left(\left[\left(\Phi^l\right)^T, \left(\Phi^u\right)^T\right]^T \right. \\ &\quad \left. \cdot \left[\left(V^l\right)^T, \left(V^u\right)^T\right]\right) = \sum_{v=1}^M \left[\text{Tr}\left(X^{(v)} X^{(v)T}\right) \right. \\ &\quad \left. - 2 \text{Tr}\left(X^{(v)l} V^l U^{(v)T}\right) - 2 \text{Tr}\left(X^{(v)u} V^u U^{(v)T}\right) \right. \\ &\quad \left. + \text{Tr}\left(U^{(v)} \left(V^l\right)^T V^l U^{(v)T}\right) + \text{Tr}\left(U^{(v)} \left(V^u\right)^T \right. \right. \\ &\quad \left. \left. \cdot V^u U^{(v)T}\right) + \text{Tr}\left(\Psi^{(v)} U^{(v)T}\right) \right] + \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) \\ &\quad - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) + \text{Tr}\left(\Phi^l \left(V^l\right)^T \right. \\ &\quad \left. + \Phi^u \left(V^u\right)^T\right). \end{aligned} \quad (14)$$

Separating the terms associated with V^l and V^u , the above equation can be written as

$$L = L(V^l) + L(V^u) \quad (15)$$

$$\begin{aligned} L(V^l) &= \sum_{v=1}^M \left[-2 \text{Tr}\left(X^{(v)l} V^l U^{(v)T}\right) \right. \\ &\quad \left. + \text{Tr}\left(U^{(v)} \left(V^l\right)^T V^l U^{(v)T}\right) \right] + \alpha \text{Tr}\left(\left(V^l\right)^T L V^l\right) \\ &\quad - \beta \text{Tr}\left(\left(V^l\right)^T L^P V^l\right) + \text{Tr}\left(\Phi^l \left(V^l\right)^T\right) + \text{const} \end{aligned} \quad (16)$$

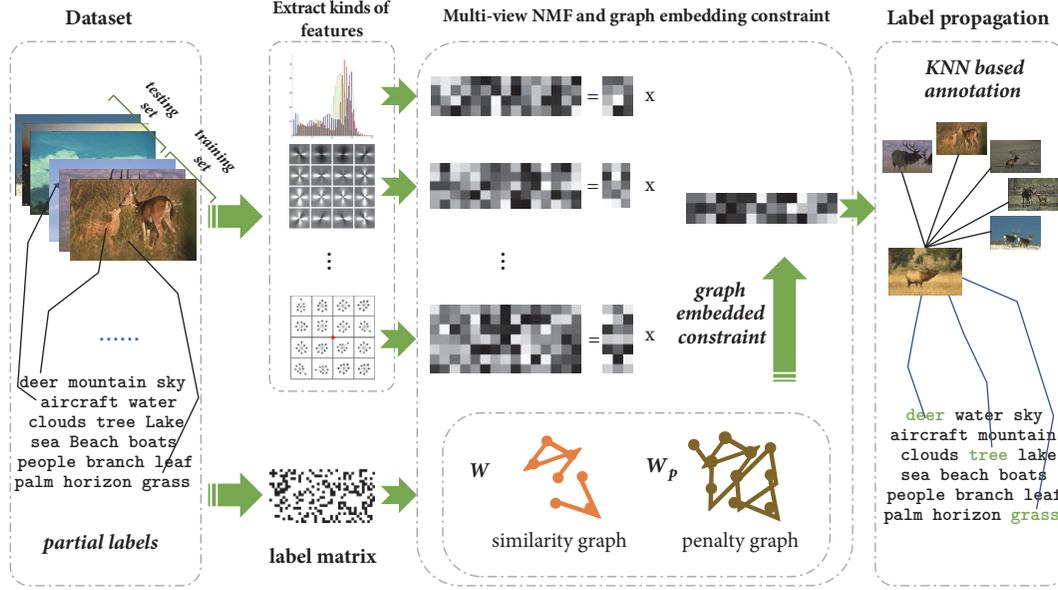


FIGURE 1: Schematic diagram of the GENMF model.

$$\begin{aligned}
 L(V^u) &= \sum_{v=1}^M \left[-2Tr \left(X^{(v)u} V^u U^{(v)T} \right) \right. \\
 &+ Tr \left(U^{(v)} (V^u)^T V^u U^{(v)T} \right) \left. \right] + Tr \left(\Phi^u (V^u)^T \right) \\
 &+ const
 \end{aligned} \quad (17)$$

The partial derivatives of L with respect to V^l and V^u are as follows:

$$\begin{aligned}
 \frac{\partial L}{\partial V^l} &= \sum_{v=1}^M \left[-2 \left(X^{(v)l} \right)^T U^{(v)} + 2V^l U^{(v)T} U^{(v)} \right] \\
 &+ 2\alpha L V^l - 2\beta L^p V^l + \Phi^l
 \end{aligned} \quad (18)$$

$$\frac{\partial L}{\partial V^u} = \sum_{v=1}^M \left[-2 \left(X^{(v)u} \right)^T U^{(v)} + 2V^u U^{(v)T} U^{(v)} \right] + \Phi^u \quad (19)$$

Using the KKT conditions $\psi_{ij}^{(v)} u_{ij}^{(v)} = 0$ and $\varphi_{ij} v_{ij} = 0$ (i.e., $\psi_{ij}^{(v)} = 0$ and $\varphi_{ij} = 0$), consider formulae (13), (18), and (19) and let the derivatives equal 0; the following three equations can be obtained:

$$- \left(X^{(v)} V \right)_{ij} u_{ij} + \left(U^{(v)} V^T V \right)_{ij} u_{ij} = 0 \quad (20)$$

$$\begin{aligned}
 \sum_{v=1}^M \left[-2 \left(X^{(v)l} \right)^T U^{(v)} + 2V^l U^{(v)T} U^{(v)} \right]_{ij} v_{ij} \\
 + \left(\alpha L V^l - \beta L^p V^l \right)_{ij} v_{ij} = 0
 \end{aligned} \quad (21)$$

$$\sum_{v=1}^M \left[-2 \left(X^{(v)u} \right)^T U^{(v)} + 2V^u U^{(v)T} U^{(v)} \right]_{ij} v_{ij} = 0 \quad (22)$$

The following update rules can be obtained through the above three equations:

$$u_{ik}^{(v)} \leftarrow u_{ik}^{(v)} \frac{\left(X^{(v)} V \right)_{ik}}{\left(U^{(v)} V^T V \right)_{ik}} \quad (23)$$

$$v_{jk}^l \leftarrow$$

$$v_{jk}^l \frac{\left(\sum_{v=1}^M \left(X^{(v)l} \right)^T U^{(v)} + \alpha W V^l + \beta D^p V^l \right)_{jk}}{\left(\sum_{v=1}^M V^l \left(U^{(v)} \right)^T U^{(v)} + \alpha D V^l + \beta W^p V^l \right)_{jk}} \quad (24)$$

$$v_{jk}^u \leftarrow v_{jk}^u \frac{\left(\sum_{v=1}^M \left(X^{(v)u} \right)^T U^{(v)} \right)_{jk}}{\left(\sum_{v=1}^M V^u \left(U^{(v)} \right)^T U^{(v)} \right)_{jk}} \quad (25)$$

It is mentioned in [34] that in order to ensure the convexity of the loss function, β needs to be taken as an appropriately small value, which is suggested by $\beta = 10^{-4}$. Besides, [35] gives a modified strategy to the original update rules to ensure convergence. The same strategy can be applied to the derived update rules.

3.4. Framework of the GENMF. The schematic diagram of the proposed GENMF model can be illustrated as in Figure 1. First, multiview features are extracted from images as the input matrix X in (10). Equations (1)-(8) are utilized to build graph embedding regularization terms as the input matrices L and L^p in (10). Then, $U^{(v)}$ and V are updated iteratively by using updated equations (23) to (25) until the maximum number of iterations is reached or the loss value is within the permissible range. Finally, the new features V^u of the test set and the training set features V^l are input to the KNN-based labelling algorithm to obtain the predicted labels. The flowchart of the algorithm is shown in Figure 2.

Input: Image set $I = [I_{train}, I_{test}]$ and label matrix Y^l of the training set.
Output: Predicted label matrix Y^{pre} for the test set I_{test} .
(1) Extract different feature $X^{(v)} \in \mathbb{R}_+^{D^{(v)} * N}$ for image set I ;
(2) Construct Laplacian graph L and L^P ;
(3) Initialize $U^{(v)} \in \mathbb{R}_+^{D^{(v)} * K}$ and $V \in \mathbb{R}_+^{N * K}$ randomly;
(4) do
(5) Update $U^{(v)}$ and V based on equation (23)-(25);
(6) while the terminating condition is not satisfied
(7) Input V into 2PKNN [21] image annotation algorithm;
(8) Output predicted labels Y^{pre} for the test set.

ALGORITHM 1: Multiview NMF with graph embedding for image annotation.

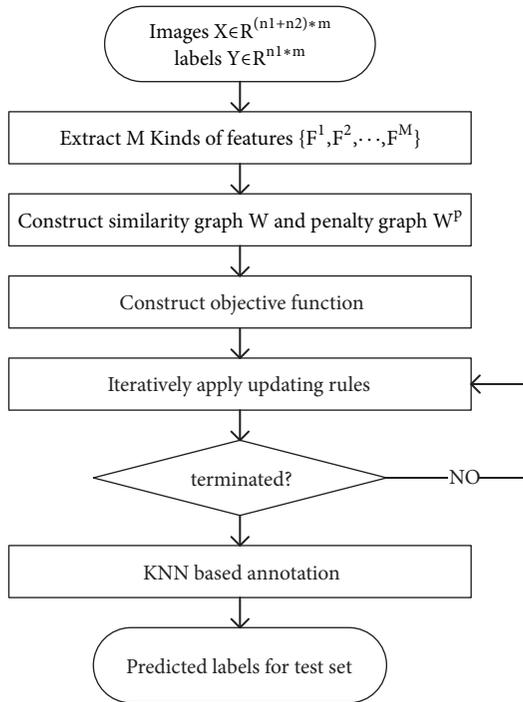


FIGURE 2: Flowchart of the GENMF.

Algorithm 1 gives the pseudocode of the GENMF.

4. Experimental Studies

4.1. Dataset and Experiment Design. The main purpose of the proposed algorithm is to improve the performance of automatic image annotation by fusing the multiview features and reducing the feature dimension, which makes it better to represent semantic concepts under semantic constraints in new low-dimensional feature spaces. So this paper selects the dataset Corel5k with 15 different features, and Corel5k consists of 4500 images for training and 499 images for test, which is available on <http://lear.inrialpes.fr>. The 15 features are all low-level image features including Gist, DenseSift, DenseSiftV3H1, HarrisSift, HarrisSiftV3H1, DenseHue, DenseHueV3H1, HarrisHue, HarrisHueV3H1,

Rgb, RgbV3H1, Lab, LabV3H1, Hsv, and HsvV3H1. In the experiment, we select a local feature DenseSiftV3H1, a global feature Gist, and a color feature Hsv.

In the experiments, the multiple features except Gist are regularized through L2-normalization, and the normalized features are input into the GENMF to obtain low-dimensional representations. Then the low-dimensional feature vectors are input into the 2PKNN annotation algorithm to obtain the predicted labels for the test set. The performance of the algorithm is evaluated in terms of four metrics Pre, Rec, F1, and N+. Table 1 lists the parameters used in the experiments.

4.2. Experimental Results

4.2.1. Convergence Curve of Loss Function. Figure 3 shows the convergence curves of loss function with different parameters. It can be observed that, after about 300 iterations, the trend of the loss curve tends to be stable.

4.2.2. The Influence of Different T_u and T_l . The relation matrix $W^s \in \mathbb{R}^{4500 * 4500}$ can be established according to formula (2). Observed by experimental methods, the maximum value of W^s is 12.9554 and the minimum value of W^s is 0. The values of $T_u = \{1, 2, \dots, 10\}$ and $T_l = \{0, \dots, T_u\}$ are traversed, where $T_u \geq T_l$. Figure 4 shows the changes in the performance of the annotation when the different values of parameters are selected. On the whole, when $T_u = 2$ and $T_l = 1$, the algorithm obtains the highest F1 value. Thus, in the following experiments T_u is taken as 2 and T_l is taken as 1.

4.2.3. The Influence of Different α . Figure 5 shows the varying curve of Pre, Rec, F1, and N+ in the case of $K = 300$ with different α values. Figure 5-1 shows that the annotation accuracy increases first and then decreases with the increase of α . When $\alpha = 1000$, the accuracy reaches the highest value. Figure 5-2 shows that the recall rate generally increases first and then decreases. When $\alpha = 2000$, the recall rate reaches the highest value. From Figure 5-3, it can be seen that the F1 value also increases first and then decreases with the increase of α , but a concave point appears at $\alpha = 1500$. When $\alpha = 1000$, the F1 value reaches the highest value. In Figure 5-4, the N+ value fluctuates in the interval $[0, 1500]$, and its value reaches the highest value at $\alpha = 2000$ and decreases afterwards.

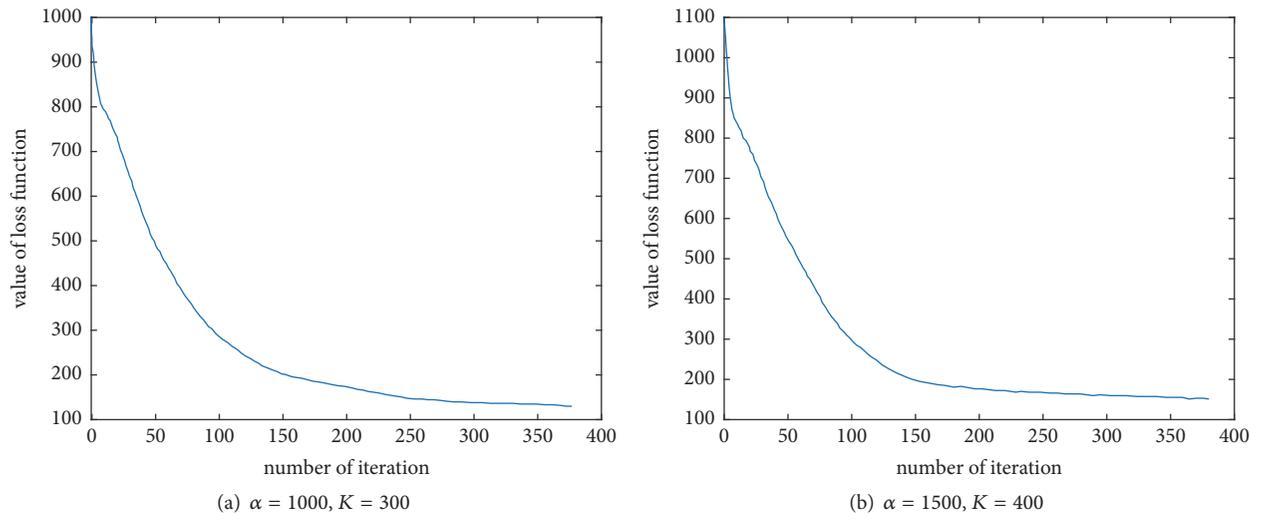


FIGURE 3: Convergence curves of loss function.

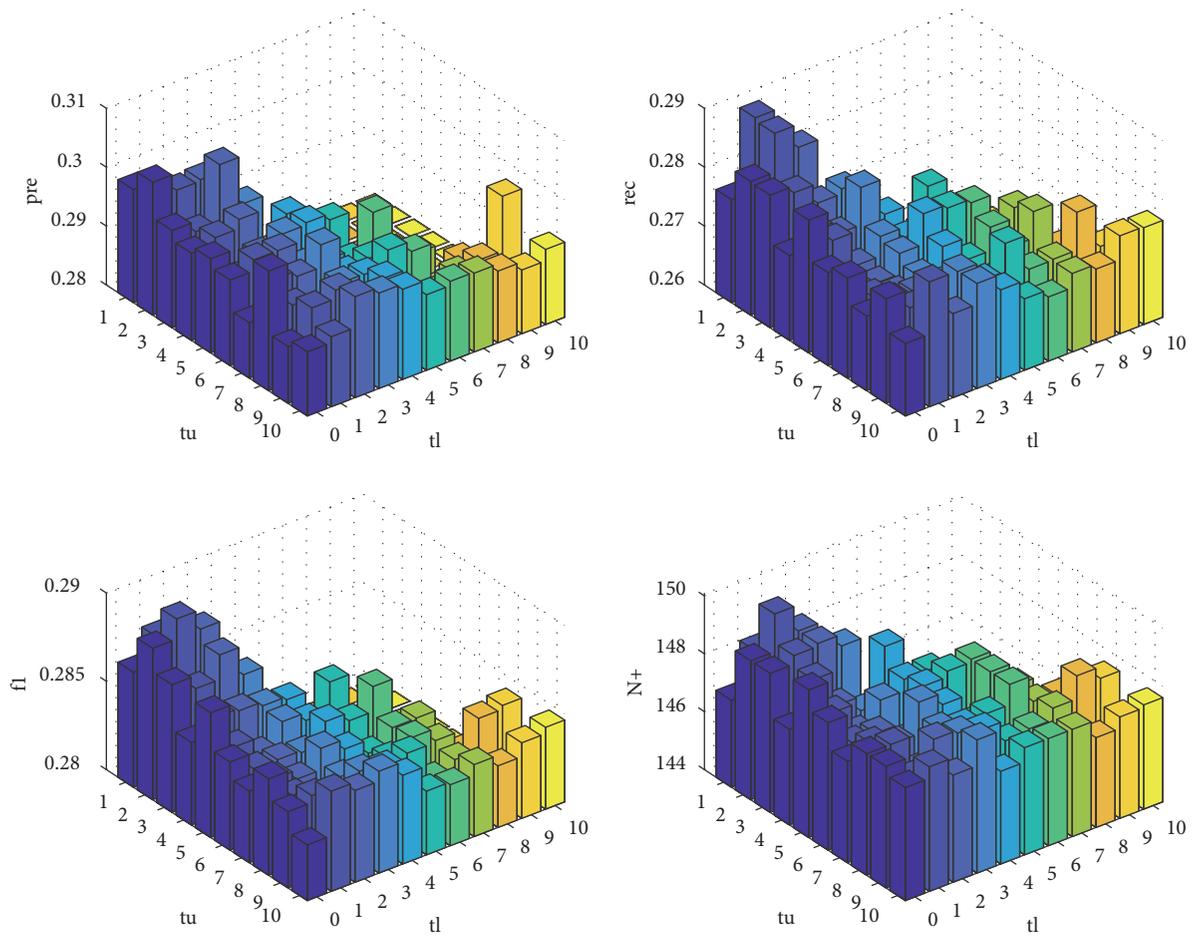


FIGURE 4: Impact of different values for T_u and T_l .

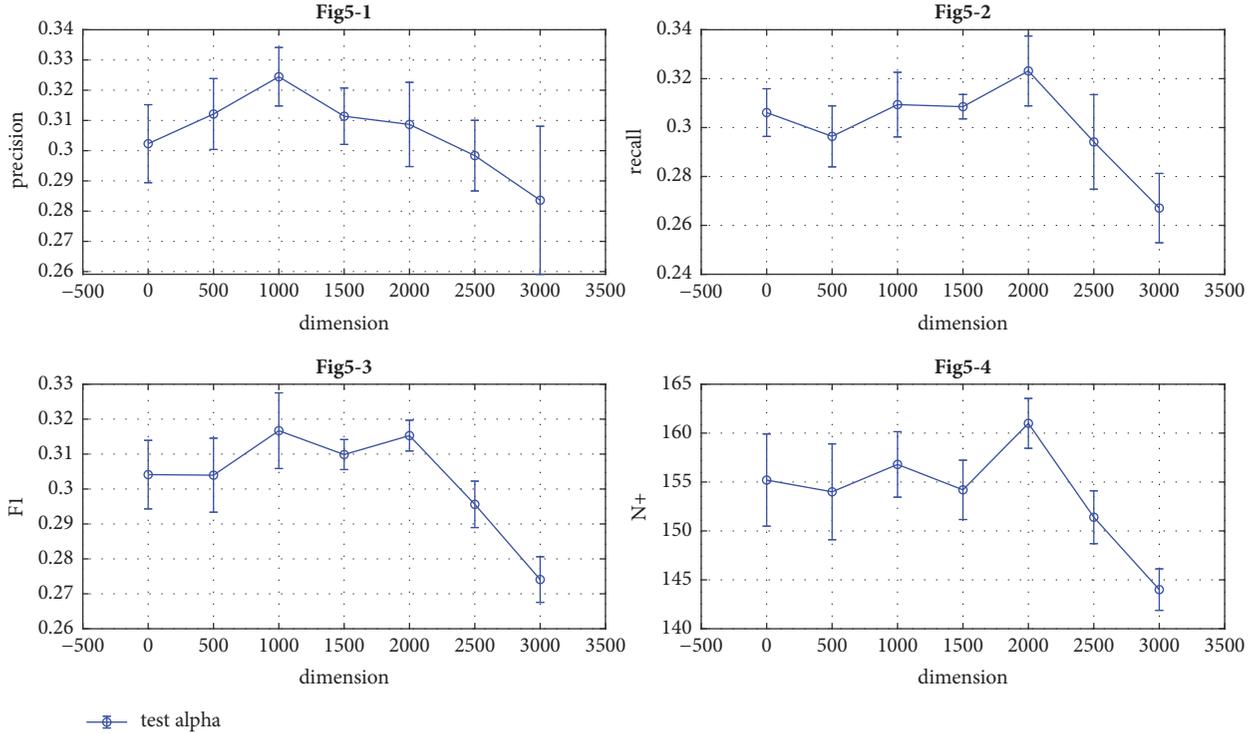
FIGURE 5: Curve of Pre, Rec, F1, and N+ with different α values.

TABLE 1: Parameters required in the algorithm and their ranges of values.

Notation	Description	Range of values
α	Weight for graph embedding terms	{0, 500, 1000, 1500, 2000, 2500, 3000}
K	Dimension of the new features	{100, 200, 300, 400, 500, 600, 700, 800}
T_u	Label-relevant coefficient	{1, 2, ..., 10}
T_l	Label-irrelevant coefficient	{0, 1, ..., T_u }

4.2.4. The Influence of Different Feature Dimensions K . Figure 6 shows the annotation performance curves when α is taken as 0, 1000, and 2000, respectively, and the value of K changes from 100 to 800 with an increase of 100 each time. The three curves with different values of parameter α show the consistent trend of change. In Figure 6-1, the accuracy increases with the increase of dimension because more information can be retained, and the curve becomes stable until α reaches 2000. The worst performance is at $\alpha = 0$. Figure 6-2 shows that the recall rate decreases slightly with the increase of dimension because the requirement for retrieval is higher with the increase of dimension. In Figure 6-3, F1 is reflecting the comprehensive effect of the accuracy and recall rate. It can be observed that the F1 increases in the interval [100, 300] with the increase of dimension and then tends to be stable except for $\alpha = 0$. Figure 6-4 shows that N+ value fluctuates but the overall trend is stable. In general, the performance of proposed algorithm on four metrics outperforms using the original features when $\alpha = 1000$ or $\alpha = 2000$ with dimension in the range of [200-800].

4.2.5. Comparison with Existing Annotation Algorithms. Table 2 presents the comparison results with existing annotation algorithms. RMLF [36] optimizes the final prediction tag score by fusing prediction tag scores of 15 different features. LDMKL [14] and SDMKL [14] use the different classifiers based on the nonlinear kernel of three-layer network to annotate images. 2PKNN [22] uses two steps for annotation: after dealing with data imbalance, images are annotated through a KNN-based method in data-balanced dataset. LJNMF [31], merging features [31], and Scoefficients [31] consider different kinds of NMF modeling, extract new features, and annotate images through a KNN-based method. TagProp (ML) [21] and TagProp (σ ML) [21] acquire discriminative feature fusion on the training set by designing a metric learning model and annotate images using weighted KNN method. JEC [37] is a KNN-based algorithm based on the average distance of multiple features, which is a benchmark algorithm for image annotation. MRFA [38] proposes a new semantic context modeling and learning method based on multimarkov random fields. SML [39] is a discriminative model that treats each label as one class in multiclass classification problems;

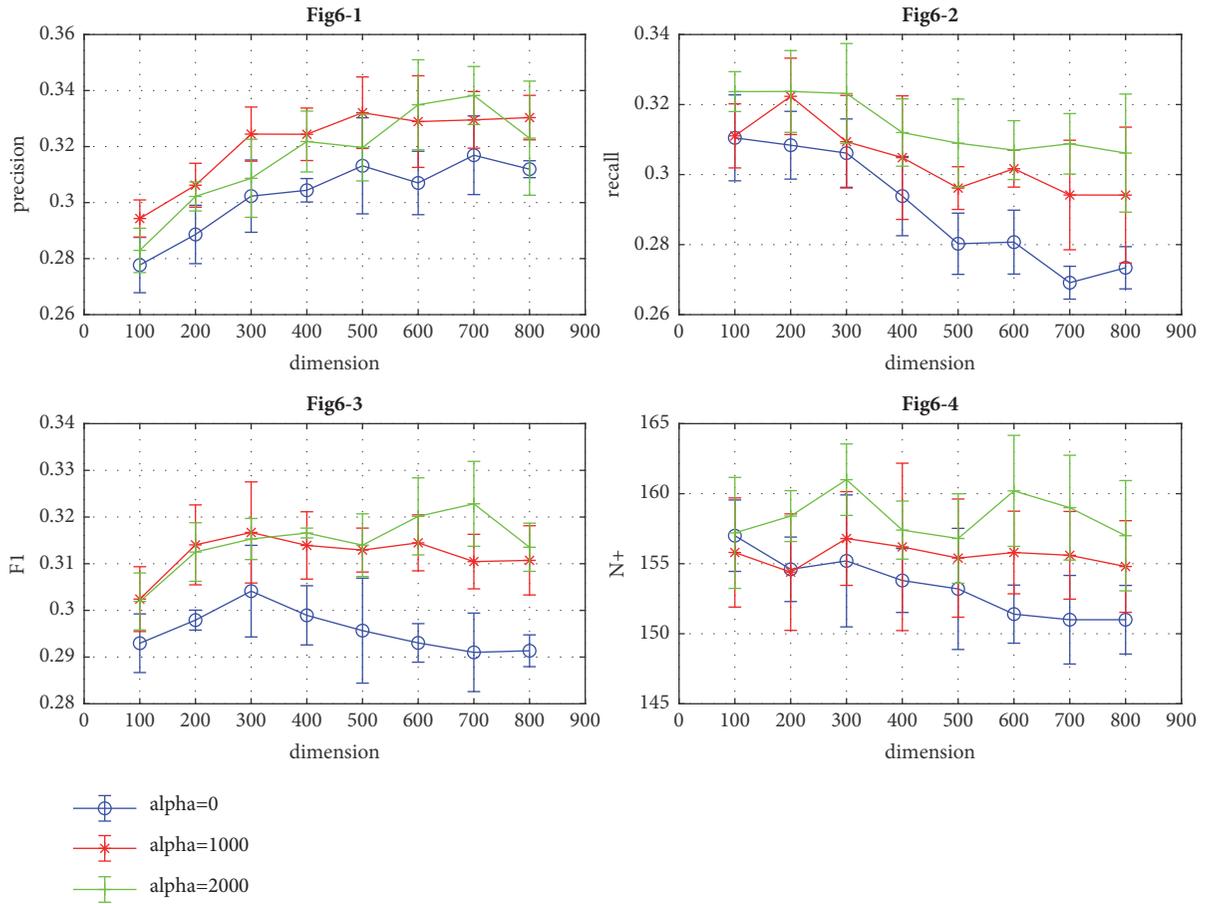


FIGURE 6: Annotation performance curves for different values of K.

TABLE 2: Comparison results with other annotation algorithms.

Methods	Pre	Rec	F1	N+
SML	23	29	25.7	137
JEC	27	32	29.3	139
GS	30	33	31.4	146
MRFA	31	36	33.3	172
TagProp(ML)	31	37	33.7	146
TagProp(σ ML)	33	42	37.0	160
RMLF	29.7	32.6	31.1	-
Merging features	33	40	36.5	-
Scoefficients	30	39	34.6	-
LJNMF(3f')	35	43	39.1	-
2PKNN(3f)	32	28	30.6	177
SDMKL	38	25	30	158
LDMKL	44	29	34.9	179
GENMF (3f)	38	39	39.2	168

GS [38] introduces the regularization-based feature selection algorithm to exploit the sparsity and clustering properties of features.

In Table 2, the note (3f) denotes using the three features selected in this paper, and the note (3f') indicates using three

features that are not the same as in this paper. The results of other algorithms are directly taken from respective literatures and all the 15 features are utilized. Our algorithm uses only three features, and it can be seen in Table 2 that the proposed GENMF achieves the competitive performance.

TABLE 3: The maximum, mean, and standard deviation of results using 10 independent runs.

metrics	Precision	Recall	F1	N+
mean	0.38	0.39	0.392	168
SD	0.017	0.010	0.012	4.50
maximum	0.41	0.40	0.398	175

4.2.6. *The Best, Average, and Standard Deviation of the Results.* Table 3 shows the best, average, and standard deviation of the results using 10 independent runs. The NMF-based algorithms have a certain randomness, and different initial values may produce different results. Table 3 shows that the influence of different initialization values is limited, but better performance could be expected if a better initialization strategy is chosen. Besides, the average time consumption of the proposed GENMF with the new low-dimensional features is 13.945 seconds to label all 499 test images, whereas utilizing the original features to label takes 34.652 seconds, which is about 2.5 times that of GENMF.

5. Conclusions

In this paper, we propose a semisupervised framework based on graph embedding and multiview nonnegative matrix factorization for automatic image annotation with multilabel images. The main purpose of the proposed algorithm is to improve the performance of automatic image annotation by fusing multiview features and reducing feature dimension, which makes it better to represent semantic concepts under semantic constraints in new low-dimensional feature spaces. For feature fusion and dimension deduction, a novel graph embedding term is constructed based on the relevant graph and the irrelevant graph. Then, the fusion of multiview features and the reduction of dimensionality are realized based on multiview NMF model. Moreover, the updated rules of the model are derived. Finally, images are annotated by using a KNN-based approach. Experimental results validate that the proposed algorithm can achieve competitive performance in terms of accuracy and efficiency.

Data Availability

The code used in this paper is released, which is written in Matlab and available at <https://github.com/MenSanYan/image-annotation>.

Conflicts of Interest

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

Acknowledgments

The authors are grateful to the support of the National Natural Science Foundation of China (61572104, 61103146, 61425002, and 61751203), the Fundamental Research Funds for the Central Universities (DUT17JC04), and the Project of

the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172017K03).

References

- [1] L. Mai, H. Jin, Z. Lin, C. Fang, J. Brandt, and F. Liu, "Spatial-Semantic Image Search by Visual Feature Synthesis," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1121–1130, Honolulu, HI, USA, July 2017.
- [2] H. Guan and W. A. Smith, "BRISKS: Binary Features for Spherical Images on a Geodesic Grid," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4516–4524, Honolulu, HI, USA, July 2017.
- [3] Y. Zhang, W. Lin, Q. Li, W. Cheng, and X. Zhang, "Multiple-level feature-based measure for retargeted image quality," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 451–463, 2018.
- [4] F. Zhang and B. W. Wah, "Fundamental principles on learning new features for effective dense matching," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 822–836, 2018.
- [5] H. Zhang, V. M. Patel, and R. Chellappa, "Hierarchical Multimodal Metric Learning for Multimodal Classification," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3057–3065, Honolulu, HI, USA, July 2017.
- [6] P. Li, Q. Wang, H. Zeng, and L. Zhang, "Local Log-Euclidean Multivariate Gaussian Descriptor and Its Application to Image Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 803–817, 2017.
- [7] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2355–2368, 2015.
- [8] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 323–330, 2004.
- [9] J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak, "Two view learning: SVM-2K, theory and practice," in *Proceedings of the 2005 Annual Conference on Neural Information Processing Systems, NIPS 2005*, pp. 355–362, December 2005.
- [10] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [11] J. Kludas, E. Bruno, and S. Marchand-Maillet, "Information Fusion in Multimedia Information Retrieval," in *Adaptive Multimedia Retrieval: Retrieval, User, and Semantics*, vol. 4918 of *Lecture Notes in Computer Science*, pp. 147–159, Springer Berlin Heidelberg, Berlin, Germany, 2008.
- [12] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA '05)*, pp. 399–402, ACM, November 2005.
- [13] Y. Gu, X. Qian, Q. Li, M. Wang, R. Hong, and Q. Tian, "Image Annotation by Latent Community Detection and Multikernel Learning," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3450–3463, 2015.
- [14] M. Jiu and H. Sahbi, "Nonlinear Deep Kernel Learning for Image Annotation," in *Proceedings of the IEEE International*

- Conference on Acoustics, Speech and Signal Processing*, pp. 1551–1555, 2017.
- [15] L. Sun, H. Ge, S. Yoshida, Y. Liang, and G. Tan, “Support vector description of clusters for content-based image annotation,” *Pattern Recognition*, vol. 47, no. 3, pp. 1361–1374, 2014.
- [16] M.-L. Zhang and L. Wu, “LIFT: Multi-label learning with label-specific features,” in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011*, pp. 1609–1614, July 2011.
- [17] M. Zand, S. Doraisamy, A. Abdul Halin, and M. R. Mustafa, “Visual and semantic context modeling for scene-centric image annotation,” *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8547–8571, 2017.
- [18] D. Tian and Z. Shi, “Automatic image annotation based on Gaussian mixture model considering cross-modal correlations,” *Journal of Visual Communication and Image Representation*, vol. 44, pp. 50–60, 2017.
- [19] J. Tian, Y. Huang, Z. Guo, X. Qi, Z. Chen, and T. Huang, “A multi-modal topic model for image annotation using text analysis,” *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 886–890, 2015.
- [20] K. Pliakos and C. Kotropoulos, “PLSA driven image annotation, classification, and tourism recommendation,” in *Proceedings of the IEEE International Conference on Image Processing*, pp. 3003–3007, 2014.
- [21] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation,” in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 309–316, IEEE, Kyoto, Japan, September–October 2009.
- [22] Y. Verma and V. Jawahar C, “Image Annotation Using Metric Learning in Semantic Neighbourhoods,” in *Proceedings of the European Conference on Computer Vision*, pp. 836–849, 2012.
- [23] M. M. Kalayeh, H. Idrees, and M. Shah, “NMF-KNN: image annotation using weighted multi-view non-negative matrix factorization,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 184–191, IEEE, Columbus, OH, USA, June 2014.
- [24] Z. Chen, M. Chen, and K. Q. Weinberger, “Marginalized denoising for link prediction and multi-label learning,” in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1707–1713, 2015.
- [25] S. Hamid Amiri and M. Jamzad, “Efficient multi-modal fusion on supergraph for scalable image annotation,” *Pattern Recognition*, vol. 48, no. 7, pp. 2241–2253, 2015.
- [26] Z. He, C. Chen, J. Bu, P. Li, and D. Cai, “Multi-view based multi-label propagation for image annotation,” *Neurocomputing*, vol. 168, no. C, pp. 853–860, 2015.
- [27] F. Su and L. Xue, “Graph learning on K nearest neighbours for automatic image annotation,” in *Proceedings of the 5th ACM International Conference on Multimedia Retrieval, ICMR 2015*, pp. 403–410, June 2015.
- [28] S. Hasija, M. J. Buragohain, and S. Indu, “Fish species classification using graph embedding discriminant analysis,” in *Proceedings of the 2017 International Conference on Machine Vision and Information Technology, CMVIT 2017*, pp. 81–86, February 2017.
- [29] X. Liu, R. Liu, F. Li, and Q. Cao, “Graph-based dimensionality reduction for KNN-based image annotation,” in *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012*, pp. 1253–1256, 2013.
- [30] J. BenAbdallah, J. C. Caicedo, F. A. Gonzalez, and O. Nasraoui, “Multimodal image annotation using non-negative matrix factorization,” in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010*, pp. 128–135, September 2010.
- [31] R. Rad and M. Jamzad, “Automatic image annotation by a loosely joint non-negative matrix factorisation,” *IET Computer Vision*, vol. 9, no. 6, pp. 806–813, 2015.
- [32] Z. Guan, L. Zhang, J. Peng, and J. Fan, “Multi-View Concept Learning for Data Representation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3016–3028, 2015.
- [33] H. Wang, C. Ding, and H. Huang, “Multi-label Linear Discriminant Analysis,” in *Proceedings of the European Conference on Computer Vision*, pp. 126–139, 2010.
- [34] N. Guan, X. Huang, L. Lan, Z. Luo, and X. Zhang, “Graph based semi-supervised non-negative matrix factorization for document clustering,” in *Proceedings of the 11th IEEE International Conference on Machine Learning and Applications, ICMLA 2012*, pp. 404–408, 2013.
- [35] C.-J. Lin, “On the convergence of multiplicative update algorithms for non-negative matrix factorization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 18, no. 6, pp. 1589–1596, 2007.
- [36] Y. Yao, X. Xin, and P. Guo, “A rank minimization-based late fusion method for multi-label image annotation,” in *Proceedings of the 23rd International Conference on Pattern Recognition, ICPR 2016*, pp. 847–852, 2017.
- [37] A. Makadia, V. Pavlovic, and S. Kumar, “A New Baseline for Image Annotation,” in *Proceedings of the European Conference on Computer Vision*, pp. 316–329, 2008.
- [38] Y. Xiang, X. Zhou, T.-S. Chua, and C.-W. Ngo, “A revisit of generative model for automatic image annotation using markov random fields,” in *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 1153–1160, June 2009.
- [39] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 394–410, 2007.

Research Article

Wind Power Prediction Based on Nonlinear Partial Least Square

Qian Wang ¹, Yang Lei ¹, and Hui Cao ²

¹School of Automation and Information Engineering, Xi'an University of Technology, Xi'an, Shaanxi 710048, China

²Shaanxi Key Laboratory of Smart Grid & the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Correspondence should be addressed to Qian Wang; wangqian77@xaut.edu.cn

Received 26 January 2018; Revised 16 March 2018; Accepted 22 April 2018; Published 26 June 2018

Academic Editor: Dingli Yu

Copyright © 2018 Qian Wang 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.

Wind power prediction is important for the smart grid safe operation and scheduling, and it can improve the economic and technical penetration of wind energy. The intermittent and the randomness of wind would affect the accuracy of prediction. According to the sequence correlation between wind speed and wind power data, we propose a method for short-term wind power prediction. The proposed method adopts the wind speed in every sliding data window to obtain the continuous prediction of wind power. Then, the nonlinear partial least square is adopted to map the wind speed under the time series to wind power. The model carries the neural network as the nonlinear function to describe the inner relation, and the outputs of hidden layer nodes are the extension term of the original independent input matrix to partial least squares regression. To verify the effectiveness of the proposed algorithm, the real data of wind power with different working conditions are adopted in experiments. The proposed method, backpropagation neural network, radial basis function neural network, support vector machine, and partial least square are performed on the real data and their effectiveness is compared. The experimental results show that the proposed algorithm has higher precision, and the real power running curves also verify that the proposed method can predict the wind power in short-term effectively.

1. Introduction

Wind power is a big proportion of clean energy that gives a powerful response to the shortage of energy and environment pollution. The intermittent and the randomness of wind cause that the system-balancing costs among all generation and demand participants are very high. Therefore, it is necessary that the short-term prediction of wind power should be attached much more attention to for higher accuracy in different situation [1, 2].

Generally, wind power prediction could be divided into three kinds: physical model [3–6], statistical model, and also the hybrid model. The physical one has a good performance for longer horizons and complicated terrain, because it adopts numerical weather prediction (NWP) and improves its resolution to accurately predict a certain point (such as wind turbines at each) of weather parameters. However, there are some limitations in both theory and application, such as the need of knowing physical properties, the huge amount of computation, and high cost of physical circumstances

forecasting. The statistical models which purely carry previous data over time to make a prediction of that kind are effective for short-term forecasts. And also the hybrid model uses the data coming from NWP, like the wind speed and wind direction, to make the regression of wind power. It performs well in wind power prediction [7, 8]. In detail, the partial least squares (PLS) approach is a common multivariate regression algorithm for linear system and could yield the statistical model of prediction issues [9–11]. While wind data is inherently nonlinear, PLS regression may not always catch the function of wind power output and historical wind speed input [12–14]. Support vector machine (SVM) is a common regression algorithm based on the structural risk minimization with mapping the data into a high-dimensional feature space [15, 16]; due to their dot-product form, linear kernel SVMs are able to be transformed into a compact form by exchanging summation. Neural network (NN) learns from the samples without focusing on mathematical derivation and could outperform the simpler model structures, such as the PLS model. It could obtain good results and has

been widely adopted for wind power prediction [17–19]. Among NN, the backpropagation neural networks (BPNN) are often used for prediction problem [20, 21], which adopts the mean square error and the gradient descent to modify the weights of the neurons. The more flexible radial basis function neural networks (RBFNN) in training algorithm and network resources are adopted in prediction [22–24]. RBFNN are also easy to integrate with other regression analysis models, such as SVM [25]. The original NN also has the problem of insufficient accuracy; especially in different working conditions with different terrains and climate, it is promising to use NN ensembles with feature selection to increase the accuracy and robustness [26–28].

In this paper, a wind power prediction method which combines the neural network and NPLS considering the sequence of data is proposed. The proposed method adopts the wind speed in every sliding data window to obtain the continuous prediction of wind power. The nonlinear partial least squares is adopted to map the wind speed under the time series to wind power. In detail, the model carries the neural network as the nonlinear function to describe the inner relation, and the outputs of hidden layer nodes are the extension term of the original independent input matrix to partial least squares regression. PLS performs on the output matrix to establish the NPLS model based on NN extending input to make the prediction that how wind power is driven by wind speed in a short term. Backpropagation neural network and radial basis function neural network are chosen to realize the NPLS model as BPE-NPLS and RBE-NPLS, respectively, and the two NPLS model are compared with PLS, BPNN, RBFNN, and SVM to perform on real data. The effectiveness of the these models are evaluated by the root-mean-squares errors of cross-validation (RMSECV), the root-mean-squares errors of prediction (RMSEP), the squared correlation coefficient of prediction (R_p^2), and the squared correlation coefficient of cross-validation (R_{CV}^2). The experiments are recorded from the typical dispersed wind farm in Northwest China; three different kinds of wind power fluctuation conditions are discussed.

The organization of this paper is as follows: Section 2 reviews the related work and relevant algorithms. The experimental datasets and procedure are provided in detail in Section 3. In Section 4, the experiments results are discussed. Finally, Section 5 concludes the paper.

2. Problem Statement

A typical wind power generator system is shown in Figure 1; the main function of the gear box is to transmit the power generated by the wind turbine blade to the generator under the action of wind and get the corresponding speed to generate. The control system is the center of the modern wind power generator. Wind turbine control system is based on wind speed, wind direction control, the wind power generator system so that it could run at a steady voltage and frequency, automatic grid and off-grid, alarm for any abnormal situation, and automatic shutdown if necessary.

In operation, the wind speed always fluctuates astatically, resulting in insufficient grid connected generators, so that a reasonable forecast and control of wind speed would make the wind turbine more efficient to generate electricity. While wind speed is the amount of random variation, the output of wind power is also being related to the performance of wind speed. The prediction of wind power and the accuracy of prediction method are especially important to the safety operation of wind power grid. Being of statistical wind power prediction model, a number of wind speed values in a fixed period of time usually are chosen as the input, corresponding to the wind power output in the next short term. However, the mapping between a fixed number of wind speeds and wind power in the next short-term is not always good in prediction performance. Especially, the increase in the length of the data would require a higher calculation capacity, and it will also be mixed with more data outliers adding to the burden of preprocessing. We propose the wind power prediction method considering the datasets matrix mapping, and the time sequence prediction of the sliding data window is carried out instead of the original one-time form prediction in a short term.

3. The Proposed Method

In this paper, a combined NPLS model integrated with the neural network is proposed and the structure of the model is shown as Figure 2. The input of neural network is the continuous historical wind speed $\mathbf{X} \in \mathbf{R}^{m \times m}$ in the sliding data window that assumes a unit sliding window consists of n historical wind speed and there are m sliding data windows in a short-term prediction. \mathbf{X}' is the input for PLS regression after extended conversion process, the outputs of hidden nodes of neural network are the extension term \mathbf{G} . Therefore, the wind power $\mathbf{Y} \in \mathbf{R}^{m \times 1}$ based on the historical wind speed dataset can be obtained as follows:

$$Y = XA + GH + B = [X \ G \ 1] \begin{bmatrix} A \\ H \\ B \end{bmatrix} \quad (1)$$

It assumes that the number of hidden nodes is h , then $\mathbf{G} = [g_1(x), g_2(x), \dots, g_h(x)] \in \mathbf{R}^{n \times h}$. $\mathbf{A} = [a_1, a_2, \dots, a_m]^T \in \mathbf{R}^{m \times s}$ is the linear weighting matrix of \mathbf{X} ; $\mathbf{H} = [h_1, h_2, \dots, h_h]^T \in \mathbf{R}^{h \times s}$ is the nonlinear weighting matrix of the outputs of the hidden nodes of neural network; $\mathbf{1} \in \mathbf{R}^{n \times 1}$ is the unit column vector; $\mathbf{B} = [b_1, b_2, \dots, b_s] \in \mathbf{R}^{1 \times s}$ is the bias vector.

The external PLS part firstly uses the external model to obtain the feature vectors \mathbf{t} and \mathbf{u} , then takes the regression of \mathbf{Y} on \mathbf{t} , and obtains the regression model of \mathbf{Y} on \mathbf{t} . The regression equations can be demonstrated by the following:

$$\begin{aligned} X' &= TP^T + E = \sum_{i=1}^a t_i p_i^T + E \\ Y &= UQ^T + F = \sum_{i=1}^a u_i q_i^T + F \end{aligned} \quad (2)$$

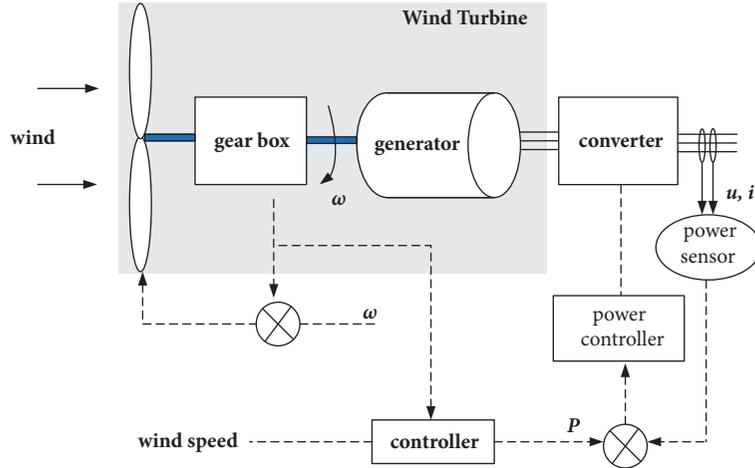


FIGURE 1: Diagram of wind power generation system.

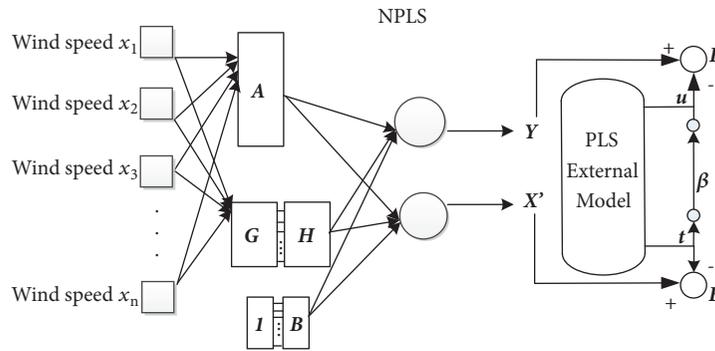


FIGURE 2: Structure of the combined model.

where $\mathbf{T} = (t_1, t_2, \dots, t_a) \in \mathbf{R}^{n \times a}$ and $\mathbf{U} = (u_1, u_2, \dots, u_a) \in \mathbf{R}^{n \times a}$ are the latent variables of \mathbf{X}' and \mathbf{Y} ; $\mathbf{P} = (p_1, p_2, \dots, p_a) \in \mathbf{R}^{m \times a}$ and $\mathbf{Q} = (q_1, q_2, \dots, q_a) \in \mathbf{R}^{s \times a}$ are the loading matrices of \mathbf{X}' and \mathbf{Y} , respectively; a is the number of latent vectors; $\mathbf{E} \in \mathbf{R}^{n \times m}$ and $\mathbf{F} \in \mathbf{R}^{n \times s}$ are the residual error matrices of \mathbf{X}' and \mathbf{Y} , respectively. $(\cdot)^T$ is a transpose operation.

Considering t_i and u_i should represent the \mathbf{X}' and \mathbf{Y} well, and t_i should have the interpretation for u_i . The relation between \mathbf{T} and \mathbf{U} can be presented by the following:

$$\mathbf{U} = \mathbf{T}\boldsymbol{\beta} \quad (3)$$

where $\boldsymbol{\beta} = \text{diag}(\beta_1, \beta_2, \dots, \beta_a) \in \mathbf{R}^{a \times a}$ is the regression coefficient matrix.

The optimization criteria for PLS is as follows:

$$J = \max_{\|w_i\|=1, \|c_i\|=1} \sqrt{\text{var}(t_i) \text{var}(u_i) \text{Corr}(t_i, u_i)} \quad (4)$$

where $\mathbf{w} = (w_1, w_2, \dots, w_a)$ and $\mathbf{c} = (c_1, c_2, \dots, c_a)$ are the weight matrices of \mathbf{X}' and \mathbf{Y} , respectively.

Let $\mathbf{E}_0 = \mathbf{X}'$ and $\mathbf{F}_0 = \mathbf{Y}$; then the optimization problem of PLS can be written as follows:

$$J_1 = \max_{\|w_i\|=1, \|c_i\|=1} (w_1^T \mathbf{E}_0^T \mathbf{F}_0 c_1) \quad (5)$$

Using the Lagrangian method to solve the problem, the Lagrangian function is defined as follows:

$$L(w_1, E_0, F_0, c_1) = w_1^T E_0^T F_0 c_1 - \eta_1 (w_1^T w_1 - 1) - \eta_2 (c_1^T c_1 - 1) \quad (6)$$

where η_1 and η_2 are Lagrange multipliers.

Assume $\partial L / \partial w_1 = \partial L / \partial c_1 = \partial L / \partial \eta_1 = \partial L / \partial \eta_2 = 0$, $\eta = 2\eta_1 = 2\eta_2$, and $\lambda_1 = \eta^2$, and the solution of Lagrange's function constructed in formula (6) can be obtained:

$$\begin{aligned} E_0^T F_0 F_0^T E_0 w_1 &= \lambda_1 w_1 \\ F_0^T E_0 E_0^T F_0 c_1 &= \lambda_1 c_1 \end{aligned} \quad (7)$$

where w_1 and c_1 are the corresponding eigenvectors for the same maximum eigen value λ_1 of $E_0^T F_0 F_0^T E_0$ and $F_0^T E_0 E_0^T F_0$, respectively. Thus, $t_1 = E_0 w_1$, $u_1 = F_0 c_1$.

Reconstruct \mathbf{E}_0 and \mathbf{F}_0 with the following equations:

$$\begin{aligned} E_0 &= t_1 p_1^T + E_1 \\ F_0 &= t_1 b_1^T + F_1 \end{aligned} \quad (8)$$

where $p_1 = E_0^T t_1 / (t_1^T t_1)$ and $b_1 = F_0^T t_1 / (t_1^T t_1)$ are the loading vectors; \mathbf{E}_1 and \mathbf{F}_1 are the residual error matrices.

After the iteration process being performed, the regression equation is as follows:

$$\begin{aligned}
 Y &= t_1 b_1^T + t_2 b_2^T + \cdots + t_a b_a^T + F = X' w_1 b_1^T \\
 &+ X' (1 - w_1 p_1^T) w_2 b_2^T + \cdots + X' (1 - w_1 p_1^T) \\
 &\cdot (1 - w_2 p_2^T) \cdots (1 - w_{a-1} p_{a-1}^T) w_a b_a^T + F \quad (9) \\
 &= X' (w_1 * b_1^T + w_2 * b_2^T + \cdots + w_a * b_a^T) + F \\
 &= X' B_{PLS} + F
 \end{aligned}$$

where F is the residual error matrix.

To implement the proposed NPLS model, BP neural networks and RBF neural networks are adopted in the external NPLS model (BPE-NPLS and RBE-NPLS), and the PLS, BPNN, and RBFNN are also put into prediction for compare. For BPNN, the activation function of the hidden layer is a tangent s-type function:

$$g(x) = \frac{2}{(1 + e^{-2x})} - 1 \quad (10)$$

where x is the input of the hidden layer.

The Gaussian function of the hidden layer of RBFNN can be expressed as follows:

$$g(x) = \exp\left(-\frac{\|x - c\|^2}{\delta^2}\right) \quad (11)$$

where c is the central vector of the hidden node; δ is the corresponding width parameter.

In the next section, we will verify the proposed algorithm to predict short-term wind power on real data.

4. Experiments

Due to the obvious influence of daily, seasonal, and annual variations of wind resources, the volatility of wind energy is very large and unstable. So it requires data from different scenarios and different fluctuations to validate the model.

To evaluate the performance of proposed method, the prediction model is built for the single wind turbine of the typical dispersed wind farm in Northwest China (the capacity is 1.5 MW, double-fed wind turbines). There are three selected wind turbine working scenarios in the experiments, illustrated in Figures 3–5, respectively. In order to compare the performance of different prediction methods and avoid the influence of other factors, the single wind turbine in the same region is selected at different times, reflecting the test scenes of three typical wind fluctuation characteristics.

4.1. Condition I. Recorded from 2013/2/3, Winter, 00 a.m.–12 a.m., wind speed and wind power vary widely during day and night; the wind is large at night, while in daytime it is smaller; the mean wind power for 5 minutes is 727.877 kW and the mean wind speed for 5 minutes is 9.051 m/s.

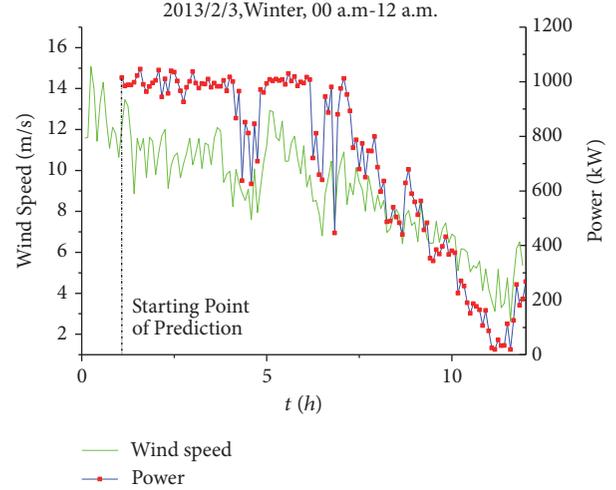


FIGURE 3: Wind turbine working conditions I.

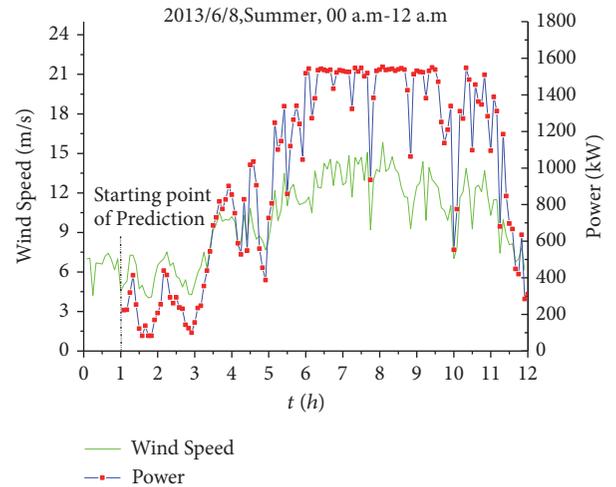


FIGURE 4: Wind turbine working conditions II.

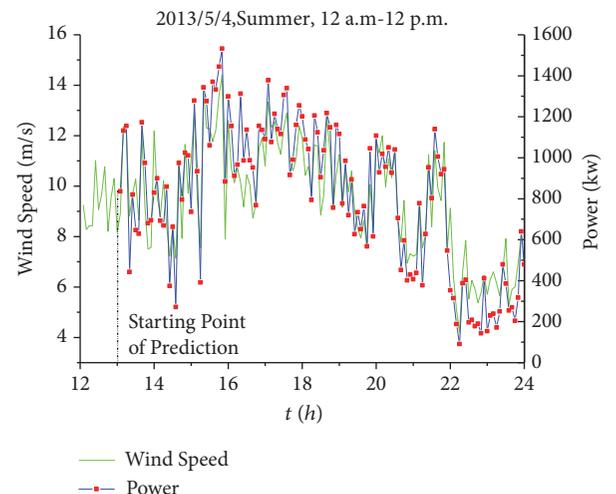


FIGURE 5: Wind turbine working conditions III.

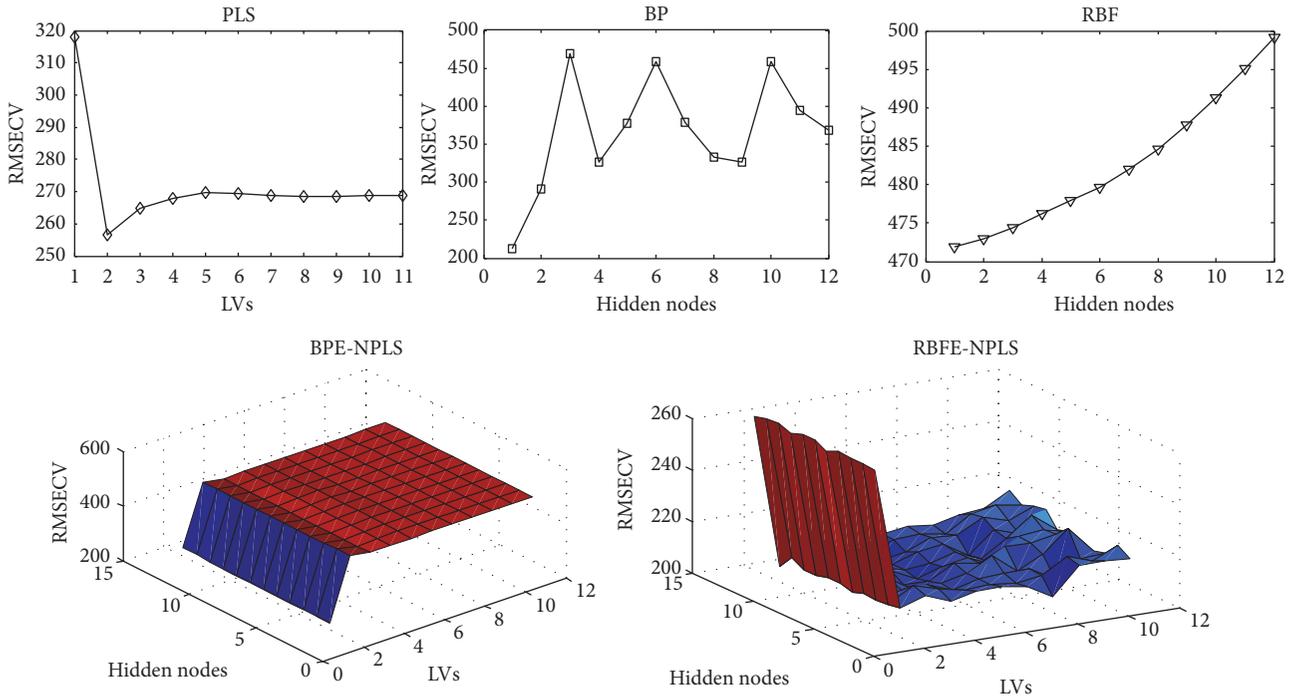


FIGURE 6: Cross-validation errors for condition I: error curves of PLS, BP, and RBF. Error surfaces of BPE-NPLS and RBFN-NPLS.

4.2. *Condition II.* Recorded from 2013/6/8, Summer, 00 a.m.–12 a.m., rich in wind energy, wind power is also high but gentle; the mean wind power for 5 minutes is 919.994 kW and the mean wind speed for 5 minutes is 9.795 m/s.

4.3. *Condition III.* Shown as Figure 5, recorded from 2013/5/4, Summer, 12 a.m.–12 p.m., moderate wind, wind speed and power changes fast. The mean wind power for 5 minutes is 805.351 kW and the mean wind speed for 5 minutes is 9.246 m/s.

In the experiments, both the wind speed and wind power data are raw data mingles with null data recorded from the dispersed wind farm in time series. All data are recorded by five-minute interval and the null data have been removed before. The training set for the input of the proposed prediction is selected from the previous 12 hours using the shutters grouping strategy, which splits the data into 12 wind speed points corresponding to a wind power point, and there are 132 sets of series that constitute a short-time prediction period. After training process, the following wind speed in a sliding data window is used to predict the wind power. During each sliding data window in prediction, twelve continuous wind speeds are the input of the prediction model to obtain the next thirteenth wind power.

The parameter setting range could also be determined by the resolution, the optimal latent Variables (LVs) of PLS and hidden nodes of ANN should be within the sliding window width. Here there are sets 1 to 11 for LVs selection range and sets 1 to 12 for BP and RBF hidden nodes selection in the experiments. The same training set and prediction set are also implemented on SVM, where the kernel function type of SVM selects the Gauss type [16].

5. Results and Discussion

For the experiments of the paper, PLS, BPNN, RBFNN, SVM, BPE-NPLS, and RBFN-NPLS are implemented in MATLAB R2010a. The running environment for all of the calculations is a general-purpose personal computer with an Intel i3-4150 CPU and 8 GB of RAM.

5.1. *Condition I.* For condition I in discussion, the prediction parameters of different regression models are determined by the minimum RMSECV values. The prediction error of the cross-validation among PLS, BP, RBF, BPE-NPLS, and RBFN-NPLS is illustrated in Figure 6. The optimal prediction parameters of PLS, BP, and RBF are always less than 3, and the minimum error on the top of the surface of BPE-NPLS is 1 and 4 for LVs and hidden nodes, respectively; RBFN-NPLS is 8 and 2 for LVs and hidden nodes, respectively.

The detailed numerical analysis results are summarized in Table 1. In this situation where the wind is rich in the evening rich while weaker in daytime, RBFN-NPLS offers a relatively ideal predictive capability and has the superiority effectiveness compared to the poor performance of RBF models. SVM performs better than neural networks under condition I, and the performance is slightly worse than that of NPLS. The RMSECV and RMSEP of RBFN-NPLS are the lowest, the R_{cv}^2 and R_p^2 values are higher than the other methods, and the R_{cv}^2 values are slightly 0.34% inferior to SVM.

In addition, the hidden nodes are selected to be 1 for both BP and RBF network. Width of SVM Gauss kernel is 0.6. For BPE-NPLS, the number of latent variables and the number of

TABLE I: Analytical results for working condition I.

Model	RMSECV	RMSEP	R_{CV}^2	R_p^2	LVs	Hidden nodes	Width
PLS	256.6612	265.5835	0.6903	0.6670	2	-	-
BP	212.8876	248.9909	0.7744	0.6842	-	1	-
RBF	471.8434	429.7947	0.7739	0.0259	-	1	-
SVM	328.6061	240.5229	0.7750	0.7250	-	-	0.6
BPE-NPLS	288.2057	232.0449	0.6262	0.7112	1	4	-
RBFE-NPLS	206.2611	225.5973	0.7724	0.7324	8	2	-

The bold means the minimum error or the maximum correlation among models.

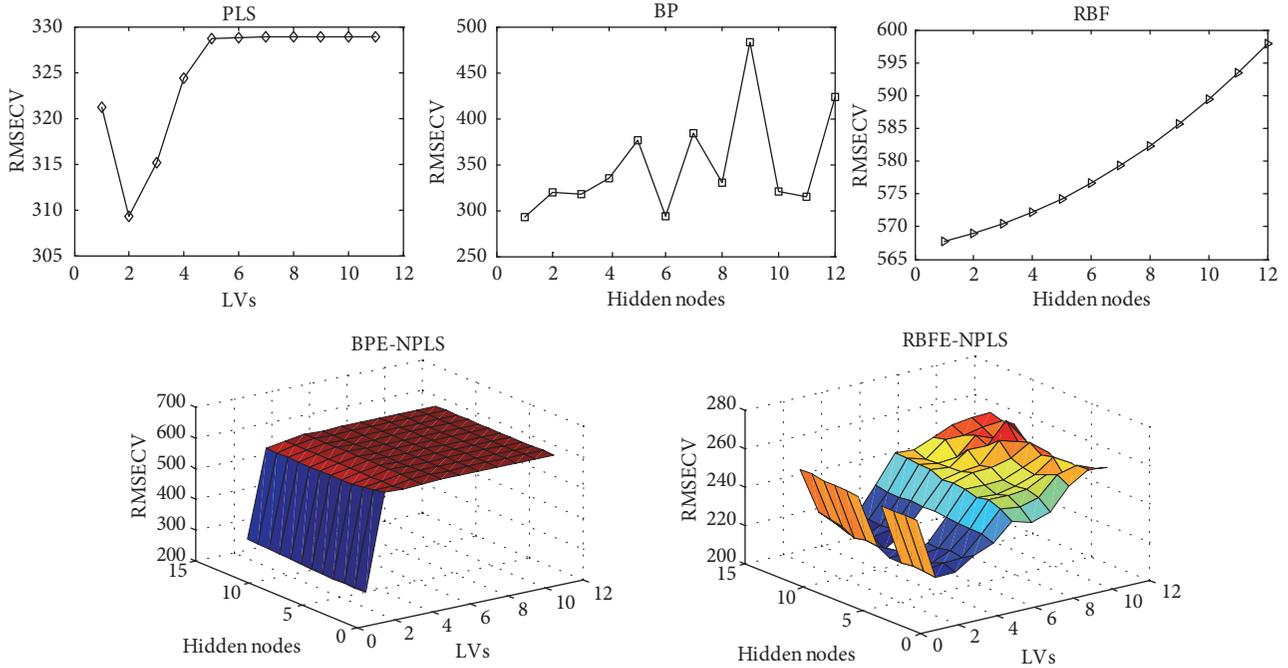


FIGURE 7: Cross-validation errors for condition II: error curves of PLS, BP, and RBF. Error surfaces of BPE-NPLS and RBFE-NPLS.

nodes in hidden layer are 1 and 4 in this circumstance. And for RBFE-NPLS, those are chosen as 8 and 2, respectively.

5.2. Condition II. For condition II in discussion, the prediction error of the cross-validation among PLS, BP, RBF, BPE-NPLS, and RBFE-NPLS are illustrated in Figure 7. The optimal prediction parameters of PLS, BP, and RBF are always less than 3, and the minimum error on the top of the surface of BPE-NPLS is 1 and 2 for LVs and hidden nodes, respectively; RBFE-NPLS are 2 and 5 for LVs and hidden nodes, respectively.

The numerical analysis results are summarized in Table 2. In this situation where wind energy always is rich, wind power is also high but gentle in Summer, RBFE-NPLS still offers a relatively ideal predictive capability and has the superiority effectiveness compared to the poor performance of RBF and SVM. The RMSECV and RMSEP of RBFE-NPLS are the lowest, the R_{cv}^2 and R_p^2 values are higher than the other methods, and the R_{cv}^2 values are 2.58% inferior to RBF network. Original NN model with the generalization capabilities, particularly with flexible structures such as RBF,

proved that simpler model structures (such as the PLS model) may outperform them, and the proposed NPLS method could improve the accuracy of prediction effectively. In this time, NN model outperforms SVM. The R_{cv}^2 values of BP are 6.96% higher than SVM, and the R_p^2 values of BP are 24.97% higher than SVM.

In addition, the hidden nodes are selected to be 1 and 1 for BP and RBF network, respectively. Width of SVM Gauss kernel is 0.6. For BPE-NPLS, the number of latent variables and the number of nodes in hidden layer are 1 and 2 in this circumstance. And for RBFE-NPLS, those are chosen as 2 and 5, respectively.

5.3. Condition III. For condition III, the prediction error of the cross-validation among PLS, BP, RBF, BPE-NPLS, and RBFE-NPLS are illustrated in Figure 8. The optimal prediction parameters of PLS, BP, and RBF are always less than 3, and the minimum error on the top of the surface of BPE-NPLS is 1 and 1 for LVs and hidden nodes, respectively; RBFE-NPLS are 2 and 7, respectively.

TABLE 2: Analytical results for working condition II.

Model	RMSECV	RMSEP	R_{CV}^2	R_p^2	LVs	Hidden nodes	Width
PLS	309.3051	278.4153	0.7418	0.7999	2	-	-
BP	258.7904	250.9290	0.7527	0.7908	-	1	-
RBF	477.7540	460.5628	0.8429	0.2303	-	1	-
SVM	508.0016	336.4237	0.7037	0.6328	-	-	0.6
BPE-NPLS	286.3022	234.4676	0.7636	0.7876	1	2	-
RBFE-NPLS	217.8454	207.9319	0.8217	0.8341	2	5	-

The bold means the minimum error or the maximum correlation among models.

TABLE 3: Analytical results for working condition III.

Model	RMSECV	RMSEP	R_{CV}^2	R_p^2	LVs	Hidden nodes	Width
PLS	203.736	203.952	0.4632	0.4973	2	-	-
BP	262.628	249.336	0.2177	0.2821	-	2	-
RBF	209.364	265.404	0.4682	0.0438	-	1	-
SVM	229.985	238.792	0.4882	0.4727	-	-	0.6
BPE-NPLS	271.416	197.988	0.2900	0.4229	1	1	-
RBFE-NPLS	203.809	190.784	0.4910	0.4875	2	7	-

The bold means the minimum error or the maximum correlation among models.

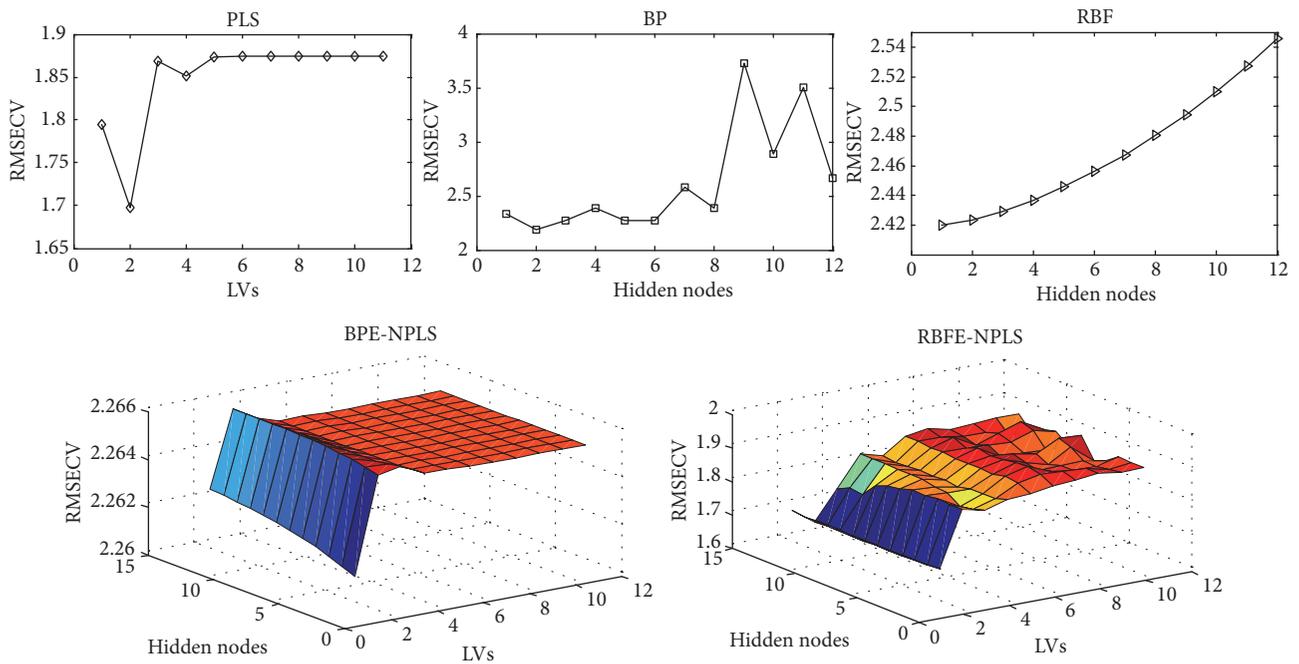


FIGURE 8: Cross-validation errors for condition III: error curves of PLS, BP, and RBF. Error surfaces of BPE-NPLS and RBFE-NPLS.

The detailed numerical analysis results are summarized in Table 3. In this situation where there are moderate wind and the wind speed and power changes slowly, PLS has the ideal RMSECV and R_p^2 , choosing PLS as a benchmark, 0.04% and 1.97% superior to that of RBFE-NPLS. RBFE-NPLS has the ideal value of R_{cv}^2 and RMSEP. Generally, PLS offers a relatively ideal predictive capability in this condition; NPLS

model outperforms neural networks. The performance of SVM is close to that of RBFE-NPLS. The R_{cv}^2 values of SVM are only 0.57% lower than RBFE-NPLS, and the SVM R_p^2 are 3.13% lower than RBFE-NPLS.

In addition, the hidden nodes are selected to be 2 and 1 for BP and RBF network, respectively. Width of SVM Gauss kernel is still chosen as 0.6. For BPE-NPLS, the number of

latent variables and the number of nodes in hidden layer are 1 and 1 in this circumstance. And for RBFN-NPLS, those are chosen as 2 and 7, respectively.

It is noteworthy that the R_p^2 and R_{cv}^2 values are generally low for both the other two conditions across all models. These low prediction performance are due to the anomalous fluctuation characteristics of wind farm no matter which strategy is employed. A number of factors could contribute to these limitations of the prediction process, such as the instability of the grid power flow and the fluctuation of the wind itself. Here, changing the scale of the data sliding window or transferring the short-term into an ultra-short-term forecast cycle may make the analysis more precise.

While wind data is inherently nonlinear, in our experiments the nonlinear ANN models are inferior to the linear PLS models for most components. The poor performance of the ANN models may be related to the size of the experimental datasets available (number of training samples) relative to its dimensionality and the level of noise. The characteristics of the NPLS paradigm, which essentially involves the estimation of relatively simple local models and the interpolation functions that interpolate between them, make it a more robust in the discussed circumstances, and the RBFN-NPLS method shows good all round performance in the experiments.

6. Conclusions

This paper proposes a wind power prediction algorithm structure based on PLS and chooses BPNN, RBFNN to implement the framework. Some advantages are concluded as follows: first, it uses the learning ability of the neural network to implement and refine the NPLS framework to predict the wind power. Secondly, it has higher prediction capability compared with neural network and PLS method themselves. Thirdly, by the hybrid learning process, all the best parameters of membership functions are obtained. The experiments results of proposed models verify that RBFN-NPLS could better adapt to different wind situation than other discussed methods. Since the training time may be affected by the number of inputs, in the future research work, we will use some feature extraction approaches to further improve the training performance of the proposed algorithm.

Data Availability

The data of this article are derived from the measured data of the Lang'er Gou wind farm. The wind farm is located in the southeast of Dingbian County, Yulin, Shaanxi Province, with an elevation of 1440 m to 1710 m. The capacity of the single wind turbine is 2.5 kW. If there are any needs for accessing the data of the study, contact the author of this article, Dr. Qian Wang (wangqian77@xaut.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China with Project no. 51507135 and State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources with Grant no. LAPS15011.

References

- [1] J. Wang, Y. Song, F. Liu, and R. Hou, "Analysis and application of forecasting models in wind power integration: a review of multi-step-ahead wind speed forecasting models," *Renewable & Sustainable Energy Reviews*, vol. 60, pp. 960–981, 2016.
- [2] A. Kusiak, Z. Zhang, and A. Verma, "Prediction, operations, and condition monitoring in wind energy," *Energy*, vol. 60, pp. 1–12, 2013.
- [3] K. Pan, Z. Qian, and N. Chen, "Probabilistic Short-Term Wind Power Forecasting Using Sparse Bayesian Learning and NWP," *Mathematical Problems in Engineering*, vol. 2015, Article ID 785215, 11 pages, 2015.
- [4] M. B. Ozkan and P. Karagoz, "A novel wind power forecast model: Statistical hybrid wind power forecast technique (SHWIP)," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 375–387, 2015.
- [5] S. Buhan, Y. Özkazanç, and I. Çadirci, "Wind Pattern Recognition and Reference Wind Mast Data Correlations With NWP for Improved Wind-Electric Power Forecasts," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 991–1004, 2016.
- [6] L. Dong, L. Wang, S. F. Khahro, S. Gao, and X. Liao, "Wind power day-ahead prediction with cluster analysis of NWP," *Renewable & Sustainable Energy Reviews*, vol. 60, pp. 1206–1212, 2016.
- [7] M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," *Renewable & Sustainable Energy Reviews*, vol. 13, no. 4, pp. 915–920, 2009.
- [8] I. Munteanu and G. Besançon, "Identification-based prediction of wind park power generation," *Journal of Renewable Energy*, vol. 97, pp. 422–433, 2016.
- [9] E. Taslimi Renani, M. F. M. Elias, and N. A. Rahim, "Using data-driven approach for wind power prediction: A comparative study," *Energy Conversion and Management*, vol. 118, pp. 193–203, 2016.
- [10] X. Yuan, Q. Tan, X. Lei, Y. Yuan, and X. Wu, "Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine," *Energy*, vol. 129, pp. 122–137, 2017.
- [11] D. J. Morrow, J. Fu, and S. M. Abdelkader, "Experimentally validated partial least squares model for dynamic line rating," *IET Renewable Power Generation*, vol. 8, no. 3, pp. 260–268, 2014.
- [12] S. Fang and H.-D. Chiang, "Improving supervised wind power forecasting models using extended numerical weather variables and unlabelled data," *IET Renewable Power Generation*, vol. 10, no. 10, pp. 1616–1624, 2016.
- [13] Y. Noorollahi, M. A. Jokar, and A. Kalhor, "Using artificial neural networks for temporal and spatial wind speed forecasting in Iran," *Energy Conversion and Management*, vol. 115, pp. 17–25, 2016.
- [14] U. B. Filik, "A New Hybrid Approach for Wind Speed Prediction Using Fast Block Least Mean Square Algorithm and Artificial

- Neural Network,” *Mathematical Problems in Engineering*, vol. 2016, 9 pages, 2016.
- [15] J. W. Zeng and W. Qiao, “Short-term solar power prediction using a support vector machine,” *Journal of Renewable Energy*, vol. 52, pp. 118–127, 2013.
- [16] G. Fasshauer and M. McCourt, “Machine Learning,” in *Kernel-based Approximation Methods using MATLAB*, pp. 335–360, 2015.
- [17] G. N. Kariniotakis, G. S. Stavrakakis, and E. F. Nogaret, “Wind power forecasting using advanced neural networks models,” *IEEE Transactions on Energy Conversion*, vol. 11, no. 4, pp. 762–767, 1996.
- [18] T. Li, Y. Li, M. Liao, W. Wang, and C. Zeng, “A New Wind Power Forecasting Approach Based on Conjugated Gradient Neural Network,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 8141790, 8 pages, 2016.
- [19] G. W. Chang, H. J. Lu, Y. R. Chang, and Y. D. Lee, “An improved neural network-based approach for short-term wind speed and power forecast,” *Journal of Renewable Energy*, vol. 105, pp. 301–311, 2017.
- [20] T. Ren, S. Liu, G. Yan, and H. Mu, “Temperature prediction of the molten salt collector tube using BP neural network,” *IET Renewable Power Generation*, vol. 10, no. 2, pp. 212–220, 2016.
- [21] C. Ren, N. An, J. Wang, L. Li, B. Hu, and D. Shang, “Optimal parameters selection for BP neural network based on particle swarm optimization: a case study of wind speed forecasting,” *Knowledge-Based Systems*, vol. 56, pp. 226–239, 2014.
- [22] Z.-Q. Wu, W.-J. Jia, L.-R. Zhao, and C.-H. Wu, “Maximum wind power tracking based on cloud RBF neural network,” *Journal of Renewable Energy*, vol. 86, pp. 466–472, 2016.
- [23] R. Cavoretto, T. Schneider, and P. Zulian, “OpenCL based parallel algorithm for RBF-PUM interpolation,” *Journal of Scientific Computing*, vol. 74, no. 1, pp. 267–289, 2018.
- [24] G. S. Bhatia and G. Arora, “Radial basis function methods for solving partial differential equations-A review,” *Indian Journal of Science and Technology*, vol. 9, no. 45, pp. 1–18, 2016.
- [25] H. Cao, T. Naito, and Y. Ninomiya, “Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification,” in *Proceedings of the 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA’08*, 2008.
- [26] N. Haouas and P. R. Bertrand, “Wind farm power forecasting,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 163565, 5 pages, 2013.
- [27] S. Fang and H.-D. Chiang, “A High-Accuracy Wind Power Forecasting Model,” *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1589–1590, 2017.
- [28] J. Jung and R. P. Broadwater, “Current status and future advances for wind speed and power forecasting,” *Renewable & Sustainable Energy Reviews*, vol. 31, pp. 762–777, 2014.

Research Article

Multisource Data-Driven Modeling Method for Estimation of Intercity Trip Distribution

Yilin Li,¹ Haiquan Wang ,^{1,2} Jiejie Zhao,^{2,3} and Bowen Du ^{2,3}

¹College of Software, Beihang University, Beijing 100044, China

²State Key Laboratory of Software Development Environment, Beihang University, Beijing 100044, China

³School of Computer and Engineering, Beihang University, Beijing 100044, China

Correspondence should be addressed to Bowen Du; dubowen@buaa.edu.cn

Received 22 February 2018; Revised 2 May 2018; Accepted 8 May 2018; Published 11 June 2018

Academic Editor: Li Xu

Copyright © 2018 Yilin Li 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.

Traditional intercity trip distribution modeling methods are merely derived from household travel survey due to its limitation to partial or inaccurate information. With the development of information construction, reliable historical data can be easily collected from different sources, such as sensor and statistical data. In this study, a data-driven method based on Poisson distribution theory is proposed to estimate intercity trip distribution using sensor data and various city features. A Poisson model, which reveals the deep correlation between city feature variables and trip distribution, is initially formulated. The L1-norm approach and the coordinate descent algorithm are then adopted in selecting related features and estimating model parameters, respectively, to reduce the complexity of the model. Finally, a k-means clustering method is used to analyze the latent correlation between city features and improve the availability of the model. The methodology is tested on a realistic dataset containing the highway trips of 17 cities in Shandong Province, China. The city feature variables have 66 dimensions, including economic index and population indicator. In comparison with traditional gravity model, which regards population as the most important factor affecting city attraction, our result shows that one of the core positive factors is the economic feature, such as gross regional domestic product. Moreover, the dimension of city features in the developed model decreases from 66 to 13 dimensions. The model developed in this study performs well in replicating the observed intercity origin-destination matrix.

1. Introduction

Intercity trip distribution analyzes and estimates the number of trips between the departure and the destination cities, considering the cities' differences in body mass index and degree of urban obstruction. It is a crucial part of transportation analysis on city interaction and helps us gain a better understanding of intercity human mobility.

Lei et al. [1] divided intercity trip distribution analysis and modeling methods into two groups, namely, top-down and bottom-up approaches. Top-down approaches begin with city socioeconomic, land use, and demographic information, and only few data are utilized to calibrate model parameters. These models rely on experts' knowledge and behavior postulates and are robust to the change of inputs, thereby guaranteeing intercity trip distribution estimation with only few detailed data. However, this group of models

only considers few factors, whereas the actual situation is considerably more complicated with different effects in different regions. Conversely, bottom-up approaches, also called data-driven method, directly estimate flows of trip distribution from available data sources. Determining the manner in which relative data is collected is a critical issue in these approaches.

Traditional dominant methodology, which is referred to as top-down approaches, including the gravity models and destination choice models, is modeled by following different behavior postulates and theories. Gravity model, which is one of the most well-known models, is widely used in estimating the flow among different regions [2–4]. It is formulated based on the postulates that the observed trip distribution is governed by a power decay effect and proportional to the population of origin-destination (OD) cities. Several studies have considered other factors that affect

city's production-attraction capability and decay effects, such as generalized cost [2], tertiary percentage, tourism revenue, the convenience of other traffic modes [3], city size, and trip purpose [4]. Other types of models, such as destination choice models, are based on discrete choice and random utility theories at disaggregate level, such as intervening opportunity, radiation, and population weight opportunity models. These models conclude that the factor of distance has no effect on the travelers' destination choice, and only intervening opportunities play a surrogate role. Thus, the models estimate the commuting flows only by population. However, the conclusions about the performances of these two types of models vary in different studies. Mishra [5] showed that destination choice models perform well in a statewide situation, whereas Lenormand et al. [6] showed that gravity model performs well in estimating commuting flows. Yan et al. [7] developed a general model that only requires population as an input in predicting human mobility in different scales.

In previous studies, household travel survey sample data have been widely used in analyzing trip distribution and continue to be a mainstay of trip distribution modeling [8]. Pitombo [9] applied nonparametric decision tree algorithms to analyze individual destination choices based on OD survey dataset. However, the information obtained from these surveys is limited and partial. Conversely, information derived from sensors and statistics is reliable and sufficiently wide to cover an entire area. Rasouli and Nikraz [10] estimated the distribution of journey to work trips using census data sourced from the Australian Bureau of Statistics based on a generalized regression neural network algorithm. Perrakis et al. [11] also utilized census data to estimate OD flows based on a statistical Bayesian approach. Bekhor [8] applied cellular phone technology to analyze long-distance trips.

Recent development in intelligent transportation allows passive data collection from several travelers and offers a wide variety of data, including GPS and fare data. Mario [12] reviewed studies utilizing passively collected data and summarized the applications conducted based on these data, which include OD estimation. Moreover, city statistical data have become richer and more accurate, and multidimensional data, such as population, economic, and traffic network, can be considered. Traditional top-down methods are limited when detailed data are absent, and the fitness of model to complex reality is not sufficient, which inspires us to consider more factors and different scenarios. The reliable historical traffic data and multidimensional city statistical data allow us to estimate and analyze trip distribution using data-driven method without behavior postulates.

The present work proposes a data-driven method to estimate intercity trip distribution through reliable traffic historical data and statistical data of urban multidimensional characteristics. The method described in this study follows two main steps (Figure 1). First, a Poisson model is proposed to reveal the correlation between city features and trip distribution according to the theory of statistics. Next, the L1-norm approach is applied to select the feature, and the coordinate descent algorithm is used to estimate parameters of the model to solve the problem raised by its complexity. A k-means

clustering method is then used to analyze and interpret all the city features in the light of the feature selection result. Finally, an attribute-optional trip distribution model is developed to enable the replacement of similar attributes in the same cluster. In summary, the main contributions of this paper include the following:

- (1) Considering the samples estimated are biased and insufficient even if the sampling rates are large, we utilize the historical OD data to estimate the trip distribution.
- (2) This theoretical attempt is one of the first to consider such complicated and comprehensive features, which include not only population and distance that always considered in the intercity trip distribution models but also urban economic index and transportation capacity used in estimating intercity trip distribution. Meanwhile, we propose a solution to the problem that high dimension of feature causes overfitness and low generalization of model. Besides, considering some features are hard to be obtained, the features selected by the method are alternative.
- (3) According to the assumption that OD flows are independently Poisson distributed[11], we develop the model based on statistical theory and big data science without relying on experts' knowledge.
- (4) We evaluate the methodology with real-world OD data collected from 17 cities in Shandong Province, and the result outperforms the traditional gravity model as a baseline approach.

As a case study, the new approach is applied to highway intercity trips for Shandong Province in China, which includes 17 cities using expressway networking toll data and Shandong statistical data sourced from Shandong Bureau of Statistics. Accordingly, two different models are developed, namely, the trip distribution model based on Poisson theory and gravity model. Quantified evaluation and comparison of the two models are performed. The result shows that the Poisson model performs better than the traditional gravity model (R^2 score is improved from 0.39 to 0.69). The real city attraction is not population, which the traditional intercity trip distribution model always considers, but the economic feature, such as gross regional domestic product).

The rest of the paper is arranged as follows. Section 2 presents the model formulation and estimation. Section 3 shows the clustering method used to find the latent classes in city features. The case study and result analysis are presented in Section 4. Finally, Section 5 presents the conclusions and provides directions for further research.

2. Trip Distribution Model Based on Poisson

In this section, the model is formulated by Poisson distribution based on statistical theory, which describes the relationship between intercity distribution and city features, as shown by Step 1 of the methodology in Figure 1. L1-norm and coordinate descent algorithm are utilized in selecting city

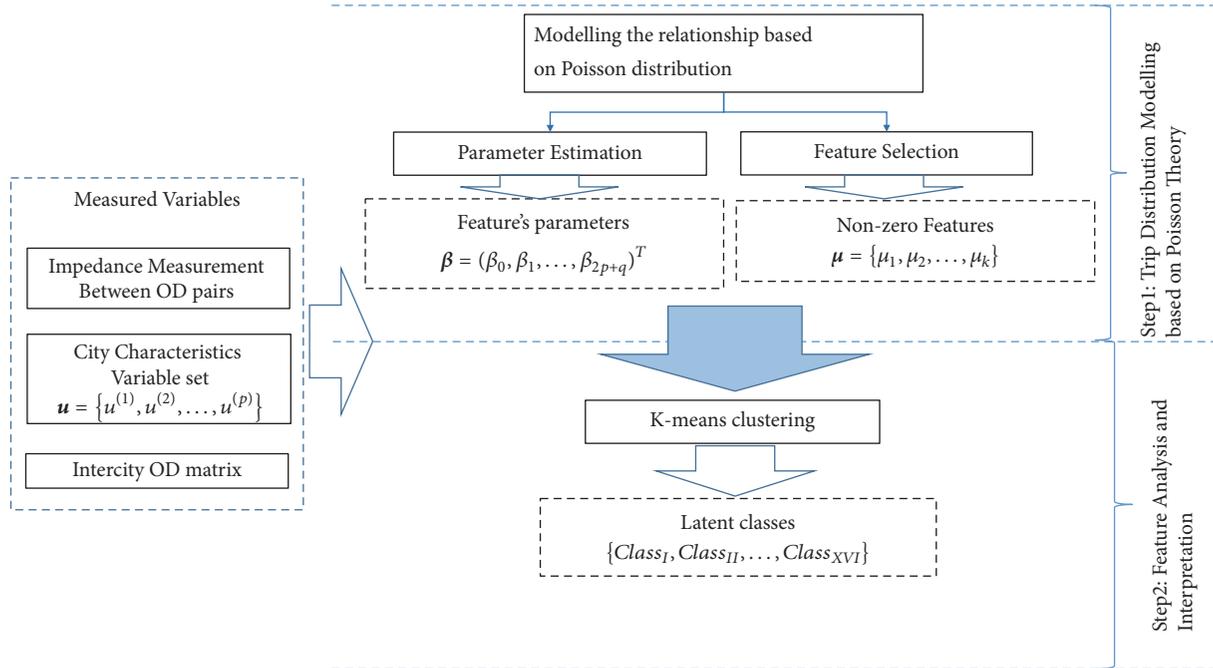


FIGURE 1: Proposed method for estimating intercity trip distribution.

features and estimate parameters to solve the problem raised by the high complexity of the model.

For computational convenience, the flows of intercity trip distribution are presented as a vector. Considering the nature of variables, we divide the variables into two groups. One group presents the city features, and the other group presents the relationship between two different cities. n is denoted as the number of trip distribution samples, p denotes the number of dimensions of city attraction features, and q denotes the number of variables presenting relationship between two different cities. Let $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ denote the vector of observed flows of intercity trip distribution; $\beta = (\beta_0, \beta_2, \dots, \beta_p)^T$ the vector of parameters; and \mathbf{X} the design matrix of dimensionality $n \times (2p + q + 1)$ containing the p origin-city feature variables, p destination-city feature variables, and q relationship variables between origin and destination cities and intercept, with $x_i = (x_{i0}, x_{i2}, \dots, x_{i(2p+q)})^T$ as the i th row of \mathbf{X} related to flow y_i and $i = 1, 2, \dots, n$.

2.1. Model Formulation. According to the likelihood assumption that these intercity trips occur with a known constant rate and independently of time since the last trip, Poisson distribution is used to describe the probability of a given number of intercity trips occurring in a fixed interval of time. Thus, $y_i \sim Pois(\mu_i)$ for $i = 1, 2, \dots, n$. In the context of general linear models, the common assumption of link function in Poisson is logarithmic function. According to the logarithmic link function μ_i , the Poisson mean of y_i is formulated as $x_i^T \beta = \ln(\mu_i)$ for $i = 1, 2, \dots, n$. Thus, $(\mu_i) = \exp(x_i^T \beta) = \exp(\beta_0) \cdot \exp(x_{i1}\beta_1) \cdot \exp(x_{i2}\beta_2) \cdots \exp(x_{i(2p+q+1)}\beta_{2p+q+1})$. The probability of observing y is defined as follows:

$$p(\mathbf{y} | \beta) = \prod_{i=1}^n \frac{\exp[-\exp(x_i^T \beta)] \exp(x_i^T \beta)^{y_i}}{y_i!}. \quad (1)$$

According to the definition of Poisson distribution, the expected value of y_i is equal to μ_i , that is, $E(y_i | \beta) = \exp(x_i^T \beta)$. Using the maximum likelihood estimation for β , the likelihood assumption is

$$\begin{aligned} L(\beta; \mathbf{y}) &= p(\mathbf{y} | \beta) \\ &= \prod_{i=1}^n \frac{\exp[-\exp(x_i^T \beta)] \exp(x_i^T \beta)^{y_i}}{y_i!}. \end{aligned} \quad (2)$$

In practice, working with the natural logarithm of the likelihood function called log-likelihood is often convenient.

$$\begin{aligned} l(\beta; \mathbf{y}) &= \ln L(\beta; \mathbf{y}) \\ &= \sum_{i=1}^n \left(-\exp(x_i^T \beta) + y_i(x_i^T \beta) - \frac{n(1 + y_i)}{2} \right) \end{aligned} \quad (3)$$

$$\hat{\beta} \in \left\{ \arg \max_{\beta} l(\beta; \mathbf{y}) \right\} \quad (4)$$

$\hat{\beta}$ must be estimated to develop the trip distribution model.

2.2. Feature Selection and Parameter Estimation. According to the formulation simplified in Section 2.1, the parameters can be estimated by using a regression method. A regular term L1 is added to the loss function to improve the prediction accuracy and interpretability of the model. This method selects only a subset of the provided covariates called lasso

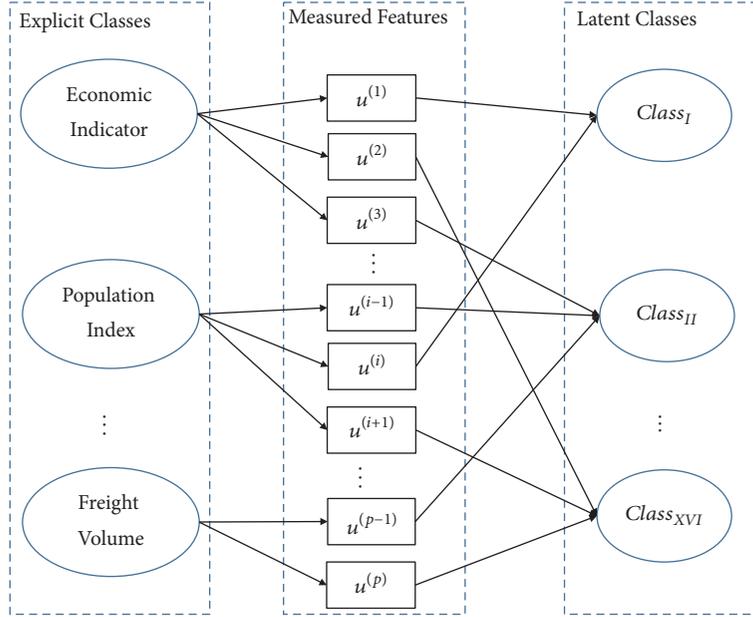


FIGURE 2: Relationship between explicit and latent classes.

regression to be used in the final model. The lost function is expressed as follows:

$$J = -\sum_{i=1}^n \left(-\exp(x_i^T \beta) + y_i (x_i^T \beta) - \frac{n(1 + y_i)}{2} \right) + \lambda \|\beta\|_1. \quad (5)$$

Coordinate descent algorithm is utilized to calculate the lost function.

Step 1 (initialization). Set starting value $\beta^0 = \vec{0}$. The number in parentheses presents the number of iterations.

Step 2 (loss minimization).

for each $k = 1, 2, \dots, N$ **do**

for each $j = 1, 2, \dots, 2p + q + 1$ **do**

Only consider $\beta_j^{(k)}$ as a variable, and the other dimensions of β are constants, formulated as follows:

$$\beta_j^{(k)} = \arg \min_{\beta_j} L(\beta_1^{(k)}, \beta_2^{(k)}, \dots, \beta_{j-1}^{(k)}, \beta_j, \beta_{j+1}^{(k-1)}, \dots, \beta_{2p+q+1}^{(k-1)}). \quad (6)$$

Set $\beta' = (\beta_1^{(k)}, \beta_2^{(k)}, \dots, \beta_{j-1}^{(k)}, \beta_j^{(k-1)}, \beta_{j+1}^{(k-1)}, \dots, \beta_{2p+q+1}^{(k-1)})$.

Update β_j until convergence.

The update iterative formula follows:

$$J \leftarrow J - \eta \frac{\partial J}{\partial \beta_j}, \quad (7)$$

$$\frac{\partial J}{\partial \beta_j} = \sum_{j=1}^n (x_{ij} \exp(x_i^T \beta') + y_i x_{ij}) + \lambda \times \text{sgn}(\beta_j).$$

end for
end for

Step 3 (calculation). Calculate the difference between $\beta^{(k-1)}$ and $\beta^{(k)}$, and the change of L . If both are sufficiently small, then algorithm convergence is achieved. Otherwise, go back to Step 2.

3. Feature Analysis and Interpretation

In Section 2, a trip distribution model is developed according to the Poisson distribution. However, the model is difficult to explain, and the coefficient estimates are not unique if covariates are collinear. The result must be further analyzed and explained as Step 2 in Figure 1. K-means method is applied to classify all the relevant features, including features selected and not selected from Section 2, to analyze and explain the result of feature selection. The clustering result called latent classes can be used to improve the availability of the model. Figure 2 illustrates the relationship between the explicit classes of city features and the latent classes derived from the k-means clustering method. If the nonzero features are unavailable, then they can be replaced with the method in the same latent classes. The experiment is presented in Section 4.

For computational convenience, let $\mathbf{u} = \{u^{(1)}, u^{(2)}, \dots, u^{(p)}\}$ denote all the city features considered in the trip

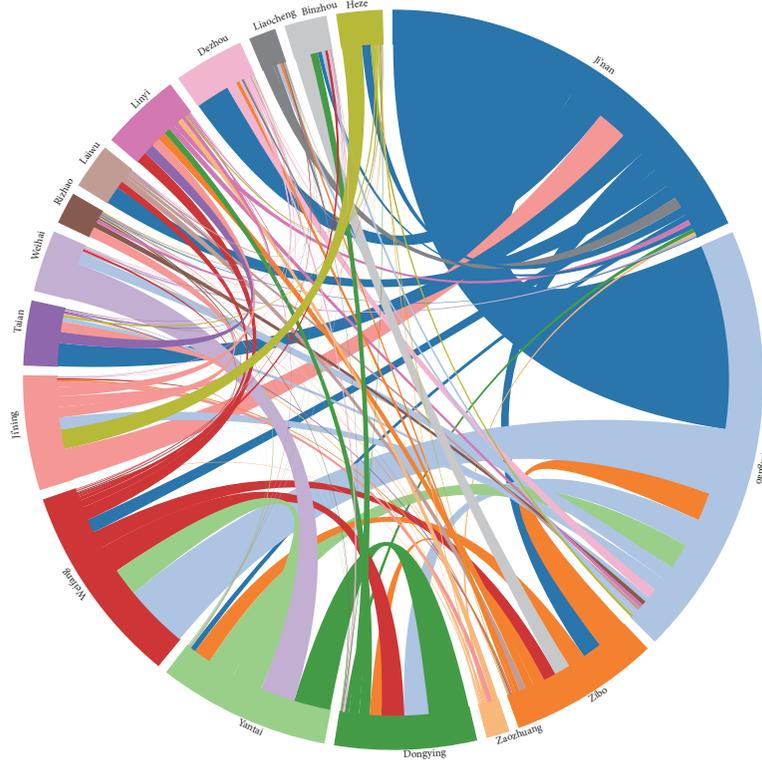


FIGURE 3: Chordal graph of intercity trip distribution. (The proportion of the graph presents the departure number from this city, and the arc presents the relationship between two cities.)

distribution model, and $\boldsymbol{\mu} = \{\mu_1, \mu_2, \dots, \mu_k\}$, which is a subset of \mathbf{u} , denote the nonzero features in the lasso regression. m is the number of cities; thus, the number of trip distribution sample n equals $m^2 - m$, and the dimension of $u^{(i)}$ is m .

K-means clustering method has always been used for cluster analysis in data mining. However, the number of clusters k and the initial set of centroid $\mathbf{m}^{(0)} = \{m_1^{(0)}, m_2^{(0)}, \dots, m_k^{(0)}\}$, which are input parameters, are difficult to determine, and an inappropriate choice may yield unsatisfactory results. In this study, k and $\mathbf{m}^{(0)}$ are determined by the result of lasso regression, which provides a good initial set for the k-means clustering method. The nonzero feature set is the initial set of the centroid, and the number of nonzero features is k .

Input: Feature set: All the city features ($p \times m$): $\mathbf{u} = \{u^{(1)}, u^{(2)}, \dots, u^{(p)}\}$, the dimension of $u^{(i)}$ is m

Initial set of centroid: $\mathbf{m}^{(0)} \leftarrow \boldsymbol{\mu}$

Number of clusters: $k \leftarrow \text{size of } \boldsymbol{\mu}$

Output: Class number l_i of each city feature $u^{(i)}$

for each u_i in feature set \mathbf{u} **do**

Normalize u_i : $u'_i = u_i / \sqrt{\sum_{j=1}^m u_{i,j}^2}$

end for

for each μ_i in feature set $\boldsymbol{\mu}$ **do**

Normalize μ_i : $u^{(i)'} = u^{(i)} / \sqrt{\sum_{j=1}^m (u_j^{(i)})^2}$

end for

repeat until convergence

for each $u^{(i)'}$ in \mathbf{u}' **do**

$$c^{(i)} := \arg \min \|u^{(i)'} - m_j\|^2 \quad (8)$$

end for

for each m_j in \mathbf{m}

$$m_j := \frac{\sum_{i=1}^m 1 \{c^{(i)} = j\} u^{(i)'}}{\sum_{i=1}^m 1 \{c^{(i)} = j\}} \quad (9)$$

end for

4. Data

As one of the richest provinces in China, Shandong owns 17 cities, 15 of which are in the list of top 100 cities in China in 2015. The balanced development in the economy of these cities is convenient for the analysis of intercity trip distribution. As a coastal province in China, Shandong is located in the lower reaches of the Yellow River. The Grand Canal of China enters Shandong from the northwest and leaves on the southwest. An easy communication network of water channels, roads, railways, and air transportation covers all the cities. All the cities are connected by more than 5,000 kilometers of expressways in total. Most cities can reach each other within half a day.

TABLE 1: Set of city characteristics.

Explicit Classes	Concrete Features	Index
Number of Administration Division	number of county units	$u^{(1)}$
	number of municipal districts	$u^{(2)}$
	number of county-level cities	$u^{(3)}$
	number of counties	$u^{(4)}$
	number of town units	$u^{(5)}$
	number of community offices	$u^{(6)}$
	number of townships	$u^{(7)}$
	number of towns	$u^{(8)}$
Per Capita Consumption	per capita consumption of all households (yuan)	$u^{(9)}$
	per capita consumption of rural households (yuan)	$u^{(10)}$
	per capita consumption of urban households (yuan)	$u^{(11)}$
Freight Volume	total freight volume (kiloton)	$u^{(12)}$
	highway freight volume (kiloton)	$u^{(13)}$
	waterway freight volume (kiloton)	$u^{(14)}$
	total freight turnover volume (megaton-kilometer)	$u^{(15)}$
	highway freight turnover volume (megaton-kilometer)	$u^{(16)}$
	waterway freight turnover volume (megaton-kilometer)	$u^{(17)}$
Economic Indicator	gross domestic product (GDP, 100 million yuan)	$u^{(18)}$
	value-added of primary industry (100 million yuan)	$u^{(19)}$
	value-added of secondary industry (100 million yuan)	$u^{(20)}$
	value-added of tertiary industry (100 million yuan)	$u^{(21)}$
	value-added of industry (100 million yuan)	$u^{(22)}$
	per capita GDP (yuan)	$u^{(23)}$
Number of Private Vehicles	total number of private vehicles	$u^{(24)}$
	number of private buses	$u^{(25)}$
	number of private large buses	$u^{(26)}$
	number of private medium buses	$u^{(27)}$
	number of private light buses	$u^{(28)}$
	number of private mini buses	$u^{(29)}$
	number of private trucks	$u^{(30)}$
	number of private large trucks	$u^{(31)}$
	number of private medium trucks	$u^{(32)}$
	number of private light trucks	$u^{(33)}$
	number of private mini trucks	$u^{(34)}$
number of private other vehicles	$u^{(35)}$	
Number of Operation Vehicles	total number of operation vehicles	$u^{(36)}$
	number of operation buses	$u^{(37)}$
	number of operation trucks	$u^{(38)}$
Passenger Volume	total passenger volume (person)	$u^{(39)}$
	highway passenger volume (person)	$u^{(40)}$
	waterway passenger volume (person)	$u^{(41)}$
	total passenger turnover volume (person*kilometer)	$u^{(42)}$
	highway passenger turnover volume (person*kilometer)	$u^{(43)}$
waterway passenger turnover volume (person*kilometer)	$u^{(44)}$	

TABLE 1: Continued.

Explicit Classes	Concrete Features	Index
Population Index	total population (person)	$u^{(45)}$
	total number of households (household)	$u^{(46)}$
	rural population (person)	$u^{(47)}$
	urban population (person)	$u^{(48)}$
Number of Civilian Vehicles	total number of civilian vehicles	$u^{(49)}$
	number of civilian buses	$u^{(50)}$
	number of civilian large buses	$u^{(51)}$
	number of civilian medium buses	$u^{(52)}$
	number of civilian light buses	$u^{(53)}$
	number of civilian mini buses	$u^{(54)}$
	number of civilian trucks	$u^{(55)}$
	number of civilian large trucks	$u^{(56)}$
	number of civilian medium trucks	$u^{(57)}$
	number of civilian light trucks	$u^{(58)}$
	number of civilian mini trucks	$u^{(59)}$
Highway Network Evaluation Index	number of civilian other vehicles	$u^{(60)}$
	highway mileage (kilometer)	$u^{(61)}$
	graded highway mileage (kilometer)	$u^{(62)}$
	advanced course highway mileage (kilometer)	$u^{(63)}$
	expressway mileage (kilometer)	$u^{(64)}$
	all-weather highway mileage (kilometer)	$u^{(65)}$
	highway density (kilometer/million square kilometers)	$u^{(66)}$

The analysis, which is aimed at the highway intraprovince trip distribution of all the 17 cities in Shandong, is based on the dataset of expressway networking toll conducted in the 17 cities of Shandong and statistical data sourced from Shandong Bureau of Statistics. The expressway networking toll data of approximately 280,000 records include information, such as entry time, departure time, name of toll gate, and vehicle type. According to the additional geographical information of toll gates, we can determine the cities where these toll gates belong and calculate the number of intercity trip distributions and the average travel time.

4.1. Trip Distribution OD Matrix. The initial data of OD city matrix is sourced from Shandong intraprovince expressway toll. In Figure 3, chordal graph of intercity trip distribution illustrates the intercity OD matrix, and Figure 3 shows chordal graph of intercity trip distribution.

4.2. Feature Variables. Variables are classified into city characteristics and impedance measurement between the OD pairs. Most of the researches[9] use distance to describe impedance measurement, but we think the intercity average travel time is a better measurement of intercity travel costs, as average travel time not only depends on not only the distance between cities, but also the conditions of traffic. Besides, the dimension of city characteristics which is included in our research is 66 (listed in Table 1). Those are derived from Shandong Bureau of Statistics. All variables are transformed

in logarithmic scale, so that the multiplicative interpretation of the models remains on natural scale.

5. Case Study

This section initially presents the performance of the intercity trip distribution model and clustering method, followed by the result analysis of model and clusters.

5.1. Estimation of Distribution Model. K-fold cross-validation is applied on the intercity trip distribution model, and data are divided into 10 groups in the cross-validation. R^2 score between the observed and predicted outputs is utilized to estimate the goodness-of-fit of the model according to the following equation:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (10)$$

The proposed model and the traditional gravity model are compared. Table 2 summarizes the modeling results for the cross-validation and final model. It shows that the proposed model provides better results than the gravity model. Figure 4 illustrates the goodness-of-fit results of the two models; the measured value is on the horizontal axis, and the predicted value is on the vertical axis. The result predicted by the model is denser than the gravity model. Figure 5 shows the density distribution of the 10 groups of cross-validation. The peak of

TABLE 2: Performance comparison between gravity model and trip distribution model.

	R^2 score of cross validation		R^2 score of model
	Mean	Variance	
Gravity model	0.28	0.46	0.39
Trip distribution model	0.63	0.29	0.69

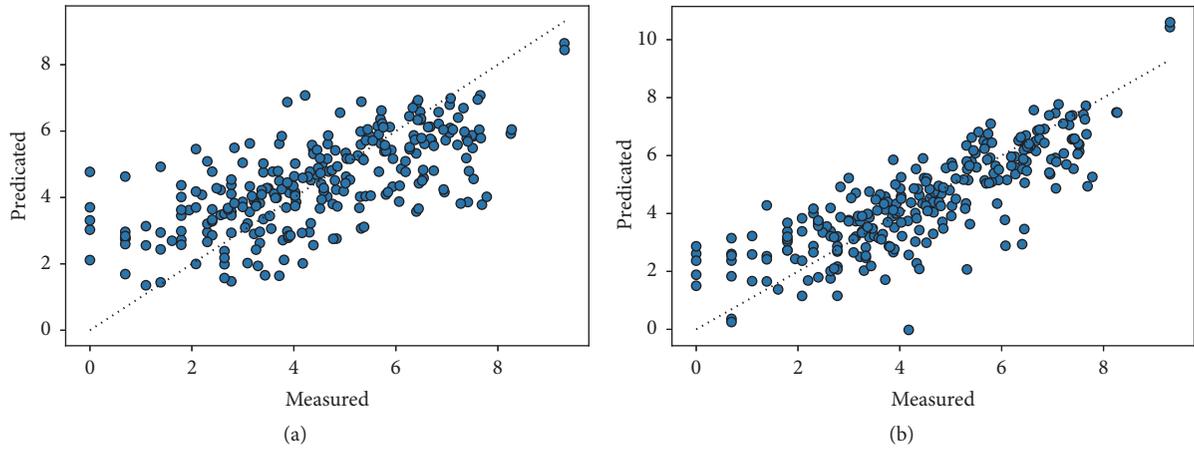


FIGURE 4: Fitting results of predicted and measured value. (a) is the gravity model; (b) is the trip distribution model.

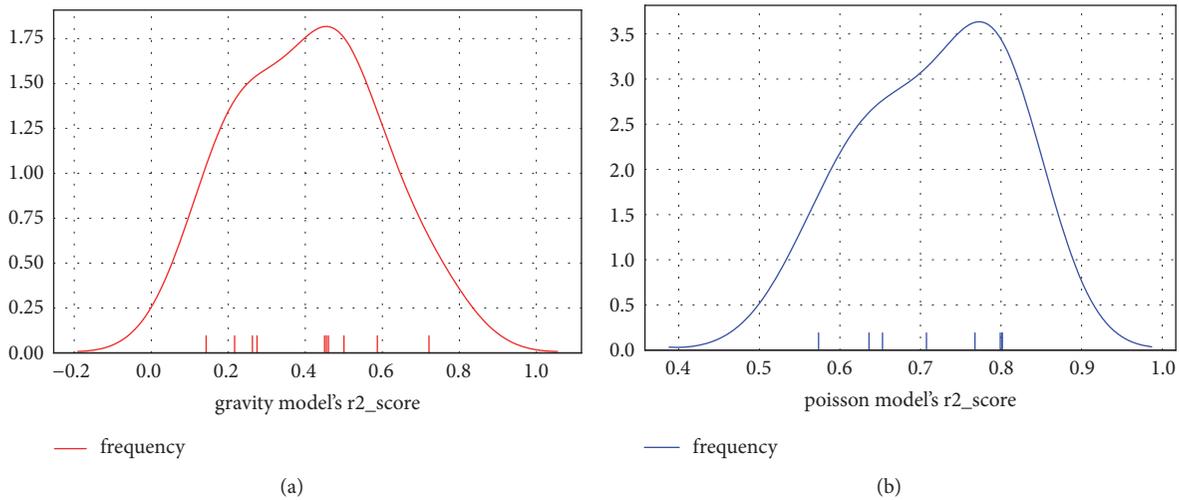


FIGURE 5: Density profile of R^2 score for cross-validation. (a) is the gravity model; (b) is the trip distribution model.

the gravity model's curve is approximately 0.3 and that of the intercity trip distribution model is approximately 0.7.

5.2. *Validation of Parameter Selectivity.* A random selection policy is applied on the clustering method to estimate the clustering methods and validate the selectivity of the parameters. The R^2 score is used to validate the result. According to the clustering results, the features relative to the nonzero features are selected to take the place of the nonzero features to fit the model, and R^2 score is used to estimate the random sampling results. This process is repeated 500 times to estimate the clustering methods. Table 3 and Figure 6 indicate that the mean score of the k-means is high and that the variance is low.

TABLE 3: Performance of k-means clustering method.

	Mean	Variance
K-means clustering	0.619	0.00131

5.3. *Case-Specific Inferences.* The results of the model are applied to the nonzero parameters, and the effects of these parameters on the expected trip distribution are additive on the logarithmic scale. Thus, the interpretation of the result is explicit. The positive parameter values correspond to an increasing effect, whereas values less than 0 mean a decreasing effect. Table 4 shows the parameters and classes of nonzero features and Table 5 shows the k-means cluster

TABLE 4: Parameters and classes of nonzero features.

	Non-zero features	Bi	Class
Departure city	number of civilian medium buses	0.584554605	I
	number of civilian large buses	-0.122094917	II
	number of civilian mini buses	0.024162375	XVI
	total freight volume (kiloton)	0.0971418	V
	highway freight volume (kiloton)	0.309173839	V
	number of municipal districts	0.157563009	I
	highway passenger volume	-0.154281739	IX
	waterway passenger volume	-0.067118113	X
	waterway passenger turnover volume	0.148509538	XI
	expressway mileage (kilometer)	0.367166985	I
	number of private large buses	0.374970381	XIII
	number of private mini buses	-0.026213636	XIV
	number of private large trucks	-0.068492342	XV
Destination city	number of civilian medium buses	0.341385231	I
	number of civilian large buses	-0.000768006	II
	number of civilian mini buses	0.144286726	XVI
	highway freight volume (kiloton)	0.276919115	V
	number of municipal districts	0.337366294	I
	number of counties	0.013490561	VII
	number of towns	-0.065913292	VIII
	highway passenger volume	-0.023081118	IX
	waterway passenger volume	-0.097300382	X
	waterway passenger turnover volume	0.194774387	XI
	expressway mileage (kilometer)	0.999713089	I
	number of private large buses	0.249855688	XIII
	number of private mini buses	-0.699258902	XIV
number of private large trucks	0.145211855	XVI	
Average travel time		-1.76090707	

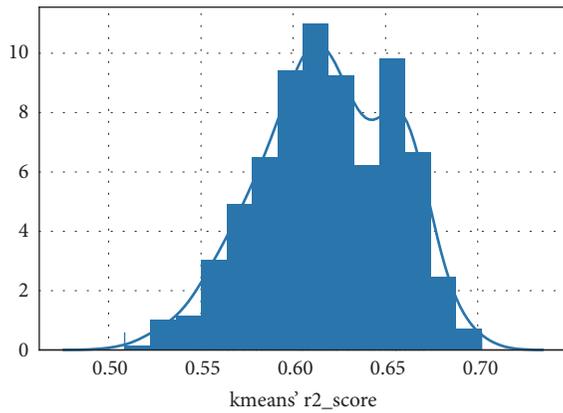


FIGURE 6: Density profile of R^2 score through random sampling.

results. The effects are analyzed through the combination of the results of the model and cluster.

Class I, which presents the city’s economic level, shows the positive effect of the intercity trip distribution on origin and destination cities. This result is intuitively consistent because the cities with developed economies interact actively with other cities. Class IV, which is relevant about the

total and highway freight volumes, also shows the positive effects. By contrast, the passenger volume has almost no effect (Classes IX and XII). The phenomenon presents that a large part of the intercity trip distribution of Shandong freeway is freight, which is relative to the highly developed rail network and even some roadway passenger volume (Class X). Class XIV, which presents the freight turnover volume, shows a negative effect because the negative effect of distance has greater influence than the positive effect of freight volume. The number of counties (Class VII) shows a positive effect, which corresponds to the city size. The result most inconsistent with previous research is Class IV, which presents the population and has no effect on the intercity trip distribution. The result can be explained by the complexity of the intercity trip distribution, and it can prove the real effects are city economy and city size. The average time cost is the core traffic impedance factor. Similar to previous studies, the trip distribution shows a power law time cost decay effect.

6. Conclusions

In this study, a trip distribution modeling method from complex city characteristics was investigated based on Poisson distribution theory, and the application of lasso regression

TABLE 5: Results of k-means clustering method.

Number of latent classes	City characteristics in the classes
I	number of civilian large buses number of civilian medium buses number of municipal districts number of community offices expressway mileage (kilometer) GDP (100 million yuan) value-added of secondary industry (100 million yuan) value-added of tertiary industry (100 million yuan) value-added of industry (100 million yuan)
II	number of civilian large trucks number of operation vehicles number of operation trucks
III	highway freight turnover volume (kiloton)
IV	number of county units number of town units total population (person) total number of households (household) urban population (person)
V	number of civilian trucks total freight volume (kiloton) highway freight volume (kiloton)
VI	per capita consumption of all households (yuan) per capita consumption of rural households (yuan) per capita consumption of urban households (yuan) per capita GDP (yuan)
VII	number of counties
VIII	number of civilian other vehicles number of towns highway mileage (kilometer) graded highway mileage (kilometer) all-weather highway mileage (kilometer) highway density (kilometer/million square kilometers) number of private other vehicles rural population (person)
IX	total passenger volume (person) highway passenger volume (person) advanced course highway mileage (kilometer) value-added of primary industry (100 million yuan)
X	waterway passenger volume (person)
XI	waterway passenger turnover volume (person)
XII	total number of civilian vehicles number of civilian buses number of civilian light buses umber of operation buses total passenger turnover volume (person-kilometer) highway passenger turnover volume (person-kilometer) total number of private vehicles number of private buses number of private medium buses number of private light buses
XIII	number of private large buses
XIV	number of civilian medium trucks number of civilian light trucks number of civilian mini trucks number of private medium buses number of private mini buses total freight turnover volume (megaton-kilometer) number of private trucks number of private light trucks

TABLE 5: Continued.

Number of latent classes	City characteristics in the classes
XV	number of private large trucks
XVI	number of civilian mini buses number of private mini trucks

for feature selection and parameter estimation was presented. Finally, a clustering algorithm was applied on the model. The conclusion drawn was that the nonzero features could be replaced with other relevant features.

Traditional approaches, namely, gravity and destination choice models, consider the population of departure and destination cities. However, population was not a statistically significant variable for our models. The methodology can solve the problem of previous studies on the selection of explanatory variables. Moreover, it provides a new approach that selects the variables by machine learning algorithms, not by experts. Besides, the model performs well in generalization because the features selected are alternative. A comparison of different models show that our method is more effective than the gravity model. Furthermore, our method has no prerequisite. The trip distribution model based on Poisson model, which has good fitness and wide inferential capabilities, can be an effective and alternative to the gravity model.

Given the limitation of the difficulty of data collection for intercity distribution, the proof of the proposed method is insufficient, and city characteristics can be further diversified. Future research may collect more features, further verify the availability of the model, and quantitatively analyze the relationship between features.

Data Availability

Shandong statistic data was sourced from Shandong Bureau of Statistics.

Conflicts of Interest

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

Acknowledgments

This research was supported by the National Key R&D program under Grant no. 2016YFC0801700, Beijing Municipal Science and Technology Project no. Z171100000917016, and the National Natural Science Foundation Project under Grant no. U1636208.

References

- [1] L. Zhang, F. Southworth, C. Xiong, and A. Sonnenberg, "Methodological Options and Data Sources for the Development of Long-Distance Passenger Travel Demand Models: A Comprehensive Review," *Transport Reviews*, vol. 32, no. 4, pp. 399–433, 2012.
- [2] D. M. Yao and D. Y. Jiang, "The research on area classification in intercity rail traffic forecast based on cluster analysis," *Advanced Materials Research*, vol. 945-949, pp. 3296–3299, 2014.
- [3] Y. Xiao, F. Wang, Y. Liu, and J. Wang, "Reconstructing Gravitational Attractions of Major Cities in China from Air Passenger Flow Data, 2001-2008: A Particle Swarm Optimization Approach," *The Professional Geographer*, vol. 65, no. 2, pp. 265–282, 2013.
- [4] M. M. M. Abdel-Aal, "Calibrating a trip distribution gravity model stratified by the trip purposes for the city of Alexandria," *Alexandria Engineering Journal*, vol. 53, no. 3, pp. 677–689, 2014.
- [5] S. Mishra, Y. Wang, X. Zhu et al., "Comparison Between Gravity and Destination Choice Models for Trip Distribution in Maryland," in *Proceedings of the Transportation Research Board Meeting*, 2013.
- [6] M. Lenormand, A. Bassolas, and J. J. Ramasco, "Systematic comparison of trip distribution laws and models," *Journal of Transport Geography*, vol. 51, pp. 158–169, 2016.
- [7] X. Yan, W. Wang, Z. Gao, and Y. Lai, "Universal model of individual and population mobility on diverse spatial scales," *Nature Communications*, vol. 8, no. 1, 2017.
- [8] S. Bekhor, Y. Cohen, and C. Solomon, "Evaluating long-distance travel patterns in Israel by tracking cellular phone positions," *Journal of Advanced Transportation*, vol. 47, no. 4, pp. 435–446, 2013.
- [9] C. S. Pitombo, A. D. de Souza, and A. Lindner, "Comparing decision tree algorithms to estimate intercity trip distribution," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 16–32, 2017.
- [10] M. Rasouli and H. Nikraz, "Application of generalised regression neural networks in trip distribution modelling," *Road & Transport Research*, vol. 23, no. 3, pp. 13–25, 2014.
- [11] K. Perrakis, D. Karlis, M. Cools, D. Janssens, K. Vanhoof, and G. Wets, "A Bayesian approach for modeling origin-destination matrices," *Transportation Research Part A: Policy and Practice*, vol. 46, no. 1, pp. 200–212, 2012.
- [12] M. B. Rojas, E. Sadeghvaziri, and X. Jin, "Comprehensive review of travel behavior and mobility pattern studies that used mobile phone data," *Transportation Research Record*, vol. 2563, pp. 71–79, 2016.

Research Article

An Improved Method for Cross-Project Defect Prediction by Simplifying Training Data

Peng He ^{1,2}, Yao He,¹ Lvjun Yu,¹ and Bing Li ³

¹School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

²Hubei Province Engineering Technology Research Center for Educational Informationization, Wuhan 430062, China

³School of Computer, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Peng He; penghe@whu.edu.cn

Received 10 December 2017; Revised 23 February 2018; Accepted 15 April 2018; Published 7 June 2018

Academic Editor: Dingli Yu

Copyright © 2018 Peng He 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.

Cross-project defect prediction (CPDP) on projects with limited historical data has attracted much attention. To the best of our knowledge, however, the performance of existing approaches is usually poor, because of low quality cross-project training data. The objective of this study is to propose an improved method for CPDP by simplifying training data, labeled as *TDSelector*, which considers both the similarity and the number of defects that each training instance has (denoted by *defects*), and to demonstrate the effectiveness of the proposed method. Our work consists of three main steps. First, we constructed *TDSelector* in terms of a linear weighted function of instances' similarity and *defects*. Second, the basic defect predictor used in our experiments was built by using the Logistic Regression classification algorithm. Third, we analyzed the impacts of different combinations of similarity and the normalization of *defects* on prediction performance and then compared with two existing methods. We evaluated our method on 14 projects collected from two public repositories. The results suggest that the proposed *TDSelector* method performs, on average, better than both baseline methods, and the AUC values are increased by up to 10.6% and 4.3%, respectively. That is, the inclusion of *defects* is indeed helpful to select high quality training instances for CPDP. On the other hand, the combination of Euclidean distance and linear normalization is the preferred way for *TDSelector*. An additional experiment also shows that selecting those instances with more bugs directly as training data can further improve the performance of the bug predictor trained by our method.

1. Introduction

Software defect prediction is one of the most active research topics in Software Engineering. Most early studies usually trained predictors (also known as prediction models) from the historical data on software defects/bugs in the same software project and predicted defects in its upcoming release versions [1]. This approach is referred to as Within-Project Defect Prediction (WPDP). However, WPDP has an obvious drawback when a project has limited historical defect data.

To address the above issue, researchers in this field have attempted to apply defect predictors built for one project to other projects [2–7]. This method is termed Cross-Project Defect Prediction (CPDP). The main purpose of CPDP is to predict defect-prone instances (such as classes) in a project based on the defect data collected from other projects on

those public software repositories like PROMISE (<http://openscience.us/repo/>). The feasibility and potential usefulness of cross-project predictors built with a number of software metrics have been validated [1, 3, 5, 6], but how to improve the performance of CPDP models is still an open issue.

Peters et al. [5] argued that selecting appropriate training data from a software repository became a major issue for CPDP. Moreover, some researchers also suggested that the success rate of CPDP models could be drastically improved when using a suitable training dataset [1, 7]. That is to say, the selection of training data of quality could be a key breakthrough on the above issue. Thus, the construction of an appropriate training dataset gathered from a large number of projects on public software repositories is indeed a challenge for CPDP [7].

As far as we know, although previous studies on CPDP have taken different types of software metrics into account during the process of selecting relevant training samples, none of them considered the number of defects contained in each sample (denoted by *defects*). But in fact, we argue that it is also an important factor to consider. Fortunately, some studies have empirically demonstrated the relevance of *defects* to prediction. For example, “modules with faults in the past are likely to have faults in the future” [8], “17% to 54% of the high-fault files of release i are still high-fault in release $i + 1$ ” [9], “cover 73%–95% of faults by selecting 10% of the most fault prone source code file” [10], and “the number of defects found in the previous release of file correlates with its current defect count on a high level” [11].

Does the selection of training data considering *defects* improve the performance of CPDP models? If the answer is “Yes”, on the one hand, it is helpful to validate the feasibility of CPDP; on the other hand, it will contribute to better software defect predictors by making full use of those defect datasets available on the Internet.

The objective of our work is to propose an improved method of training data selection for CPDP by introducing the information of *defects*. Unlike the prior studies similar to our work, such as [5, 12], which focus mainly on the similarity between instances from training set and test set, this paper gives a comprehensive account of two factors, namely, similarity and *defects*. Moreover, the proposed method, called *TDSelector*, can automatically optimize their weights to achieve the best result. In brief, our main contributions to the current state of research on CPDP are summarized as follows.

(1) Considering both similarity and *defects*, we proposed a simple and easy-to-use training data selection method for CPDP (i.e., *TDSelector*), which is based on an improved scoring scheme that ranks all possible training instances. In particular, we designed an algorithm to calculate their weights automatically, so as to obtain the best prediction result.

(2) To validate the effectiveness of our method, we conducted an elaborate empirical study based on 15 datasets collected from PROMISE and AEEEM (<http://bug.inf.usi.ch>), and the experimental results show that, in a specific CPDP scenario (i.e., *many-to-one* [13]), the *TDSelector*-based defect predictor outperforms its rivals that were built with two competing methods in terms of prediction precision.

With these technical contributions, our study could complement previous work on CPDP with respect to training data selection. In particular, we provide a reasonable scoring scheme as well as a more comprehensive guideline for developers to choose appropriate training data to train a defect predictor in practice.

The rest of this paper is organized as follows. In Section 2, we reviewed the related work of this topic; Section 3 presents the preliminaries to our work; Section 4 describes the proposed method *TDSelector*, Section 5 introduces our experimental setup, and Section 6 shows the primary experimental results; a detailed discussion of some issues including potential threats to the validity of our study is presented in Section 7; in the end, Section 8 summarizes this paper and presents our future work.

2. Related Work

2.1. Cross-Project Defect Prediction. Many studies were carried out to validate the feasibility of CPDP in the last five years. For example, Turhan et al. [12] proposed a cross-company defect prediction approach using defect data from other companies to build predictors for target projects. They found that the proposed method increased the probability of defect detection at the cost of increasing false positive rate. Ni et al. [14] proposed a novel method called FeSCH and designed three ranking strategies to choose appropriate features. The experimental results show that FeSCH can outperform WPDP, ALL, and TCA+ in most cases, and its performance is independent of the used classifiers. He et al. [15] compared the performance between CPDP and WPDP using feature selection techniques. The results indicated that for reduced training data WPDP obtained higher precision, but CPDP in turn achieved a better recall or F -measure. Some researchers have also studied the performance of CPDP based on ensemble classifiers and then validated their effects on this issue [16, 17].

Ryu et al. [18] proposed a transfer cost-sensitive boosting method by considering both distributional characteristics and the class imbalance for CPDP. The results show that their method significantly improves CPDP performance. They also [19] proposed a multiobjective naive Bayes learning technique under CPDP environments by taking into account the class-imbalance contexts. The results indicated that their approaches performed better than the single-objective ones and WPDP models. Li et al. [20] compared some famous data filters and proposed a method called HSBF (hierarchical select-based filter) to improve the performance of CPDP. The results demonstrate that the data filter strategy can indeed improve the performance of CPDP significantly. Moreover, when using appropriate data filter strategy, the defect predictor built from cross-project data can outperform the predictor learned by using within-project data.

Zhang et al. [21] proposed a universal CPDP model, which was built using a large number of projects collected from SourceForge (<https://sourceforge.net/>) and Google Code (<https://code.google.com/>). Their experimental results showed that it was indeed comparable to WPDP. Furthermore, CPDP is feasible for different projects that have heterogeneous metric sets. He et al. [22] first proposed a CPDP-IFS approach based on the distribution characteristics of both source and target projects to overcome this problem. Nam and Kim [23] then proposed an improved method called HDP, where metric selection and metric matching were introduced to build a defect predictor. Their empirical study on 28 projects showed that about 68% of predictions using the proposed approach outperformed or were comparable to WPDP with statistical significance. Jing et al. [24] proposed a unified metric representation (UMR) for heterogeneous defect data. More researches can be found in [25–27]. The experiments on 14 public heterogeneous datasets from four different companies indicated that the proposed approach was more effective in addressing the problem.

2.2. Training Data Selection for CPDP. As mentioned in [5, 28], a fundamental issue for CPDP is to select the most appropriate training data for building quality defect predictors. He et al. [29] discussed this problem in detail from the perspective of data granularity, i.e., release level and instance level. They presented a two-step method for training data selection. The results indicated that the predictor built based on naive Bayes could achieve fairly good performance when using the method together with Peter filter [5]. Porto and Simao [30] proposed an Instance Filtering method by selecting the most similar instances from the training dataset, and the experimental results of 36 versions of 11 open-source projects show that the defect predictor built from cross-project data selected by Feature Selection and Instance Filtering can have generally better performances both in classification and in ranking.

With regard to the data imbalance problem of defect datasets, Jing et al. [31] introduced an effective feature learning method called SDA to provide effective solutions for class-imbalance problems of both within-project and cross-project types, by employing the semisupervised transfer component analysis (SSTCA) method to make the distributions of source and target data consistent. The results indicated that their method greatly improved WPDP and CPDP performance. Ryu et al. [32] proposed a method of hybrid instance selection using nearest neighbor (HISNN). Their results suggested that those instances, which had strong local knowledge, could be identified via nearest neighbors with the same class label. Poon et al. [33] proposed a credibility theory based naive Bayes (CNB) classifier to establish a novel reweighting mechanism between the source projects and target projects, so that the source data could simultaneously adapt to the target data distribution and retain its own pattern. The experimental results demonstrate the significant improvement in terms of the performance metrics considered achieved by CNB over other CPDP approaches.

The above-mentioned existing studies aimed at reducing the gap in prediction performance between WPDP and CPDP. Although they are making progress towards the goal, there is clearly a lot of room for improvement. For this reason, in this paper, we proposed a selection approach to training data based on an improved strategy for instance ranking instead of a single strategy for similarity calculation, which was used in many prior studies [1, 5, 7, 12].

3. Preliminaries

In our context, a defect dataset S contains m instances, which is represented as $S = \{I_1, I_2, \dots, I_m\}$. Instance I_i is an object class represented as $I_i = \{f_{i1}, f_{i2}, \dots, f_{in}\}$, where f_{ij} is the j th metric value of instance I_i and n is the number of metrics (also known as features). Given a source dataset S_s and a target dataset S_t , CPDP aims to perform a prediction in S_t using the knowledge extracted from S_s , where $S_s \neq S_t$ (see Figure 1(a)). In this paper, source and target datasets have the same set of metrics, and they may differ in distributional characteristics of metric values.

To improve the performance of CPDP, several strategies used to select appropriate training data have been put forward

(see Figure 1(b)); e.g., Turhan et al. [12] filtered out those irrelevant training instances by returning k -nearest neighbors for each test instance.

3.1. An Example of Training Data Selection. First, we introduce a typical method for training data selection at the instance level, and a simple example is used to illustrate this method. For the strategy for other levels of training data selection, such as at the release level, please refer to [7].

Figure 2 shows a training set S_s (including five instances) and a test set S_t (including an instance). Here, each instance contains five metrics and a classification label (i.e., 0 or 1). An instance is defect-free (label = 0) only if its *defects* are equal to 0; otherwise, it is defective (label = 1). According to the k -nearest neighbor method based on Euclidean distance, we can rank all the five training instances in terms of their distances from the test instance. Due to the same nearest distance from test instance I_{test} , it is clear that three instances I_1 , I_2 , and I_5 are suitable for use as training instances when k is set to 1. For the three instances, I_2 and I_5 have the same metric values, but I_2 is labeled as a defective instance because it contains a bug. In this case, I_1 will be selected with the same probability as that of I_2 , regardless of the number of defects they include.

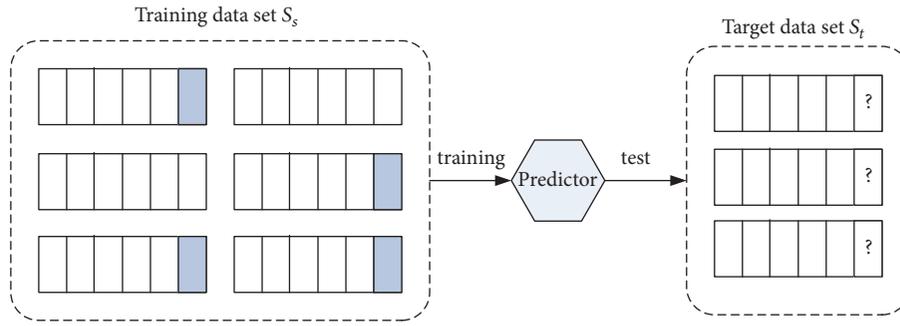
In this way, those instances most relevant to the test one can be quickly determined. Clearly, the goal of training data selection is to preserve the representative training instances in S_s as much as possible.

3.2. General Process of Training Data Selection. Before presenting our approach, we describe a general selection process of training data, which consists of three main steps: TDS (training dataset) setup, ranking, and duplicate removal.

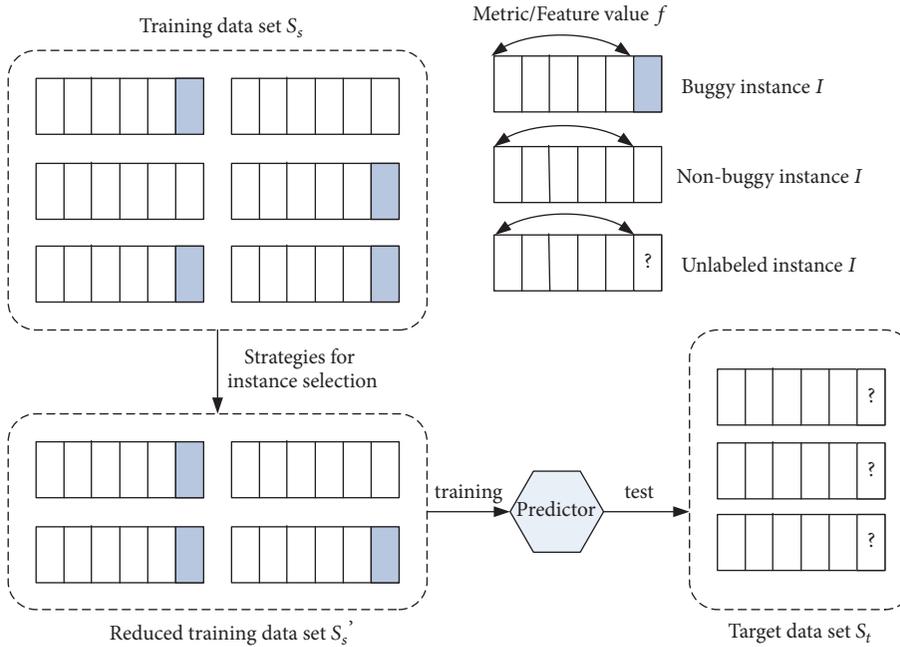
TDS Setup. For each target project with little historical data, we need to set up an initial TDS where training data are collected from other projects. To simulate this scenario of CPDP, in this paper, any defect data from the target project must be excluded from the initial TDS. Note that different release versions of a project actually belong to the same project. A simple example is visualized in Figure 3.

Ranking. Once the initial TDS is determined, an instance will be treated as a metric vector I , as mentioned above. For each test instance, one can calculate its relevance to each training instance and then rank these training instances in terms of their similarity based on software metrics. Note that a wide variety of software metrics, such as source code metrics, process metrics, previous defects, and code churn, have been used as features for CPDP approaches to improve their prediction performance.

Duplicate Removal. Let l be the size of test set. For each test instance, if we select its k -nearest neighbors from the initial TDS, there are a total of $k \times l$ candidate training instances. Considering that these selected instances may not be unique (i.e., a training instance can be the nearest neighbor of multiple test instances), after removing the duplicate ones, they form the final training set which is a subset of the initial TDS.



(a) General CPDP

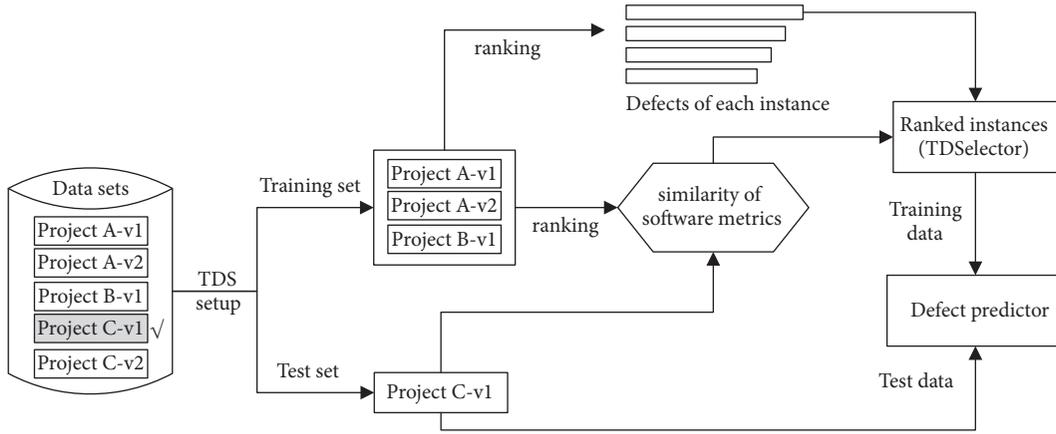


(b) Improved CPDP using training data selection

FIGURE 1: Two CPDP scenarios used in this paper.

		f_1	f_2	f_3	f_4	Label (defects)										
S_s	I_1	0.1	0	0.5	0	1(3)	distance(I_i, I_{test})	<table border="1"> <thead> <tr> <th>rank</th> <th>instance</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>I_1, I_2, I_5</td> </tr> <tr> <td>2</td> <td>I_4</td> </tr> <tr> <td>3</td> <td>I_3</td> </tr> </tbody> </table>	rank	instance	1	I_1, I_2, I_5	2	I_4	3	I_3
	rank	instance														
	1	I_1, I_2, I_5														
	2	I_4														
	3	I_3														
I_2	0.1	0	0	0.5	1(1)											
I_3	0.4	0.3	0	0.1	0											
I_4	0	0	0.4	0	0											
I_5	0.1	0	0	0.5	0											
S_t	I_{test}	0.1	0	0.5	0.5	?										

FIGURE 2: An example of the selection of training instances.

FIGURE 3: The overall structure of *TDSSelector* for CPDP.

4. Our Approach *TDSSelector*

To improve the prediction performance of CPDP, we leverage the following observations.

Similar Instances. Given a test instance, we can examine its similar training instances that were labeled before. The defect proneness shared by similar training instances could help us identify the probability that a test instance is defective. Intuitively, two instances are more likely to have the same state if their metric values are very similar.

Number of Defects (*defects*). During the selection process, when several training instances have the same distance from a test instance, we need to determine which one should be ranked higher. According to our experiences in software defect prediction and other researchers' studies on the quantitative analysis of previous defect prediction approaches [34, 35], we believe that more attention should be paid to those training instances with more defects in practice.

The selection of training data based on instance similarity has been used in some prior studies [5, 12, 35]. However, to the best of our knowledge, the information about *defects* has not been fully utilized. So, in this paper, we attempt to propose a training data selection approach combining such information and instance similarity.

4.1. Overall Structure of *TDSSelector*. Figure 3 shows the overall structure of the proposed approach to training data selection named *TDSSelector*. Before selecting appropriate training data for CPDP, we have to set up a test set and its corresponding initial TDS. For a given project treated as the test set, all the other projects (except the target project) available at hand are used as the initial TDS. This is the so-called *many-to-one* (M2O) scenario for CPDP [13]. It is quite different from the typical *O2O* (*one-to-one*) scenario, where only one randomly selected project is treated as the training set for a given target project (namely, test set).

When both of the sets are given, the ranks of training instances are calculated based on the similarity of software metrics and then returned for each test instance. For the

initial TDS, we also collect each training instance's *defects* and thus rank these instances by their *defects*. Then, we rate each training instance by combining the two types of ranks in some way and identify the top-*k* training instances for each test instance according to their final scores. Finally, we use the predictor trained with the final TDS to predict defect proneness in the test set. We describe the core component of *TDSSelector*, namely, scoring scheme, in the following subsection.

4.2. Scoring Scheme. For each instance in training set and test set which is treated as a vector of features (namely, software metrics), we calculate the similarity between them in terms of similarity index (such as cosine similarity, Euclidean distance, and Manhattan distance, as shown in Table 1). Training instances are then ranked by the similarity between each of them and a given test instance.

For instance, the cosine similarity between a training instance I_p and the target instance I_q is computed via their vector representations, described as follows:

$$\text{Sim}(I_p, I_q) = \frac{\vec{I}_p \cdot \vec{I}_q}{\|I_p\| \times \|I_q\|} = \frac{\sum_{i=1}^n (f_{pi} \times f_{qi})}{\sqrt{\sum_{i=1}^n f_{pi}^2} \times \sqrt{\sum_{i=1}^n f_{qi}^2}}, \quad (1)$$

where \vec{I}_p and \vec{I}_q are the metric vectors for I_p and I_q , respectively, and f_{*i} represents the i th metric value of instance I_* .

Additionally, for each training instance, we also consider the factor *defects* in order to further enrich the ranking of its relevant instances. The assumption here is that the more the previous *defects*, the richer the information of an instance. So, we propose a scoring scheme to rank those candidate training instances, defined as below:

$$\text{Score}(I_p, I_q) = \alpha \text{Sim}(I_p, I_q) + (1 - \alpha) N(\text{defect}_p), \quad (2)$$

where defect_p represents the *defects* of I_p , α is a weighting factor ($0 \leq \alpha \leq 1$) which is learned from training data using Algorithm 1 (see Algorithm 1), and $N(\text{defect}_p)$ is a function used to normalize *defects* with values ranging from 0 to 1.

Optimizing the parameter α

Input:

- (1) Candidate TDS $S_s = \{I_{s1}, I_{s2}, \dots, I_{sm}\}$, test set $S_t = \{I_{t1}, I_{t2}, \dots, I_{tl}\}$ ($m > l$),
- (2) $defect_s = \{defect(I_{s1}), defect(I_{s2}), \dots, defect(I_{sm})\}$, and $k = 10$

Output:

- (3) α ($\alpha \in [0, 1]$)

Method:

- (4) Initialize $\alpha = 0$, $S_s(\alpha) = \emptyset$;
- (5) **While** ($\alpha \leq 1$) **do**
- (6) **For** $i = 1; i \leq l; i++$
- (7) **For** $j = 1; j \leq m; j++$
- (8) $Score(I_{ti}, I_{sj}) = \alpha Sim(I_{ti}, I_{sj}) + (1 - \alpha)N(defect(I_{sj}))$;
- (9) **End For**
- (10) $descSort(\{Score(I_{ti}, I_{sj}) \mid j = 1 \dots m\})$; //sort m training instances in descending order
- (11) $S_s(\alpha) = S_s(\alpha) \cup \{\text{Top-}k \text{ training instances}\}$; //select the top k instances
- (12) **End For**
- (13) $AUC \leftarrow S_s(\alpha) \xrightarrow{CPDP} S_t$; //prediction result
- (14) $\alpha = \alpha + 0.1$;
- (15) **End While**
- (16) **Return** ($\alpha \mid \max_{\alpha} AUC$);

ALGORITHM 1: Algorithm of parameter optimization.

TABLE 1: Similarity indexes and normalization methods used in this paper.

Similarity	
Cosine	$\cos(X, Y) = \frac{\sum_{k=1}^n x_k y_k}{\sqrt{\sum_{k=1}^n x_k^2} \sqrt{\sum_{k=1}^n y_k^2}}$
Euclidean distance	$d(X, Y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$
Manhattan distance	$d(X, Y) = \sum_{k=1}^n x_k - y_k $
Normalization	
Linear	$N(x) = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$
Logistic	$N(x) = \frac{1}{1 + e^{-x}} - 0.5$
Square root	$N(x) = 1 - \frac{1}{\sqrt{1 + x}}$
Logarithmic	$N(x) = \log_{10}(x + 1)$
Inverse cotangent	$N(x) = \frac{\arctan(x) * 2}{\pi}$

Normalization is a commonly used data preprocessing technique for mathematics and computer science [36]. Graf and Borer [37] have confirmed that normalization can improve prediction performance of classification models. For this reason, we normalize the *defects* of training instances when using *TDSelector*. As you know, there are many normalization methods. In this study, we introduce five typical normalization methods used in machine learning [36, 38]. The description and formulas of the five normalization methods are listed in Table 1.

For each test instance, the top- k training instances ranked in terms of their scores will be returned. Hence, the final

TDS is composed by merging the sets of the top- k training instances for each test instance when those duplicate instances are removed.

5. Experimental Setup

5.1. Research Questions. Our experiments were conducted to find empirical evidence that answers the following three research questions.

RQ1: Does the Consideration of Defects Improve the Performance of CPDP? Unlike the previous methods [1, 5, 7, 12, 29], *TDSelector* ranks candidate training instances in terms of both *defects* and metric-based similarity. To evaluate the effectiveness of the proposed method considering the additional information of *defects*, we tested *TDSelector* according to the experimental data described in Section 5.2. According to (2), we also empirically analyzed the impact of the parameter α on prediction results.

RQ2: Which Combination of Similarity and Normalization Is More Suitable for TDSelector? Equation (2) is comprised of two parts, namely, similarity and the normalization of *defects*. For each part, several commonly used methods can be adopted in our context. To fully take advantage of *TDSelector*, one would wonder which combination of similarity and normalization should be chosen. Therefore, it is necessary to compare the effects of different combinations of similarity and normalization methods on prediction results and to determine the best one for *TDSelector*.

RQ3: Can TDSelector-Based CPDP Outperform the Baseline Methods? Cross-project prediction has attracted much research interest in recent years, and a few CPDP approaches using training data selection have also been proposed, e.g.,

Peter filter based CPDP [5] (labeled as baseline1) and TCA+ (Transfer Component Analysis) based CPDP [39] (labeled as baseline2). To answer the third question, we compared *TDSelector*-based CPDP proposed in this paper with the above two state-of-the-art methods.

5.2. Data Collection. To evaluate the effectiveness of *TDSelector*, in this paper, we used 14 open-source projects written in Java on two online public software repositories, namely, PROMISE [40] and AEEEM [41]. The data statistics of the 14 projects in question are presented in Table 2, where *#Instance* and *#Defect* are the numbers of instances and defective instances, respectively, and *% Defect* is the proportion of defective instances to the total number of instances. Each instance in these projects represents a file of object class and consists of two parts, namely, software metrics and *defects*.

The first repository, PROMISE, was collected by Jureczko and Spinellis [40]. The information of defects and 20 source code metrics for the projects on PROMISE have been validated and used in several previous studies [1, 7, 12, 29]. The second repository, AEEEM, was collected by D'Ambros et al. [41], and each project on it has 76 metrics, including 17 source code metrics, 15 change metrics, 5 previous defect metrics, 5 entropy-of-change metrics, 17 entropy-of-source-code metrics, and 17 churn-of-source-code metrics. AEEEM has been successfully used in [23, 39].

Before performing a cross-project prediction, we need to determine a target dataset (test set) and its candidate TDS. For PROMISE (10 projects), each one in the 10 projects was selected to be the target dataset once, and then we set up a candidate TDS for CPDP, which excluded any data from the target project. For instance, if Ivy is selected as test project, data from the other nine projects was used to construct its initial TDS.

5.3. Experiment Design. To answer the three research questions, our experimental procedure, which is designed under the context of M2O in the CPDP scenario, is described as follows.

First, as with many prior studies [1, 5, 15, 35], all software metric values in training and test sets were normalized by using the *Z*-score method, because these metrics are different in the scales of numerical values. For the 14 projects on AEEEM and PROMISE, their numbers of software metrics are different. So, the training set for a given test set was selected from the same repository.

Second, to examine whether the consideration of *defects* improves the performance of CPDP, we compared our approach *TDSelector* with NoD, which is a baseline method considering only the similarity between instances, i.e., $\alpha = 1$ in (2). Since there are three similarity computation methods used in this paper, we designed three different *TDSelectors* and their corresponding baseline methods based on similarity indexes. The prediction results of each method in question for the 15 test sets were analyzed in terms of mean value and standard deviation. More specifically, we also used Cliff's delta (δ) [42], which is a nonparametric effect size measure of how often the values in one distribution are larger than the

values in a second distribution, to compare the results generated through our approach and its corresponding baseline method.

Because Cliff did not suggest corresponding δ values to represent small, medium, and large effects, we converted Cohen's *d* effect size to Cliff's δ using *coh2delta* R package (<https://rdrr.io/cran/orddom/man/coh2delta.html>). Note that Table 3 contains descriptors for magnitude of $d = 0.01$ to 2.0.

Third, according to the results of the second step of this procedure, 15 combinations based on three typical similarity methods for software metrics and five commonly used normalization functions for *defects* were examined by the pairwise comparison method. We then determined which combination is more suitable for our approach according to mean, standard deviation, and Cliff's delta effect size.

Fourth, to further validate the effectiveness of the *TDSelector*-based CPDP predictor, we conducted cross-project predictions for all the 15 test sets using *TDSelector* and two competing methods (i.e., baseline1 and baseline2 introduced in Section 5.1). Note that the *TDSelector* used in this experiment was built with the best combination of similarity and normalization.

After this process is completed, we will discuss the answers to the three research questions of our study.

5.4. Classifier and Evaluation Measure. As an underlying machine learning classifier for CPDP, Logistic Regression (LR), which was widely used in many defect prediction literatures [4, 23, 39, 43–46], is also used in this study. All LR classifiers were implemented with Weka (<https://www.cs.waikato.ac.nz/ml/weka/>). For our experiments, we used the default parameter setting for LR specified in Weka unless otherwise specified.

To evaluate the prediction performance of different methods, in this paper, we utilized the area under a Receiver Operating Characteristic curve (AUC). AUC is equal to the probability that a classifier will identify a randomly chosen defective class higher than a randomly chosen defect-free one [47], known as a useful measure for comparing different models. Compared with traditional accuracy measures, AUC is commonly used because it is unaffected by class imbalance and independent of the prediction threshold that is used to decide whether an instance should be classified as a negative instance [6, 48, 49]. The AUC value of 0.5 indicates the performance of a random predictor, and higher AUC values indicate better prediction performance.

6. Experimental Results

6.1. Answer to RQ1. We compared our approach considering *defects* with the baseline method NoD that selects training data in terms of cosine similarity. Table 5 shows that, on average, *TDSelector* does achieve an improvement in AUC value across the 15 test sets. Obviously, the average growth rates of AUC value vary from 5.9% to 9.0% when different normalization methods for *defects* were utilized. In addition, all the δ values in this table are greater than 0.2, which

TABLE 2: Data statistics of the projects used in our experiments.

Repository	Project	Version	#Instance	#Defect	% Defect
PROMISE	Ant	1.7	745	166	22.3%
	Camel	1.6	965	188	19.5%
	Ivy	2.0	352	40	11.4%
	Jedit	3.2	272	90	33.1%
	Lucene	2.4	340	203	59.7%
	Poi	3.0	442	281	63.6%
	Synapse	1.2	256	86	33.6%
	Velocity	1.4	196	147	75.0%
	Xalan	2.6	885	411	46.4%
AEEEEEM	Xerces	1.4	588	437	74.3%
	Equinox	1.1.2005–6.25.2008	324	129	39.8%
	Eclipse JDT core (Eclipse)	1.1.2005–6.17.2008	997	206	20.7%
	Apache Lucene (Lucene2)	1.1.2005–10.8.2008	692	20	2.9%
	Mylyn	1.17.2005–3.17.2009	1,862	245	13.2%
	Eclipse PDE UI (Pde)	1.1.2005–9.11.2008	1,497	209	14.0%

TABLE 3: The mappings between different values and their effectiveness levels.

Effect size	d	δ
Very small	0.01	0.008
Small	0.20	0.147
Medium	0.50	0.33
Large	0.80	0.474
Very large	1.20	0.622
Huge	2.0	0.811

indicates that each group of 15 prediction results obtained by our approach has a greater effect than that of NoD. In other words, our approach outperforms NoD. In particular, for Jedit, Velocity, Eclipse, and Equinox, the improvements of our approach over NoD are substantial. For example, when using the linear normalization method, the AUC values for the four projects are increased by 30.6%, 43.0%, 22.6%, and 39.4%, respectively; moreover, the logistic normalization method for Velocity achieves the biggest improvement in AUC value (namely, 61.7%).

We then compared *TDSelector* with the baseline methods using other widely used similarity calculation methods, and the results obtained by using Euclidean distance and Manhattan distance to calculate the similarity between instances are presented in Tables 6 and 7. *TDSelector*, compared with the corresponding NoD, achieves the average growth rates of AUC value that vary from 5.9% to 7.7% in Table 6 and from 2.7% to 6.9% in Table 7, respectively. More specifically, the highest growth rate of AUC value in Table 6 is 43.6% for Equinox and in Table 7 is 39.7% for Lucene2. Besides, all Cliff’s delta (δ) effect sizes in these two tables are also greater than 0.1. Hence, the results indicate that our approach can, on average, improve the performance of those baseline methods without regard to *defects*.

TABLE 4: Analyzing the factors similarity and normalization.

Factor	Method	Mean	St.d	δ
Similarity	Cosine similarity	0.704	0.082	-0.133
	Euclidean distance	0.719	0.080	-
	Manhattan distance	0.682	0.098	-0.193
Normalization	Linear	0.706	0.087	-0.012
	Logistic	0.710	0.078	-
	Square root	0.699	0.091	-0.044
	Logarithmic	0.700	0.086	-0.064
	Inverse cotangent	0.696	0.097	-0.056

In short, during the process of training data selection, the consideration of *defects* for CPDP can help us to select higher quality training data, thus leading to better classification results.

6.2. *Answer to RQ2.* Although the inclusion of *defects* in the selection of training data of quality is helpful for better performance of CPDP, it is worthy to note that our method completely failed in Mylyn and Pde when computing the similarity between instances in terms of Manhattan distance (see the corresponding maximum AUC values in Table 7). This implies that the success of *TDSelector* depends largely on the reasonable combination of similarity and normalization methods. Therefore, which combination of similarity and normalization is more suitable for *TDSelector*?

First, we analyzed the two factors (i.e., similarity and normalization) separately. For example, we evaluated the difference among cosine similarity, Euclidean distance, and Manhattan distance, regardless of any normalization method used in the experiment. The results, expressed in terms of mean and standard deviation, are shown in Table 4, where they are grouped by factors.

TABLE 5: The best prediction results obtained by the CPDP approach based on *TDSelector* with Cosine similarity. NoD represents the baseline method; + denotes the growth rate of AUC value; the maximum AUC value of different normalization methods is underlined; each number shown in bold indicates that the corresponding AUC value rises by more than 10%.

Cosine similarity	Ant	Xalan	Camel	Ivy	Jedit	Lucene	Poi	Synapse	Velocity	Xerces	Eclipse	Equinox	Lucene2	Mylyn	Pde	Mean ± Std	δ
Linear																	
α	0.7	0.9	0.9	1.0	0.9	1.0	0.9	1.0	0.6	0.9	0.8	0.6	0.7	0.7	0.5		
AUC	<u>0.813</u>	<u>0.676</u>	<u>0.603</u>	<u>0.793</u>	<u>0.700</u>	<u>0.611</u>	<u>0.758</u>	<u>0.741</u>	<u>0.512</u>	<u>0.742</u>	<u>0.783</u>	<u>0.760</u>	<u>0.739</u>	<u>0.705</u>	<u>0.729</u>	<u>0.711 ± 0.081</u>	<u>0.338</u>
+	6.3%	3.7%	1.9%	-	30.6%	-	3.0%	-	43.0%	0.3%	22.6%	39.4%	4.1%	5.9%	4.0%	9.0%	
Logistic																	
α	0.7	0.5	0.7	1	0.7	0.6	0.6	0.6	0.5	0.5	0	0.4	0.7	0.5	0.5		
AUC	<u>0.802</u>	<u>0.674</u>	<u>0.595</u>	<u>0.793</u>	<u>0.665</u>	<u>0.621</u>	<u>0.759</u>	<u>0.765</u>	<u>0.579</u>	<u>0.745</u>	<u>0.773</u>	<u>0.738</u>	<u>0.712</u>	<u>0.707</u>	<u>0.740</u>	<u>0.711 ± 0.070</u>	<u>0.351</u>
+	4.8%	3.4%	0.5%	-	24.1%	1.6%	3.1%	3.2%	61.7%	0.7%	21.0%	35.5%	0.3%	6.2%	5.6%	9.0%	
Square root																	
α	0.7	0.7	0.6	0.6	0.7	0.6	0.7	0.9	0.5	1	0.4	0.6	0.6	0.6	0.6		
AUC	<u>0.799</u>	<u>0.654</u>	<u>0.596</u>	<u>0.807</u>	<u>0.735</u>	<u>0.626</u>	<u>0.746</u>	<u>0.762</u>	<u>0.500</u>	<u>0.740</u>	<u>0.774</u>	<u>0.560</u>	<u>0.722</u>	<u>0.700</u>	<u>0.738</u>	<u>0.697 ± 0.091</u>	<u>0.249</u>
+	4.4%	0.3%	0.7%	1.8%	37.1%	2.5%	1.4%	2.8%	39.7%	-	21.0%	2.8%	1.7%	5.3%	5.3%	6.9%	
Logarithmic																	
α	0.6	0.6	0.9	1.0	0.7	1.0	0.7	0.7	0.5	0.9	0.5	0.5	0.6	0.6	0.6		
AUC	<u>0.798</u>	<u>0.662</u>	<u>0.594</u>	<u>0.793</u>	<u>0.731</u>	<u>0.611</u>	<u>0.748</u>	<u>0.744</u>	<u>0.500</u>	<u>0.758</u>	<u>0.774</u>	<u>0.700</u>	<u>0.755</u>	<u>0.702</u>	<u>0.741</u>	<u>0.707 ± 0.083</u>	<u>0.351</u>
+	4.3%	1.5%	0.3%	-	36.4%	-	1.6%	0.4%	39.7%	2.4%	21.2%	28.5%	6.3%	5.5%	5.8%	8.5%	
Inverse cotangent																	
α	0.7	1.0	1.0	1.0	0.7	1.0	0.7	1.0	0.6	0.7	0	0.7	0.7	0.7	0.7		
AUC	<u>0.798</u>	<u>0.652</u>	<u>0.592</u>	<u>0.793</u>	<u>0.659</u>	<u>0.611</u>	<u>0.749</u>	<u>0.741</u>	<u>0.500</u>	<u>0.764</u>	<u>0.773</u>	<u>0.556</u>	<u>0.739</u>	<u>0.695</u>	<u>0.734</u>	<u>0.690 ± 0.092</u>	<u>0.213</u>
+	4.3%	-	-	-	22.9%	-	1.8%	-	39.7%	3.2%	21.0%	2.1%	4.1%	4.4%	4.8%	5.9%	
NoD (α = 1)	0.765	0.652	0.592	0.793	0.536	0.611	0.736	0.741	0.358	0.740	0.639	0.543	0.709	0.665	0.701	0.652 ± 0.113	

TABLE 6: The best prediction results obtained by the CPDP approach based on *TDSelector* with Euclidean distance.

	Ant	Xalan	Camel	Ivy	Jedit	Lucene	Poi	Synapse	Velocity	Xerces	Eclipse	Equinox	Lucene2	Mylyn	Pde	Mean \pm St.d	δ
Euclidean distance																	
Linear																	
α	0.9	0.9	1.0	0.9	0.9	0.8	1.0	1.0	0.8	0.8	0	0.6	1.0	0.8	0.8		
AUC	0.795	0.727	0.598	0.826	0.793	0.603	0.714	0.757	0.545	0.775	0.773	0.719	0.722	0.697	0.744	0.719 \pm 0.080	0.369
+	1.3%	6.8%	-	0.9%	32.2%	1.9%	-	-	11.7%	5.2%	17.6%	43.0%	-	1.1%	9.6%	7.7%	
Logistic																	
α	0.7	0.8	0.4	0.7	0.7	0.5	0.6	0.9	0.9	0.9	0	0.7	1.0	1.0	0.9		
AUC	0.787	0.750	0.603	0.832	0.766	0.613	0.716	0.767	0.556	0.745	0.773	0.698	0.722	0.690	0.730	0.717 \pm 0.075	0.360
+	0.3%	10.1%	0.8%	1.6%	27.7%	3.5%	0.3%	1.3%	13.9%	1.1%	17.6%	38.8%	-	-	7.5%	7.2%	
Square root																	
α	0.7	0.8	1.0	0.7	0.8	0.6	0.7	0.7	0.7	1.0	0.7	0.8	1.0	1.0	0.9		
AUC	0.796	0.743	0.598	0.820	0.720	0.618	0.735	0.786	0.564	0.737	0.774	0.696	0.722	0.690	0.750	0.715 \pm 0.076	0.342
+	1.4%	9.1%	-	0.1%	20.0%	4.4%	2.9%	3.8%	15.6%	-	17.8%	38.4%	-	-	10.5%	7.0%	
Logarithmic																	
α	0.7	0.8	1.0	1.0	0.8	0.6	1.0	1.0	0.9	0.9	0.9	0.8	1.0	1.0	0.9		
AUC	0.794	0.746	0.598	0.819	0.722	0.607	0.714	0.757	0.573	0.739	0.778	0.722	0.722	0.690	0.748	0.715 \pm 0.072	0.324
+	1.1%	9.5%	-	-	20.3%	2.5%	-	-	17.4%	0.3%	18.5%	43.6%	-	-	10.3%	7.0%	
Inverse cotangent																	
α	0.8	0.9	0.6	0.8	0.8	0.7	1.0	0.8	0.6	0.7	0	0.9	0.9	1.0	0.9		
AUC	0.796	0.749	0.603	0.820	0.701	0.623	0.714	0.787	0.538	0.750	0.773	0.589	0.763	0.690	0.722	0.708 \pm 0.084	0.280
+	1.4%	10.0%	0.8%	0.1%	16.8%	5.2%	-	4.0%	10.2%	1.8%	17.6%	17.0%	5.6%	-	6.4%	5.9%	
NoD ($\alpha=1$)	0.785	0.681	0.598	0.819	0.600	0.592	0.714	0.757	0.488	0.737	0.657	0.503	0.722	0.690	0.678	0.668 \pm 0.096	

TABLE 7: The best prediction results obtained by the CPDP approach based on *TDSelector* with Manhattan distance.

	Ant	Xalan	Camel	Ivy	Jedit	Lucene	Poi	Synapse	Velocity	Xerces	Eclipse	Equinox	Lucene2	Mylyn	Pde	Mean ± St.d	δ
Manhattan distance																	
Linear																	
α	0.8	0.9	0.9	1.0	0.9	0.9	1.0	1.0	0.8	1.0	0	0.8	0.9	1.0	1.0		
AUC	<u>0.804</u>	0.753	0.599	0.816	0.689	0.626	0.695	0.748	0.500	0.749	<u>0.773</u>	0.633	<u>0.692</u>	0.695	0.668	0.696 ± 0.084	0.187
+	1.3%	7.0%	0.3%	-	7.3%	6.3%	-	-	7.8%	-	11.6%	19.0%	39.7%	-	-	5.6%	
Logistic																	
α	0.7	0.7	0.8	0.8	0.8	0.7	0.7	0.9	0.6	0.7	0	0.9	0.9	1.0	1.0		
AUC	0.799	<u>0.760</u>	0.607	<u>0.830</u>	0.674	0.621	<u>0.735</u>	<u>0.794</u>	0.520	0.756	<u>0.773</u>	<u>0.680</u>	0.559	0.695	0.668	0.705 ± 0.084	0.249
+	0.6%	8.0%	1.7%	<u>1.7%</u>	5.0%	5.4%	5.8%	6.1%	12.1%	0.9%	11.6%	27.9%	12.7%	-	-	6.9%	
Square root																	
α	0.9	0.9	0.9	1.0	0.8	0.8	0.9	0.8	0.9	1.0	0	1.0	0	1.0	1.0		
AUC	0.795	0.755	0.604	0.816	<u>0.693</u>	<u>0.627</u>	0.704	0.750	0.510	0.749	<u>0.773</u>	0.532	0.523	0.695	0.668	0.680 ± 0.1	0.164
+	0.1%	7.2%	1.2%	-	7.9%	6.5%	1.3%	0.3%	9.9%	-	11.6%	-	4.6%	-	-	3.1%	
Logarithmic																	
α	1.0	0.9	0.9	1.0	0.9	1.0	1.0	0.8	0.9	0.9	0	1.0	0	1.0	1.0		
AUC	0.794	0.755	0.603	0.816	0.664	0.589	0.695	0.763	0.524	0.756	<u>0.773</u>	0.532	0.523	0.695	0.668	0.677 ± 0.102	0.116
+	-	7.2%	1.0%	-	3.4%	-	-	2.0%	12.9%	0.9%	11.6%	-	4.6%	-	-	2.7%	
Inverse cotangent																	
α	1.0	0.9	0.9	0.9	0.9	0.8	0.9	1.0	0.7	0.8	0	1.0	0	1.0	1.0		
AUC	0.794	0.749	0.608	0.821	0.667	0.609	0.710	0.748	0.500	0.758	<u>0.773</u>	0.532	0.523	0.695	0.668	0.677 ± 0.103	0.133
+	-	6.4%	1.8%	0.6%	3.9%	3.4%	2.2%	-	7.8%	1.2%	11.6%	-	4.6%	-	-	2.7%	
NoD (α=1)	0.794	0.704	0.597	0.816	0.642	0.589	0.695	0.748	0.464	0.749	0.693	0.532	0.500	0.695	0.668	0.659 ± 0.105	

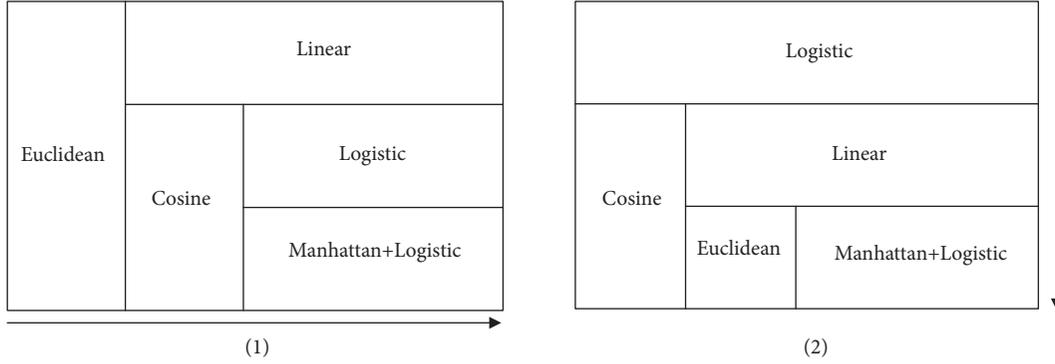


FIGURE 4: A guideline for choosing suitable similarity indexes and normalization methods from two aspects of similarity (see (1)) and normalization (see (2)). The selection priority is lowered along the direction of the arrow.

If we do not take into account normalization, Euclidean distance achieves the maximum mean value 0.719 and the minimum standard deviation value 0.080 among the three similarity indexes, followed by cosine similarity. Therefore, Euclidean distance and cosine similarity are the first and second choices of our approach, respectively. On the other hand, if we do not take into account similarity index, the logistic normalization method seems to be the most suitable method for *TDSector*, indicated by the maximum mean value 0.710 and the minimum standard deviation value 0.078, and it is followed by the linear normalization method.

Therefore, the logistic normalization method is the preferred way for *TDSector* to normalize *defects*, while the linear normalization method is a possible alternative method. It is worth noting that the evidence that all Cliff’s delta (δ) effect sizes in Table 4 are negative also supported the result. Then, a simple guideline for choosing similarity indexes and normalization methods for *TDSector* from two different aspects is presented in Figure 4.

Then, we considered both factors. According to the results in Tables 5, 6, and 7 grouped by different similarity indexes, *TDSector* can obtain the best result 0.719 ± 0.080 , 0.711 ± 0.070 , and 0.705 ± 0.084 when using “Euclidean + Linear” (short for Euclidean distance + linear normalization), “Cosine + Logistic” (short for cosine similarity + logistic normalization), and “Manhattan + Logistic” (short for Manhattan distance + logistic normalization), respectively. We also calculated the value of Cliff’s delta (δ) effect size for every two combinations under discussion. As shown in Table 8, according to the largest number of positive δ values in this table, the combination of Euclidean distance and the linear normalization method can still outperform the other 14 combinations.

6.3. Answer to RQ3. A comparison between our approach and two baseline methods (i.e., baseline1 and baseline2) across the 15 test sets is presented in Table 9. It is obvious that our approach is, on average, better than the two baseline methods, indicated by the average growth rates of AUC value (i.e., 10.6% and 4.3%) across the 15 test sets. The *TDSector* performs better than baseline1 in 14 out of 15 datasets, and it has an advantage over baseline2 in 10 out of 15 datasets. In

particular, compared with baseline1 and baseline2, the highest growth rates of AUC value of our approach reach up to 65.2% and 64.7%, respectively, for Velocity. We also analyzed the possible reason in terms of the defective instances of simplified training dataset obtained from different methods. Table 10 shows that the proportion of defective instances in each simplified training dataset is very close. However, according to instances with more than one defect among these defective instances, our method can return more, and the ratio approximates to twice as large as that of the baselines. Therefore, a possible explanation for the improvement is that the information about defects was more fully utilized due to the instances with more defects. The result further validated that the selection of training data considering defects is valuable.

Besides, the negative δ values in this table also indicate that our approach outperforms the baseline methods from the perspective of distribution, though we have to admit that the effect size 0.009 is too small to be of interest in a particular application.

In summary, since the *TDSector*-based defect predictor outperforms those based on the two state-of-the-art CPDP methods, our approach is beneficial for training data selection and can further improve the performance of CPDP models.

7. Discussion

7.1. Impact of Top- k on Prediction Results. The parameter k determines the number of the nearest training instances of each test instance. Since k was set to 10 in our experiments, here we discuss the impact of k on prediction results of our approach as its value is changed from 1 to 10 with a step value of 1. As shown in Figure 5, for the three combinations in question, selecting the k -nearest training instances (e.g., $k \leq 5$) for each test instance in the 10 test sets from PROMISE, however, does not lead to better prediction results, because their best results are obtained when k is equal to 10.

Interestingly, for the combinations of “Euclidean + Linear” and “Cosine + Linear”, a similar trend of AUC value changes is visible in Figure 6. For the five test sets from AEEEM, they achieve stable prediction results when k ranges from four to eight, and then they reach peak performance

TABLE 8: Pairwise comparisons between a given combination and each of the 15 combinations in terms of Cliff's delta (δ) effect size.

	Cosine similarity					Euclidean distance					Manhattan distance				
	Linear	Logistic	Square root	Logarithmic	Inverse cotangent	Linear	Logistic	Square root	Logarithmic	Inverse cotangent	Linear	Logistic	Square root	Logarithmic	Inverse cotangent
Cosine + Linear	-	0.018	0.084	0.000	0.116	-0.049	-0.036	-0.004	-0.013	-0.009	0.138	0.049	0.164	0.178	0.169
Euclidean + Linear	0.049	0.102	0.111	0.062	0.164	-	0.036	0.040	0.058	0.089	0.209	0.102	0.249	0.276	0.244
Manhattan + Logistic	-0.049	-0.022	0.022	-0.013	0.111	-0.102	-0.076	-0.080	-0.049	-0.031	0.053	-	0.124	0.151	0.147

TABLE 9: A comparison between our approach and two baseline methods for the data sets from PROMISE and AEEEM. The comparison is conducted based on the best prediction results of all the three methods in question.

Test set	Baseline1	Baseline2	Euclidean + Linear	δ	
Ant	0.785	0.803	1.3%	-1.0%	
Xalan	0.657	0.675	10.7%	7.7%	
Camel	0.595	0.624	0.5%	-4.2%	
Ivy	0.789	0.802	4.7%	3.0%	
Jedit	0.694	0.782	14.3%	1.4%	Baseline1 vs. TDSelector: -0.409
Lucene	0.608	0.701	-0.8%	-14.0%	
Poi	0.691	0.789	3.3%	-9.5%	
Synapse	0.740	0.748	2.3%	1.2%	
Velocity	0.330	0.331	65.2%	64.7%	
Xerces	0.714	0.753	8.5%	2.9%	
Eclipse	0.706	0.744	10.2%	4.6%	
Equinox	0.587	0.720	23.1%	0.3%	Baseline2 vs. TDSelector: -0.009
Lucene2	0.705	0.724	2.5%	-0.2%	
Mylyn	0.631	0.646	9.3%	6.8%	
Pde	0.678	0.737	10.4%	1.5%	
Avg.	0.663	0.705	10.6%	4.3%	

TABLE 10: Comparison with the defect instances of simplified training dataset obtained from different methods on Velocity project.

	$\frac{\#defect_instances}{\#instances}$	$\frac{\#instances(defects > 1)}{\#defect_instances}$
	Baseline1	0.375
Baseline2	0.393	0.291
TDSelector	0.376	0.487

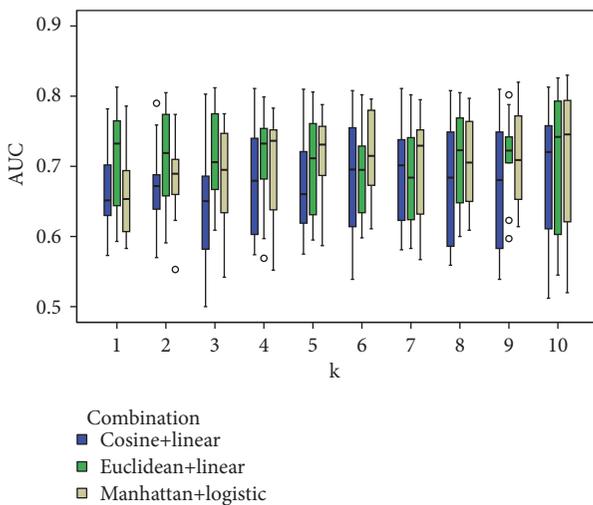


FIGURE 5: The impact of k on prediction results for the 10 test sets from PROMISE.

when k is equal to 10. The combination of “Manhattan + Logistic”, by contrast, achieves the best result as k is set to 7. Even so, the best result is still worse than those of the other two combinations.

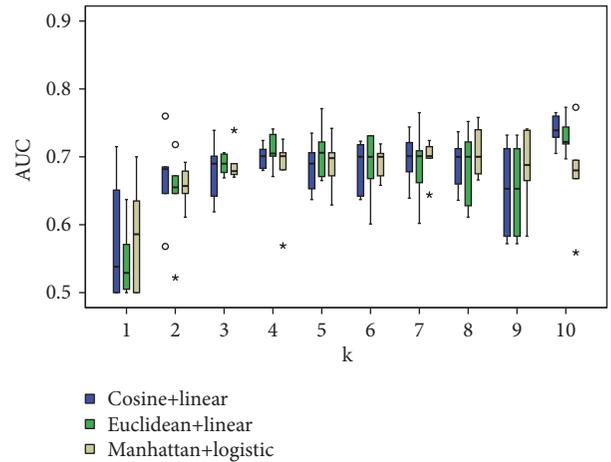


FIGURE 6: The impact of k on prediction results for the 5 test sets from AEEEM.

7.2. *Selecting Instances with More Bugs Directly as Training Data.* Our experimental results have validated the impact of *defects* on the selection of training data of quality in terms of AUC, and we also want to know whether the direct selection of defective instances with more bugs as training instances, which simplifies the selection process and reduces computation cost, would achieve better prediction performance. The result of this question is of particular concern for developers in practice.

According to Figure 7(a), for the 15 releases, most of them contain instances with no more than two bugs. On the other hand, the ratio of the instances that have more than three defects to the total instances is less than 1.40% (see Figure 7(b)). Therefore, we built a new TDSelector based on the number of bugs in each instance, which is referred to as

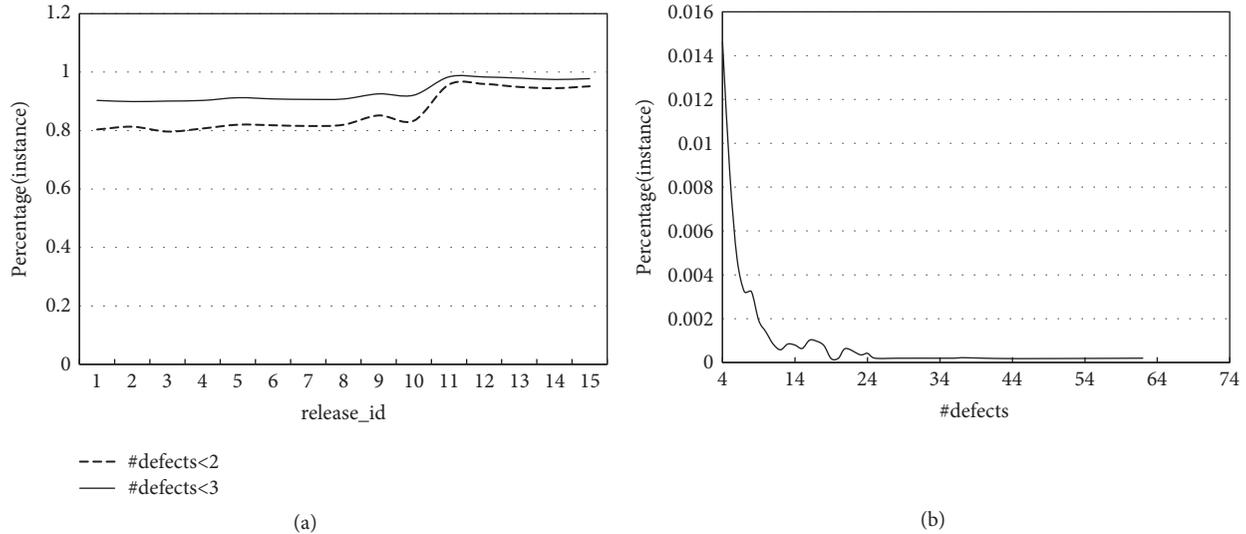


FIGURE 7: Percentage of defective instances with different numbers of bugs. (a) is shown from the viewpoint of a single dataset (release), while (b) is shown from the viewpoint of the whole dataset used in our experiments.

TDSelector-3. That is to say, those defective instances that have at least three bugs were chosen directly from an initial TDS as training data, while the remaining instances in the TDS were selected in light of (2). All instances from the two parts then form the final TDS after removing redundant ones.

Figure 8 shows that the results of the two methods differ from dataset to dataset. For Ivy and Xerxes collected from PROMISE, TDSelector outperforms TDSelector-3 in all the three scenarios, but only slightly. On the contrary, for Lucene and Velocity from PROMISE, the incremental AUC values obtained by using TDSelector-3 with “Cosine + Linear” reach up to 0.109 and 0.235, respectively. As shown in Figure 8, on average, TDSelector-3 performs better than the corresponding TDSelector, and the average AUC values for “Cosine + Linear”, “Euclidean + Linear”, and “Manhattan + Logistic” are improved by up to 3.26%, 2.57%, and 1.42%, respectively. Therefore, the direct selection of defective instances that contain quite a few bugs can, overall, further improve the performance of the predictor trained by our approach. In other words, those valuable defective instances can be screened out quickly according to a threshold for the number of bugs in each training instance (namely, three in this paper) at the first stage. Our approach is then able to be applied to the remaining TDS. Note that the automatic optimization method for such a threshold for *TDSelector* will be investigated in our future work.

7.3. Threats to Validity. In this study, we obtained several interesting results, but potential threats to the validity of our work remain.

Threats to *internal validity* concern any confounding factor that may affect our results. First, the raw data used in this paper were normalized by using the *Z*-score method, while the baseline method TCA+ provides four normalization methods [39]. Second, unlike TCA+, *TDSelector* does not introduce any feature selection method to process software

metrics. Third, the weighting factor α changes with a step size 0.1, when Algorithm 1 calculates the maximum value of AUC. There is no doubt that a smaller step size will result in greater calculation time. Fourth, we trained only one type of defect predictor based on the default parameter settings configured by the tool Weka, because LR has been widely used in previous studies. Hence, we are indeed aware that the results of our study would change if we use different settings of the above three factors.

Threats to *statistical conclusion validity* focus on whether conclusions about the relationship among variables based on the experimental data are correct or reasonable [50]. In addition to mean value and standard deviation, in this paper, we also utilized Cliff’s delta effect size instead of hypothetical test methods such as the Kruskal–Wallis *H* test [51] to compare the results of different methods, because there are only 15 datasets collected from PROMISE and AEEEM. According to the criteria that were initially suggested by Cohen and expanded by Sawilowsky [52], nearly all of the effect size values in this paper belong to *small* ($0.147 \leq \delta < 0.33$) and *very small* ($0.008 \leq \delta < 0.147$). This indicates that there is no significant difference in AUC value between different combinations in question, though some perform better in terms of mean value and standard deviation. However, it is clear that our method obviously performs better than baseline, indicated by $|\delta| = 0.409 > 0.33$.

Threats to *external validity* emphasize the generalization of the obtained results. First, the selection of experimental datasets—in addition to AEEEM and PROMISE—is the main threat to validate the results of our study. All the 14 projects used in this paper are written in Java and from the Apache Software Foundation and the Eclipse Foundation. Although our experiments can be repeated with more open-source projects written in other programming languages and developed with different software metrics, the empirical results may be different from our main conclusions. Second, we utilized only three similarity indexes and five

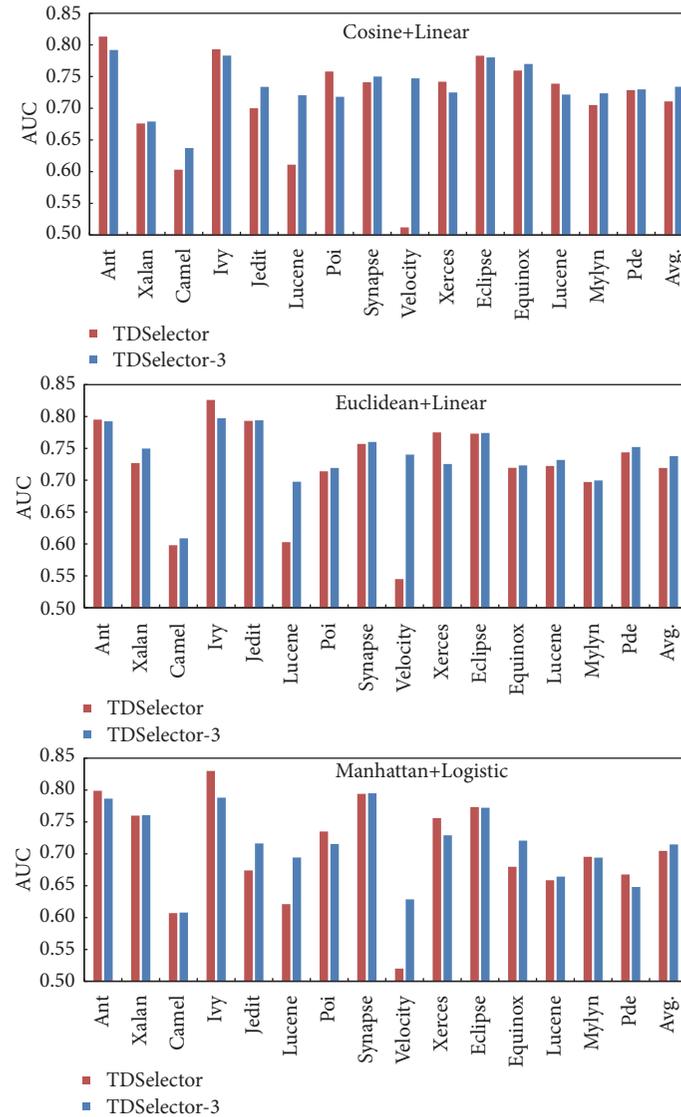


FIGURE 8: A comparison of prediction performance between TDSelector-3 and the corresponding TDSelector. The last column in each of the three plots represents the average AUC value.

normalization methods when calculating the score of each candidate training instance. Therefore, the generalizability of our method for other similarity indexes (such as Pearson Correlation Coefficient and Mahalanobis distance [53]) and normalization methods has yet to be tested. Third, to compare our method with TCA+, defect predictors used in this paper were built using LR, implying that the generalizability of our method for other classification algorithms remains unclear.

8. Conclusion and Future Work

This study aims to train better defect predictors by selecting the most appropriate training data from those defect datasets available on the Internet, to improve the performance of cross-project defect predictions. In summary, the study has been conducted on 14 open-source projects and consists of (1) an empirical validation on the usability of the number of

defects that an instance includes for training data selection, (2) an in-depth analysis of our method *TDSelector* with regard to similarity and normalization, and (3) a comparison between our proposed method and the benchmark methods.

Compared with those similar previous studies, the results of this study indicate that the inclusion of *defects* does improve the performance of CPDP predictors. With a rational balance between the similarity of test instances with training instances and *defects*, *TDSelector* can effectively select appropriate training instances, so that *TDSelector*-based defect predictors, built by using LR, achieve better prediction performance in terms of AUC. More specifically, the combination of Euclidean distance and linear normalization is the preferred way for *TDSelector*. In addition, our results also demonstrate the effectiveness of the proposed method according to a comparison with the baseline methods in the context of M2O in CPDP scenarios. Hence, we believe that our approach can be helpful for developers when they are

required to build suitable predictors quickly for their new projects, because one of our interesting findings is that those candidate instances with more bugs can be chosen directly as training instances.

Our future work mainly includes two aspects. On the one hand, we plan to validate the generalizability of our study with more defect data from projects written in different languages. On the other hand, we will focus on more effective hybrid methods based on different selection strategies such as feature selection techniques [32]. Last but not least, we also plan to discuss the possibility of considering not only the number of defects but also time variables for training data selection (such as bug-fixing time).

Conflicts of Interest

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

Acknowledgments

The authors greatly appreciate Dr. Nam and Dr. Pan, the authors of [39], for providing them with the TCA source program and teaching them how to use it. This work was supported by the Natural Science Foundation of Hubei province (no. 2016CFB309) and the National Natural Science Foundation of China (nos. 61272111, 61273216, and 61572371).

References

- [1] Z. He, F. Shu, Y. Yang, M. Li, and Q. Wang, "An investigation on the feasibility of cross-project defect prediction," *Automated Software Engineering*, vol. 19, no. 2, pp. 167–199, 2012.
- [2] L. C. Briand, W. L. Melo, and J. Wüst, "Assessing the applicability of fault-proneness models across object-oriented software projects," *IEEE Transactions on Software Engineering*, vol. 28, no. 7, pp. 706–720, 2002.
- [3] Y. Ma, G. Luo, X. Zeng, and A. Chen, "Transfer learning for cross-company software defect prediction," *Information and Software Technology*, vol. 54, no. 3, pp. 248–256, 2012.
- [4] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction: A large scale experiment on data vs. domain vs. process," in *Proceedings of the Joint 12th European Software Engineering Conference and 17th ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC-FSE'09*, pp. 91–100, nld, August 2009.
- [5] F. Peters, T. Menzies, and A. Marcus, "Better cross company defect prediction," in *Proceedings of the 10th International Working Conference on Mining Software Repositories, MSR 2013*, pp. 409–418, usa, May 2013.
- [6] F. Rahman, D. Posnett, and P. Devanbu, "Recalling the "imprecision" of cross-project defect prediction," in *Proceedings of the the ACM SIGSOFT 20th International Symposium*, p. 1, Cary, North Carolina, November 2012.
- [7] S. Herbold, "Training data selection for cross-project defect prediction," in *Proceedings of the the 9th International Conference*, pp. 1–10, Baltimore, Maryland, October 2013.
- [8] T. M. Khoshgoftaar, E. B. Allen, R. Halstead, G. P. Trio, and R. M. Flass, "Using process history to predict software quality," *The Computer Journal*, vol. 31, no. 4, pp. 66–72, 1998.
- [9] T. J. Ostrand and E. J. Weyuker, "The distribution of faults in a large industrial software system," *ACM SIGSOFT Software Engineering Notes*, vol. 27, no. 4, p. 55, 2002.
- [10] S. Kim, T. Zimmermann, E. J. Whitehead Jr., and A. Zeller, "Predicting faults from cached history," in *Proceedings of the 29th International Conference on Software Engineering (ICSE '07)*, pp. 489–498, IEEE Computer Society, Washington, DC, USA, May 2007.
- [11] T. Gyimothy, R. Ferenc, and I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Transactions on Software Engineering*, vol. 31, no. 10, pp. 897–910, 2005.
- [12] B. Turhan, T. Menzies, A. B. Bener, and J. Di Stefano, "On the relative value of cross-company and within-company data for defect prediction," *Empirical Software Engineering*, vol. 14, no. 5, pp. 540–578, 2009.
- [13] M. Chen and Y. Ma, "An empirical study on predicting defect numbers," in *Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering, SEKE 2015*, pp. 397–402, usa, July 2015.
- [14] C. Ni, W. Liu, Q. Gu, X. Chen, and D. Chen, "FeSCH: A Feature Selection Method using Clusters of Hybrid-data for Cross-Project Defect Prediction," in *Proceedings of the 41st IEEE Annual Computer Software and Applications Conference, COMPSAC 2017*, pp. 51–56, ita, July 2017.
- [15] P. He, B. Li, X. Liu, J. Chen, and Y. Ma, "An empirical study on software defect prediction with a simplified metric set," *Information and Software Technology*, vol. 59, pp. 170–190, 2015.
- [16] T. Wang, Z. Zhang, X. Jing, and L. Zhang, "Multiple kernel ensemble learning for software defect prediction," *Automated Software Engineering*, vol. 23, no. 4, pp. 1–22, 2015.
- [17] J. Y. He, Z. P. Meng, X. Chen, Z. Wang, and X. Y. Fan, "Semi-supervised ensemble learning approach for cross-project defect prediction," *Journal of Software. Ruanjian Xuebao*, vol. 28, no. 6, pp. 1455–1473, 2017.
- [18] D. Ryu, J.-I. Jang, and J. Baik, "A transfer cost-sensitive boosting approach for cross-project defect prediction," *Software Quality Journal*, vol. 25, no. 1, pp. 1–38, 2015.
- [19] D. Ryu and J. Baik, "Effective multi-objective naïve Bayes learning for cross-project defect prediction," *Applied Soft Computing*, vol. 49, pp. 1062–1077, 2016.
- [20] Y. Li, Z. Huang, Y. Wang, and B. Fang, "Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements," *IEEE Access*, vol. 5, pp. 25646–25656, 2017.
- [21] F. Zhang, A. Mockus, I. Keivanloo, and Y. Zou, "Towards building a universal defect prediction model," in *Proceedings of the 11th International Working Conference on Mining Software Repositories, MSR 2014*, pp. 182–191, ind, June 2014.
- [22] P. He, B. Li, and Y. Ma, *Towards Cross-Project Defect Prediction with Imbalanced Feature Sets*, 2014.
- [23] J. Nam and S. Kim, "Heterogeneous defect prediction," in *Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015*, pp. 508–519, ita, September 2015.
- [24] X. Jing, F. Wu, X. Dong, F. Qi, and B. Xu, "Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning," in *Proceedings of the 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015*, pp. 496–507, ita, September 2015.

- [25] Z. Li, X.-Y. Jing, X. Zhu, H. Zhang, B. Xu, and S. Ying, "On the Multiple Sources and Privacy Preservation Issues for Heterogeneous Defect Prediction," *IEEE Transactions on Software Engineering*, 1 page, 2017.
- [26] Z. Li, X.-Y. Jing, F. Wu, X. Zhu, B. Xu, and S. Ying, "Cost-sensitive transfer kernel canonical correlation analysis for heterogeneous defect prediction," *Automated Software Engineering*, pp. 1–45, 2017.
- [27] Z. Li, X. Jing, X. Zhu, and H. Zhang, "Heterogeneous Defect Prediction Through Multiple Kernel Learning and Ensemble Learning," in *Proceedings of the 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 91–102, Shanghai, September 2017.
- [28] Z. He, F. Peters, T. Menzies, and Y. Yang, "Learning from open-source projects: An empirical study on defect prediction," in *Proceedings of the 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2013*, pp. 45–54, usa, October 2013.
- [29] P. He, B. Li, D. Zhang, and Y. Ma, *Simplification of Training Data for Cross-Project Defect Prediction*, *arXiv preprint*, 2014.
- [30] F. Porto and A. Simao, "Feature Subset Selection and Instance Filtering for Cross-project Defect Prediction-Classification and Ranking," *CLEI Electronic Journal*, vol. 19, no. 4, p. 17, 2016.
- [31] X.-Y. Jing, F. Wu, X. Dong, and B. Xu, "An Improved SDA Based Defect Prediction Framework for Both Within-Project and Cross-Project Class-Imbalance Problems," *IEEE Transactions on Software Engineering*, vol. 43, no. 4, pp. 321–339, 2017.
- [32] D. Ryu, J.-I. Jang, and J. Baik, "A hybrid instance selection using nearest-neighbor for cross-project defect prediction," *Journal of Computer Science and Technology*, vol. 30, no. 5, pp. 969–980, 2015.
- [33] W. N. Poon, K. E. Bennin, J. Huang, P. Phannachitta, and J. W. Keung, "Cross-project defect prediction using a credibility theory based naive bayes classifier," in *Proceedings of the 17th IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, pp. 434–441, cze, July 2017.
- [34] M. D'Ambros, M. Lanza, and R. Robbes, "Evaluating defect prediction approaches: a benchmark and an extensive comparison," *Empirical Software Engineering*, vol. 17, no. 4–5, pp. 531–577, 2012.
- [35] B. Turhan, A. Tosun Misirli, and A. Bener, "Empirical evaluation of the effects of mixed project data on learning defect predictors," *Information and Software Technology*, vol. 55, no. 6, pp. 1101–1118, 2013.
- [36] J. Han and M. Kamber, *Data Mining : Concepts and Techniques*, Elsevier/Morgan Kaufmann, Waltham, Mass, USA, 3rd edition, 2012.
- [37] A. Graf and S. Borer, "Normalization in Support Vector Machines," in *Proceedings of Dagm-Symposium on Pattern Recognition*, vol. 2191, pp. 277–282, Springer-Verlag, 2001, no. 7.
- [38] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Learning," *Enformatika*, vol. 1, no. 1, pp. 111–117, 2006.
- [39] J. Nam, S. J. Pan, and S. Kim, "Transfer defect learning," in *Proceedings of the 2013 35th International Conference on Software Engineering, ICSE 2013*, pp. 382–391, usa, May 2013.
- [40] M. Jureczko and D. D. Spinellis, "Using Object-Oriented Design Metrics to Predict Software Defects," in *Proceedings of the International Conference on Dependability of Computer System, Monographs of System Dependability*, pp. 69–81, 2010.
- [41] M. D'Ambros, M. Lanza, and R. Robbes, "An extensive comparison of bug prediction approaches," in *Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR '10)*, pp. 31–41, IEEE, May 2010.
- [42] G. E. Macbeth, E. Razumiejczyk, and R. D. Ledesma, "Cliff's delta calculator: A non-parametric effect size program for two groups of observations," *Universitas Psychologica*, vol. 10, no. 2, pp. 545–555, 2011.
- [43] T. Lee, D. G. Han, S. Kim, and H. P. In, "Micro interaction metrics for defect prediction," in *Proceedings of the 19th ACM SIGSOFT Symposium on Foundations of Software Engineering, SIGSOFT/FSE'11*, pp. 311–321, hun, September 2011.
- [44] A. Meneely, L. Williams, W. Snipes, and J. Osborne, "Predicting failures with developer networks and social network analysis," in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT '08)*, pp. 13–23, ACM, November 2008.
- [45] R. Moser, W. Pedrycz, and G. Succi, "A Comparative analysis of the efficiency of change metrics and static code attributes for defect prediction," in *Proceedings of the 30th International Conference on Software Engineering 2008, ICSE'08*, pp. 181–190, deu, May 2008.
- [46] E. Shihab, A. Mockus, Y. Kamei, B. Adams, and A. E. Hassan, "High-impact defects: A study of breakage and surprise defects," in *Proceedings of the 19th ACM SIGSOFT Symposium on Foundations of Software Engineering, SIGSOFT/FSE'11*, pp. 300–310, hun, September 2011.
- [47] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [48] E. Giger, M. D'Ambros, M. Pinzger, and H. C. Gall, "Method-level bug prediction," in *Proceedings of the 6th ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2012*, pp. 171–180, swe, September 2012.
- [49] Q. Song, Z. Jia, M. Shepperd, S. Ying, and J. Liu, "A general software defect-proneness prediction framework," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 356–370, 2011.
- [50] P. C. Cozby, "Methods in behavioral research," in *McGraw-Hill Higher Education*, 2011.
- [51] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.
- [52] S. S. Sawilowsky, "New Effect Size Rules of Thumb," *Journal of Modern Applied Statistical Methods*, vol. 8, no. 2, pp. 597–599, 2009.
- [53] S. S. Choi, S. H. Cha, and C. C. Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal of Systemics Cybernetics Informatics*, vol. 8, no. 1, pp. 43–48, 2010.

Research Article

J Wave Autodetection Using Analytic Time-Frequency Flexible Wavelet Transformation Applied on ECG Signals

Deng-ao Li , Jie Zhou , Jumin Zhao , and Xinyan Liu

College of Information and Computer, Taiyuan University of Technology, Taiyuan, China

Correspondence should be addressed to Deng-ao Li; lidengao@tyut.edu.cn

Received 18 November 2017; Revised 16 March 2018; Accepted 12 April 2018; Published 31 May 2018

Academic Editor: Li Xu

Copyright © 2018 Deng-ao Li 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 a new important index of the electrocardiogram (ECG) of ventricular bipolar play, J wave plays an increasingly significant role in the clinical diagnosis. The existence of J wave hints at potential crisis of fatal disease and even death. Nowadays, however, it can hardly meet the clinical needs where the diagnosis of J wave variation only depends on experience of clinicians. Therefore, a new technique which is capable of detecting J wave using analytic time-frequency flexible wavelet transformation (ATFFWT) is proposed in this paper. We have used ATFFWT to decompose the processed ECG signals into the desired subbands. Further, Fuzzy Entropy (FE) is computed from each subband to capture more hidden and meaningful information. Feature scoring method is applied to select optimal feature set. Finally, the extracted features are fed to Least Squares-Support Vector Machine (LS-SVM) classifier. The 10-fold cross validation is used to obtain reliable and stable performance and to avoid the overfitting of the model. Our proposed algorithm has achieved accuracy of 97.61% for Morlet Wavelet (MW) kernel in comparison to 97.56% for Radial Basis Function (RBF) kernel. The developed effective algorithm can be used to design an expert system to aid clinicians in their regular diagnosis.

1. Introduction

Nowadays, cardiovascular diseases (CVDs) cause nearly one-third of all deaths worldwide. CVDs remain a leading cause of health loss for all regions of the world and a major barrier to long-term sustainable development of mankind [1]. Nearly 17 million people die due to cardiovascular diseases globally every year [2]. J wave is regarded as a new important index of the electrocardiogram (ECG) of ventricular bipolar play, and it plays an increasingly significant role in the clinical diagnosis of cardiovascular diseases. A series of diseases or conditions that can produce J waves in the ECG continues to rise [3].

The J wave, also referred to as an Osborn wave, is a deflection immediately following the QRS complex of the surface ECG, which is usually partially buried inside the QRS, often appearing as a J-point elevation; it represents the end of depolarization and the start of bipolarization [4]. The presence of J wave may lead to early repolarization syndrome (ERS), pericarditis, idiopathic ventricular fibrillation (IVF),

Brugada syndrome (BrS), and even sudden unexplained nocturnal death syndrome [4]. From a mechanistic point of view, these syndromes should be referred to as the J wave syndromes [4]. J wave and J wave syndrome are high-risk early warning indicators of sudden cardiac death [5]. The appearances of prominent J waves in the ECG have long been reported in cases of hypothermia and hypercalcemia [6]. More recently, accentuation of the J wave has been associated with life-threatening ventricular arrhythmias. Although typical J waves usually are accentuated with bradycardia or long pauses, the opposite has also been described [5, 7]. J waves are often seen in young males with no apparent structural heart diseases, whereas intraventricular conduction delay is often observed in older individuals or those with a history of myocardial infarction or cardiomyopathy [5, 7].

J wave is mixed in the normal ST segment, coupled with the small amplitude, existence of noise, and baseline wander. Accordingly, the diagnosis of J wave variation and minute changes in the ECG signals only depends on the clinicians' experience, which can not meet demand at present

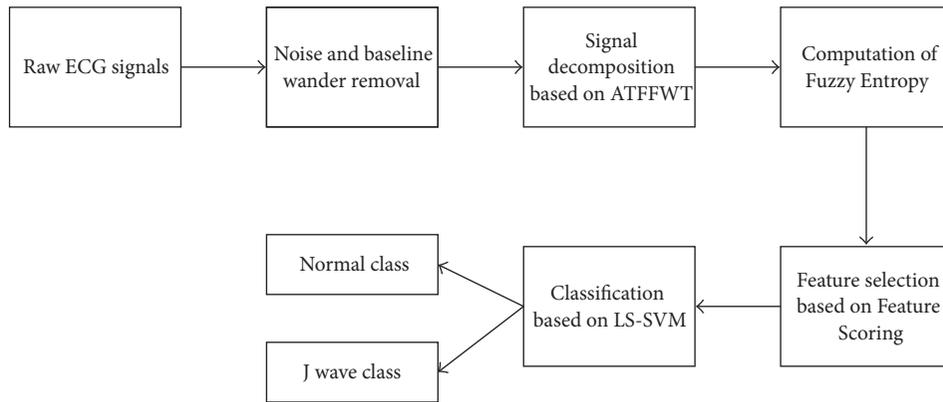


FIGURE 1: Steps used for automatic J wave detection.

clinical, and is also apt to be misdiagnosed. Therefore, it is pretty essential and necessary for us to analyze J wave from the perspective of computer aided method with advanced digital signal processing techniques and machine learning algorithms, which can help to capture the subtle and hidden information in the ECG signals and realize the accurate and automatic diagnosis of J wave. Such an automated system will provide tremendous assistance to the clinicians in their routine screening of cardiac patients [8–12]. In literature, although the method of processing ECG signals has been very mature, methods of detecting J wave signals from ECG signal were relatively less and the detection precision is undesirable. The authors in [13] proposed the method based on locating a break point in the descending limb of the terminal QRS. They added new logic to Glasgow ECG analysis program to automate the detection of J wave. A technique based on digital 12-lead electrocardiogram is used in [14]. ECG signals were automatically processed with the GE Marquette 12-SL program 2001 version (GE Marquette, Milwaukee, WI). And the functional data analysis techniques were applied to the processed ECG signals. In [15], five features including three time-domain features and two wavelet-based features are defined; these features are found significantly different in discriminate J wave and normal classes. Thereafter, Principle Component Analysis (PCA) is used to reduce the dimension of these features. An approach for J wave autodetection based on SVM is proposed. Curving Fitting (CF) and wavelet transform (WT) are used for feature extraction from J wave and healthy ECG segments data in [16], which have shown effective variations for J wave and normal subjects. In [17], a novel J wave detection method based on massive ECG data and MapReduce is presented. The power spectrum and the cumulative probability of ECG signals are computed as features and Decision Tree (DT) is applied for classification.

Compared with the existing methods of J wave detection, the main contributions of this work are listed as follows: (1) we built a J wave detection expert system with FE entropy on the coefficient of ATFFWT as feature and LS-SVM as a classifier, and it owned high accurate rate; (2) we use lower number of features because feature scoring method is applied

(only entropy features) to obtain the desirable accuracy; the detection of J wave will be fast; (3) it is the first time that tenfold cross validation method is applied, which makes our algorithm more reliable and robust. The objective of present work is to develop a noninvasion marker for some cardiac diseases clinically that can automatically, accurately, and fast detect J wave. To achieve this, the ECG signals are segmented into beats firstly. Then, to cope with the nonlinear and nonstationary nature of ECG signals, analytic time-frequency flexible wavelet transformation (ATFFWT) is employed to decompose the signals in terms of subbands signals. Further, Fuzzy Entropy (FE) is computed on decomposed subband. Then, feature scoring method is performed on these FE to select more meaningful features and improve the classification performance. Finally, these clinically significant features are fed to Least Squares-Support Vector Machine (LS-SVM) classifier with different kernel functions. The followed steps of the proposed method are shown in Figure 1.

2. Materials and Methods

2.1. Data Acquisition and Preprocessing. In this work, the normal ECG signals (normal class) were recorded from 58 normal subjects, which are acquired from MIT/BIH Normal Sinus Rhythm (NSR) database [12] and Fantasia [12, 18]. From the MIT/BIH NSR dataset, we have used 18 subjects (5 males and 13 females). From the Fantasia dataset, we have obtained the ECG records of 40 subjects (20 young and 20 old). The ECG signals with J wave (abnormal class) of 15 patients (10 males and 5 females) are obtained from Shanxi Dayi Hospital; the sampling frequency is 257 Hz. The age of all the subjects varies from 20 to 71 years. ECG signals of lead II were applied in our study. All ECG signals we used were initially inspected by experienced cardiologists. To avoid the inclusion of noise, artifacts, and baseline wander, the digitized ECG signals were preprocessed by using eight levels' Daubechies 6 (db6) basis function of wavelet [8].

2.2. Beats Segmentation. In order to segment the ECG signals into single beats, R-points should be detected firstly. In the

TABLE 1: Total number of beats used in this work.

Type	Number of beats
J wave	25,468
Normal	169,426
Total	194,894

present study, Pan-Tompkins algorithm is carried out on each preprocessed ECG signal to detect R-peaks [8, 19]. Then ECG beats are segmented or selected through the detected R-points [20]. We chose 64 samples before the R-point and 105 samples after R-point for each ECG beat. Hence, there is a segment of 170 samples for one ECG beat, and it consists of P, QRS, T, and U waves. The number of ECG beats segmented for J wave and normal in this work are shown in Table 1.

2.3. Decomposition of the Beats Based on ATFFWT. Wavelet transform (WT) is a powerful mathematical tool for processing nonstationary signal [10]. WT and its various improved methods are still playing a significant role in the signal processing field, because it enjoys fine time-frequency concentration and multiresolution analysis property. However, WT and some of its improved methods are suffering from a number of limitations and shortcomings. For instance, the continuous WT (CWT) suffers restricted computational efficiency, the discrete WT (DWT) suffers from the limitation of having constant time-frequency covering at all scales, and it also suffers shift-variance and poor resolution at its high frequency subbands [21]. These limitations are addressed by a newly introduced time-frequency analysis tool called ATFFWT [11], which has desirable properties such as shift-invariance, flexible time-frequency covering, and tunable oscillatory bases [11]. It has been applied to the weak fault features detection in rotating machinery [21] and the diagnosis of coronary artery disease using HRV signals [9]. ATFFWT is realized by the iterated filter bank (IFB) which consists of one low-pass channel and two high-pass channels; one of these high-pass channels analyzes “positive frequencies,” while the other analyzes “negative frequencies” [11]. J th level decomposition of ATFFWT can be achieved by using IFB [11].

Frequency response of the low-pass filter can be given by the following mathematical equations [11]:

$$H(\omega) = \begin{cases} \sqrt{pq}, & |\omega| < w_p \\ \sqrt{pq}\theta\left(\frac{\omega - w_p}{w_s - w_p}\right), & w_p \leq \omega \leq w_s \\ \sqrt{pq}\theta\left(\frac{\pi - (\omega - w_p)}{w_s - w_p}\right), & -w_s \leq \omega \leq -w_p \\ 0, & |\omega| > w_s, \end{cases} \quad (1)$$

where p and q are up and down sampling parameters for low-pass filter, respectively. w_s and w_p represent the stop band and

pass band frequencies of the low-pass filter and are shown as [11]

$$w_p = \frac{(1 - \beta)\pi}{p} + \frac{\varepsilon}{p}, \quad (2)$$

$$w_s = \frac{\pi}{q}.$$

The other used filter is the high-pass filter and can be defined mathematically as [11]

$$G(\omega) = \begin{cases} \sqrt{rs}\theta\left(\frac{\pi - (\omega + w_0)}{w_1 - w_0}\right), & w_0 < \omega < w_1 \\ \sqrt{rs}, & w_1 \leq \omega \leq w_2 \\ \sqrt{rs}\theta\left(\frac{\omega - w_2}{w_3 - w_2}\right), & w_2 \leq \omega \leq w_3 \\ 0, & \omega \in [0, w_0) \cup (w_2, 2\pi), \end{cases} \quad (3)$$

where r and s are up and down sampling parameters for high-pass filter, respectively. The other parameters are described as follows [11]:

$$w_0 = \frac{(1 - \beta)\pi}{r} + \frac{\varepsilon}{r},$$

$$w_1 = \frac{p\pi}{qr},$$

$$w_2 = \frac{\pi - \varepsilon}{r}, \quad (4)$$

$$w_3 = \frac{\pi + \varepsilon}{r},$$

$$\varepsilon \leq \left(\frac{p - q + \beta q}{p + q}\right)\pi.$$

In this work, the transition band $\theta(\omega)$ is chosen as [11]

$$\theta(\omega) = \frac{[1 + \cos(\omega)]\sqrt{2 - \cos(\omega)}}{2} \quad \omega \in [0, \pi]. \quad (5)$$

The parameters p , q , r , s , and β provide flexibility to control wavelets with the attractive quality-factor (Q -factor) Q , dilation factor d , and redundancy factor R . β and ε are the nonnegative constants. These parameters are not independent of each other and are given as [11]

$$Q \approx \frac{2 - \beta}{\beta},$$

$$d \approx \frac{p}{q},$$

$$\beta \leq 1, \quad (6)$$

$$R \approx \left(\frac{r}{s}\right) \frac{1}{1 - d},$$

$$R > \frac{\beta}{1 - d}.$$

The perfect reconstruction filter bank can be achieved by satisfying the following condition [11]:

$$\begin{aligned} |\theta(w)|^2 + |\theta(\pi - w)|^2 &= 1, \\ 1 - \frac{p}{q} \leq \beta \leq \frac{r}{s}. \end{aligned} \quad (7)$$

Hilbert transform pairs of the wavelet bases can be obtained owing to this type of separation of positive and negative frequencies. These characters make ATFFWT flexible by allowing one to control the Q-factor, redundancy, and dilation factor [11]. Recently, it has been applied for characterization of coronary artery disease [22, 23], myocardial infarction ECG signals [23], and detection of congestive heart failure using heart rate variability (HRV) signals [24]. Matlab toolbox of ATFFWT method is available at <http://web.itu.edu.tr/ibayram/AnDWT/>.

2.4. Nonlinear Feature Extraction from the Detail Coefficients.

Fuzzy Entropy is extracted as nonlinear feature from the each beat segment from the standard ECG signals. Therefore, Fuzzy Entropy is computed on the real value of detail coefficients at each level. As an improvement of the sample entropy algorithm, the similarity measures are fuzzed by Fuzzy Entropy (FE) using an exponential function. The steps for the calculation of the FE are as follows [25].

(1) The sampling sequences of length N are extracted from the detail coefficients:

$$\{u(i) : 1 \leq i \leq N\}. \quad (8)$$

(2) A set of m -dimensional vectors refactored and generated in sequential order $X_i^m = \{u(i), u(i+1), \dots, u(i+m-1)\} - u^0(i)$ ($i = 1, \dots, N-m$). $u^0(i)$ represents the mean value and can be defined as [25]

$$u^0(i) = \frac{1}{m} \sum_{j=0}^{m-1} u(i+j). \quad (9)$$

(3) The distance d_{ij}^m between the sequences X_i^m and X_j^m is defined as the largest difference, and it can be expressed as

$$\begin{aligned} d_{ij}^m &= d[X_i^m, X_j^m] \\ &= \max_{k \in (0, m-1)} \{|u(i+k) - u^0(i) - u(j+k) - u^0(j)|\} \quad (10) \\ &\quad (i, j \in [1, N-m], j \neq i). \end{aligned}$$

(4) The similarity degree D_{ij}^m between the sequences X_i^m and X_j^m is computed by applying the fuzzy function as follows [25]:

$$D_{ij}^m = \mu(d_{ij}^m, n, r) = \exp\left(-\frac{(d_{ij}^m)^n}{r}\right), \quad (11)$$

where μ , n , and r stand for the fuzzy function, the gradient, and the width of the exponential function boundary, respectively.

(5) The function $Q^m(n, r)$ is defined, and it is shown as follows [25]:

$$Q^m(n, r) = \frac{1}{N-m} \sum_{i=1}^{N-m} \left(\frac{1}{N-m-1} \sum_{j=1, j \neq i}^{N-m} D_{ij}^m \right). \quad (12)$$

(6) Finally, FE can be computed according to the formula as follows [25]:

$$\begin{aligned} \text{Fuzzy En}(m, n, r, N) &= \ln[Q^m(n, r)] \\ &\quad - \ln[Q^{m+1}(n, r)]. \end{aligned} \quad (13)$$

FE more easily identifies the abnormal activities in the signal. Therefore, it is applied to discriminate nonfocal and focal electroencephalogram signals [26], characterize the surface electromyogram signals [25], and diagnose epilepsy [27].

2.5. Features Selection Based on Feature Scoring.

Feature selection is widely used in pattern classification and regression to remove the redundant and uncorrelated features in feature space, thus reducing the computational load and improving the classification performance [28]. In this investigation, we employed a feature scoring algorithm to choose the optimal set. This algorithm is also known as mutual information based feature scoring [28]. In this algorithm, score value is calculated for each feature to reflect its usefulness. The larger S^q , the higher the dependency between the feature values and the class labels. The score value of each feature is evaluated according to the following formula [29, 30]:

$$S^q = \sum_{x^q} \sum_y P(x^q, y) \log \frac{P(x^q, y)}{P(y)P(x^q)}. \quad (14)$$

The feature matrix $X \in R^{p \times q}$ consists of p number of beat samples and q number of feature attributes. We gave the class label vector y , $y \in R^p$ for the feature matrix. The values of y are 1 and -1 corresponding to the class label of normal class and J wave class. The score value of q th feature in the feature vector x_p is calculated by using the probability of q th feature $P(x^q)$ and the probability of class level $P(y)$. In this work, the FE feature vectors and the new feature vectors obtained using feature scoring are fed to LS-SVM.

2.6. Classification Based on LS-SVM. The LS-SVM [31], an excellent binary classifier derived from SVM, has successful application of pattern recognition and nonlinear function fitting. However, some problems exist in SVM such as the parameter selection for hyperplane, and the size of matrix is greatly influenced by the number of training samples in Quadratic Programming (QP) problem solving, resulting in the huge solution dimension. Therefore, the improved method LS-SVM makes up for the limitation of SVM. They minimize the classification error by constructing a hyperplane in higher dimensional space and maximize the

TABLE 2: p values of features computed from each level of ATFFWT decomposition for J wave and normal classes.

Features	Level 2	Level 3	Level 4	Level 5	Level 6
FE p value	8.328×10^{-19}	3.943×10^{-19}	1.791×10^{-19}	1.293×10^{-19}	4.531×10^{-18}

TABLE 3: Mean (μ) and standard deviation (σ) of features computed from each level of ATFFWT decomposition for J wave and normal classes.

Level of decomposition	Features	J wave class ($\mu \pm \sigma$)	Normal class ($\mu \pm \sigma$)
Level 1	FE _{D1}	0.2787 \pm 0.0370	0.2254 \pm 0.1098
Level 2	FE _{D2}	0.2638 \pm 0.2549	0.2396 \pm 0.0723
Level 3	FE _{D3}	0.3081 \pm 0.2995	0.2602 \pm 0.0259
Level 4	FE _{D4}	0.2807 \pm 0.0578	0.2340 \pm 0.1492
Level 5	FE _{D5}	0.1032 \pm 0.0216	0.0984 \pm 0.0743

distance of any one of the classes. The classification decision function of LS-SVM is defined mathematically as [31]

$$J = \text{sign} \left[\sum_{m=1}^M \alpha_m \omega_m K(z, z_m) + b \right], \quad (15)$$

where $K(z, z_m)$ represents a kernel function, α_m denotes the Lagrangian multiplier, z_m is the M th input vector, ω_m is the target vector, and b represents bias term. In this work, we selected Radial Basis Function (RBF) and Morlet Wavelet (MW) as the kernel function.

The expression of the RBF kernel is given as [32]

$$K(z, z_m) = e^{-\|z - z_m\|^2 / 2\sigma^2}, \quad (16)$$

where σ is the kernel parameter and it controls the width of the RBF kernel function. MW kernel can be represented as follows [33, 34]:

$$K(z, z_m) = \prod_{n=1}^D \cos \left[v_0 \frac{z^n - z_m^n}{l} \right] e^{-\|z^n - z_m^n\|^2 / 2l^2}, \quad (17)$$

where D is the dimension of the feature set and l denotes the scale factor of MW kernel. LS-SVM is widely used in the diagnosis of diabetes [35], analysis of the heart sound signals [36], and classification of focal EEG class [37, 38], seizure class [26, 39], and glaucoma using fundus images [40].

3. Results and Discussion

In this present study, we have segmented ECG signals into beats. Then, to extract features, the ECG beats are employed to ATFFWT decomposition method to obtain a series of subbands signals. The plots of real values of coefficients are shown in Figures 2(a) and 2(b) for J wave and normal classes, respectively. These subband signals are obtained from ATFFWT-based decomposition of the ECG beats. In Figure 2, CN is the decomposition detail coefficient and CO stands for the magnitude of the coefficient. L1a and L1b are described as detail coefficients of first level. Similarly, L2a and L1b, L3a and L3b, L4a and L4b, and L5a and L5b represent second-level, third-level, fourth-level, and fifth-level detail coefficients, respectively. The parameters of ATFFWT are selected by way

of trial and error experimentation to reach the maximum classification accuracy. In the present study, the values of the parameters are $p = 5$, $q = 6$, $r = 1$, $s = 2$, and $\beta = 0.8(r/s)$ [9, 21]. Therefore, $d = 5/6$, $Q = 4$, and $R = 3$. The high Q -factor denotes that desirable frequency analysis of ECG signals is acquired. In order to select the desirable decomposition level, p values from subbands at different levels of ATFFWT-based decomposition are computed using Kruskal-Wallis test [41]. It should be noted that the lowest p value indicates the best discrimination ability of the feature. The lowest p values are observed at the fifth level of decomposition, as can be seen in Table 2. There was no significant improvement when the decomposition is increased from fifth to sixth level. Accordingly, we select fifth level of decomposition for analyzing J wave and normal ECG signals.

FE is computed from detail coefficients, and the parameters demanded to compute FE are also chosen by trial and error experimentation, thus getting maximum classification accuracy; the values of the parameters m , n , and r are selected to be 5, 2, and 0.3 separately [27]. Mean (μ) and standard deviation (σ) of the features computed from various levels of ATFFWT decomposition for J wave and normal classes are offered in Table 3, and we can observe from the table that, for J wave classes, μ and σ of the FE features have higher values at each decomposition level as compared to normal classes. Boxplots for FE at various levels of decomposition are depicted in Figures 3(a)–3(e). Feature scoring method is employed for selection, thus removing irrelevant and redundant features, and search for the optimal feature subset. In this work, the dimension of the feature vector is five, and it derives from the FE computation of 5 levels subbands. The score value of each feature evaluated using feature scoring method is shown in Figure 4. It can be observed that FE_{D1}, FE_{D4}, and FE_{D5} have higher score values than those of FE_{D2} and FE_{D3} features.

Accuracy (ACC), sensitivity (SEN), and specificity (SPE) are computed at each level of decomposition and are tabulated in Table 4. It is obviously seen that the significant improvement in classification performance occurs at second level of decomposition. The average accuracy, average sensitivity, and average specificity of classification are found to be 97.56%, 95.69%, and 95.78% for RBF kernel and 97.61%, 97.76%, and 95.82% for MW kernel functions, respectively.

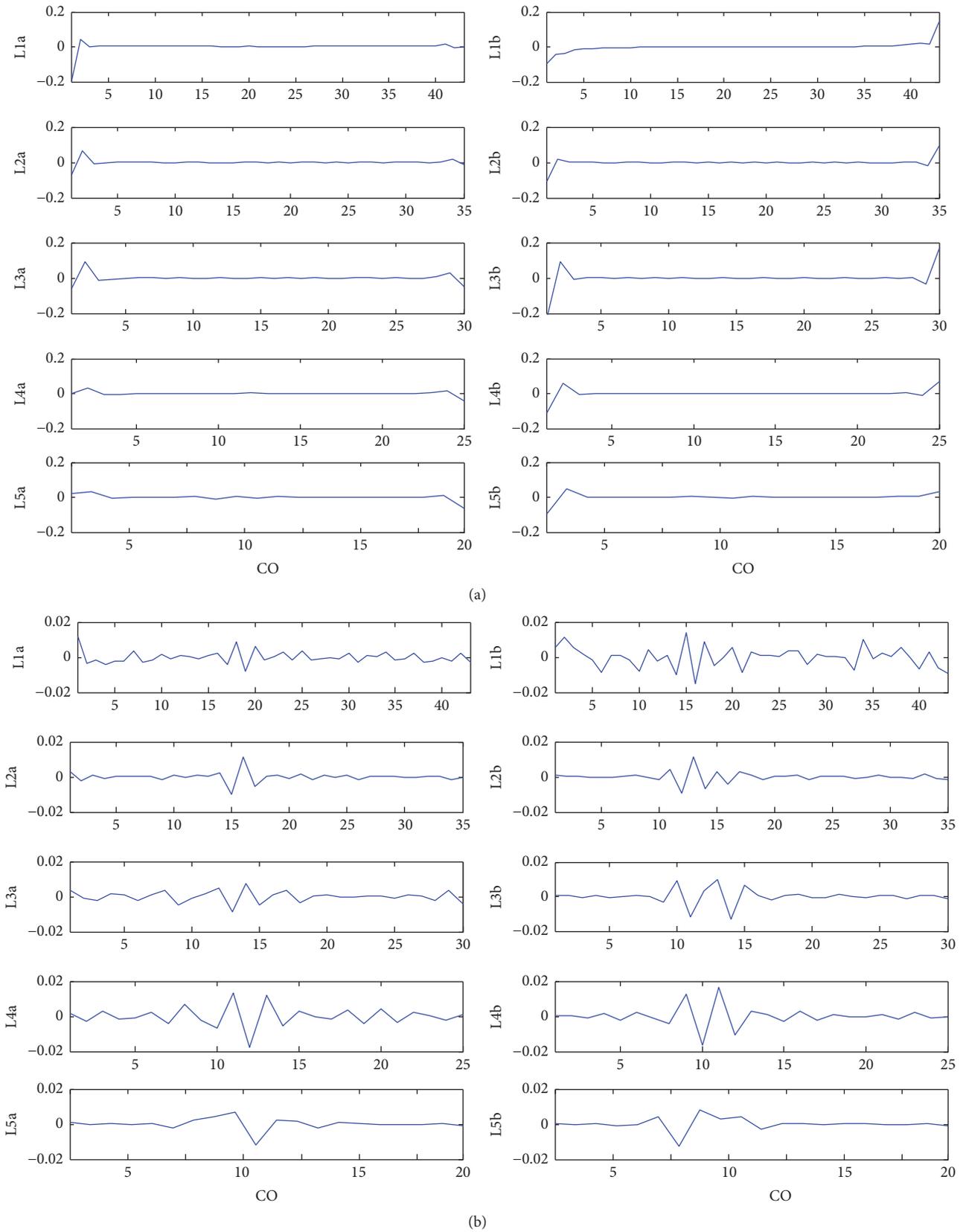


FIGURE 2: Plots of real coefficients of ATFFWT decomposition: (a) normal class and (b) J wave class.

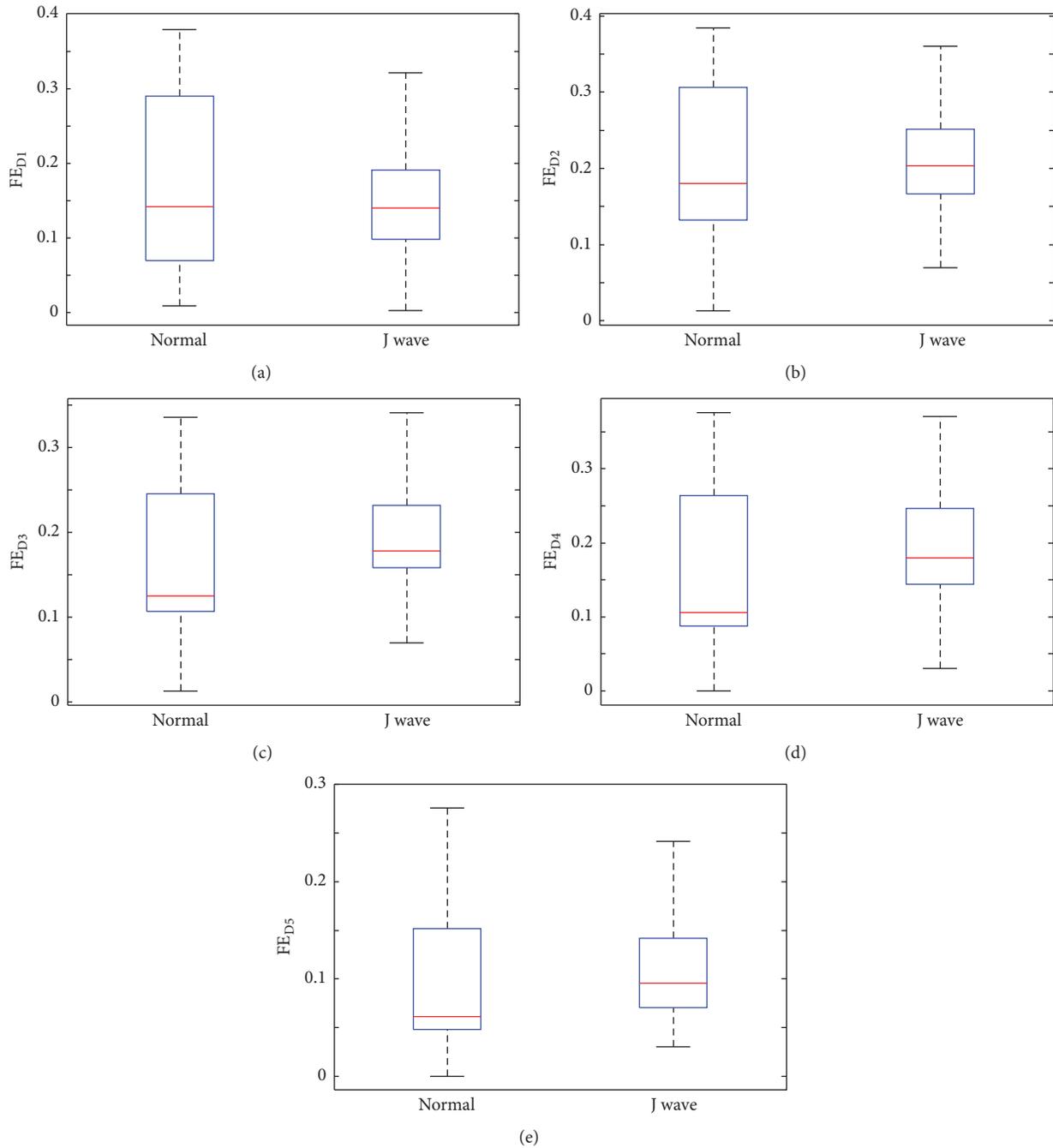


FIGURE 3: Boxplots for features at each decomposition level of normal ECG and J wave: (a) Level 1, (b) Level 2, (c) Level 3, (d) Level 4, and (e) Level 5.

TABLE 4: Classification performance of LS-SVM classifier using different kernel functions.

Feature number	Kernel function	ACC (%)	SEN (%)	SPE (%)
All features	RBF	96.73	95.18	95.17
	MW	96.79	95.47	95.29
Feature selection	RBF	97.56	95.69	95.78
	MW	97.61	97.76	95.82

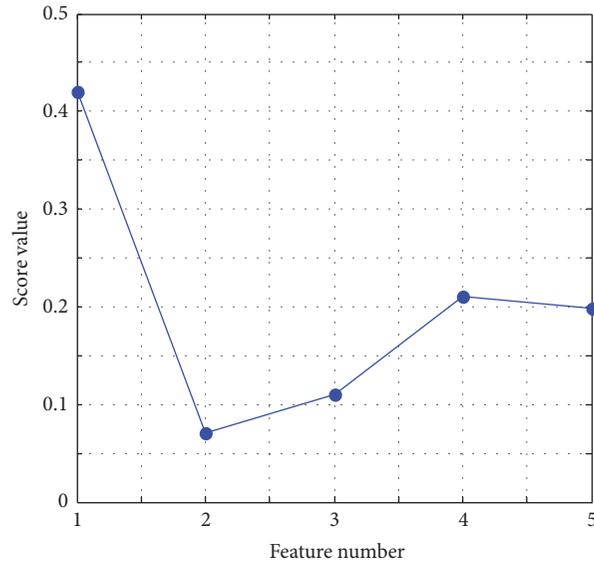


FIGURE 4: Score value of each feature using feature scoring.

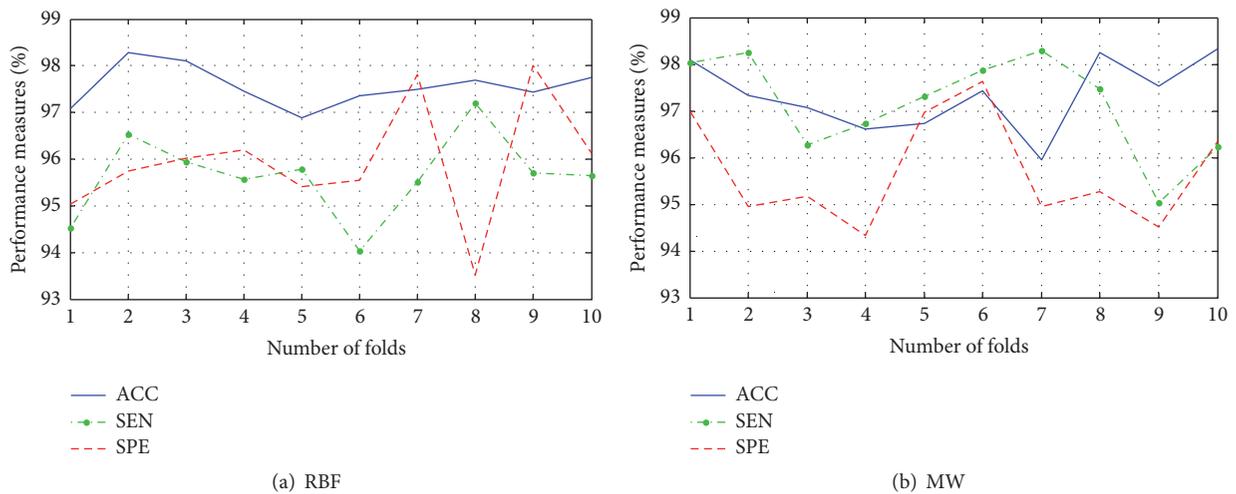


FIGURE 5: Plots showing performance measures versus the number of folds for LS-SVM classifier for kernel function: (a) RBF; (b) MW.

We can also observe that after using feature selection method the values of average accuracy, average sensitivity, and average specificity are higher compared with using all features. The values of the kernel parameters we selected are $\sigma = 0.9$ for RBF kernel and $l = 1.1$ and $\nu_0 = 0.35$ for MW kernel. In order to gain a stable and reliable classification performance of LS-SVM, tenfold cross validation technique is applied to avoid the possibility of overfitting of the model [42, 43]. That is, the dataset of features is divided into 10 subsets randomly and consists of one testing subset and nine training subsets. The performances are averaged after ten iterations. ACC, SEN, and SPE for tenfold cross validation method can be seen in Figures 5(a) and 5(b) for RBF and MW kernels, respectively.

Currently, however, only few research teams have studied the automatic detection and classification of J wave from the perspective of signal processing and machine learning.

In [13], a method for automated detection of J wave is performed. Their method is based on locating a break point in the descending limb of the terminal QRS. New logic was added to Glasgow ECG analysis program to automate the detection of J wave. The high sensitivity of 90.5% and specificity of 96.5% are achieved using this method. A technique based on digital 12-lead electrocardiogram is proposed for the automated J wave detection in [14]. ECG signals were automatically processed with the GE Marquette 12-SL program 2001 version (GE Marquette, Milwaukee, WI). Thereafter, the functional data analysis techniques were applied to the processed ECG signals. The detection sensitivity and specificity are 89% and 86%, respectively. The two methods have no parameter of accuracy. In [15], a method for automated J wave detection and characterization based on feature extraction is developed. First, five features including

TABLE 5: Comparison of proposed method with other existing methods of J wave automatic detection.

Authors	Applied methods	Classification	ACC (%)	SEN (%)	SPE (%)
Clark et al. (2014)	Glasgow ECG analysis program with new logic		91.3	89.5	94.5
Wang et al. (2015)	GE Marquette 12-SL program 2001 version	Functional data analysis	89.6	88.45	87.8
Li et al. (2015)	Time-domain features and discrete wavelet transform (DWT)	Hidden Markov model (HMM)	93.35	91.32	92.2
Li et al. (2015)	Curve fitting and wavelet transform (WT)	SVM	92.58	93.21	93.87
Li et al. (2016)	Time-domain features, power spectrum, and cumulative probability	Decision tree (DT)	96.03	95.1	96.25
In the present work	ATFFWT and Fuzzy Entropy	LS-SVM	97.61	97.76	95.82

three time-domain features and two wavelet-based features are defined. Then, PCA is used to reduce the dimension of these features. Highest classification accuracy of 92.5% is attained using HMM. An approach for J wave autodetection is based on support vector machine [16]. CF and WT are used for feature extraction from ECG segments data. Also 92.5% classification accuracy is reported using SVM classifier. In [17], a novel J wave detection method based on massive ECG data and MapReduce is presented. The power spectrum and the cumulative probability of ECG signals are computed as features. Further, DT is applied to classify and recognize ECG signals. Lastly, they implemented the above process under the parallel programming model MapReduce to handle the massive ECG data and achieved an accuracy of 96.23%.

In the present work, we have developed a novel methodology for automatic detection of J wave. ATFFWT is applied to decompose the processed ECG signals into the desired subbands to capture the hidden useful information from ECG segment beats of 100 normal and 15 J wave subjects. We have used ATFFWT-based decomposition due to its desirable property; it is able to extract more meaningful information and is suitable for biomedical signals analysis. Furthermore, FE is computed on the decomposed subbands to fetch the information from detail coefficients at each level. FE can measure the similarity based on exponential function in the time series [25]. Feature scoring method is employed for selection, thus removing irrelevant and redundant features, and searching for the optimal feature subset. Finally, these clinically significant features are fed to a LS-SVM classifier with different kernel functions. We have observed an accuracy of 97.56% and 97.61% using RBF and MW kernel functions, respectively. Finally, to verify the effectiveness of our method, we have evaluated all methods with the latest collected data, and the summary of performance of other existing methods of J wave automatic detection is shown in Table 5.

4. Conclusion

An automated detection of J wave from ECG signals with high speed and accuracy is a great challenge task. In this study, a new technique is proposed to detect the J wave automatically using ECG segment beats. ATFFWT decomposition method and FE extraction are employed to catch the hidden information from ECG signals. Feature scoring method is used to optimize the classification performance. Highest classification performance is founded using LS-SVM classifier with tenfold cross validation procedure while training and testing. The limitation of this work is small dataset; we have used only 15 J wave subjects. Before the developed effective algorithm can be used to design an expert system to aid clinicians in their regular diagnosis, it needs to be tested using large dataset. And another limitation is that utilization of a fixed beat length is not always optimum because of fast and slow varying heart rhythms. Better methods of adaptive beat size segmentation are needed to study. In the future, the work could be extended in three aspects: (1) it would be of interest to develop an expert model for filter parameters p , q , r , s , and β optimization of ATFFWT; (2) it is highly desirable that the large dataset would be used to evaluate the proposed technique; (3) the method can be used for other biosignals application, such as electroencephalograph (EEG) and electromyogram (EMG).

Conflicts of Interest

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

Acknowledgments

This work was supported by the National High Technology Research and Development Program (863 Program) of China

[Grant no. 2015AA016901, High Linearity Laser Diode Array and High Saturation Power Photodiode Array]; the National Natural Science Foundation of China [Grant no. 61371062, Research on the Theory and Key Technology of J Wave Extraction in ECG Signal]; and the International Cooperation Project of Shanxi Province [Grant no. 201603D421014, Three-Dimensional Reconstruction Research of Quantitative Multiple Sclerosis Demyelination].

References

- [1] G. Roth A, C. Johnson, A. Abajobir et al., "Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015," *Journal of the American College of Cardiology*, 2017.
- [2] N. D. Wong, "Epidemiological studies of CHD and the evolution of preventive cardiology," *Nature Reviews Cardiology*, vol. 11, no. 5, pp. 276–289, 2014.
- [3] M. J. Junttila, S. J. Sager, J. T. Tikkanen, O. Anttonen, H. V. Huikuri, and R. J. Myerburg, "Clinical significance of variants of J-points and J-waves: early repolarization patterns and risk," *European Heart Journal*, vol. 33, no. 21, pp. 2639–2643, 2012.
- [4] C. Antzelevitch and G.-X. Yan, "J wave syndromes," *Heart Rhythm*, vol. 7, no. 4, pp. 549–558, 2010.
- [5] M. Badri, A. Patel, and G.-X. Yan, "Cellular and ionic basis of J-wave syndromes," *Trends in Cardiovascular Medicine*, vol. 25, no. 1, pp. 12–21, 2015.
- [6] H. V. Huikuri, "Separation of Benign from Malignant J waves," *Heart Rhythm*, vol. 12, no. 2, pp. 384–385, 2015.
- [7] Y. Aizawa, M. Sato, H. Kitazawa et al., "Tachycardia-dependent augmentation of "notched J waves" in a general patient population without ventricular fibrillation or cardiac arrest: not a repolarization but a depolarization abnormality?" *Heart Rhythm*, vol. 12, no. 2, pp. 376–383, 2015.
- [8] R. J. Martis, U. R. Acharya, and L. C. Min, "ECG beat classification using PCA, LDA, ICA and discrete wavelet transform," *Biomedical Signal Processing and Control*, vol. 8, no. 5, pp. 437–448, 2013.
- [9] M. Kumar, R. B. Pachori, and U. R. Acharya, "An efficient automated technique for CAD diagnosis using flexible analytic wavelet transform and entropy features extracted from HRV signals," *Expert Systems with Applications*, vol. 63, pp. 165–172, 2016.
- [10] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.
- [11] I. Bayram, "An analytic wavelet transform with a flexible time-frequency covering," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1131–1142, 2013.
- [12] A. L. Goldberger, L. A. N. Amaral, L. Glass et al., "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals [электронныйресурс]," *Circulation*, vol. 101, pp. e215–e220, 2000.
- [13] E. N. Clark, I. Katibi, and P. W. Macfarlane, "Automatic detection of end QRS notching or slurring," *Journal of Electrocardiology*, vol. 47, no. 2, pp. 151–154, 2014.
- [14] Y. Wang, H.-T. Wu, I. Daubechies, Y. Li, E. H. Estes, and E. Z. Soliman, "Automated J wave detection from digital 12-lead electrocardiogram," *Journal of Electrocardiology*, vol. 48, no. 1, pp. 21–28, 2015.
- [15] D. Li, Y. Bai, and J. Zhao, "A method for automated J wave detection and characterisation based on feature extraction," in *Big Data Computing and Communications*, pp. 421–433, Springer International Publishing, 2015.
- [16] D. Li, X. Liu, and J. Zhao, "An approach for J wave auto-detection based on support vector machine," in *Big Data Computing and Communications*, pp. 453–461, Springer International Publishing, 2015.
- [17] D. Li, W. Ma, and J. Zhao, "A novel J wave detection method based on massive ECG data and MapReduce," in *International Conference on Big Data Computing and Communications*, pp. 399–408, Springer International Publishing, 2016.
- [18] N. Iyengar, C.-K. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz, "Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 271, no. 4, pp. R1078–R1084, 1996.
- [19] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 32, no. 3, pp. 230–236, 1985.
- [20] U. R. Acharya, V. K. Sudarshan, J. E. W. Koh et al., "Application of higher-order spectra for the characterization of Coronary artery disease using electrocardiogram signals," *Biomedical Signal Processing and Control*, vol. 31, pp. 31–43, 2017.
- [21] C. L. Zhang, B. Li, B. Q. Chen et al., "Weak fault signature extraction of rotating machinery using flexible analytic wavelet transform," *Mechanical Systems and Signal Processing*, vol. 64, pp. 162–187, 2015.
- [22] M. Kumar, R. B. Pachori, and U. R. Acharya, "Characterization of coronary artery disease using flexible analytic wavelet transform applied on ECG signals," *Biomedical Signal Processing and Control*, vol. 31, pp. 301–308, 2017.
- [23] U. R. Acharya, H. Fujita, M. Adam et al., "Automated characterization and classification of coronary artery disease and myocardial infarction by decomposition of ECG signals: a comparative study," *Information Sciences*, vol. 377, pp. 17–29, 2017.
- [24] M. Kumar, R. B. Pachori, and U. R. Acharya, "Use of accumulated entropies for automated detection of congestive heart failure in flexible analytic wavelet transform framework based on short-term HRV signals," *Entropy*, vol. 19, no. 3, article 92, 2017.
- [25] W. T. Chen, Z. Z. Wang, H. B. Xie, and W. Yu, "Characterization of surface EMG signal based on fuzzy entropy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 2, pp. 266–272, 2007.
- [26] R. Sharma and R. B. Pachori, "Classification of epileptic seizures in EEG signals based on phase space representation of intrinsic mode functions," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1106–1117, 2015.
- [27] U. R. Acharya, H. Fujita, V. K. Sudarshan, S. Bhat, and J. E. W. Koh, "Application of entropies for automated diagnosis of epilepsy using EEG signals: a review," *Knowledge-Based Systems*, vol. 88, pp. 85–96, 2015.
- [28] J. Pohjalainen, O. Räsänen, and S. Kadioglu, "Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits," *Computer Speech and Language*, vol. 29, no. 1, pp. 145–171, 2015.
- [29] C. Camara, K. Warwick, R. Bruña, T. Aziz, F. del Pozo, and F. Maestú, "A fuzzy inference system for closed-loop deep brain stimulation in Parkinson's disease," *Journal of Medical Systems*, vol. 39, no. 11, article 155, 2015.

- [30] S. Aydin, M. A. Tunga, and S. Yetkin, "Mutual information analysis of sleep EEG in detecting psycho-physiological insomnia," *Journal of Medical Systems*, vol. 39, no. 5, p. 43, 2015.
- [31] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [32] A. H. Khandoker, D. T. H. Lai, R. K. Begg, and M. Palaniswami, "Wavelet-based feature extraction for support vector machines for screening balance impairments in the elderly," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 4, pp. 587–597, 2007.
- [33] V. Bajaj and R. B. Pachori, "Classification of seizure and nonseizure EEG signals using empirical mode decomposition," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1135–1142, 2012.
- [34] M. Zavar, S. Rahati, M.-R. Akbarzadeh-T, and H. Ghasemifard, "Evolutionary model selection in a wavelet-based support vector machine for automated seizure detection," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10751–10758, 2011.
- [35] R. B. Pachori, M. Kumar, and P. Avinash, "An improved online paradigm for screening of diabetic patients using RR-interval signals," *Journal of Mechanics in Medicine and Biology*, vol. 16, no. 01, Article ID 1640003, 2016.
- [36] S. Patidar and R. B. Pachori, "Classification of cardiac sound signals using constrained tunable-Q wavelet transform," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7161–7170, 2014.
- [37] R. Sharma, R. B. Pachori, and U. Rajendra Acharya, "An integrated index for the identification of focal electroencephalogram signals using discrete wavelet transform and entropy measures," *Entropy*, vol. 17, no. 8, pp. 5218–5240, 2015.
- [38] R. Sharma, R. B. Pachori, and U. R. Acharya, "Application of entropy measures on intrinsic mode functions for the automated identification of focal electroencephalogram signals," *Entropy*, vol. 17, no. 2, pp. 669–691, 2015.
- [39] M. Sharma, R. B. Pachori, and U. Rajendra Acharya, "A new approach to characterize epileptic seizures using analytic time-frequency flexible wavelet transform and fractal dimension," *Pattern Recognition Letters*, vol. 94, pp. 172–179, 2017.
- [40] S. Maheshwari, R. B. Pachori, and U. R. Acharya, "Automated diagnosis of glaucoma using empirical wavelet transform and correntropy features extracted from fundus images," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 803–813, 2017.
- [41] P. E. McKight and J. Najab, "Kruskal-Wallis test," *Corsini Encyclopedia of Psychology*, 2010.
- [42] S.-N. Yu and Y.-H. Chen, "Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1142–1150, 2007.
- [43] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *International Joint Conferences on Artificial Intelligence*, vol. 14, no. 2, pp. 1137–1145, 1995.

Research Article

Predicting Traffic Dynamics with Driver Response Model for Proactive Variable Speed Limit Control Algorithm

Huixuan Ye,¹ Lili Tu,¹ and Jie Fang ^{1,2}

¹College of Civil of Engineering, Fuzhou University, Fuzhou, Fujian, China

²University of Alberta, 3-019 NREF, Edmonton, AB, Canada T6G 2W2

Correspondence should be addressed to Jie Fang; fangjie@fzu.edu.cn

Received 21 December 2017; Revised 24 February 2018; Accepted 22 March 2018; Published 6 May 2018

Academic Editor: Li Xu

Copyright © 2018 Huixuan Ye 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.

Variable Speed Limit (VSL) control contributes to potential crash risk reduction by suggesting a suitable dynamic speed limit to achieve more stable and uniform traffic flow. In recent studies, researchers adopted macroscopic traffic flow models and perform prediction-based optimal VSL control. The response of drivers to the advised VSL is one of the most critical parameters in VSL-controlled speed dynamics modeling, which significantly affects the accuracy of traffic state prediction as well as the control reliability and performance. Nevertheless, the variations of driver responses were not explicitly modeled. Thus, in this research, the authors proposed a dynamic driver response model to formulate how the drivers respond to the advised VSL during various traffic conditions. The model was established and calibrated using field data to quantitatively analyze the dynamics of drivers' desired speed regarding the advised VSL and current traffic state variables. A proactive VSL control algorithm incorporating the established driver response model was designed and implemented in field-data-based simulation study. The design proactive control algorithm modifies VSL in real-time according to the traffic state prediction results, aiming to reduce potential crash risks over the experiment site. By taking into account the real-time driver response variations, the VSL-controlled traffic state dynamics was more accurately predicted. The experimental results illustrated that the proposed control algorithm effectively reduces the crash probabilities in the traffic network.

1. Introduction

Active Transportation and Demand Management (ATDM) has become a well-known concept in the last decade. ATDM aims to mitigate daily recurrent traffic congestion with the support of real-time traffic data monitoring, processing, and control optimizing. It relies on fast emerging technology for traffic data capturing and information sharing in order to facilitate coordinated control and achieve optimal traffic conditions at the network level. Along with other active traffic control strategies, Variable Speed Limit (VSL) control is one of the most intensively studied control measures. The static speed limit posted on a highway section defines the highest driving speed allowed under prevailing traffic conditions. Nevertheless, such static speed limits may fail to suggest the most appropriate operating speed during certain extreme circumstances [1], including the following: a drop in capacity due to road construction, a change in road conditions due to

severe weather (fog, snow storm, etc.), or high traffic demand generated by large events. Earlier studies reported that, by implementing VSL to enhance speed homogeneity, traffic conditions can be improved in terms of both safety [1–3] and mobility [4, 5].

Given that the nature of VSL control is to alter the operating speed of traffic flow, the response of drivers to the advised VSL is a critical factor in control performance. In reality, it is arbitrary to assume that traffic flow will follow the advised VSL value strictly. However, in the existing literature, other than not presenting the assumption of how the speed dynamic responds to VSL control, only a few studies in the past explicitly investigated how the speed dynamic responds to VSL control. Hellinga and Mandelzys [6] assumed four different possible levels of driver responses (low, medium, high, and very high) to the VSL and then examined the sensitivity of VSL's control performance over operational and safety benefits subject to these categories. Several other studies [7]

adopted a similar approach to assume how certain percentages of drivers will respond to the advised VSL in simulation.

However, in the authors' opinion, the mechanism of driver responses to the VSL is a sophisticated system and it can hardly be represented accurately by simply assuming that the advised VSL will be complied with by a constant proportion of drivers. Instead, the response of flow speed to the VSL should vary according to the current traffic conditions. Moreover, such a dynamic response mechanism needs to be taken into consideration while anticipating future traffic states in prediction-based VSL control algorithms, such as [3, 8]. Therefore, in this study, the authors propose to establish a macroscopic VSL response model, which describes the change in drivers' responses to the implemented VSL dynamically, according to the current traffic conditions. This relationship was formulated to associate drivers' desired speed dynamically with traffic state variables, such as current volume or occupancy. The model was conducted through a regression of field experiment data. For model evaluation, this paper adopts the prediction-based VSL control algorithm described in the authors' earlier article [3] and has further improved its traffic state prediction module.

The remainder of this paper is organized into sections: Section 2 is a brief literature review of existing VSL control strategies and related speed response modeling efforts; Section 3 presents the proposed driver response modeling approach and the corresponding VSL control algorithm; Section 4 presents a field-data-based VISSIM simulation study to reproduce real-world traffic conditions and implement the proposed control algorithm. The VSL-controlled network performance is compared with the baseline condition (uncontrolled case) to evaluate the efficiency of the proposed control algorithm. Lastly, Section 5 presents conclusions and future work.

2. Literature Review

Realizing the importance of drivers' responses to VSL control, previous studies have investigated the correlation between driver compliance and VSL control. Real-world field implementation was examined first in both Europe and North America to analyze the effects. In general, field implementation in Europe was enforced by police through means of cameras or other monitoring methods, which resulted in overall higher compliance compared to the advisory-only VSL control in North America.

One of the earliest VSL driver response studies in Europe was implemented on a 20 kilometer freeway section over A2 motorway near Amsterdam, Netherlands [9]. The study investigated drivers' responses to the installed VSL control system, which lowered the speed limit from over 90 km/h to 70 km/h. Although the study reported that drivers follow the speed limit in general, the traffic flow dynamics under VSL control were not quantitatively analyzed to support the argument. Another study in Europe [10] was conducted in the UK on M25 motorway, which was automatically enforced by overhead cameras. Similarly, the study reported improved driver compliance with the VSL, although the increase was not quantified.

In North America, VSL field implementation was conducted in Washington [11]. In the Washington test, 25 VSL signs were installed on both directions of I-90 to reduce the traveling speed through construction zones. However, due to the existing traffic congestion, the study did not analyze the direct response to VSL control. The Minnesota study adopted a two-stage speed reduction scheme at work zone bottlenecks along I-494 in Twin Cities. The field data demonstrated that the average speed difference (1-minute interval) was reduced by 25 to 35 percent. It also reported that drivers' compliance was better when there was a larger gap between the current speed and advised speed limit. Another speed response test was conducted on a six-mile work zone section over I-80 near Wanship, Utah. The results showed that traffic flow followed the VSL closely and that there was no statistically significant difference in between. Chang et al. worked on integrating variable speed control and travel time information for alleviating day-to-day recurrent congestion on a highway corridor [12]. The VSL control and travel time information was updated based on real-time traffic condition collected by license-plate-recognition system.

More literature on this topic can be found among simulation-based studies. One reprehensive study is Hellinga and Mandelzys's work [6] in 2011, which tested server compliance levels to evaluate the sensitivity of the safety and operational impacts of VSL on driver compliance in PARAMICS. Four levels of driver response (low, medium, high, and very high) to the VSL were assumed and used to testify the impact of the response on control performance. They reported that safety is positively correlated with driver compliance, while the analysis of travel time went the other way.

The study mentioned above implemented VSL control based on logics using predefined thresholds, such as a decision tree [6]. Another approach to implementing VSL control is to adopt a macroscopic traffic flow model and perform prediction-based optimal control. Hegyi et al.'s work [8, 13, 14] is among the earliest studies on this approach. The author adopted the Model Predictive Control (MPC) framework to facilitate VSL control based on the prediction results from an extended version of the famous macroscopic traffic flow model: METANET. The control vector (advised VSL) was introduced into the original model to replace the speed expectation, and a driver compliance modifier (as a percentage) was attached to the control vector.

Carlson et al. [15] used a similar framework which adopts a macroscopic traffic flow model (modified METANET model) and performed prediction-based optimal control. The impact of driver compliance was not explicitly included in the prediction stage, and thus it was not clear how it affected the control performance.

The work by Hadiuzzaman et al. [7] further extended the study regarding the compliance level using MPC-based control. Multiple compliance levels were assumed for VSL control and the control benefits were measured. It was reported that VSL control has benefits in terms of both mobility and safety aspects over most compliance level assumptions, but the nature of the dynamic driver response is still not clear. Yang et al.'s recent work [16, 17] proposed to incorporate Kalman filter in the VSL control algorithm design, which was used

to surpass the uncertainty brought into the system by the driver compliance. The predicted result was used to optimize VSL targeting at travel time minimization and speed variance minimization.

In summary, there are only a few studies available to examine the critical relationship between driver response and VSL control. The existing literature either neglects this critical effect [15] or addresses it by assuming that a fixed proportion of drivers will follow the advised VSL [6–8, 18, 19], regardless of changing traffic state variables. This has two major drawbacks: (1) Assuming one (or several) fixed driver response proportion is not necessarily accurate for all scenarios. In reality, the drivers' response (change of desired speed) to the VSL will change along with current traffic conditions and the corresponding VSL. For example, during a congested period, drivers might follow a VSL of 60 km/h much better than during free-flow conditions. (2) As a sophisticated and critical effect, the drivers' response was not considered before optimizing the control. That is to say, the anticipated control performance provided by the existing prediction-based optimal control algorithms did not quantitatively measure the impact of driver compliance to the VSL. A model of drivers' responses must be included in the macroscopic traffic flow model for predicting the future traffic state after control effort.

Therefore, in this paper, the authors propose to analyze the mechanism of drivers' responses to the advised VSL based on the field data and establish a model to describe how traffic flow dynamics change according to different traffic states and the VSL control signal. Furthermore, this model has been introduced to improve a prediction-based VSL control algorithm by considering the impact of drivers' responses and providing more precise evidence for traffic state prediction before control optimization and implementation.

3. Prediction-Based VSL Control

In order to evaluate the performance of the proposed driver response model, an enhanced prediction-based active traffic control algorithm previously proposed by the authors in [3, 20] was adopted in this study. For the readers' convenience, here the authors briefly introduce the basic concept of the control algorithm and where the driver response model proposed in this paper fits in.

As indicated in Figure 1, the controller of the designated algorithm includes three major steps: traffic state predicting, control performance assessment, and optimization. The adopted control algorithm optimizes the control plan based on the current traffic conditions (traffic state variables). The traffic state predictions are estimated using the combined application of the macroscopic traffic flow model and the driver response model. Considering the drivers' dynamic response to the control inputs, these two models incorporate both the current traffic state $x(k)$ and the control plan $u(k)$ to predict the traffic state at the consecutive time step. The estimated traffic states and their associated control inputs are then evaluated by the system assessment model to select the optimal control input $u^*(k)$ with the best system performance. After applying the selected control input $u^*(k)$ to the traffic network, the updated traffic state variables are sent to

the controller again to optimize the control input for the next time step.

In addition to the driver response model, a macroscopic traffic flow model is needed for predicting traffic states. A classic macroscopic traffic flow model, such as METANET, divides a freeway into discrete links to analyze their spatial-temporal aggregated characteristics at divided time steps. Lu et al.'s model [18] is a modified version of the METANET model: the fundamental diagram assumption in the original METANET model is replaced by introducing the VSL control variable VSL ($u_m(k)$) directly into the model.

Lu's model is composed of the following equations:

$$\begin{aligned} \rho_m(k+1) &= \rho_m(k) \\ &+ \frac{T}{L_m \lambda_m} [q_{m-1}(k) - q_m(k) + q_{m,on} - q_{m,off}] \end{aligned} \quad (1)$$

$$\begin{aligned} v_m(k+1) &= v_m(k) + \frac{T}{\tau} (u_m(k) - v_m(k)) \\ &+ \frac{T}{L_m} v_m(k) (v_{m-1}(k) - v_m(k)) \\ &- \frac{\mu T}{\tau L_m} \left(\frac{\rho_{m+1}(k) - \rho_m(k)}{\rho_m(k) + \kappa} \right), \end{aligned} \quad (2)$$

where m represents the index of the links, while k represents the index of the time steps; $v_m(k)$, $q_m(k)$, and $\rho_m(k)$ are the speed, volume, and density of link m at the current time step k ; T is the length of the time step; L_m and λ_m are the length and number of lanes in link m ; $q_{m,on}$ and $q_{m,off}$ are the volume at the on-/off-ramp m ; $u_m(k)$ is the advised VSL value at link m ; τ , μ , and κ are the model parameters.

Equation (1) is the flow conservation equation. This equation expresses the flow conservation and calculation to measure the density at the consecutive time step. On the other hand, (2) interprets the mechanism of anticipating the speed dynamics. The velocity in the forthcoming time step was predicted by modifying the current link speed with three different terms: the relaxation term, convection term, and anticipation term. Among these three terms, the relaxation term measures the impact of the drivers' desired speed on the traffic flow. In the original METANET model, this term assumes that drivers will tend to accelerate/decelerate towards a desired speed (which is the value from the fundamental diagram assumption). In VSL-controlled environment, while the control signal (advised VSL) is missing, the original speed-density relationship assumption (FD diagram) proposed in METANET does not necessarily represent the driver response accurately.

Lu et al.'s and a couple of other later research works [3, 7, 18] replaced the desired speed directly with the advised VSL value $u_m(k)$, assuming that traffic will follow the advised VSL value strictly. As explained before, to more accurately reflect the real-world situation, this anticipation term should

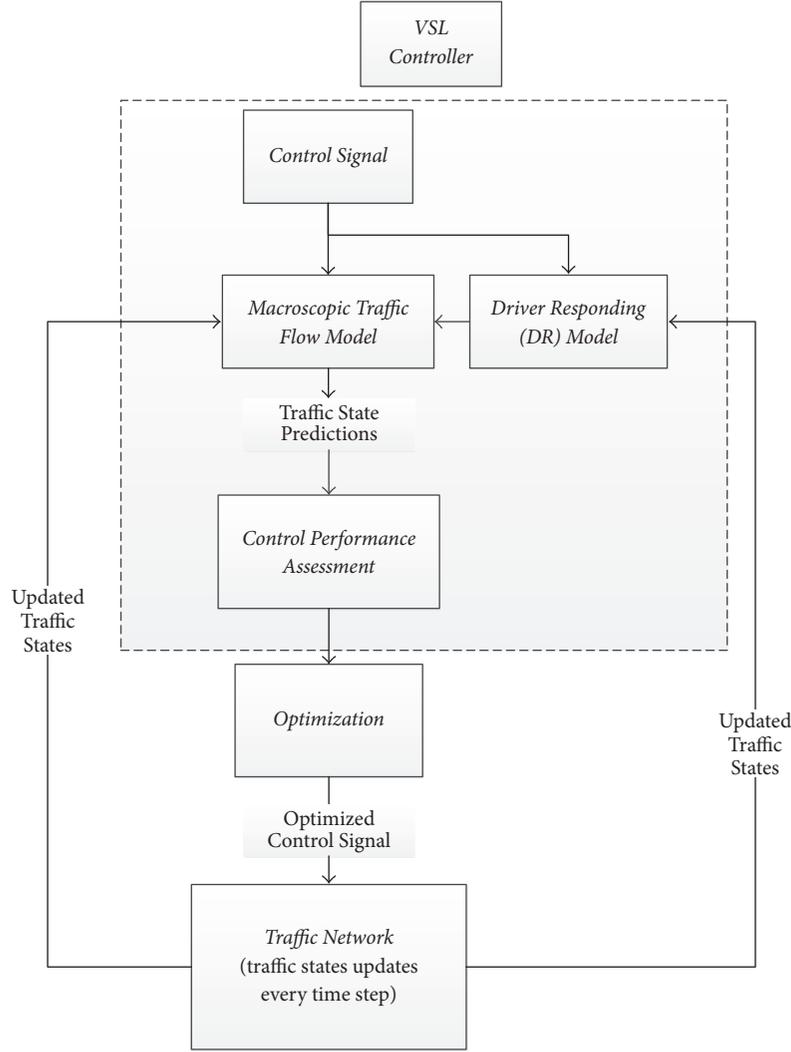


FIGURE 1: Established driver response (DR) model and adopted control algorithm.

be qualified to treat driver responses differently under different traffic conditions. Thus, in this study, the authors use the proposed driver response model to modify this term. The modified speed dynamic equation is expressed in the equation below:

$$\begin{aligned}
 v_m(k+1) = & v_m(k) + \frac{T}{\tau} (\text{DR}(k+1) - v_m(k)) \\
 & + \frac{T}{L_m} v_m(k) (v_{m-1}(k) - v_m(k)) \quad (3) \\
 & - \frac{\mu T}{\tau L_m} \left(\frac{\rho_{m+1}(k) - \rho_m(k)}{\rho_m(k) + \kappa} \right)
 \end{aligned}$$

Basically, this modified model adopted the proposed driver response model (DR($k+1$)) to replace the control variable in the relaxation term. By introducing this model, the drivers' desired speed does not simply equal the advised VSL (as in the original model). Instead, the drivers' responses to

the advised VSL under current traffic conditions are formulated. This modification will further improve the accuracy of traffic state prediction under various traffic conditions.

3.1. Modeling the Driver Response Mechanism

3.1.1. Model Formulation. To more accurately reflect the dynamic driver response mechanism in VSL control, the regression technique was employed to model the relationship among traffic status, control signal (advised VSL value), and drivers' response (the drivers' desired speed). It should be noted that the field-collected flow speed can only be considered as the desired speed when traffic is uncongested. Otherwise, the flow speed is restrained by the traffic conditions and hardly reflects the true desired speed associated with the current VSL value. Moreover, it has been proven that VSL mainly works to improve traffic operations by postponing the traffic breakdown [12]. After the traffic breakdown happens, drivers fall into a stop-and-go situation other than following the advised VSL. Thus, to design a driver response model that

works for VSL prediction and control optimization purpose, the main focus of the model should be investigating the data above the jammed region, which is the function of VSL control to lower the driving speed beforehand and postpone traffic breakdown. Therefore, in this work, the authors mainly focused on examining the scenarios with relatively high VSL values (60 km/h–80 km/h) to avoid having noise in the observed data.

During the test, a total of five variables were considered to establish the model, as expressed in

$$\begin{aligned} DR(k+1) \\ = R(u(k), v(k), v^\wedge(k), q(k), \rho(k), \rho(k+1)). \end{aligned} \quad (4)$$

And four model formulations were tested to examine which better represents this dynamic driver response mechanism:

$$DR(k+1) = C + ax_1 + bx_2 + \dots + zx_p \quad (I)$$

$$DR(k+1) = ax_1 + bx_2 + \dots + zx_p \quad (II)$$

$$DR(k+1) = C + a(x_1)^x + b(x_2)^y + \dots + z(x_p)^z \quad (III)$$

$$DR(k+1) = a(x_1)^x + b(x_2)^y + \dots + z(x_p)^z, \quad (IV)$$

where DR represents the driver responses. In order to incorporate the driver response into a macroscopic traffic flow model, it was represented by the averaged individual vehicle driving speed.

k represents the index of time steps; $u(k)$ is the advised VSL value; $v(k)$ is the flow speed at time step k ; $v^\wedge(k)$ is the difference between current speed $v(k)$ and advised VSL $u(k)$; $q(k)$ is the volume at time step k ; $\rho(k)$ and $\rho(k+1)$ are the observed density at time step k and predicted density at time step $k+1$, respectively.

3.1.2. Data Preparation and Processing. The data for modeling the driver responses were collected at a major connector highway installed with DSDSs (Dynamic Speed Display Signs) in Edmonton, Alberta, Canada. See Figure 2 for the demonstration of the experiment site.

The demonstrated highway corridor has a static speed limit of 80 kilometers per hour posted. The DSDSs were installed with sensors at the four locations (sites 1–4) as illustrated on the map to record the drivers' dynamic response to the advised speed limit: individual speed, volume, and occupancy (aggregated by 60-second interval). Among them, sites 2 and 4 were considered as advanced warning for drivers to decelerate from the original speed limit, and traffic state variables were collected at sites 2 and 4 to analyze their responses. A warning sign was placed about 400 m beforehand of the lowered speed limit sign. The whole speed decreasing segment is about 1.6 km, in which the driver will need to break from 80 to 60 kph. The distance among the signs was set to be consistent with the length of road segment in the field validation test. In the field, it is observed that most drivers were able to adjust their speed within the setup region, since

its length is similar to the typical advance warning sign of off-ramp and that on the freeway.

Traffic flow characteristics and vehicle classification data were collected for 63 continuous days with DSDS in use and 38 days without. The dataset obtained while the DSDSs were on was used. 28 days of data were used for model parameter calibration and 27 days of data were used for model validation (both with and without DSDS in use). Among the selected variables, segment speed and speed limit were directly measured, while segment density required postprocessing. The simplified equation below was used for calculating the segment density based on the available data (occupancy):

$$\rho = \text{occ.} \times \frac{52.8}{(Lv + Ls)}; \quad (5)$$

$$Lv = N_{\text{bin}1}L_{\text{bin}1} + N_{\text{bin}2}L_{\text{bin}2} + \dots + N_{\text{bin}n}L_{\text{bin}n} \quad Ls = 0.5 \text{ ft},$$

where occ. is the occupancy data collected by the sensor; ρ is the segment density; Lv is the averaged vehicle length during one data collection interval; $N_{\text{bin}i}$ is the vehicle count within one interval for classification i ; $L_{\text{bin}i}$ is the average vehicle length for classification i , which is the median value of the upper bound and lower bound of each classification.

3.1.3. Driver Response Model Calibration and Validation. In order to establish the model and calibrate the model parameters, both linear and nonlinear regression techniques were performed to analyze the aforementioned model formulations (I)~(IV) and the results are demonstrated in Table 1.

According to the regression analysis results, model formulation (II) outperformed the other three model formulations with the highest adjusted R square value of 0.92. Therefore it was selected as the formulation of the proposed model. One underlying reason of this selection is in order to exclude the disturbances from congestion (enforced speed reductions); this research analyzed the traffic data out of the jammed region. Thus, the driver response is more introductive but nonlinear as in the breakdown state. As mentioned before, the driver responses among this transition region between free-flow and congestion should be focused on the VSL control study in order to benefit the road users.

Furthermore, a significance level of 0.05 was selected as the criterion for including a particular variable in the final model, which excluded two candidate variables: the volume rate $q(k)$ and density $\rho(k)$. The advised VSL value $u(k)$, current speed $v(k)$, the predicted segment density $\rho(k+1)$ according to the flow changes, and $v^\wedge(k)$ the difference between current speed and advised speed were found to be statistically significant in the model. Since $u(k)$ and $v(k)$ are correlated to $v^\wedge(k)$, the one with lower significance level ($v(k)$, significance level = 0.03) was removed and the model was recalibrated. The final model and model estimates were shown in Table 1(b).

As shown in Table 1(b) of parameter estimation results, the magnitude of the parameter indicates that the speed limit has a larger impact than the current traveling speed. The estimated coefficient of density is negative, which suggests that drivers tend to lower their speed when the traffic is dense. Figure 3 shows the detailed model validation results.

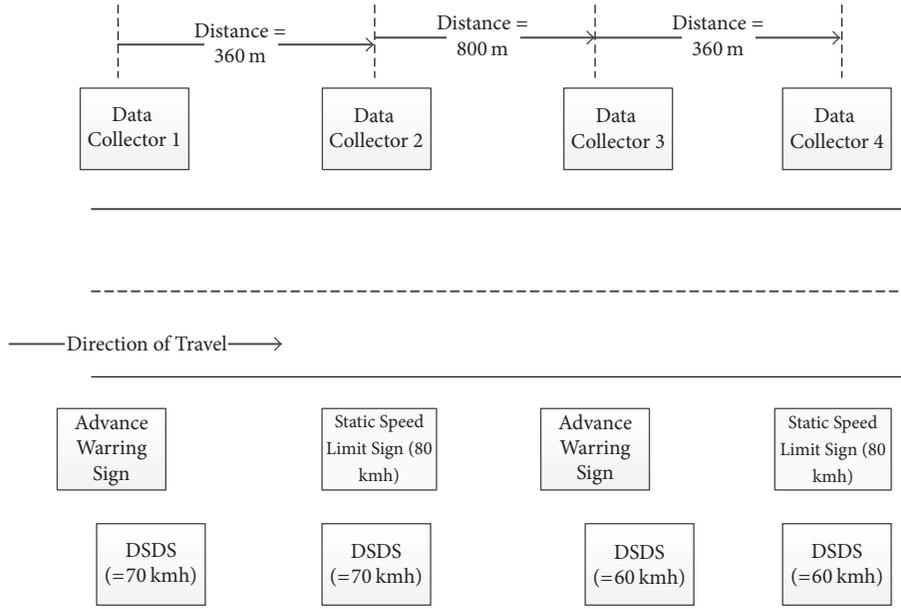


FIGURE 2: Experiment site layout in Edmonton, AB, Canada.

TABLE 1

(a) Statistics of examined model formulations

Variable	Model formulation (I)	Model formulation (II)	Model formulation (III)	Model formulation (IV)
R square	0.81	0.92	0.71	0.84
Adopted variable	Con, $u(k)$, $v(k)$, $v^\wedge(k)$, $\rho(k+1)$	$u(k)$, $v^\wedge(k)$, $\rho(k+1)$	Con, $u(k)$, $v(k)$, $v^\wedge(k)$, $\rho(k+1)$	$u(k)$, $v^\wedge(k)$, $\rho(k+1)$

(b) Parameter estimates for the adopted model (formulation (II))

Variable	Parameter estimate	Standard error	Significance	R square
$u(k)$	0.659	0.016	0.01	
$v^\wedge(k)$	0.474	0.008	0.01	0.92
$\rho(k+1)$	-0.577	0.038	0.03	

For model validation, the proposed model was adopted to predict the drivers' responses in the validation dataset. As demonstrated in Figure 3, the model-predicted value matches the field data closely (mean error < 1 km/h, std = 6.16 km/h). Also, there is no significant bias observed since the median and average drop within a 0–4 km/h range. Therefore, it is concluded that the proposed model can be used to formulate the dynamic driver response mechanism. Therefore the following calibrated driver response model DR (6) can be substitute into the speed dynamics model (3) for real-time VSL-controlled speed predictions.

$$\begin{aligned} DR(k+1) = & 0.659u(k) + 0.474v^\wedge(k) \\ & - 0.577\rho(k+1). \end{aligned} \quad (6)$$

3.2. VSL Control Optimization. As shown in the control algorithm (Figure 1), the optimization module in the adopted VSL control algorithm takes the evaluation results from the network performance assessment module to determine the optimized VSL control input. Although the proposed traffic

prediction model process and updates the prediction result at 20-second interval, in practice, it is unrealistic for drivers to adjust their desire in such high frequency. Altering the advised VSL by such short interval will confuse the drivers and therefore encouraging speed violations instead. Thus, in the control algorithm, the VSL control signal was optimized every 5 minutes according to predicted network safety performance. The network performance is determined by a precursor collision prediction model the authors presented in their previous research [3, 5] and the following equation:

$$\begin{aligned} P(u(k)) \\ = \frac{\exp(\text{con} + \alpha SV_2 + \beta \log AO_1 + \gamma \log SS_2)}{1 + \exp(\text{con} + \alpha SV_2 + \beta \log AO_1 + \gamma \log SS_2)}, \end{aligned} \quad (7)$$

where SV_2 is the standard deviation of volume in the last 5–10 minutes;

AO_1 is the average occupancy in the last 0–5 minutes before;

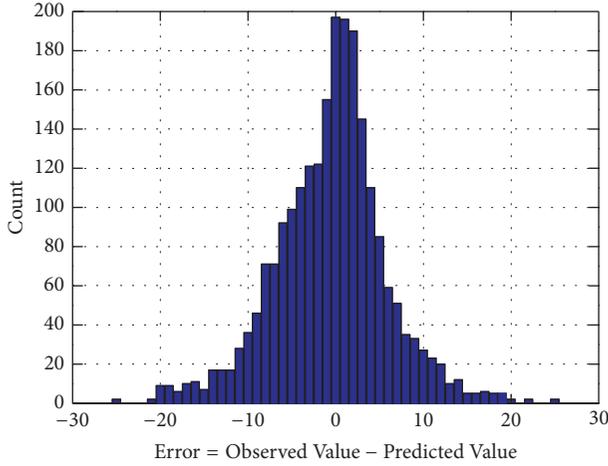


FIGURE 3: Validation results of proposed driver response model (one week of data, error units: km/h).

SS_2 is the standard deviation of speed 5–10 minutes prior to the incident;

Con, α , β , and γ are the model parameters to be calibrated.

Equation (6) was established through the logistic regression technique to identify the critical traffic state variables for modeling the probability of traffic incidents. For more details of how the crash probability was established, system constraints, and control plan optimization techniques of the control algorithm, the authors refer to their previous study [3, 5].

The optimization aims to minimize the overall system safety risk, which is a problem of minimizing the objective function J over the examined prediction horizon N_p (the time window of the predicted traffic states, set at 10 minutes in this study).

$$J(u(k)) = \sum_{j=1}^{N_p-1} \sum_{i=1}^M P_{i,j}(u_{i,j}(k)). \quad (8)$$

The objective function in (8) measures the summation of measured incident probability over all of the m analyzed links within the prediction horizon. The collision probabilities were evaluated at every link and at all time steps within the prediction horizon N_p , separately for each feasible control input. The average overall collision probability was then used to determine the optimal VSL control input.

The authors adopted a simple SQP (Sequence Quadratic Program) to solve the control optimization problem. In the selected experiment site, the adopted SQP method was able to converge the optimization within minutes, which makes it suitable for online application. The computation complexity may vary depending on the required optimization accuracy and the number of control variables. For more discussion regarding the optimization technique, the authors would refer to another article that works on solving complex multi-objective optimization problem in active freeway control [21].

4. Evaluation and Results Analysis

4.1. Experimental Design. To assess the efficiency of using the proposed model in the prediction-based VSL control algorithm, a field-data-based simulation study was conducted. A local freeway corridor allocated in the same region as the model calibration site was chosen for evaluation. The selected corridor is the northbound segment of Whitemud Drive (WMD), which is a major highway that serves as the inner ring road of Edmonton, Alberta, Canada (Figure 4).

According to the local Department of Transportation (DOT), the WMD corridor carries heavy commuter trips of more than 40,000 in AADT (Annual Average Daily Traffic). Over 30 loop detectors installed along the corridor monitor the real-time traffic conditions along the corridor at 60-second intervals. The detector report shows that the corridor suffers from daily recurrent congestion during peak hours for most weekdays; the flow speed is lower than 50 km/h for more than 70% of the workday peak hours. For evaluation, the selected corridor was projected into the microscopic traffic simulation tool VISSIM with a complete morning peak hour demand profile (collected from the field by the loops, 6:00–9:00 AM). To model the network, the selected experiment site was divided into 13 links (as shown in Figure 4). The underlying principle of segmentation is that, within each link, traffic characteristics should be consistent, for example, the same lane number and speed limit. Furthermore, for each link, it should possess at least one loop detection station, from which the 20-second interval loop detector data were processed to represent the real-time traffic condition on that link. The simulation platform was calibrated by these corresponding field-collected loop detector data.

In the selected corridor, two bottleneck locations were identified, and thus four VSL signs were placed before and after each of the identified bottlenecks and optimized by the proposed VSL algorithm with the enhanced driver response model. In order to accurately reproduce the driver behavior during VSL control, the driver behavior profile on the site was calibrated for different VSL values according to the aforementioned DSDS data. Since the DSDS data and the WMD corridor are in the same area (6–8 km apart) and both carry mostly commuting traffic, the authors assumed that the characteristics of their driver responses to the VSL control were similar. With such calibration, the simulated traffic flow will respond to the VSL differently instead of receiving the same traffic flow characteristics for all VSL values; this procedure will make the simulation work more realistic and close to the actual field situation.

4.2. Result Analysis. To appraise the proposed VSL control algorithm, the safety performance of the selected corridor without VSL control was studied as the baseline condition and compared with the VSL-controlled scenario. To quantitatively perform the evaluation, the collision probability (CP) was taken to measure the network-wide safety performance. Using the adopted network safety performance model (7), CP was measured for each link at each time step and the average overall CP was recorded for comparison. It should be noted that the first half-hour of the study period was considered

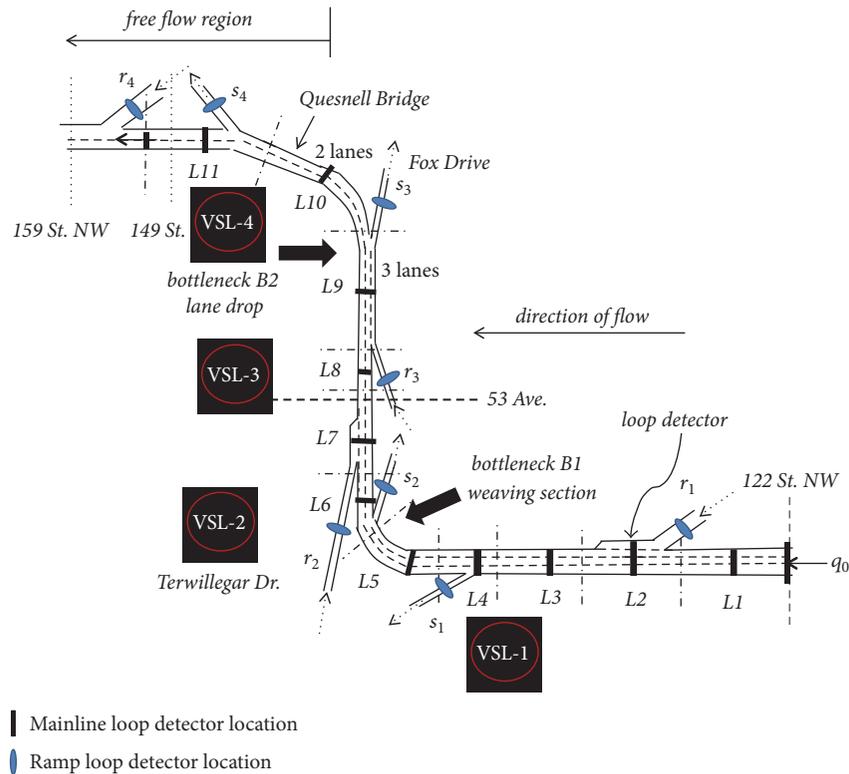


FIGURE 4: Layout of the experimental site.

as the warm-up period and excluded from the comparisons; links 1 and 11 were outside of the VSL-controlled area, so they were excluded from the comparison as well. Figure 5 and Table 2 illustrate the detailed comparison results.

First of all, by examining the average CP across the entire study area, it can be concluded that the implemented VSL control has an overall positive impact on the experiment corridor. The average CP in the VSL-controlled scenario was reduced by 32%, relatively (from 19% to 13%). Table 2 further splits the experiment duration into five 30-minute segments. The comparison results show that, for the first hour, there was no major difference between the two tested scenarios. The reason is, for that period, that there was no heavy traffic demand in the network and the safety performance of the corridor was fine (CP less than 10%). Thus, the control algorithm was not active to restrict the traffic flow. Nevertheless, during the congestion period (8:00–9:00 AM) where the CP went over 25% due to the oversaturated demand and congested traffic flow (high density), the corridor CP level was effectively controlled by implementing the proposed VSL system. The overall CP of the corridor was significantly reduced by over 35%, relatively.

Secondly, for individual link evaluation results, all the examined links show a positive result by reporting effectively reduced crash potential. Among the tested links, the most significant CP reduction was observed at link 6 (49%, relatively). Link 6 is one of the major identified bottlenecks in the evaluated corridor. Thus, here the authors take link 6 as an example to further demonstrate how the implemented VSL control works.

Figure 6 illustrates the change in CP for link 6 with and without VSL control. The demonstrated link suffers serious congestion without VSL (pale dash line). The speed profile drops to 20 km/h after the peak demand hits the bottleneck around 7:30 AM and does not recover until the end of the experiment (approximately 8:40 AM). On the other hand, relying on the proposed macroscopic traffic flow model and driver response model, the VSL control algorithm successfully predicted the oversaturated demand and forthcoming traffic breakdown in advance. As a response, the control algorithm lowered the speed limit gradually before the traffic breakdown at approximately 7:15 AM. This scheme helped the traffic stream to slow down step by step, which brought down the crash risk caused by the abrupt traffic breakdown. Furthermore, the control strategy effectively mitigated the scale of the congestion (flow speed recovered at about 8:00 AM instead of after 8:40 AM), which also lowered the CP by reducing the length of the high density region. The observed CP reduction is the joint effort of corridor-wide coordinated multilink control, and it is consistent with the corridor-level CP reduction observed in Table 2.

Another interesting fact is the experiment reflects the dynamic response mechanism of the drivers to VSL: at the very outset, when the control algorithm lowered the speed limit before traffic breakdown (7:10 to 7:25 AM), the drivers did not follow the advised VSL considerably. Some even disobeyed the advised speed limit. However, later on when traffic was more congested (7:30 to 8:00 AM), the drivers appeared to follow the VSL more strictly or even drive lower than the advised value. This observation proves that the

TABLE 2: Comparison of crash probability with and without proposed VSL control.

Crash probability (%)	6:30–7:00 AM	7:00–7:30 AM	7:30–8:00 AM	8:00–8:30 AM	8:30–9:00 AM	Overall
Uncontrolled	0.6	8.6	27.4	29.0	25.2	19.8
VSL control	0.6	8.3	23.1	18.6	13.3	13.4
CP reduction (absolute)	0	0.3	4.3	10.4	11.9	6.4
CP reduction (relative)	0.0	3.4	15.7	35.7	47.1	32.3

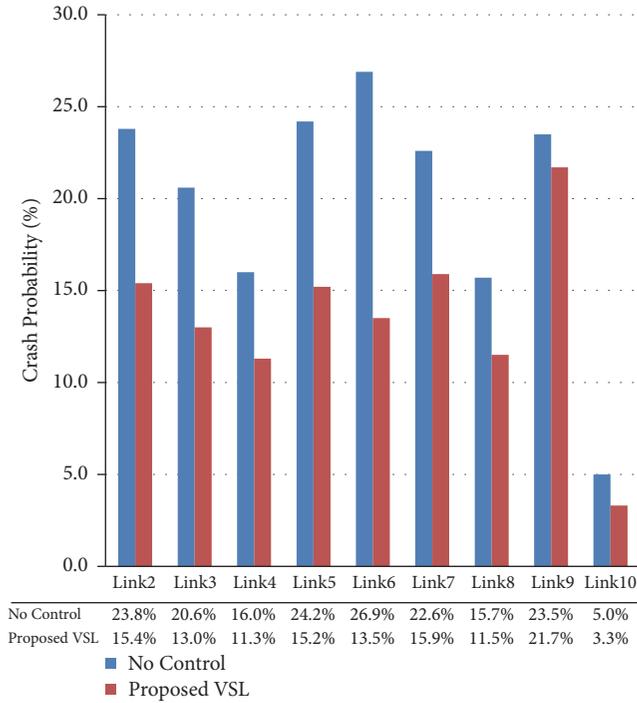


FIGURE 5: Comparison of crash probability profile with and without proposed VSL control.

conducted experiment and proposed control algorithm capture the nature of the dynamic response mechanism and provide a more realistic control practice. The authors give credit to the field data calibrated driver response model and simulations.

5. Conclusions and Future Work

In this research study, the authors proposed a dynamic VSL driver response modeling approach. Taking the current VSL value and traffic state variables as the independent variables, the model formulates the critical factor in safety oriented VSL control: the drivers’ sophisticated response to various advised VSLs. The analysis of field data revealed how the current flow speed, density, and VSL value were associated with the desired speed. After calibration, the established model showed a close match to the field observation in the model validation process.

Furthermore, the proposed model was adopted to incorporate dynamic driver response mechanism modeling into a prediction-based optimal VSL control algorithm. For model evaluation, this control algorithm was used to implement VSL control over a congested highway corridor. The experiment

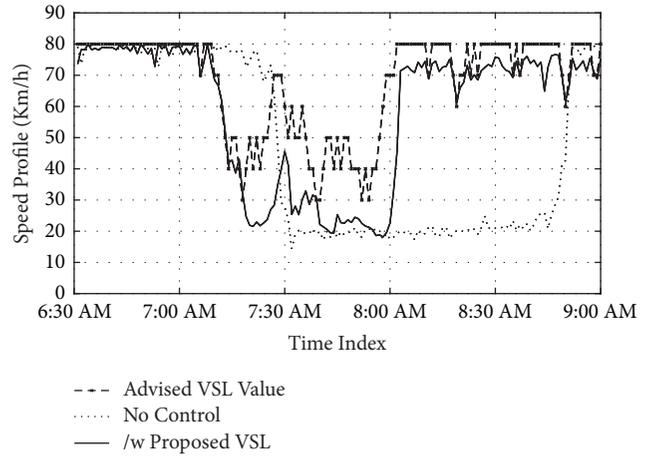


FIGURE 6: Speed profile of link 6 during experiment.

result demonstrated that the implemented VSL control has an overall positive impact on the experiment corridor. The average CP in the VSL-controlled scenario was reduced by 32%, relatively. By the exertion of the proposed macroscopic traffic flow model and driver response model, the VSL control algorithm successfully predicted the oversaturated demand and forthcoming traffic breakdown in advance. Thus, the control algorithm was able to reduce the probability of crashes, by (1) forcing the traffic flow to slow down gradually beforehand and (2) mitigating the scale of the congestion period. Moreover, the conducted experiment and proposed control algorithm were able to capture the nature of the dynamic driver response mechanism and produce a more realistic control practice.

For future studies, the authors plan to evaluate the proposed VSL control algorithm in real-world applications. The WMD corridor evaluated in this paper has been equipped with proposed VSL signs and the dynamic control system. The authors look forward to reporting the field experiment results.

Conflicts of Interest

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

Acknowledgments

This work was jointly supported by the Fujian Science and Technology Department and Fujian Provincial Education

Department, which provided support for the projects entitled “Traffic Flow Prediction and Evaluation Method of Urban Expressway Under Intelligent Traffic Control Environment” (Award no. 2015J05100) and “Multi-Objective Online Comprehensive Evaluation Method of Network Operation State” (Award no. JA15081), respectively.

References

- [1] M. Abdel-Aty, J. Dilmore, and L. Hsia, “Applying variable speed limits and the potential for crash migration,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1953, pp. 21–30, 2006.
- [2] P. Rämä, “Effects of weather-controlled variable speed limits and warning signs on driver behavior,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1689, pp. 53–59, 1999.
- [3] J. Fang, M. Hadiuzzaman, A. Karim, Y. Luo, and T. Z. Qiu, “Variable speed limit control strategy with collision probability assessments based on traffic state prediction,” *Transportation Research Record*, vol. 2435, pp. 11–18, 2014.
- [4] M. Abdel-Aty, R. J. Cunningham, V. V. Gayah, and L. Hsia, “Dynamic variable speed limit strategies for real-time crash risk reduction on freeways,” *Transportation Research Record*, vol. 2078, no. 1, pp. 108–116, 2008.
- [5] M. Tazul Islam, M. Hadiuzzaman, J. Fang, T. Qiu, and K. El-Basyouny, “Assessing mobility and safety impacts of a variable speed limit control strategy,” *Transportation Research Record*, no. 2364, pp. 1–11, 2013.
- [6] B. Hellinga and M. Mandelzys, “Impact of driver compliance on the safety and operational impacts of freeway variable speed limit systems,” *Journal of Transportation Engineering*, vol. 137, no. 4, pp. 260–268, 2011.
- [7] M. Hadiuzzaman, J. Fang, M. A. Karim, Y. Luo, and T. Z. Qiu, “Modeling driver compliance to VSL and quantifying impacts of compliance levels and control strategy on mobility and safety,” *Journal of Transportation Engineering*, vol. 141, no. 12, Article ID 04015028, 2015.
- [8] A. Hegyi, M. Burger, B. De Schutter, J. Hellendoorn, and T. J. J. Van Den Boom, “Towards a practical application of model predictive control to suppress shock waves on freeways,” in *Proceedings of the 2007 9th European Control Conference, ECC 2007*, pp. 1764–1771, July 2007.
- [9] E. van den Hoogen and S. Smulders, “Control by variable speed signs: results of the Dutch experiment,” in *Proceedings of the Seventh International Conference on Road Traffic Monitoring and Control*, pp. 145–149, London, UK, 1994.
- [10] UK Highways Agency, *M25 Controlled Motorway Summary Report*, Highways Agency Publications Group, Bristol, UK, 2006.
- [11] Washington State DOT, “I-90—Two-way transit and HOV operations—Variable speed limit signs,” 2009, <http://www.wsdot.wa.gov/Projects/I90/TwoWayTransit/vsl.html>.
- [12] G.-L. Chang, S. Y. Park, and J. Paracha, “Intelligent transportation system field demonstration: integration of variable speed limit control and travel time estimation for a recurrently congested highway,” *Transportation Research Record*, no. 2243, pp. 55–66, 2011.
- [13] A. Hegyi and S. P. Hoogendoorn, “Dynamic speed limit control to resolve shock waves on freeways - Field test results of the SPECIALIST algorithm,” in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, ITSC 2010*, pp. 519–524, Funchal, Portugal, September 2010.
- [14] A. Hegyi, *Model Predictive Control for Integrating Traffic Control Measures*, Netherlands TRAIL Research School, 2004.
- [15] R. C. Carlson, I. Papamichail, M. Papageorgiou, and A. Messmer, “Optimal mainstream traffic flow control of large-scale motorway networks,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 2, pp. 193–212, 2010.
- [16] X. Yang, Y. Lu, and G.-L. Chang, “Exploratory analysis of an optimal variable speed control system for a recurrently congested freeway bottleneck,” *Journal of Advanced Transportation*, vol. 49, no. 2, pp. 195–209, 2015.
- [17] X. Yang and Y. C. Lu, “Development of optimal variable speed limit control system for freeway work zone operations,” *Transportation Research Board Meeting*, vol. 94, pp. 149–160, 2014.
- [18] X.-Y. Lu, P. Varaiya, R. Horowitz, D. Su, and S. E. Shladover, “Novel freeway traffic control with variable speed limit and coordinated ramp metering,” *Transportation Research Record*, no. 2229, pp. 55–65, 2011.
- [19] D. G. Hidalgo and M. Fontaine, “Impact of a variable speed limit system on driver speeds during low-visibility conditions,” in *Proceedings of the 97th Transportation Research Board Annual Meeting*, (No. 18-03222), Washington, DC, USA, 2018.
- [20] J. Fang, E. Hadiuzzaman, T. Qiu, M. Hadiuzzaman, and E. Yin, “DynaTAM: an online algorithm for performing simultaneously optimized proactive traffic control for freeways,” *Canadian Journal of Civil Engineering*, vol. 41, no. 4, pp. 315–322, 2014.
- [21] J. Fang, L. Tu, and S. M. Easa, “Ramp metering control using multi-objective optimization: mobility and safety,” in *Proceedings of the 97th Transportation Research Board Annual Meeting*, (No. 18-01958), Washington, DC, USA, 2018.

Research Article

Statistical Process Monitoring with Biogeography-Based Optimization Independent Component Analysis

Xiangshun Li , Di Wei, Cheng Lei, Zhiang Li, and Wenlin Wang

School of Automation, Wuhan University of Technology, Wuhan, China

Correspondence should be addressed to Xiangshun Li; lixiangshun@whut.edu.cn

Received 26 December 2017; Revised 14 March 2018; Accepted 26 March 2018; Published 30 April 2018

Academic Editor: Sarah Spurgeon

Copyright © 2018 Xiangshun Li 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.

Independent Component Analysis (ICA), a type of typical data-driven fault detection techniques, has been widely applied for monitoring industrial processes. FastICA is a classical algorithm of ICA, which extracts independent components by using the Newton iteration method. However, the choice of the initial iterative point of Newton iteration method is difficult; sometimes, selection of different initial iterative points tends to show completely different effects for fault detection. So far, there is still no good strategy to get an ideal initial iterative point for ICA. To solve this problem, a modified ICA algorithm based on biogeography-based optimization (BBO) called BBO-ICA is proposed for the purpose of multivariate statistical process monitoring. The Newton iteration method is replaced with BBO here for extracting independent components. BBO is a novel and effective optimization method to search extremes or maximums. Comparing with the traditional intelligent optimization algorithm of particle swarm optimization (PSO) and so on, BBO behaves with stronger capability and accuracy of searching for solution space. Moreover, numerical simulations are finished with the platform of DAMADICS. Results demonstrate the practicability and effectiveness of BBO-ICA. The proposed BBO-ICA shows better performance of process monitoring than FastICA and PSO-ICA for DAMADICS.

1. Introduction

Industrial processes play an extremely significant role in the development of economy. Once any part of the process breaks down, the production efficiency will be greatly influenced, and more seriously disasters may be caused. Therefore, it is important for plants to detect and diagnose the fault as early as possible. Then, proper measures would be taken for minimising economic loss.

Process monitoring technologies can be divided into three types: model-based, knowledge-based, and data-driven method [1]. More detailed classifications are shown in the Figure 1.

Model-based method is a type of process monitoring technology, which is established on mathematical models of industrial process systems. To some extent, the efficiency of the process monitoring is largely dependent on the accuracy of mathematical models. Nevertheless, it is difficult to get accurate models from complex industry processes. In

addition, it is troublesome to establish different models for different industrial processes.

The knowledge-based method requires large amount of empirical knowledge. Machines can learn from these knowledge and make their own judgements and decisions on the basis of the actual situations. For a complex industrial process, it is impossible to consider all the possibilities and hard to get some knowledge of industrial processes.

Data-driven method can determine whether there is a fault according to the operation data obtained by large amount of sensors. With the arrival of the era of big data, a large number of data of running information of industrial processes have been stored in computers; however, these data have not been fully utilized. So, mining useful information from a large amount of data has become very important. Data-driven method of process monitoring without needing models and prior knowledge has attracted more and more attention in the field of process monitoring.

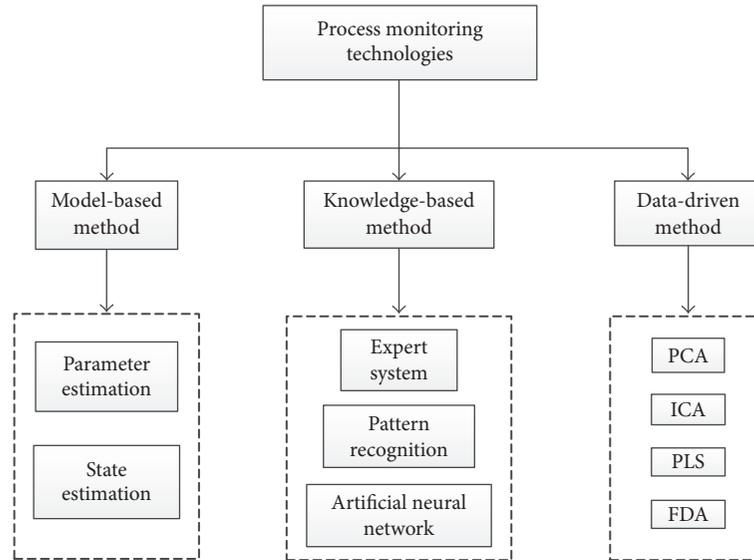


FIGURE 1: Categories of process monitoring technologies.

ICA, also as a method of data-driven monitoring techniques, has been widely applied into several fields [2] involving image processing, signal extraction [3], and medical science [4, 5] including electroencephalograph (EEG) [6] and neuroscience [7]. ICA is originally proposed for Blind Source Separation (BSS) [8–10] and the mathematical description of ICA is defined [11]. ICA is also introduced for process monitoring [12]. The classical FastICA algorithm based on Newton iteration method is employed to extract independent components [13]. More recently, several process monitoring methods based on FastICA have been proposed. Kernel Independent Component Analysis (KICA) [14] behaves with better performance in detecting faults of nonlinear characteristic of process systems. Considering the autocorrelation characteristics of process variables in time series, Dynamic Independent Component Analysis (DICA) [15] is utilized to eliminate the sensitivity to dynamic feature of processes. Canonical Variate Analysis (CVA) algorithm is combined with ICA [16] for data preprocessing, which improves the efficiency of fault detection. Fault classifier based on Support Vector Machines (SVM) and ICA [17] is used to reveal the hidden factors from massive process data. There is no mathematical expression of noise in the traditional ICA. Aiming at this point, a Noisy-ICA method [18] is proposed to fix the classical ICA model. Dynamic Bayesian Independent Component Analysis [19] is proposed to detect faults of multimode process. Probability is introduced to improve performance of ICA [20]. Jarque-Bera test is used to drive the observed data into different data blocks [21]; then DICA and DPCA are used to handle different data blocks.

Although the ICA algorithm has been improved by many scholars, the influence of the initial iteration point on the performance of the algorithm is neglected. It is worth noting that the effect of the FastICA algorithm largely depends on the choice of the initial iterative point. Different initial

iterative points would cause completely different convergence properties. Mostly, it is difficult to obtain an ideal initial iterative point and it is inconvenient and unadvisable to obtain the initial iterative point by trying many times.

When the FastICA algorithm is applied to process monitoring, different initial iterative points would lead to opposite result of diagnosis. This is unacceptable in practice. Taking the safety system of aircraft as an example, the alarm of the safety system means that there is a fault in the aircraft. If the result of diagnosis is unreliable, it would result in the panic of the passengers when there is a false alarm. Furthermore, the safety of passengers is not guaranteed. Thus, to some extent, the result of diagnosis of FastICA algorithm is unreliable because of sensitivity of initial iterative point.

Biogeography-based optimization (BBO) is a heuristic optimization method based on the collective behavior simulation of biological population, which is firstly proposed and compared with seven other traditional intelligent optimization algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimization Algorithm (PSO), and Ant Colony Algorithm (ACO) [22]. The performance of various algorithms on 14 standard function benchmarks show that BBO is a more efficient method to solve optimization problems.

Though PSO-ICA has been proposed [23] to separate independent components, it cannot separate independent components accurately, which would behave with poor performance in the process monitoring.

A method named BBO-ICA is proposed in this paper to monitor processes by replacing the Newton iteration method for independent components extraction by BBO and achieve more accurate results. A comparison among FastICA, BBO-ICA, and PSO-ICA in the performance of process monitoring is made in DAMADICS benchmark [24]. This paper is organized as follows: Section 2 gives the basic procedure of process monitoring based on ICA.

Then, BBO-ICA method is proposed in Section 3. A simulation study on the DAMADICS benchmark is presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Process Monitoring Based on ICA

2.1. ICA Algorithm. Independent Component Analysis (ICA) is a statistical and computational technique, which can reveal hidden factors from data of signals, measured variables, and so on. ICA was originally proposed to solve the Cocktail Party Problem [25], which can separate noises, different voices, and background music signals from measured mixed signals.

It is assumed that there are m sensors and each of the sensors measures one of the process parameters. At each sampling interval, m measured variables $\mathbf{x}(k) = [x_1, x_2, \dots, x_m]^T$ are obtained. Measured signals are often mixed with noise and multiple source signals. Suppose that m measured variables can be expressed as linear combinations of l ($l \leq m$) unknown independent components s_1, s_2, \dots, s_l . The mathematical relationship between them can be expressed by

$$\mathbf{x} = \sum_{j=1}^l a_j s_j(k) = \mathbf{A} \mathbf{s}(k). \quad (1)$$

Here, $\mathbf{A} \in \mathcal{R}^{m \times l}$ is a mixing matrix, which is unknown and full-rank. It is usually assumed that l is equal to m and samples number is n ; the expression can be rewritten as

$$\begin{aligned} \mathbf{X} &= \mathbf{A} \mathbf{S} \quad (\mathbf{A} \in \mathcal{R}^{m \times n}) \\ \mathbf{X} &= [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]. \end{aligned} \quad (2)$$

Usually, the measured data \mathbf{X} need to be whitened before applying the ICA algorithm, which can eliminate the cross-correlation among measured variables. The whitened variables matrix \mathbf{Z} can be obtained by

$$\begin{aligned} \mathbf{Z} &= \mathbf{Q} \mathbf{X} = \mathbf{Q} \mathbf{A} \mathbf{S} = \mathbf{B} \mathbf{S} \\ \mathbf{B} &\stackrel{\text{def}}{=} \mathbf{Q} \mathbf{A}, \end{aligned} \quad (3)$$

where $\mathbf{Q} = \Lambda^{-1/2} \mathbf{U}^T$ and Λ and \mathbf{U} can be acquired by eigendecomposition of the covariance matrix \mathbf{R}_x , which is given by

$$\mathbf{R}_x = E(\mathbf{X} \mathbf{X}^T) = \mathbf{U} \Lambda \mathbf{U}^T, \quad (4)$$

where E represents expectations. After whitening transformation, it is easily verified that the covariance matrix $\mathbf{R}_z = E(\mathbf{Z} \mathbf{Z}^T)$ is an identity matrix and \mathbf{B} is an orthogonal matrix as verified by the following relation:

$$E(\mathbf{Z} \mathbf{Z}^T) = \mathbf{B} E(\mathbf{S} \mathbf{S}^T) \mathbf{B}^T = \mathbf{B} \mathbf{B}^T = \mathbf{I}. \quad (5)$$

Thus, (3) can also be transformed into the form of

$$\begin{aligned} \mathbf{B}^T \mathbf{Z} &= \mathbf{B}^T \mathbf{B} \mathbf{S} \implies \\ \mathbf{S} &= \mathbf{B}^T \mathbf{Z}. \end{aligned} \quad (6)$$

The object of ICA is to estimate the independent components \mathbf{S} or the mixing matrix \mathbf{A} from the measured data matrix \mathbf{X} without any knowledge of \mathbf{S} or \mathbf{A} . It is equivalent to find a demixing matrix \mathbf{W} which can reconstruct the independent components $\hat{\mathbf{S}}$ from the measured data matrix \mathbf{X} . The reconstructed system of ICA is illustrated in Figure 2.

$$\hat{\mathbf{S}} = \mathbf{W} \mathbf{X}. \quad (7)$$

Considering (3), (6), and (7), we can derive

$$\mathbf{W} = \mathbf{B}^T \mathbf{Q}. \quad (8)$$

The principle of estimating \mathbf{B} is to make $\hat{\mathbf{S}}$ independent of each other as possible. To calculate \mathbf{B} , each row vector \mathbf{b}_i of \mathbf{B} is randomly initialized and then updated so that the i th independent component $\hat{s}_i = \mathbf{b}_i^T \mathbf{Z}$ has maximum independence.

$$\begin{aligned} \mathbf{B} &= [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \dots \quad \mathbf{b}_m] \\ \hat{\mathbf{S}} &= \mathbf{B} \mathbf{Z} = \mathbf{W} \mathbf{X} = [\hat{s}_1 \quad \hat{s}_2 \quad \dots \quad \hat{s}_m]^T. \end{aligned} \quad (9)$$

Independence can be reflected by the non-Gaussianity. Negentropy is a common measure of non-Gaussianity, which is based on the information theoretic quantity of differential entropy. A flexible and reliable approximation of negentropy is as follows:

$$J(y) = [E(G(y)) - E(G(v))]^2, \quad (10)$$

where y is assumed to be of zero mean and unit variance, v is a Gaussian variable of zero mean and unit variance, and G is a nonquadratic function. By choosing G properly, one obtains good approximations of negentropy. Hyvärinen and Oja suggested three types of functions for G [3]:

$$\begin{aligned} G_1(u) &= \frac{1}{a_1} \log \cosh(a_1 u) \\ G_2(u) &= \exp\left(-\frac{a_2 u^2}{2}\right) \\ G_3(u) &= u^4, \end{aligned} \quad (11)$$

where $1 \leq a_1 \leq 2$, $a_2 = 1$. Among these three functions, G_1 is a good general-purpose contrast function and is therefore selected in this paper. So, in order to find the maximum independence of i th independent component, the problem can be simplified as

$$\mathbf{J}(\mathbf{b}_i) = E(G(\mathbf{b}_i^T \mathbf{Z})) \quad (12)$$

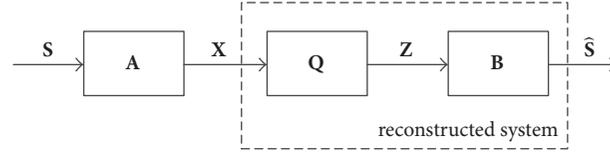


FIGURE 2: Reconstructed system of ICA.

satisfying

$$\mathbf{E} \left\{ (\mathbf{b}_i^T \mathbf{Z})^2 \right\} = 1. \quad (13)$$

The constraint can be transformed into

$$\begin{aligned} \mathbf{E} \left\{ (\mathbf{b}_i^T \mathbf{Z})^2 \right\} &= \mathbf{E} \left\{ (\mathbf{b}_i^T \mathbf{Z}) (\mathbf{b}_i^T \mathbf{Z})^T \right\} = \mathbf{E} \left\{ \mathbf{b}_i^T \mathbf{Z} \mathbf{Z}^T \mathbf{b}_i \right\} \\ &= \mathbf{b}_i^T \mathbf{E} (\mathbf{Z} \mathbf{Z}^T) \mathbf{b}_i = \mathbf{b}_i^T \mathbf{b}_i = \|\mathbf{b}_i\|^2 = 1. \end{aligned} \quad (14)$$

2.2. Process Monitoring with ICA. After solving this optimization problem, we get m independent components $\hat{\mathbf{S}}$. But we do not need to use all the independent components for process monitoring, because the independent components may contain noise and some redundant information. A standard is proposed to choose main independent components, which sorts m independent components by the Euclidean norm of each \mathbf{b}_i and selects d several dominant independent components $\hat{\mathbf{S}}_d = \mathbf{B}_d^T \mathbf{Z}$.

Two types of statistics are calculated from the process model in normal operation: the \mathbf{I}^2 -statistic for the systematic part of the process variation and the **SPE**-statistic for the residual part of the process variation, which is defined as follows:

$$\begin{aligned} \mathbf{I}^2 &= \hat{\mathbf{S}}_d^T \hat{\mathbf{S}}_d \\ \mathbf{SPE} &= \mathbf{e}^T \mathbf{e} = (\mathbf{X} - \hat{\mathbf{X}})^T (\mathbf{X} - \hat{\mathbf{X}}) \\ \hat{\mathbf{X}} &= \mathbf{Q}^{-1} \mathbf{B}_d^T \hat{\mathbf{S}} = \mathbf{Q}^{-1} \mathbf{B}_d^T \mathbf{W}_d \mathbf{X}. \end{aligned} \quad (15)$$

A limit value is also needed to determine whether the process is in control or not. Note that independent components seldom follow the Gaussian distribution. Hence, the confidence limit of \mathbf{I}^2 and **SPE** statistics cannot be determined directly from a particular approximate distribution.

Kernel Density Estimation is utilized to determine the confidence limit, which can estimate the density function of the normal \mathbf{I}^2 and **SPE** statistics. A univariate kernel estimator is given by

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K \left\{ \frac{x - x_i}{h} \right\}, \quad (16)$$

where x is the value under consideration, x_i is an observation point, h is a smoothing parameter, n is the number of observations, and $K\{\ast\}$ is a kernel function. The point, which is upper quantile of confidence level of 95%, can be obtained as the control limit of normal operating data. The flowchart of process monitoring with ICA is illustrated in Figure 3.

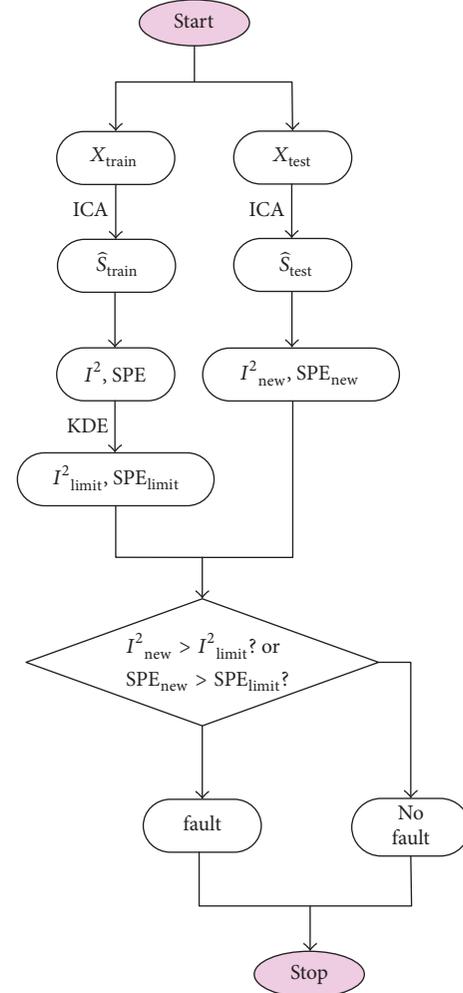


FIGURE 3: Process monitoring with ICA.

3. BBO-ICA

3.1. Biogeography. The science of biogeography can date back to nineteenth century. Naturalists, such as Darwin and Beer [26] did pioneering research on it. In 1960s, MacArthur and Wilson began [27] to focus their attention on the distribution of species among neighboring islands and were dedicated to establishing mathematical models for the extinction and migration of species, which greatly improve the development of the subject of biogeography. It was not until 2008 that this method was introduced to solve engineering problems by Simon [22].

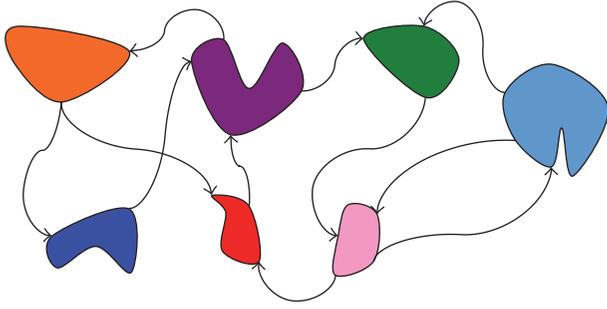


FIGURE 4: Species migrate between habitats.

In nature, thousands of species are distributed in different habitats all over the world and different habitats contain different numbers of species. Geographical areas that are well suited as residences for biological species tend to have a high habitat suitability index (HSI) [28]. Indices to measure suitability of habitats are concluded as follows: temperature, water resource, land area, vegetation, and topographic features. These factors that reflected the habitability are called suitability index variables (SIV). So, SIV and HSI can be considered as independent variables and dependent variable, respectively.

Habitat with a high HSI tends to have a large number of species. Because of saturating with species, habitat with a high HSI has high possibility of emigrating to neighboring habitats. Habitat with a low HSI would have a small number of species, which means that species have a high species immigration rate thanks to sparse populations. The immigration of new species to low HSI habitats may raise the HSI of the habitat, owing to the improvement of the biological diversity. If the habitat's HSI remains low, the species will go extinct. Therefore, low HSI habitats are more dynamic and high HSI habitats are more static in species distribution. The process of migration of species between habitats is indicated in Figure 4.

Biogeography reveals the nature's rules of distributing species, which can also be used to solve general problems. A good solution is analogous to a habitat with a high HSI, while a poor solution represents a habitat with a low HSI.

Figure 5 illustrates that the immigration rate λ and the emigration rate μ vary with number of species linearly. At the equilibrium point S_0 , the immigration rate is equal to the emigration rate. The balance would be disturbed because of a sudden spurt of immigration from neighboring habitats or a sudden burst of disease. In nature, it would spend a long time for numbers of species to regain equilibrium [29, 30].

The mathematical relationship between number of species and migration rate can be also described in the following formula:

$$\begin{aligned} \lambda_s &= I_{\max} \left(1 - \frac{s}{n}\right) \\ \mu_s &= E_{\max} \frac{s}{n}, \end{aligned} \quad (17)$$

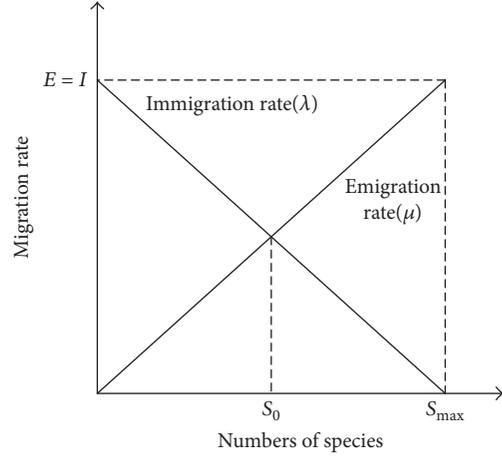


FIGURE 5: Relation between number of species and migration rate.

where λ_s and μ_s denote immigration rate and emigration rate of s species, I_{\max} and E_{\max} represent the max immigration rate and the max emigration rate, here, I_{\max} is considered equal to E_{\max} , and n is equal to the max number of species. Considering (17), we can derive

$$\lambda_s + \mu_s = E_{\max} = I_{\max}. \quad (18)$$

Now, $P_S(t)$ is introduced to represent the probability that a habitat contains s species at any time t . At time $t + \Delta t$, the probability that the number of species remains unchanged in the habitat is

$$\begin{aligned} P_S(t + \Delta t) &= P_S(t) (1 - \lambda_s \Delta t - \mu_s \Delta t) \\ &+ P_{S-1}(t) \lambda_{s-1} \Delta t + P_{S+1}(t) \mu_{s+1} \Delta t. \end{aligned} \quad (19)$$

This equation must hold for the following conditions:

- (1) If there were s species in a habitat at time t , no immigration and emigration of species would happen between t and $t + \Delta t$.
- (2) If there were $s - 1$ species in a habitat at time t , only one species would immigrate between t and $t + \Delta t$.
- (3) If there were $s + 1$ species in a habitat at time t , only one species would emigrate between t and $t + \Delta t$.
- (4) The probability of more than one immigration or emigration can be ignored.

When $\Delta t \rightarrow 0$, (19) can be transformed into the following formula:

$$\begin{aligned} \dot{P}_S(t) &= \lim_{\Delta t \rightarrow 0} \frac{P_S(t + \Delta t) - P_S(t)}{\Delta t} \\ &= -P_S(t) (\lambda_s + \mu_s) + P_{S-1}(t) \lambda_{s-1} \\ &+ P_{S+1}(t) \mu_{s+1}. \end{aligned} \quad (20)$$

3.2. Biogeography-Based Optimization. Biogeography-based optimization algorithm has been applied to solve many engineering problems such as different economic load dispatch

problems [31], parameter optimization of neural network [32], smart energy management [33], pathological brain detection [34], and optimal VAR control in the circuit [35].

Biogeography-based optimization algorithm involves two important steps: migration and mutation, which are closely related to the theory of biogeography. The procedure of emigration and immigration of each solution is used to share information among habitats probabilistically. Mutation is another significant scheme, which can increase diversity among habitats. However, best solution is easily corrupted by migration and mutation; thus, some numbers of elites are kept until next generation. Figure 6 is a general flowchart of biogeography-based optimization.

Take the following function as an example:

$$f(x) = \sum_{i=1}^m x_i, \quad -1 \leq x_i \leq 1. \quad (21)$$

BBO algorithm is described as follows:

(1) Initialize number of habitats, SIV, elite and iteration, probability of habitat migration, maximum rate of mutation, immigration, and emigration, denoted, respectively, as N , m , p , G_{\max} , P_{mod} , m_{\max} , I , and E .

(2) Initialize N solutions of the function randomly within feasible region; then, N initial habitats are obtained; the whole habitat can be written in the form of matrix:

$$H = [H_1, H_2, \dots, H_N]^T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nm} \end{bmatrix} \quad (22)$$

$\in R^{N \times m}$.

(3) HSI value for i th habitat can be calculated by

$$\text{HSI}_i = x_{i1} + x_{i2} + \cdots + x_{im} \quad (i = 1, 2, \dots, N). \quad (23)$$

(4) Sort N habitats according to HSI value; p elite solutions are kept until next generation. Initialize count number $i = j = k = e = 1$.

(5) Select a habitat H_j according to the immigration rate of i th habitat λ_i ; if $i < N$, go to step (6), $i = i + 1$; else go to step (9). The immigration rate of i th habitat λ_i is rewritten as

$$\lambda_i = I_{\max} \left(1 - \frac{k_i}{N} \right), \quad (24)$$

where k_i stands for fitness rank of i th island after sorting of HSI value.

(6) Generate a random number between 0 and 1, denoted as $\text{num_rand } 1$. If $\text{num_rand } 1 < \lambda_i$, then go to step (7); else go to step (5).

(7) Select a habitat H_j according to the emigration rate of j th habitat, which can be rewritten as

$$\mu_j = E_{\max} \frac{k_j}{N}. \quad (25)$$

(8) Replace the selected H_i 's SIV with randomly selected H_j 's SIV; if $j < N$, $j = j + 1$, go to step (7); else go to step (5).

(9) Compute the mutation probability of k th habitat, represented as m_k , $k = k + 1$.

(10) Generate a random number between 0 and 1, $\text{num_rand } 2$. If $\text{num_rand } 2 < m_k$, replace SIV of H_k with a randomly generated SIV within feasible region.

(11) Sort N habitats again according to HSI value; keep p elite solutions until next iteration.

(12) If $e < G_{\max}$, then go to step (5); else go to step (13).

(13) Output the best solution of the general problem.

3.3. BBO-ICA Algorithm. Classical FastICA has a difficult problem of acquiring an ideal initial iterative point, which has a great influence on the performance of the FastICA algorithm. Thus, in order to avoid this problem, BBO is a substitute for Newton iteration method to find independent components. BBO is an optimization method for finding optimum solution, which means that BBO-ICA can obtain only one independent component of maximum independence.

Nevertheless, ICA need to estimate several independent components. A strategy described as follows can solve this problem.

It is considered that an independent component \mathbf{s}_i is related to some information of \mathbf{X} . So, we can update the observation matrix \mathbf{X} by subtracting the information which independent components are corresponding to. It can effectively prevent the algorithm converging to the same \mathbf{b}_i . The formula can be expressed as

$$\begin{aligned} \mathbf{X} &= \mathbf{X} - \mathbf{Q}^{-1} \mathbf{b}_i \mathbf{s}_i \\ \mathbf{s}_i &= \mathbf{b}_i^T \mathbf{Z}. \end{aligned} \quad (26)$$

ICA algorithm based on biogeography-based optimization can be described as shown in Algorithm 1.

4. Simulations

4.1. Simple Example. In this section, Newton iteration method is used to solve the solutions of a simple explicit function, which is

$$f(x) = xe^x - 1. \quad (27)$$

The function is plotted in Figure 7.

The purpose of this example is to verify the influence of selection of different initial iterative points on the Newton Iterative Method. Newton iteration method is originally used to search roots, which can also get extremums by solving

$$f'(x) = 0. \quad (28)$$

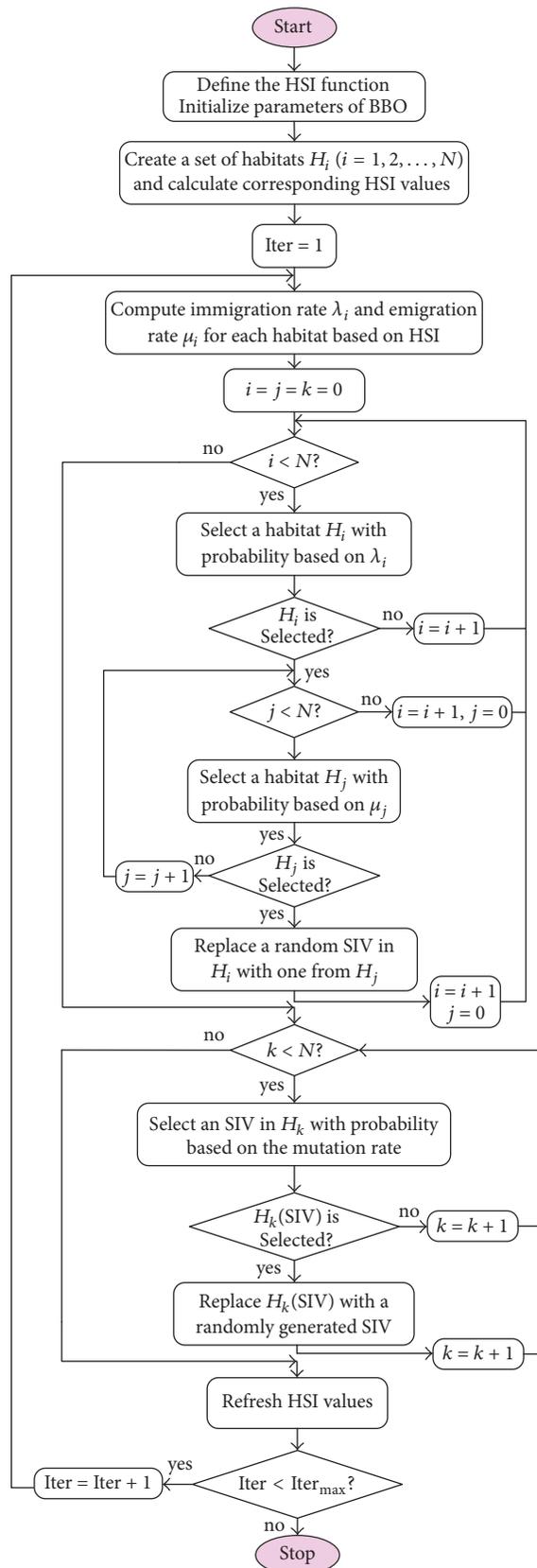


FIGURE 6: Flowchart of biogeography-based optimization.

TABLE 1: The convergence of different initial iterative points of Newton iteration method.

x_0	Convergence	Iteration times
1	0.5671	5
5	0.5671	11
-2	Nonconvergence	1000

- (1) Choose m independent components to estimate
- (2) Set counter $i = 1$
- (3) Center the mean of each variables to zero
- (4) Unitize the variance of each variables to one
- (5) Perform whitening transformation to \mathbf{X} and get \mathbf{Z}
- (6) Initialize parameters of BBO algorithm
- (7) Employ BBO to extract independent component of maximum independence and get \mathbf{bi}
- (8) Output vector \mathbf{bi} , update the observation matrix \mathbf{X}
- (9) Go back to step (5), if $i \leq m, i = i + 1$

ALGORITHM 1: Process monitoring using BBO-ICA.

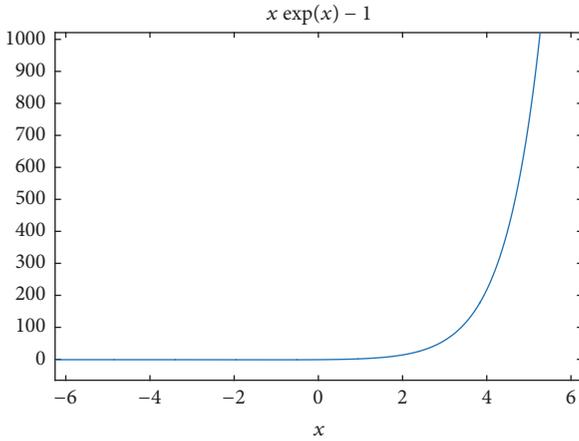


FIGURE 7: Graph of function.

This simple example is set with three different initial iterative points which are $x_0 = -2, 1,$ and $5,$ respectively. The maximum iteration is 1000 and the iteration accuracy is $1e - 6.$ The process of iteration and the convergence of different initial iterative points are illustrated in Figure 8 and Table 1.

In contrast, to find roots of $f(x)$ using Newton iteration method, we utilize BBO to find extremums of $F(x).$

$$F(x) = (x - 1)e^x - x. \quad (29)$$

It is easy to know that

$$F'(x) = f(x). \quad (30)$$

Parameters of BBO of $N, m, p, G_{\max}, P_{\text{mod}}, m_{\max}, I,$ and E are set as 50, 1, 2, 100, 1, 0.6, 1, and 1, respectively.

The process of iteration of BBO can be seen in Figure 9. It can be concluded from the Figure 8 and Table 1 that the effect of the Newton iteration method is largely dependent on the choice of the initial iterative point. Different initial

iterative points tend to result in diverse convergence rates. More surprisingly, some initial iterative points would lead to nonconvergence.

Nevertheless, the algorithm of BBO can avoid this problem because of initializing of some populations randomly. It can search the extremums of $F(x)$ accurately within 22 times of iteration. BBO is able to obtain global optimal solution in the solution space, while sacrificing the expense of operation time of program.

4.2. DAMADICS Process. In this section, the monitoring performance of the BBO-ICA method is compared with that of conventional FastICA by the DAMADICS process simulation data. Two generally used indices, fault detection rate (FDR) and false alarm rate (FAR), are mainly considered here for evaluating monitoring performance.

FDR

$$= \frac{\text{Max No. of } (I^2 > I_{\text{th}} \ \& \ \text{SPE} > \text{SPE}_{\text{th}} \ | \ f \neq 0)}{\text{total samples } (f \neq 0)} \quad (31)$$

FAR

$$= \frac{\text{Max No. of } (I^2 > I_{\text{th}} \ \& \ \text{SPE} > \text{SPE}_{\text{th}} \ | \ f = 0)}{\text{total samples } (f = 0)}.$$

DAMADICS is the abbreviation of Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems. The general structure of benchmark actuators is shown in Figure 14, which consists of three important parts: control valve, pneumatic servomotor, and positioner. The control valve is installed in the pipeline system to control the flow of the passing fluid. Pneumatic servomotor is a device, which converts electrical signal to force of pushing the movement of valve stem. Positioner can detect the actual displacement of valve stem and control the valve more precisely.

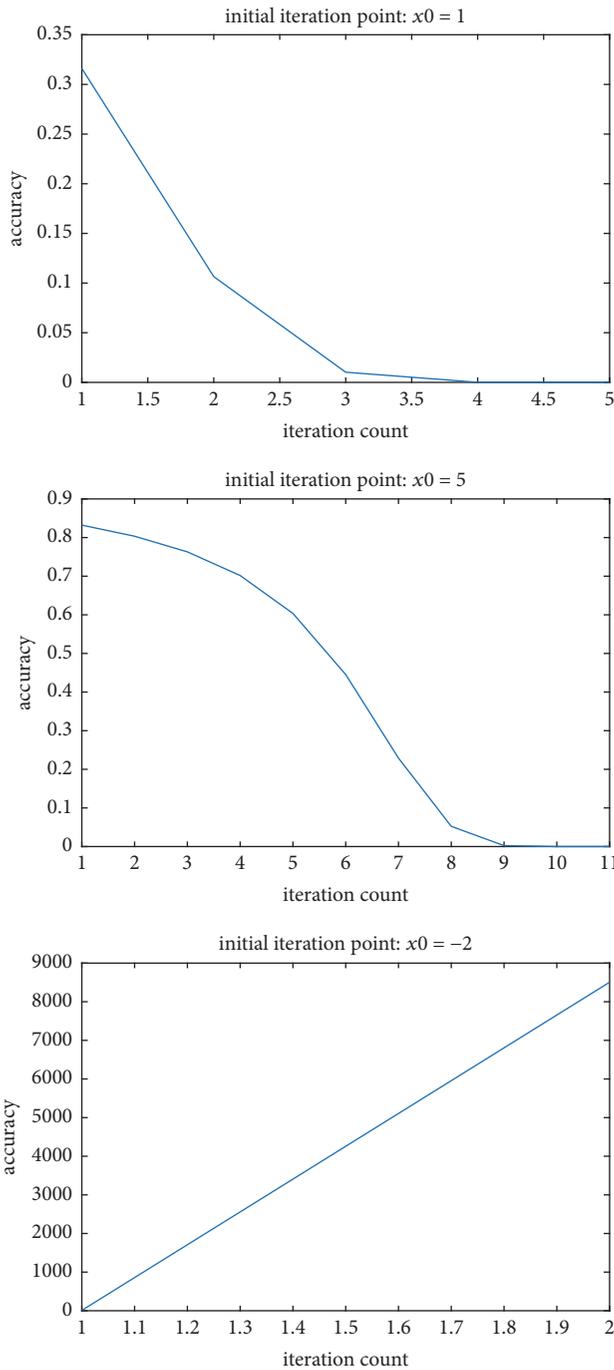


FIGURE 8: Process of iteration of different initial iterative points of Newton iteration method.

The benchmark is set with 19 different faults [30]. These faults can be classified into four classes: control valve faults (f_1 – f_7), pneumatic servomotor faults (f_8 – f_{11}), positioner faults (f_{12} – f_{14}), and general faults/external faults (f_{15} – f_{19}).

6 process variables are chosen for process monitoring, which are listed in Table 3. Only monitoring performance of several typical faults are shown here, including fault number of 7, 10, 13, and 17, which represent four different types. Faults

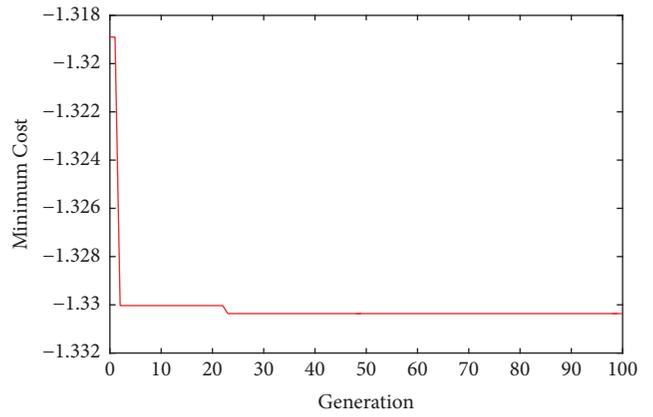


FIGURE 9: Process of iteration of BBO.

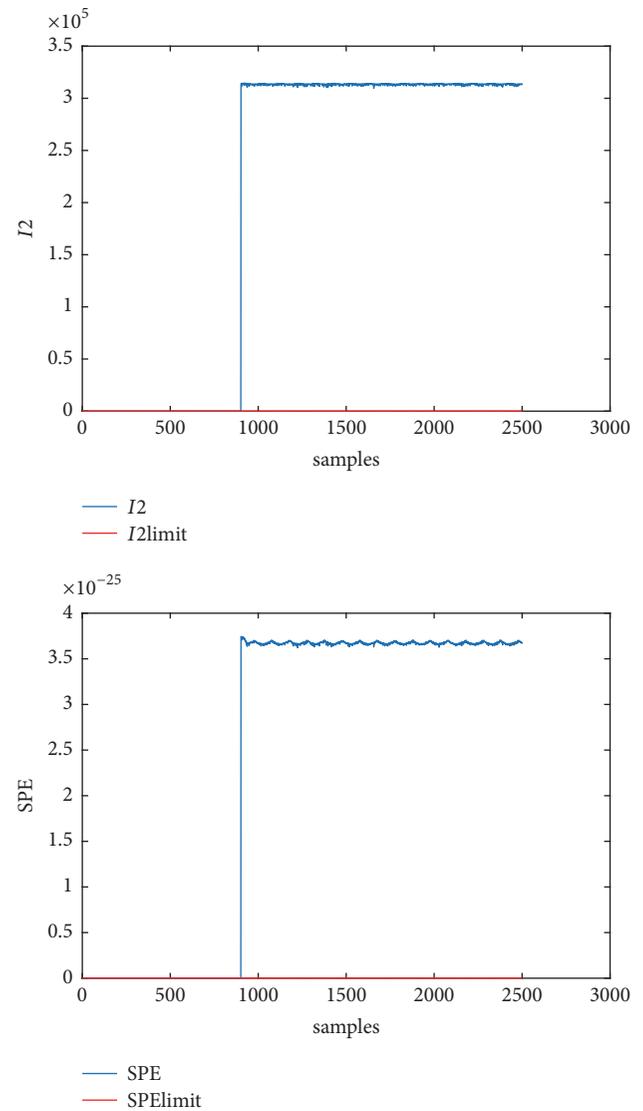
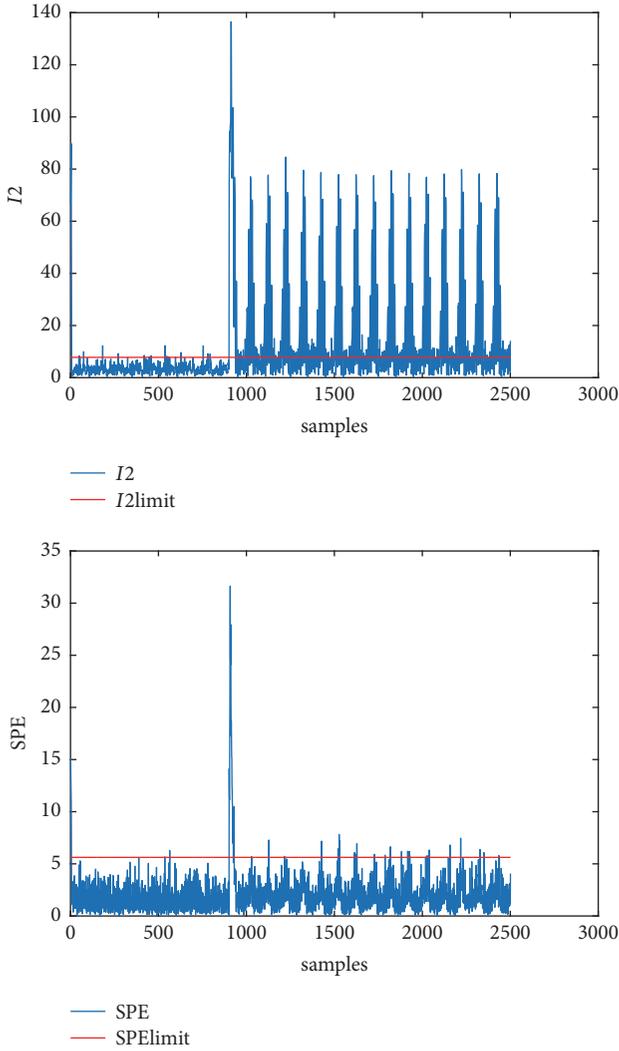
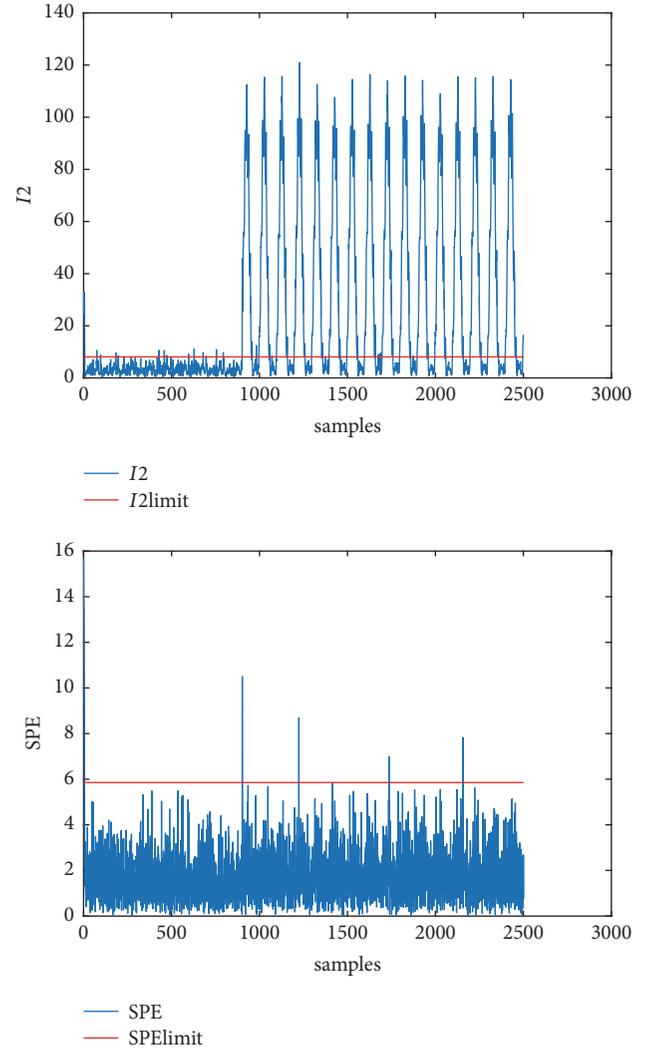


FIGURE 10: Fault 7 I_2 and SPE statistic.

are set at 900 seconds. Figures 10, 11, 12, and 13 show the monitoring statistics of I_2 and SPE of BBO-ICA. Blue lines

FIGURE 11: Fault 10 I_2 and SPE statistic.FIGURE 12: Fault 13 I_2 and SPE statistic.

denote the testing statistic of I_2 or SPE and the thresholds are shown by red lines. Faults are successfully detected after 900 seconds.

It can be seen from Table 2 that FDR and FAR of 19 kinds of faults of BBO-ICA and FastICA are compared in Table 2. With regard to FastICA, 2 initial iterative points are set to test the influence of selection of initial iterative point on the algorithm. Initial iterative point 1 is an identity matrix and the other is a null matrix. Under the circumstance of initial iterative point 2, the performance of the fault detection of FastICA algorithm is poor. Compared with the performance of process monitoring of FastICA and BBO-ICA, it can be concluded that BBO-ICA shows higher FDR and lower FAR for most of faults. Parameters of BBO of N , m , p , G_{\max} , P_{mod} , m_{\max} , I , and E are set as 100, 6, 2, 200, 1, 0.9, 1, and 1, respectively.

In addition, performance of fault detection of BBO-ICA is also compared with PSO-ICA. Most of parameters of PSO are correspondent with BBO, such as the numbers of particles and iteration. Although the PSO-ICA can also solve the

problem of selection of initial iterative point of FastICA, PSO-ICA behaves with lower FDR and higher FAR than BBO-ICA and FastICA under initial iterative point 1. So, it can be inferred that BBO have stronger capability and accuracy of searching for solution space than PSO. So, it means that BBO-ICA is a more suitable method to spread than PSO-ICA.

The proposed BBO-ICA can avoid the problem of selection of initial iterative points, which can improve the stability and robustness of the ICA algorithm. Moreover, BBO-ICA can also improve the performance of ICA algorithm. However, the BBO-ICA algorithm also has its drawbacks because of its great amount of calculation. When handling high dimensional complex problems, the process of computation will be quite slow.

It should be noted that half of faults have high FDRS and low FARs. But for some faults, that is, fault number of 3, 4, 5, 6, and so on, the efficiency of fault detection is poor, which may be attributed to little influence on variables.

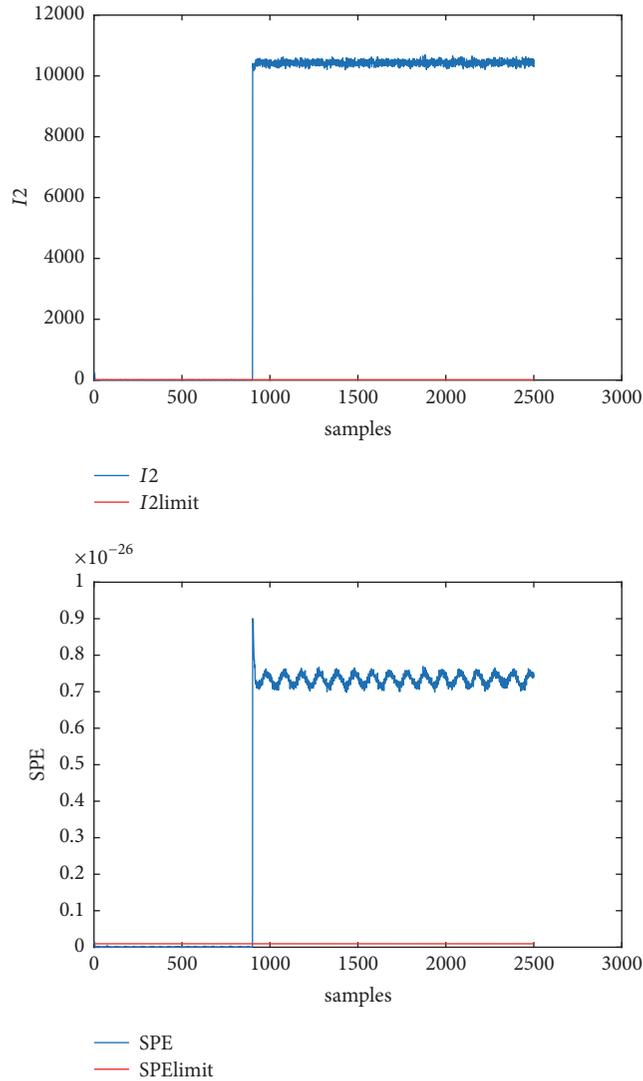


FIGURE 13: Fault 17 I_2 and SPE statistic.

TABLE 2: Process variables.

Number	Process variable
1	Process control external signal CV'
2	Values of liquid pressure on the valve inlet $P1'$
3	Values of liquid pressure on the valve outlet $P2'$
4	Stem displacement X'
5	Liquid flow rate F'
6	Liquid temperature $T1'$

5. Conclusion

BBO-ICA algorithm for process monitoring is proposed in this paper. Some drawbacks of FastICA and PSO-ICA are analysed. Compared to FastICA and PSO-ICA, the proposed BBO-ICA is more robust and stable than FastICA and

separates independent components more accurately than FastICA and PSO-ICA.

The proposed method is applied to process monitoring of the benchmark platform of DAMADICS; results demonstrate that BBO-ICA detects faults more efficiently than FastICA and PSO-ICA. Future research would be focused on the application of BBO-ICA in other subjects.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is financially supported by the Fundamental Research Funds for the Central Universities (WUT:2017II41GX).

TABLE 3: FDRs (%) and FARs (%).

Fault number	FastICA				BBO-ICA		PSO-ICA	
	Initial iterative point 1		Initial iterative point 2		FDR (%)	FAR (%)	FDR (%)	FAR (%)
	FDR (%)	FAR (%)	FDR (%)	FAR (%)				
<i>f</i> 1	61.08	2.11	0.62	1.78	61.83	2.11	51.34	2.56
<i>f</i> 2	55.9	2.11	1.25	1.78	62.84	2.0	53.34	2.67
<i>f</i> 3	1.19	2.11	0.87	1.78	1.19	1.78	1.56	2.89
<i>f</i> 4	1.12	2.11	0.99	1.78	1.19	2.11	0	0.67
<i>f</i> 5	1.37	2.11	0.87	1.78	1.06	1.89	0.43	1.22
<i>f</i> 6	1.37	2.11	0.87	1.78	1.19	1.89	1.43	2.44
<i>f</i> 7	99.94	2.11	99.94	1.78	99.94	2.11	99.94	2.89
<i>f</i> 8	1.12	2.11	0.87	1.78	1.06	2.0	1.67	1.68
<i>f</i> 9	1.12	2.11	0.87	1.78	1.06	2.0	1.06	2.67
<i>f</i> 10	53.52	2.11	0.87	1.78	47.35	2.0	40.85	2.89
<i>f</i> 11	4.06	2.11	0.99	1.78	3.94	2.11	3.5	4.0
<i>f</i> 12	0.99	2.11	0.62	1.78	0.44	1.33	0.49	1.33
<i>f</i> 13	61.09	2.11	0.62	1.78	61.83	2.11	36.16	1.78
<i>f</i> 14	1.12	2.11	0.87	1.78	1.06	1.89	2.37	2.22
<i>f</i> 15	93.94	2.11	57.15	1.78	93.07	1.56	92.44	2.56
<i>f</i> 16	38.6	2.11	0.62	1.78	33.04	1.89	34.67	3.0
<i>f</i> 17	99.94	2.11	99.94	1.78	99.94	1.44	99.94	2.33
<i>f</i> 18	61.21	2.11	0.62	1.78	62.27	2.11	61.78	2.22
<i>f</i> 19	25.86	2.11	0.56	1.78	31.48	1.78	19.49	2.22

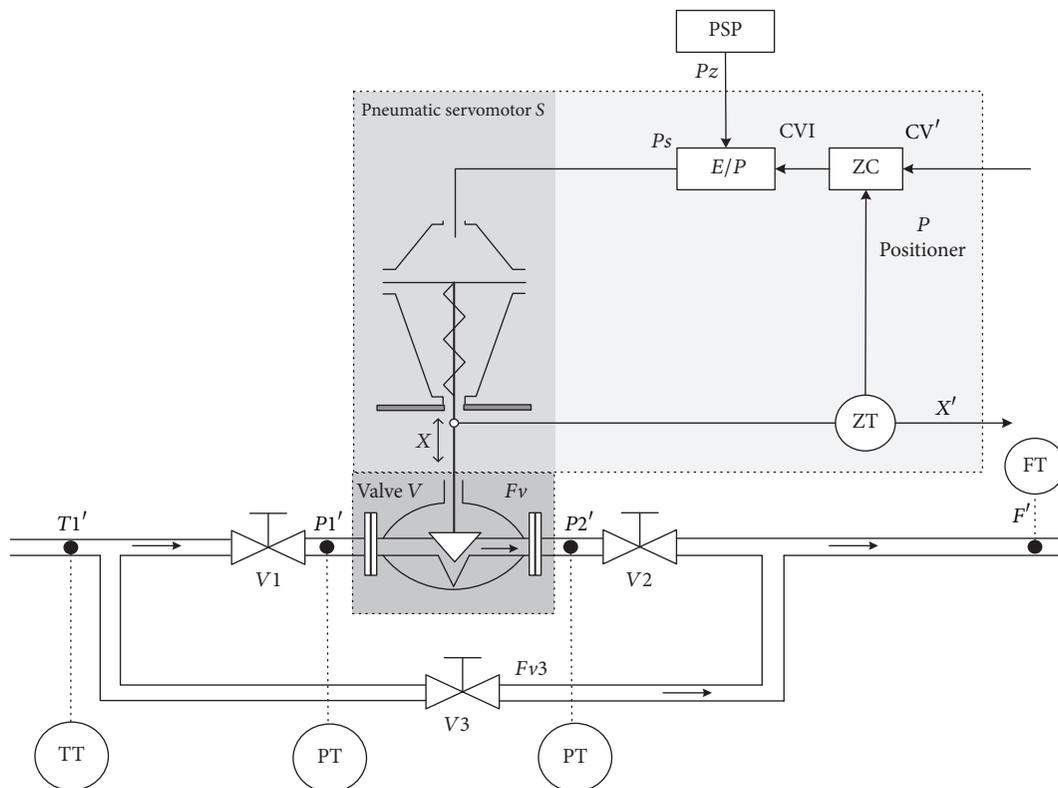


FIGURE 14: Structure of benchmark actuator system.

References

- [1] P. M. Frank, "Analytical and qualitative model-based fault diagnosis - A survey and some new results," *European Journal of Control*, vol. 2, no. 1, pp. 6–28, 1996.
- [2] G. R. Naik, "An overview of independent component analysis and its applications," *Informatica (Ljubljana)*, vol. 35, no. 1, pp. 63–81, 2011.
- [3] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 2000.
- [4] G. R. Naik, K. G. Baker, and H. T. Nguyen, "Dependence independence measure for posterior and anterior EMG sensors used in simple and complex finger flexion movements: Evaluation using SDICA," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 5, pp. 1689–1696, 2015.
- [5] J. Deslauriers, J. Ansado, G. Marrelec, J.-S. Provost, and Y. Joannette, "Increase of posterior connectivity in aging within the Ventral Attention Network: A functional connectivity analysis using independent component analysis," *Brain Research*, vol. 1657, pp. 288–296, 2017.
- [6] R. Chai, G. Naik, T. N. Nguyen et al., "Driver fatigue classification with independent component by entropy rate bound minimization analysis in an EEG-based system," *IEEE Journal of Biomedical and Health Informatics*, vol. 99, p. 1, 2016.
- [7] J. O'Muircheartaigh and S. Jbabdi, "Concurrent white matter bundles and grey matter networks using independent component analysis," *NeuroImage*, 2017.
- [8] C. Jutten and J. Herault, "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.
- [9] Y. Guo, S. Huang, Y. Li, and G. Naik, "Edge effect elimination in single-mixture blind source separation," *Circuits, Systems and Signal Processing*, vol. 32, no. 5, pp. 2317–2334, 2013.
- [10] G. Pendharkar, G. R. Naik, and H. T. Nguyen, "Using Blind Source Separation on accelerometry data to analyze and distinguish the toe walking gait from normal gait in ITW children," *Biomedical Signal Processing and Control*, vol. 13, no. 1, pp. 41–49, 2014.
- [11] P. Comon, "Independent component analysis, A new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [12] M. Kano, S. Tanaka, S. Hasebe, I. Hashimoto, and H. Ohno, "Monitoring independent components for fault detection," *AIChE Journal*, vol. 49, no. 4, pp. 969–976, 2003.
- [13] E. Bingham and A. Hyvärinen, "A fast fixed-point algorithm for independent component analysis of complex valued signals," *International Journal of Neural Systems*, vol. 10, no. 1, pp. 1–8, 2000.
- [14] J. M. Lee, S. J. Qin, and I. B. Lee, "Fault Detection of Non-Linear Processes Using Kernel Independent Component Analysis," *The Canadian Journal of Chemical Engineering*, vol. 85, no. 4, pp. 526–536, 2007.
- [15] J.-M. Lee, C. Yoo, and I.-B. Lee, "Statistical monitoring of dynamic processes based on dynamic independent component analysis," *Chemical Engineering Science*, vol. 59, no. 14, pp. 2995–3006, 2004.
- [16] Y. Yang, Y. Chen, X. Chen, and X. Liu, "Multivariate industrial process monitoring based on the integration method of canonical variate analysis and independent component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 116, pp. 94–101, 2012.
- [17] C.-C. Hsu, M.-C. Chen, and L.-S. Chen, "A novel process monitoring approach with dynamic independent component analysis," *Control Engineering Practice*, vol. 18, no. 3, pp. 242–253, 2010.
- [18] L. Cai, X. Tian, and S. Chen, "A process monitoring method based on noisy independent component analysis," *Neurocomputing*, vol. 127, pp. 231–246, 2014.
- [19] Y. Xu and X. Deng, "Fault detection of multimode non-Gaussian dynamic process using dynamic Bayesian independent component analysis," *Neurocomputing*, vol. 200, pp. 70–79, 2016.
- [20] J. Zhu, Z. Ge, and Z. Song, "Non-Gaussian Industrial Process Monitoring with Probabilistic Independent Component Analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1309–1319, 2017.
- [21] J. Huang and X. Yan, "Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and Bayesian inference," *Chemometrics and Intelligent Laboratory Systems*, vol. 148, pp. 115–127, 2015.
- [22] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [23] Y. W. Zhang and Y. Zhang, "Fault detection of non-Gaussian processes based on modified independent component analysis," *Chemical Engineering Science*, vol. 65, no. 16, pp. 4630–4639, 2010.
- [24] M. Bartyś, R. Patton, M. Syfert, S. de las Heras, and J. Quevedo, "Introduction to the DAMADICS actuator FDI benchmark study," *Control Engineering Practice*, vol. 14, no. 6, pp. 577–596, 2006.
- [25] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. Sejnowski, "Combining time-delayed decorrelation and ICA: Towards solving the cocktail party problem," in *Proceedings of the 1998 23rd IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998*, pp. 1249–1252, USA, May 1998.
- [26] C. Darwin and G. Beer, *The Origin of Species*, Dent, UK, 1951.
- [27] R. H. MacArthur and E. O. Wilson, *Theory of Island Biogeography (MPB-1)*, vol. 1, Princeton University Press, Princeton, NJ, USA, 2015.
- [28] T. A. Wesche, C. M. Goertler, and W. A. Hubert, "Modified habitat suitability index model for brown trout in southeastern wyoming," *North American Journal of Fisheries Management*, vol. 7, no. 2, pp. 232–237, 1987.
- [29] I. Hanski, "The Metapopulation Approach, Its History, Conceptual Domain, and Application to Conservation," *Metapopulation Biology*, pp. 5–26, 1997.
- [30] J. M. Kościelny and M. Z. Bartyś, "Application of Information System Theory for Actuators Diagnosis," *IFAC Proceedings Volumes*, vol. 33, no. 11, pp. 927–932, 2000.
- [31] A. Bhattacharya and P. K. Chattopadhyay, "Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1955–1964, 2010.
- [32] S. Wang, Y. Zhang, G. Ji, J. Yang, J. Wu, and L. Wei, "Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization," *Entropy*, vol. 17, no. 8, pp. 5711–5728, 2015.
- [33] K. Zhou, C. Fu, and S. Yang, "Big data driven smart energy management: From big data to big insights," *Renewable & Sustainable Energy Reviews*, vol. 56, pp. 215–225, 2016.

- [34] Y. Zhang, S. Wang, Z. Dong, P. Phillip, G. Ji, and J. Yang, "Pathological brain detection in magnetic resonance imaging scanning by wavelet entropy and hybridization of biogeography-based optimization and particle swarm optimization," *Progress in Electromagnetics Research*, vol. 152, pp. 41–58, 2015.
- [35] P. K. Roy, S. P. Ghoshal, and S. S. Thakur, "Optimal VAR control for improvements in voltage profiles and for real power loss minimization using Biogeography Based Optimization," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 830–838, 2012.

Research Article

Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets

Xingyu Zhou ¹, Zhisong Pan ¹, Guyu Hu ¹, Siqi Tang,¹ and Cheng Zhao^{1,2}

¹Army Engineering University of PLA, Nanjing 210007, China

²Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu 241003, China

Correspondence should be addressed to Zhisong Pan; hotpzs@hotmail.com

Received 6 November 2017; Revised 21 January 2018; Accepted 13 February 2018; Published 15 April 2018

Academic Editor: Qian Zhang

Copyright © 2018 Xingyu Zhou 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.

Stock price prediction is an important issue in the financial world, as it contributes to the development of effective strategies for stock exchange transactions. In this paper, we propose a generic framework employing Long Short-Term Memory (LSTM) and convolutional neural network (CNN) for adversarial training to forecast high-frequency stock market. This model takes the publicly available index provided by trading software as input to avoid complex financial theory research and difficult technical analysis, which provides the convenience for the ordinary trader of nonfinancial specialty. Our study simulates the trading mode of the actual trader and uses the method of rolling partition training set and testing set to analyze the effect of the model update cycle on the prediction performance. Extensive experiments show that our proposed approach can effectively improve stock price direction prediction accuracy and reduce forecast error.

1. Introduction

Predicting stock prices is an important objective in the financial world [1–3], since a reasonably accurate prediction has the possibility to yield high financial benefits and hedge against market risks. With the rapid growth of Internet and computing technologies, the frequency for performing operations on the stock market had increased to fractions of seconds [4, 5]. Since year of 2009 the BM&F Bovespa (the Brazilian stock exchange) has worked in high-frequency, and the number of high-frequency operations has grown from 2.5% in 2009 to 36.5% in 2013. Aldridge and Krawciw [6] estimate that in 2016 high-frequency trading on average initiated 10%–40% of trading volume in equities and 10%–15% of volume in foreign exchange and commodities. These percentages suggest that the high-frequency stock market is a global trend.

In most cases, the forecast results are assessed from two aspects: the first is forecast error (chiefly the RMSE (Root Mean Square Error) or RMSRE (Root Mean Square Relative Error)) between real price and forecast value; the second is direction prediction accuracy, which means the percentage of correct predictions of price series direction, as upward and

downward movements are what really matters for decision-making. Even small improvements in predictive performance can be very profitable [7, 8].

However, predicting stock prices is not an easy work, due to the complexity and chaotic dynamics of the markets and the many nondecidable, nonstationary stochastic variables involved [9]. Many researchers from different areas have studied the historical patterns of financial time series and have proposed various methods for forecasting stock prices. In order to achieve promising performance, most of these ways require careful selection of input variables, establishing predictive model with professional financial knowledge, and adopting various statistical methods for arbitrage analysis, which makes it difficult for people outside the financial field to use these methods to predict stock prices [10–12].

Generative adversarial network (GAN) was introduced by Goodfellow et al. [13], where images patches are generated from random noise using two networks trained simultaneously. Specifically, in GAN a discriminative net D learns to distinguish whether a given data instance is real or not, and a generative net G learns to confuse D by generating high quality data. Although this approach has been successful and applied to a wide range of fields, such as image inpainting,

semantic segmentation, and video prediction [14–16], as far as we know, it has not been used for stock forecasting.

This work uses basic technical index data as an input variable, which can be acquired directly from trading software, so that people outside the financial field can predict stock price through our method easily. This study introduces forecast error loss and direction prediction loss and shows that generative adversarial training [13] may be successfully employed for combining these losses to produce satisfying predict results, and we call this prediction architecture GAN-FD (GAN for minimizing forecast error loss and direction prediction loss). For the purpose of conforming to the practice of actual transactions, this work carries out rolling segmentation on training set and testing set of the raw data, and we will illustrate it in detail in the experimental section.

Overall, our main contributions are twofold: (1) we adapted generative adversarial network for the purpose of price prediction, which constitutes to our knowledge the first application of adversarial training to stock market, and extensive experiments show that our prediction model can achieve remarkable results and (2) we carry out rolling segmentation on training set and testing set of the raw data to investigate the effect the of model parameter update cycle on the stock forecast performance, and the experimental results show that smaller model update cycle can advance prediction performance.

In the remainder of this paper, we begin with a review of the literature on which algorithms have been used for the financial market prediction. Then we formulate the problem and propose our general adversarial network framework. Furthermore, in the experiments section, we presented the experimental analysis with the proposed model, as well as a comparison between the obtained results with those given by classical prediction models. Finally, conclusions and possible extensions are discussed.

2. Related Work

This section introduce the related work from the stock market prediction method and the generative adversarial network.

2.1. Stock Market Prediction Method. According to the research developed in this field, we can classify the techniques used to solve the stock market prediction problems to twofold.

The first category of related work is *econometric models*, which includes classical econometric models for forecasting. Common methods are the autoregressive method (AR), the moving average model (MA), the autoregressive moving average model (ARMA), and the autoregressive integrated moving average (ARIMA) [17–19]. Roughly speaking, these models take each new signal as a noisy linear combination of the last few signals and independent noise terms. However, most of them rely on some strong assumptions with respect to the noise terms (such as i.i.d. assumption, t -distribution) and loss functions, while real financial data may not fully satisfy these assumptions. By introducing a generalized autoregressive conditional heteroscedastic (GARCH) model

for conditional variances, Pellegrini et al. [20] apply ARIMA-GARCH model to the prediction of financial time series.

The second category involves *soft computing based models*. Soft computing is a term that covers artificial intelligence which mimics biological processes. These techniques include artificial neural networks (ANN) [21, 22], fuzzy logic (FL) [23], support vector machines (SVM) [24, 25], particle swarm optimization (PSO) [26], and many others. Many authors have tried to deal with fuzziness along with randomness in option pricing models [27, 28]. Carlsson and Fullér [29] were the first to study the fuzzy real options and Thavaneswaran et al. [30] demonstrated the superiority of the fuzzy forecasts and then derived the membership function for the European call price by fuzzifying the interest rate, volatility, and the initial value of the stock price. Recently there has been a resurgence of interest in deep learning, whose basic structure is best described as a multilayer neural network [31]. Some literatures have established various models based on deep neural networks to improve the prediction ability of high-frequency financial time series [32, 33]. The ability of deep neural networks to extract abstract features from data is also attractive, Chong et al. [12] applied a deep feature learning-based stock market prediction model, which extract information from the stock return time series without relying on prior knowledge of the predictors and tested it on high-frequency data from the Korean stock market. Chen et al. [34] proposed a double-layer neural network for high-frequency forecasting, with links specially designed to capture dependence structures among stock returns within different business sectors. There also exist a few studies that apply deep learning to identification of the relationship between past news events and stock market movements [35–37].

However, to our knowledge, most of these methods require expertise to impose specific restrictions on the input variables, such as combining related stocks together as entry data [12], inputting different index data to different layers of the deep neural network [34], and converting news text into structured representation as input [36]. In contrast, our proposed forecasting model directly uses the data provided by the trading software as input, which reduce the barrier for ordinary investors.

2.2. Generative Adversarial Network. Generative adversarial network (GAN) is a framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to 0.5 everywhere [13]. While G and D are defined by multilayer perceptrons in [13], most researches recently constructed G and D on the basis of Long Short-Term Memory (LSTM) [38] or convolutional neural network (CNN) [39] for a large variety of application.

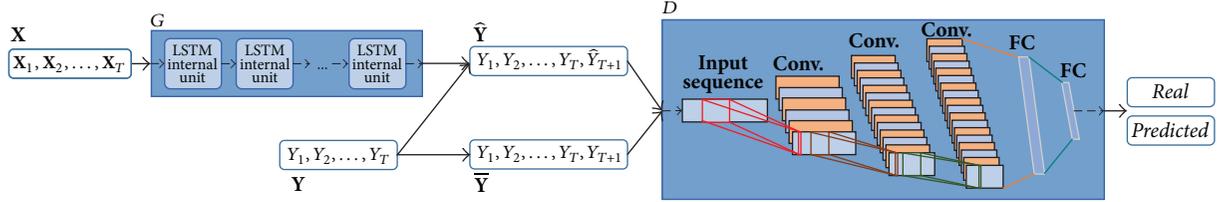


FIGURE 1: GAN-FD architecture. The generator (G) is founded on LSTM, which applies to predicting \hat{Y}_{T+1} . The discriminator (D) is based on CNN for the purpose of estimating the probability whether a sequence is real (\bar{Y}) or being predicted (\hat{Y}). *Conv.* means convolutional layer, *FC* is an abbreviation for fully connected layer. The structure of G and D can be adjusted according to the specific application.

LSTM is a basic deep learning model and capable of learning long-term dependencies. A LSTM internal unit is composed of a cell, an input gate, an output gate, and a forget gate. LSTM internal units have hidden state augmented with nonlinear mechanisms to allow state to propagate without modification, be updated, or be reset, using simple learned gating functions. LSTM work tremendously well on various problems, such as natural language text compression, handwriting recognition, and electric load forecasting.

CNN is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. A CNN consists of an input layer and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers. CNN also has many applications such as image and video recognition, recommender systems, and natural language processing.

Although there are a lot of literatures forecast stock price by using LSTM model, to the best of our knowledge, this paper is the first to adopt GAN to predict stock prices. The experimental part (Section 4.2) compares the prediction performances between GAN-FC and LSTM.

3. Forecasting with High-Frequency Data

In this section, we illuminate the details of the generative adversarial network framework for stock market forecasting with high-frequency data.

3.1. Problem Statement. Under the high-frequency trading environment, high-quality one-step forecasting is usually of great concern to algorithmic traders, providing significant information to market makers for risk assessment and management. In this article, we aim to forecast the price movement of individual stocks or the market index one step ahead, based solely on their historical price information. Our problem can be mathematically formalized as follows.

Let \mathbf{X}_t represent a set of basic indicators and Y_t denote the closing price of one stock for a 1-minute interval at time t ($t = 1, 2, \dots, T$), where T is the maximum lag of time. Given the historical basic indicators information \mathbf{X} ($\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T\}$) and the past closing price \mathbf{Y} ($\mathbf{Y} = \{Y_1, Y_2, \dots, Y_T\}$), our goal is to predict the closing price Y_{T+1} for the next 1-minute time interval. There are literatures that examined the effects of different T [7, 12, 40], but, in this work,

we just set T to 242 because each trading day contains 242-minute intervals in the China stock exchanges.

3.2. Prediction Model. The deep architecture of the proposed GAN-FD model is illustrated as in Figure 1. Since the stock data is a typical time series, we choose LSTM model, which is widely applied to time series prediction, as the generative model G to predict output \hat{Y}_{T+1} based on the input data \mathbf{X} ; that is,

$$\hat{Y}_{T+1} = G(\mathbf{X}). \quad (1)$$

The discriminative model D is based on the CNN architecture and performs convolution operations on the one-dimensional input sequence in order to estimate the probability whether a sequence comes from the dataset ($\bar{Y} = \{Y_1, Y_2, \dots, Y_T, Y_{T+1}\}$) or being produced by a generative model G ($\hat{Y} = \{Y_1, Y_2, \dots, Y_T, \hat{Y}_{T+1}\}$).

Our main intuition on why to use an adversarial loss is that it can simulate the operating habits of financial traders. An experienced trader usually predicts stock price through the available indicator data, which is the work of the generative model G , and then judges the correct probability of his own forecast with the previous stock price, as the discriminative model D does.

It is noteworthy that the structure of G and D in GAN-FD can be adjusted according to specific application, and the experimental part in this paper just proposed simple G and D framework (Section 4.2) for stock prediction. It is reasonable to believe that fine-tuning the structure of G and D can improve the predictive performance.

3.3. Adversarial Training. The training of the pair (G, D) consists of two alternated steps, described below. For the sake of clarity, we assume that we use pure SGD (minibatches of size 1), but there is no difficulty to generalize the algorithm to minibatches of size K by summing the losses over the samples.

Training G (let (\mathbf{X}, \mathbf{Y}) be a sample from the dataset). In order to make the discriminative model D as “confused” as possible, the generative model G should reduce the adversarial loss in the sense that D will not discriminate the prediction correctly. Classifying \bar{Y} into class 1 and \hat{Y} into class 0, the adversarial loss for G is

$$L_{\text{adv}}^G(\hat{Y}) = L_{\text{sce}}(D(\hat{Y}), 1), \quad (2)$$

where L_{sce} is the sigmoid cross-entropy loss, defined as

$$L_{\text{sce}}(\mathbf{A}, \mathbf{B}) = -\sum_i B_i \log(\text{sigmoid}(A_i)) + (1 - B_i) \log(1 - \text{sigmoid}(A_i)). \quad (3)$$

However, in practice, minimizing adversarial loss alone cannot guarantee satisfying predictions. Imagine that G could generate samples to “confuse” D , without being close to \hat{Y}_{T+1} , and then D will learn to discriminate these samples, leading G to generate other “confusing” samples, and so on. To address this problem, the generative model G ought to decrease the forecast error loss; that is, L_p loss

$$L_p(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}) = \|\bar{\mathbf{Y}} - \hat{\mathbf{Y}}\|_p \quad (4)$$

where $p = 1$ or $p = 2$.

Furthermore, as mentioned above, stock price direction prediction is crucial to trading, so we define direction prediction loss function L_{dpl} :

$$L_{\text{dpl}}(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}) = \left| \text{sgn}(\hat{Y}_{T+1} - Y_T) - \text{sgn}(Y_{T+1} - Y_T) \right| \quad (5)$$

where sgn represents sign function.

Combining all these losses previously defined with different parameters λ_{adv} , λ_p , and λ_{dpl} , we achieve the final loss on G :

$$L_G(\mathbf{X}, \mathbf{Y}) = \lambda_{\text{adv}} L_{\text{adv}}^G(\hat{\mathbf{Y}}) + \lambda_p L_p(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}) + \lambda_{\text{dpl}} L_{\text{dpl}}(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}). \quad (6)$$

Then we perform one SGD iteration on G to minimize $L_G(\mathbf{X}, \mathbf{Y})$ while keeping the weights of D fixed.

Training D (let (\mathbf{X}, \mathbf{Y}) be a different data sample). Since the role of D is just to determine whether the input sequence is \mathbf{Y} or $\hat{\mathbf{Y}}$, the target loss is equal to the adversarial loss on D . While keeping the weights of G fixed, we perform one SGD step on D to minimize the target loss:

$$L_D(\mathbf{X}, \mathbf{Y}) = L_{\text{adv}}^D(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}) = L_{\text{sce}}(D(\hat{\mathbf{Y}}), 0) + L_{\text{sce}}(D(\bar{\mathbf{Y}}), 1). \quad (7)$$

We train the generator and discriminator iteratively. The entire process is summarized in Algorithm 1, with minibatches of size K .

- (1) Set the learning rates ρ_D and ρ_G , and parameters $\lambda_{\text{adv}}, \lambda_p, \lambda_{\text{dpl}}$;
- (2) Initialize weights W_D and W_G .
- (3) **while** not converged **do**
- (4) **Update the generator G :**
- (5) Get K new data samples $(\mathbf{X}^{(1)}, \mathbf{Y}^{(1)}), (\mathbf{X}^{(2)}, \mathbf{Y}^{(2)}), \dots, (\mathbf{X}^{(K)}, \mathbf{Y}^{(K)})$
- (6) $W_G = W_G - \rho_G \sum_i^K \frac{\partial L_G(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})}{\partial W_G}$
- (7) **Update the discriminator D :**
- (8) Get K new data samples $(\mathbf{X}^{(1)}, \mathbf{Y}^{(1)}), (\mathbf{X}^{(2)}, \mathbf{Y}^{(2)}), \dots, (\mathbf{X}^{(K)}, \mathbf{Y}^{(K)})$
- (9) $W_D = W_D - \rho_D \sum_i^K \frac{\partial L_D(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})}{\partial W_D}$
- (10) **end while**

ALGORITHM 1: Training GAN-FD.

4. Experiments

4.1. Dataset. Next, we evaluate the performance of the proposed method based on the China stock market, ranging from January 1, 2016, to December 31, 2016. There are totally 244 trading days and each day contains 242-minute intervals, corresponding to 59048 time points. These stocks selected for the experiment should conform to three criteria: first, they should be the constituent stock of CSI 300 (the CSI 300 is a capitalization-weighted stock market index designed to replicate the performance of 300 stocks traded in the Shanghai and Shenzhen stock exchanges); second, they were not suspended during the period we just mentioned, in case accidental events bring about significant impact on their price and affect forecast results; third, their closing prices in the start time, that is, January 1, 2016, are above 30 to ensure the volatility for high-frequency exchange. This leaves 42 stocks in the sample, which are listed in Table 1. The number of increasing directions and decreasing directions for each stock's closing price per minute is also shown in Table 1, and their numbers are relatively close. The historical data was obtained from the Wind Financial Terminal, produced by Wind Information Inc. (the Wind Financial Terminal can be downloaded from <http://www.wind.com.cn>).

Many fund managers and investors in the stock market generally accept and use certain criteria for technical indicators as the signal of future market trends [12, 41]. This work selects 13 technical indicators as feature subsets by the review of domain experts and prior researches; that is, the input data \mathbf{X} at each moment (e.g., \mathbf{X}_T) consists of 13 basic indicators that can be obtained directly from almost all trading software. These basic indicators are listed in Table 2, and their parameters are using the default value of the Wind Financial Terminal. As mentioned above, \mathbf{Y} is defined as the closing price at each moment.

Most of the related articles use the traditional data partitioning method; that is, the entire dataset is directly split into training set and testing set [12, 22, 40, 42]. However,

TABLE 1: The sample stocks and their number of increasing directions and decreasing directions.

ID	Stock code	Increase	Decrease
1	000156.SZ	28927	30120
2	000432.SZ	28879	30168
3	000783.SZ	28310	30737
4	000938.SZ	28537	30510
5	000963.SZ	29192	29855
6	002007.SZ	28933	30114
7	002027.SZ	28623	30424
8	002153.SZ	28566	30481
9	002183.SZ	28795	30252
10	002195.SZ	28861	30186
11	002241.SZ	29084	29963
12	002241.SZ	28737	30310
13	002252.SZ	28696	30351
14	002292.SZ	28385	30662
15	002304.SZ	28914	30133
16	002415.SZ	29036	30011
17	002456.SZ	28671	30376
18	002475.SZ	28837	30210
19	002594.SZ	28525	30522
20	300017.SZ	28528	30519
21	300024.SZ	28411	30636
22	300072.SZ	28884	30163
23	300124.SZ	28632	30415
24	300146.SZ	29137	29910
25	600038.SH	28428	30619
26	600085.SH	28856	30191
27	600118.SH	28456	30591
28	600150.SH	28537	30510
29	600332.SH	29174	29873
30	600340.SH	28773	30274
31	600519.SH	29118	29929
32	600535.SH	29013	30034
33	600570.SH	28053	30994
34	600588.SH	28483	30564
35	600685.SH	28627	30420
36	600718.SH	28881	30166
37	600754.SH	28307	30740
38	600783.SH	28680	30367
39	601318.SH	28979	30068
40	601336.SH	28643	30404
41	601888.SH	28919	30128
42	603885.SH	28817	30230

the trading style of the stock market changes frequently; for example, investors sometimes prefer stocks with high volatility and sometimes tend to invest in technology stocks. Therefore, we should update the model parameters regularly to adapt to the change of market style. In order to make experiments closer to real transactions, we carry out rolling

TABLE 2: Basic indicators for prediction.

Indicators
Opening price
Maximum price
Minimum price
Trading volume
Turnover
Bias
Bollinger bands
Directional movement index
Exponential moving averages
Stochastic index
Moving averages
MACD
Relative strength index

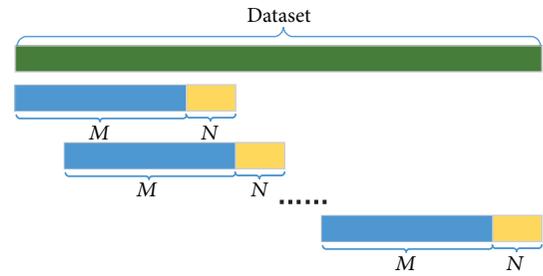


FIGURE 2: Rolling segmentation on training set and testing set. The green bar represents the entire dataset, the blue bar represents the training set for a round experiment, and the yellow bar represents the corresponding testing set.

segmentation on training set and testing set of the experimental data. As Figure 2 shows, in the beginning, we select the first M days as training set, and the next N days play the role of testing set. After the first round of experiments, we roll forward the time window for N days, that is, choosing the $(N+1)$ th day to the $(M+N)$ th day as training set and the $(M+N+1)$ th day to the $(M+2N)$ th day as testing set. Repeat until all the data has been experimented. In other words, this N can be regarded as the model update cycle, and M is the size of the corresponding training data.

4.2. Network Architecture. Given that the LSTM generator takes on the role of prediction and requires more accurate calculations of values than the CNN discriminator, we set the learning rate ρ_G to 0.0004 and ρ_D to 0.02. The LSTM cell in G contains 121 internal (hidden) units and the parameters are initialized following the normal distribution $\mathcal{N}(0, 1)$. The architecture of discriminative model D is presented in Table 3. We train GAN-FD with $p = 2$ weighted by $\lambda_{adv} = \lambda_p = \lambda_{dpl} = 1$.

4.3. Benchmark Methods. To evaluate the performance of our proposed method, we include three baseline methods for comparison. The first model is ARIMA (1, 1, 1)-GARCH(1, 1), a fitted ARIMA model that forecasts future

TABLE 3: Network architecture of discriminative model D .

Layer	Configuration
Convolution 1	Filter $32 \times 4 \times 1$, strides 2, LReLU
Convolution 2	Filter $64 \times 4 \times 1$, strides 2, LReLU, BN
Convolution 3	Filter $128 \times 4 \times 1$, strides 2, LReLU, BN
FC 1	128, leaky ReLU
FC 2	2, sigmoid

Optimizer: SGD; batch size: 121; iterations: 10000; LReLU slope: 0.01.

TABLE 4: Summary of RMSRE with different (M, N) . These figures are the average values over the 42 stocks.

	$N = 5$			$N = 10$			$N = 20$		
	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$
ARIMA-GARCH	0.0419	0.0406	0.0425	0.0529	0.0529	0.0516	0.0733	0.0657	0.0739
ANN	0.0419	0.0485	0.0531	0.0522	0.0522	0.0510	0.0739	0.0631	0.0631
SVM	0.0512	0.0450	0.0487	0.0539	0.0507	0.0527	0.0616	0.0666	0.0639
GAN-F	0.0151	0.0155	0.0157	0.0300	0.0243	0.0277	0.0326	0.0313	0.0299
GAN-D	0.0422	0.0304	0.0503	0.0625	0.0419	0.0405	0.0514	0.0598	0.0420
LSTM-FD	0.0200	0.0194	0.0180	0.0324	0.0230	0.0245	0.0321	0.0335	0.0357
GAN-FD	0.0098	0.0079	0.0101	0.0218	0.0111	0.0144	0.0333	0.0323	0.0296

values of stock time series and the GARCH model forecasts future volatilities [20]. The second one is artificial neural networks (ANN). The parameter optimization method and model architectural is setting as in [21], except that the input layer node is changed to 13 and the network outputs the predicted value instead of two patterns (0 or 1). The third one is support vector machines (SVM). An RBF kernel is used and the parameter is setting as in [25].

We also inspect our GAN-FD model from several ways. The GAN-F model is using a GAN architectural for minimizing forecast error loss, with $\lambda_{adv} = \lambda_p = 1$ and $\lambda_{dpl} = 0$. The GAN-D model is using a GAN architectural for minimizing direction prediction loss, with $\lambda_{adv} = \lambda_{dpl} = 1$ and $\lambda_p = 0$. The LSTM-FD model is a LSTM model aiming at minimizing forecast error loss and direction prediction loss, with 121 internal units in LSTM. Obviously, the main difference between LSTM-FD and GAN-FD is the presence of adversarial training.

4.4. Evaluation Metrics. For each stock at each time t , a prediction is made for the next time point $t + 1$ based on a specific method. Assume the total number of time points being tested is T_0 ; we used the following criteria to evaluate the performance of different models.

(1) *Root Mean Squared Relative Error (RMSRE)*

$$\text{RMSRE} = \sqrt{\frac{1}{T_0} \sum_{t=1}^{T_0} \left(\frac{\hat{Y}_{t+1} - Y_{t+1}}{Y_{t+1}} \right)^2}. \quad (8)$$

RMSRE is employed as an indicator for the predictive power or prediction agreement. A low RMSRE indicates that the prediction agrees with the real data (the reason why this paper uses RMSRE instead of RMSE is that RMSRE facilitates a uniform comparison of the results of 42 stocks).

(2) *Direction Prediction Accuracy (DPA)*

$$\text{DPA} = \frac{100}{T_0} \sum_{t=1}^{T_0} I_t, \quad (9)$$

where

$$I_t = \begin{cases} 1 & \text{if } (Y_{t+1} - Y_t) (\hat{Y}_{t+1} - Y_t) > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

DPA measures the percentage of accuracy relating to the series trend. A high DPA promises more winning trades.

4.5. Results. In order to investigate the effect of the model update cycle on the predictive performance, let $M \in \{10, 20, 60\}$ and $N \in \{5, 10, 20\}$. In China stock exchange market, $\{5, 10, 20, 60\}$ days represent one week, two weeks, one month, and one quarter.

Tables 4 and 5 show the average values of RMSRE and DPA with different (M, N) . The numbers clearly indicate that GAN-FD and its related methods perform better than three baseline methods in terms of RMSRE and DPA. This targeted method GAN-F brings some improvement in RMSRE, but it does not outperform three baseline methods in DPA. Contrary to GAN-F, GAN-D achieves better results in DPA but failed in RMSRE. LSTM-FD improves the results, since it combines forecast error loss with direction prediction loss for training. Finally the combination of the forecast error loss, direction prediction loss, and adversarial training, that is, GAN-FD, achieves the best RMSRE and DPA in the majority of scenarios.

Let us take a look at the effects of different (M, N) on the experiment. GAN-FD obtains the maximum average DPA (0.6956) and the minimum average RMSRE (0.0079) when

TABLE 5: Summary of DPA with different (M, N) . These figures are the average values over the 42 stocks.

	$N = 5$			$N = 10$			$N = 20$		
	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$
ARIMA-GARCH	0.5464	0.5479	0.5264	0.5280	0.5315	0.5245	0.5007	0.5214	0.5296
ANN	0.5456	0.5978	0.5738	0.5473	0.5629	0.5575	0.5205	0.5280	0.5394
SVM	0.5715	0.5490	0.5839	0.5377	0.5514	0.5576	0.5147	0.5144	0.5318
GAN-F	0.5347	0.5507	0.5281	0.4930	0.5115	0.5265	0.4880	0.5008	0.5116
GAN-D	0.6220	0.6399	0.6409	0.6117	0.6245	0.6437	0.5217	0.5517	0.5498
LSTM-FD	0.6340	0.6506	0.6509	0.6124	0.6236	0.6256	0.5546	0.5635	0.5719
GAN-FD	0.6761	0.6956	0.6793	0.6233	0.6651	0.6687	0.5535	0.5583	0.5753

TABLE 6: The number of times about the minimum RMSRE.

	$N = 5$			$N = 10$			$N = 20$		
	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$
ARIMA-GARCH	0	0	0	0	0	0	0	0	0
ANN	0	0	0	0	0	0	0	0	0
SVM	0	0	0	0	0	0	0	0	0
GAN-F	11	2	4	6	6	4	13	18	11
GAN-D	0	0	0	0	0	0	2	0	7
LSTM-FD	0	0	5	6	3	3	16	10	5
GAN-FD	31	40	33	30	33	35	11	14	19

TABLE 7: The number of times about the maximum DPA.

	$N = 5$			$N = 10$			$N = 20$		
	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$	$M = 10$	$M = 20$	$M = 60$
ARIMA-GARCH	0	0	0	0	0	0	0	6	5
ANN	0	0	0	0	0	0	2	0	3
SVM	0	0	0	0	0	0	3	1	1
GAN-F	0	0	0	0	0	0	0	0	0
GAN-D	0	1	6	5	1	7	2	6	2
LSTM-FD	1	3	4	11	4	0	18	15	12
GAN-FD	41	38	32	26	37	35	17	14	19

(M, N) is $(20, 5)$. It is interesting to note that all these methods work better when N is 5 than when N is 10 or 20, with smaller RMSRE and higher DPA. This implies that very short-term trends are best for predicting the next minute's price. Therefore, a shorter model update cycle (e.g., N is 5) is preferred. On the other hand, for the same N , different M will bring about some changes to the prediction results. From the experimental results, we suggest that M should take the value greater than N . This makes intuitive sense. If the training sample is inadequate, it would fail to train the model, especially in the volatile stock markets. We should also notice that when the training set is small while the testing set is large (i.e., (M, N) is $(10, 20)$), most of these methods perform the worst, and the DPA of these methods are no better than random guessing (i.e., 50%).

Table 6 shows the number of times for each method to achieve the minimum RMSRE over the 42 stocks. It is noticeable that the results of these three baseline methods are all zero. GAN-FD with its related methods is obviously better than these three baseline methods in RMSRE. Meanwhile,

GAN-FD obtains the minimum RMSRE 246 times, accounting for 65.08% in these 378 scenarios (42 stocks and 9 groups (M, N)). The best performance appeared when (M, N) is $(20, 5)$, with 40 stocks' minimum RMSRE coming from GAN-FD.

Table 7 shows the number of times for each method to achieve the maximum DPA over the 42 stocks. Compared with the other six methods, GAN-FD achieves the maximum DPA 269 times, accounting for 71.16% in all scenarios. When (M, N) is $(10, 5)$, the maximum DPA of 41 stocks in all 42 stocks comes from GAN-FD. Even when (M, N) is $(20, 20)$, that is, the worst performance of GAN-FD cases, GAN-FD still obtains maximum DPA in 14 stocks. From the above analyses, the performance of the GAN-FD is significantly better than the other six ways.

The results of each representation are reported in Figures 3–11. We just focus on GAN-FD. As shown in Figures 3–5, the DPA of GAN-FD ranges around 64.59%–72.24% when N is 5, and it slumps to 52.01%–62.71% when N is 20, which is presented in Figures 9–11. When N is 5, the RMSRE of GAN-FD over the 42 stocks varies between 0.48% and 1.49%,

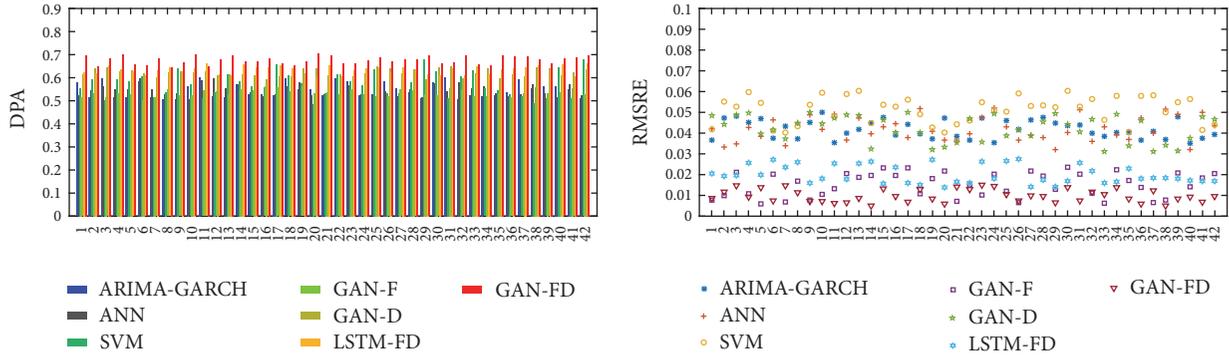


FIGURE 3: DPA and RMSRE of each stock when (M, N) is $(10, 5)$ and x -axis represents the stock ID.

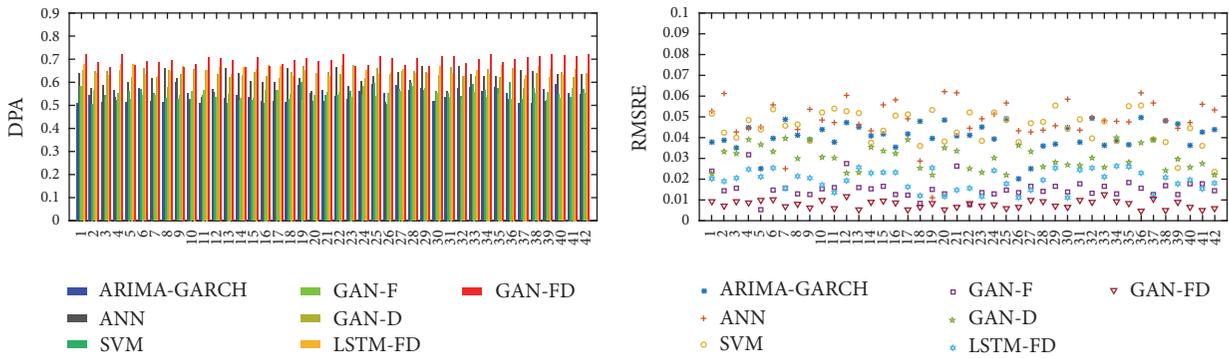


FIGURE 4: DPA and RMSRE of each stock when (M, N) is $(20, 5)$ and x -axis represents the stock ID.

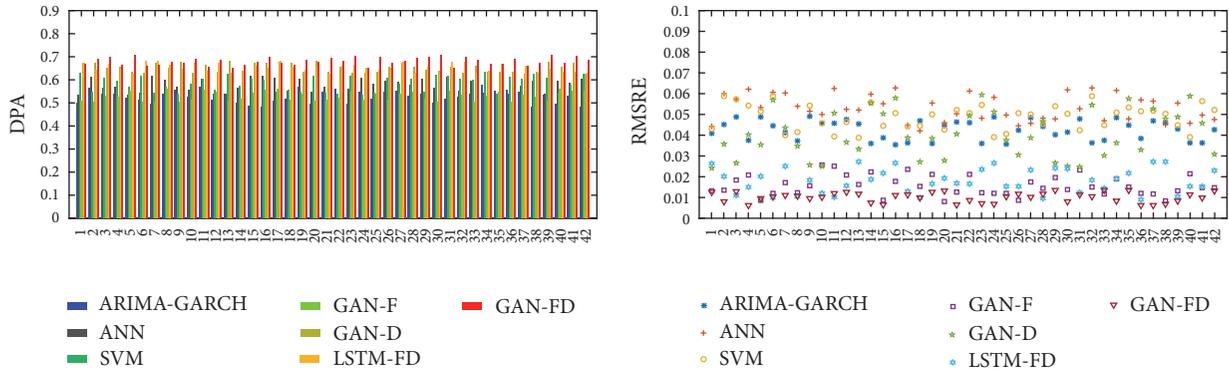


FIGURE 5: DPA and RMSRE of each stock when (M, N) is $(60, 5)$ and x -axis represents the stock ID.

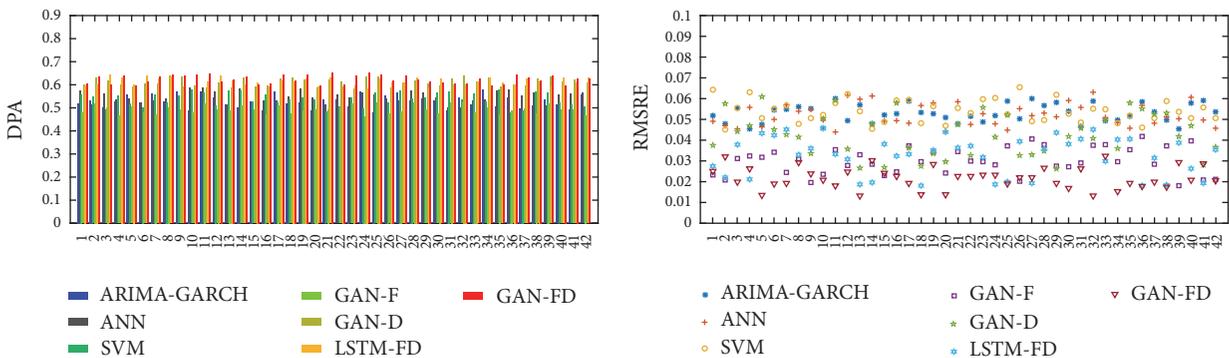


FIGURE 6: DPA and RMSRE of each stock when (M, N) is $(10, 10)$ and x -axis represents the stock ID.

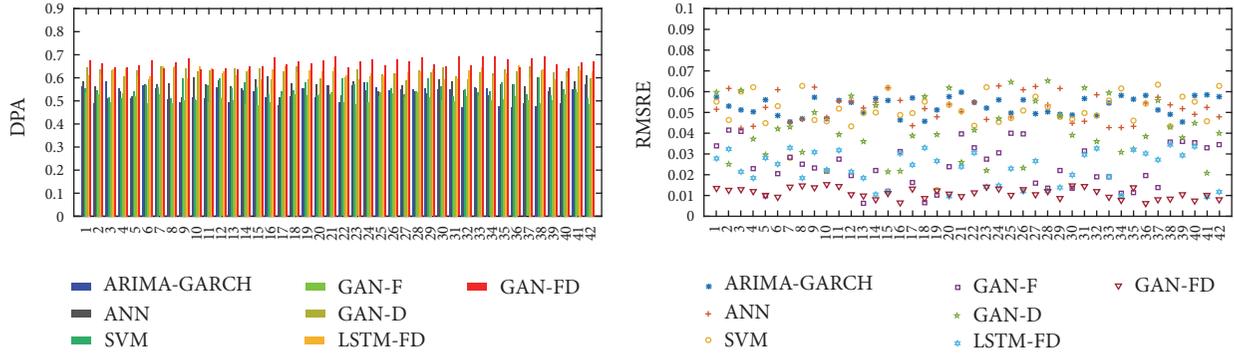


FIGURE 7: DPA and RMSRE of each stock when (M, N) is $(20, 10)$ and x -axis represents the stock ID.

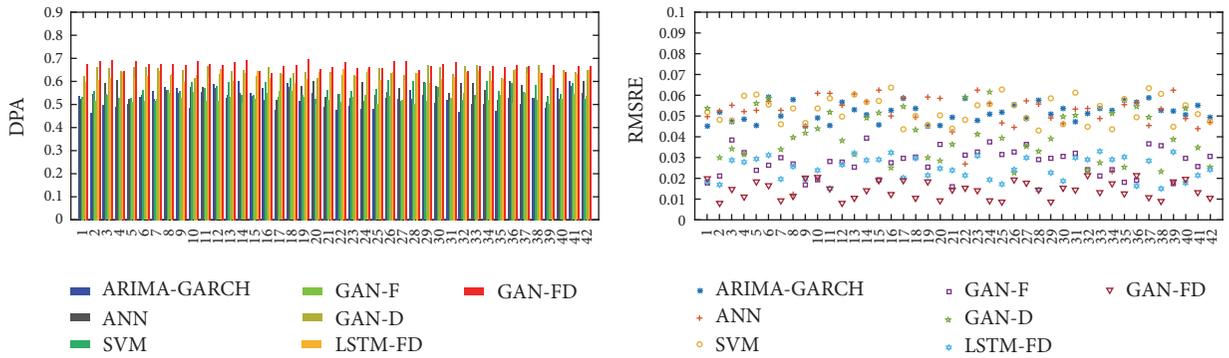


FIGURE 8: DPA and RMSRE of each stock when (M, N) is $(60, 10)$ and x -axis represents the stock ID.

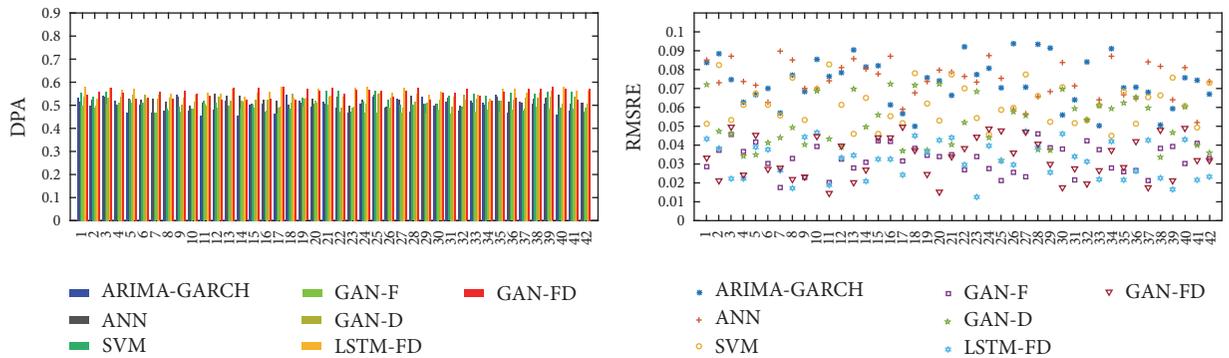


FIGURE 9: DPA and RMSRE of each stock when (M, N) is $(10, 20)$ and x -axis represents the stock ID.

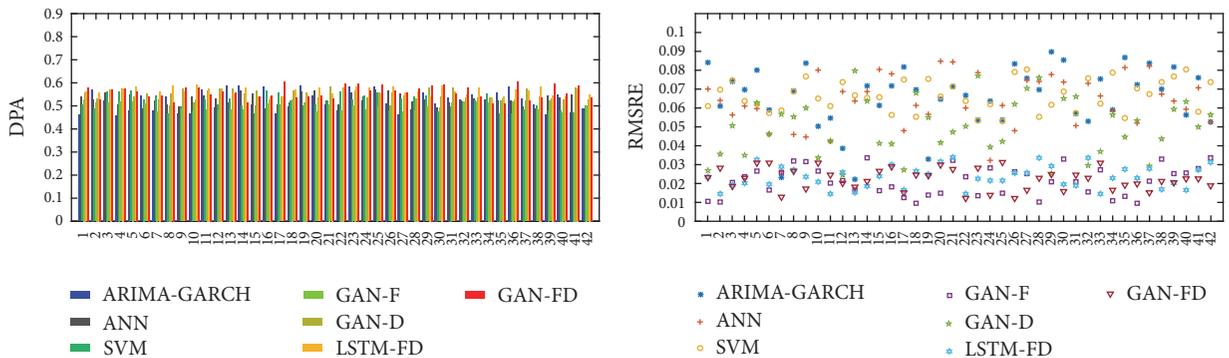


FIGURE 10: DPA and RMSRE of each stock when (M, N) is $(20, 20)$ and x -axis represents the stock ID.

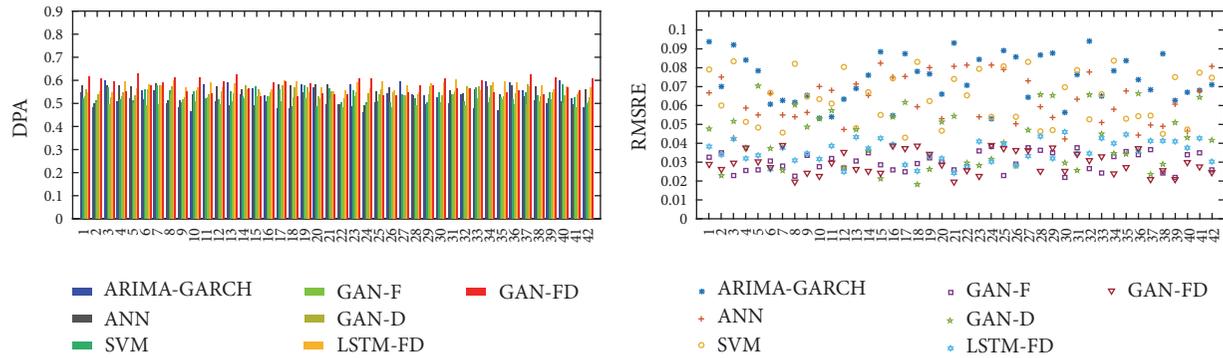


FIGURE 11: DPA and RMSRE of each stock when (M, N) is $(60, 20)$ and x -axis represents the stock ID.

which is lower than other six methods in most cases, while the volatility is smaller. However, the RMSRE of GAN-FD increases dramatically and fluctuates violently when N is 20, and it varies between 1.21% and 4.96%. This further shows that we should reduce the model update cycle N and revise the model parameters regularly to adapt to the change of market style.

5. Conclusion

In this paper, we propose an easy-to-use stock forecasting model called GAN-FD, to assist more and more nonfinancial professional ordinary investors making decisions. GAN-FD adopts 13 simple technical indexes as input data to avoid complicated input data preprocessing. Based on the deep learning network, this model achieves prediction ability superior to other benchmark methods by means of adversarial training, minimizing direction prediction loss, and forecast error loss. Moreover, the effects of the model update cycles on the predictive capability are analyzed, and the experimental results show that the smaller model update cycle can obtain better prediction performance. In the future, we will attempt to integrate predictive models under multiscale conditions.

Conflicts of Interest

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

Acknowledgments

This work is supported by the National Key Research Development Program of China (2017YFB0802800) and the National Natural Science Foundation of China (no. 61473149).

References

- [1] R. Al-Hmouz, W. Pedrycz, and A. Balamash, "Description and prediction of time series: a general framework of Granular Computing," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4830–4839, 2015.
- [2] S. Barak and M. Modarres, "Developing an approach to evaluate stocks by forecasting effective features with data mining methods," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1325–1339, 2015.
- [3] A. Booth, E. Gerding, and F. McGroarty, "Automated trading with performance weighted random forests and seasonality," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3651–3661, 2014.
- [4] A. Bagheri, H. Mohammadi Peyhani, and M. Akbari, "Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6235–6250, 2014.
- [5] Y. Son, D.-J. Noh, and J. Lee, "Forecasting trends of high-frequency KOSPI200 index data using learning classifiers," *Expert Systems with Applications*, vol. 39, no. 14, pp. 11607–11615, 2012.
- [6] I. Aldridge and S. Krawciw, "Real-Time Risk: What Investors Should Know About FinTech," in *High-Frequency Trading, and Flash Crashes*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2017.
- [7] F. A. De Oliveira, C. N. Nobre, and L. E. Zárate, "Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index - Case study of PETR4, Petrobras, Brazil," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7596–7606, 2013.
- [8] J. H. Niño-Peña and G. J. Hernández-Pérez, "Price direction prediction on high frequency data using deep belief networks," *Communications in Computer and Information Science*, vol. 657, pp. 74–83, 2016.
- [9] A. Marszałek and T. Burczynski, "Modeling and forecasting financial time series with ordered fuzzy candlesticks," *Information Sciences*, vol. 273, pp. 144–155, 2014.
- [10] X. Li, X. Huang, X. Deng, and S. Zhu, "Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information," *Neurocomputing*, vol. 142, pp. 228–238, 2014.
- [11] X. Wang, S. Bao, and J. Chen, "High-frequency stock linkage and multi-dimensional stationary processes," *Physica A: Statistical Mechanics and its Applications*, vol. 468, pp. 70–83, 2017.
- [12] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.
- [13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 28th Annual Conference*

- on *Neural Information Processing Systems 2014, NIPS 2014*, pp. 2672–2680, can, December 2014.
- [14] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” *ACM Transactions on Graphics*, vol. 36, no. 4, article no. 107, 2017.
- [15] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, *Semantic segmentation using adversarial networks*, *arXiv preprint*, arXiv, 1611.08408, 2016, arXiv:1611.08408.
- [16] M. Mathieu, C. Couprie, and Y. LeCun, *Deep multi-scale video prediction beyond mean square error*, *arXiv preprint*, arXiv, 1511.05440, 2015, arXiv:1511.05440.
- [17] J. D. Hamilton, *Time Series Analysis*, vol. 2, Princeton University Press, 1994.
- [18] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications*, Springer Texts in Statistics, Springer, New York, NY, USA, 3rd edition, 2011.
- [19] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, Springer Science & Business Media, 2013.
- [20] S. Pellegrini, E. Ruiz, and A. Espasa, “Prediction intervals in conditionally heteroscedastic time series with stochastic components,” *International Journal of Forecasting*, vol. 27, no. 2, pp. 308–319, 2011.
- [21] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, “Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange,” *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.
- [22] M. Ghiassi, J. Skinner, and D. Zimbra, “Twitter brand sentiment analysis: a hybrid system using n-gram analysis and dynamic artificial neural network,” *Expert Systems with Applications*, vol. 40, no. 16, pp. 6266–6282, 2013.
- [23] M. R. Hassan, “A combination of hidden Markov model and fuzzy model for stock market forecasting,” *Neurocomputing*, vol. 72, no. 16-18, pp. 3439–3446, 2009.
- [24] W. Huang, Y. Nakamori, and S.-Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Computers & Operations Research*, vol. 32, no. 10, pp. 2513–2522, 2005.
- [25] A. F. S. Elsir, and H. Faris, “A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 7, 2015.
- [26] R. Majhi, G. Panda, G. Sahoo, A. Panda, and A. Choubey, “Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique,” in *Proceedings of the Proceeding of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1276–1282, Hong Kong, China, June 2008.
- [27] S. S. Appadoo, *Pricing Financial Derivatives with Fuzzy Algebraic Models: A Theoretical and Computational Approach [Ph.D. thesis]*, University of Manitoba, Winnipeg, 2006.
- [28] A. Thavaneswaran, K. Thiagarajah, and S. S. Appadoo, “Fuzzy coefficient volatility (FCV) models with applications,” *Mathematical and Computer Modelling*, vol. 45, no. 7-8, pp. 777–786, 2007.
- [29] C. Carlsson and R. Fullér, “On possibilistic mean value and variance of fuzzy numbers,” *Fuzzy Sets and Systems*, vol. 122, no. 2, pp. 315–326, 2001.
- [30] A. Thavaneswaran, S. S. Appadoo, and A. Paseka, “Weighted possibilistic moments of fuzzy numbers with applications to GARCH modeling and option pricing,” *Mathematical and Computer Modelling*, vol. 49, no. 1-2, pp. 352–368, 2009.
- [31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [32] A. M. Rather, A. Agarwal, and V. N. Sastry, “Recurrent neural network and a hybrid model for prediction of stock returns,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.
- [33] R. D. A. Araújo, A. L. I. Oliveira, and S. Meira, “A hybrid model for high-frequency stock market forecasting,” *Expert Systems with Applications*, vol. 42, no. 8, pp. 4081–4096, 2015.
- [34] H. Chen, K. Xiao, J. Sun, and S. Wu, “A double-layer neural network framework for high-frequency forecasting,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 7, no. 4, article no. 11, 2017.
- [35] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, “Predicting Stock Market Trends by Recurrent Deep Neural Networks,” in *PRICAI 2014: Trends in Artificial Intelligence*, vol. 8862 of *Lecture Notes in Computer Science*, pp. 759–769, Springer International Publishing, Cham, 2014.
- [36] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” in *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, pp. 2327–2333, arg, July 2015.
- [37] Z. Zhou, J. Zhao, and K. Xu, “Can online emotions predict the stock market in China?” in *Proceedings of the International Conference on Web Information Systems Engineering*, pp. 328–342, 2016.
- [38] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [40] A. Arévalo, J. Niño, G. Hernández, and J. Sandoval, “High-frequency trading strategy based on deep neural networks,” in *Proceedings of the International conference on intelligent computing*, pp. 424–436, 2016.
- [41] K.-J. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
- [42] S. Madge, *Predicting Stock Price Direction using Support Vector Machines*, Independent Work Report Spring, 2015.

Research Article

Adaptive Learning Based Tracking Control of Marine Vessels with Prescribed Performance

Zhao Xu ¹, Shuzhi Sam Ge,² Changhua Hu ³ and Jinwen Hu⁴

¹*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi, China*

²*Department of Electrical and Computer Engineering, National University of Singapore, Singapore*

³*Xi'an Research Institute of High Technology, Xi'an, Shaanxi 710025, China*

⁴*School of Automation, Northwestern Polytechnical University, Xi'an, Shaanxi, China*

Correspondence should be addressed to Zhao Xu; zhaoxu@nwpu.edu.cn

Received 29 November 2017; Accepted 1 February 2018; Published 6 March 2018

Academic Editor: Qian Zhang

Copyright © 2018 Zhao Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel adaptive tracking controller of fully actuated marine vessels is proposed with completely unknown dynamics and external disturbances. The model of dominant dynamic behaviors and unknown disturbances of the vessel are learned by a neural network in real time. The controller is designed and it unifies backstepping and adaptive neural network techniques with predefined tracking performance constraints on the tracking convergence rate and the transient and steady-state tracking error. The stability of the proposed adaptive tracking controller of the vessel is proven with a uniformly bounded tracking error. The proposed adaptive tracking controller is shown to be effective in the tracking control of marine vessels by simulations.

1. Introduction

Tracking control of marine vessels has been widely used in civil and military applications such as shipping transportation, sea investigation, search and rescue, and security surveillance. Previous works on tracking control can be briefly classified into model-based and approximation-based approaches. Model-based methods require accurate dynamic models of the systems, which makes them difficult to be applied in reality due to the following reasons: (1) the dynamics of a vessel are generally time-varying and highly nonlinear and thus their actual mathematical model is hard to be obtained precisely using current modeling techniques; (2) vessels always suffer from unknown stochastic environmental disturbances.

Recently, adaptive NN control design techniques were presented for controlling marine vehicles with model uncertainties [1, 2], by which the dynamics of the vessels can be partially known or even completely unknown. In [3], a simple yet computationally efficient NN tracking control approach is presented for control of the fully actuated vessels. Reference [4] extends the control strategy in [3] to the underactuated vessels. Other approximation methods such as fuzzy tracking

control schemes are also proposed for marine vehicles. In [5], a fuzzy tracking controller is proposed against the model uncertainty of the vessel dynamics which can obtain with a uniformly bounded tracking error. However, one problem with the existing approximation-based control schemes is that it is of great challenge to guarantee the transient response of the vessel dynamics; for example, unsuitable initialization of the weight of NN may lead to poor transient tracking performance.

For the purpose of solving this problem, an efficient prescribed performance control method was proposed in [6, 7] to regularize the response of strict-feedback systems in advance. The main idea is to formulate the tracking control as a constrained optimization problem with predefined constraints on the transient and steady-state tracking errors. It is then transformed into a relatively easier unconstrained optimization problem by defining a transformation function that incorporates the errors and their constraints into a single function analytically. This technique has been applied in several applications [8–11]. In [12], an adaptive NN control method was proposed with guaranteed transient and steady-state tracking performances for marine surface vessels with model uncertainties. Radial basis function (RBF) NNs are

applied to approximate unknown vessel dynamics before control action applies with the assumptions of persistent excitation and recurrent orbits of the desired trajectory. However, the vessels always suffer from unknown environmental disturbances, and the estimated and stored model in memory may be violated during operating and the control performance cannot be guaranteed. Moreover, the persistent excitation condition was very restrictive making the method difficult to be applied in practice.

This paper addresses the tracking control of surface vessels with unknown system dynamics and stochastic disturbances. An adaptive NN controller is proposed to obtain desired tracking performances. First, the method can work with predefined convergence rate, maximum overshoot, and steady-state error as the desired tracking performance indices. Second, the adaptive NN control combined with backstepping technique is applied to generate control commands for tracking control with unknown model dynamics and external disturbances. Third, the stability of the control method is analyzed and the tracking error is proven to be uniformly bounded.

2. Problem Formulation

2.1. Dynamics of Surface Vessels. We consider the surge, sway, and yaw motions of the marine vessel. Let $\eta = [\eta_x, \eta_y, \eta_\psi]^T$ represent position (η_x, η_y) and heading (η_ψ) of the vessel. The vector $v = [v_x, v_y, r_\psi]^T$ represents the alongship, athwartship, and rotational velocity, respectively, in the body-fixed frame. The vessel dynamics is considered as follows:

$$\dot{\eta} = J(\eta) v \quad (1)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + \tau_d(\eta, v, t), \quad (2)$$

where

$$J(\eta) = \begin{bmatrix} \cos \psi_n & -\sin \psi_n & 0 \\ \sin \psi_n & \cos \psi_n & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

M is the diagonal inertia matrix, $C(v)$ is the matrix of Coriolis and centripetal, $D(v)$ is the damping matrix, $g(\eta)$ is the restoring force, τ is the vector of input signals, and τ_d is the time-varying unknown external disturbance and unmodeled dynamics. In this paper, we assume that M , $C(v)$, and $D(v)$ are completely unknown.

2.2. Prescribed Performance Control Preliminaries. The aim of this paper is to let the vessel position and heading in η track a smooth, bounded desired trajectory η_r with a uniformly bounded tracking error $\varepsilon(t) = \eta(t) - \eta_r(t)$. The predefined constraints of the tracking error $\varepsilon(t) = [\varepsilon_1(t), \varepsilon_2(t), \varepsilon_3(t)]^T \forall t \geq 0$ are given as follows:

$$\begin{aligned} -\kappa \varrho_i(t) < \varepsilon_i(t) < \varrho_i(t) & \quad \text{if } \varepsilon_i(0) \geq 0 \\ -\varrho_i(t) < \varepsilon_i(t) < \kappa \varrho_i(t) & \quad \text{if } \varepsilon_i(0) \leq 0, \end{aligned} \quad (4)$$

where $0 \leq \kappa \leq 1$ and $\varrho_i(t)$, $i = 1, 2, 3$, are called performance functions [13] which are smooth, bounded, and strictly positive and decreasing.

Thus, the constraints on the tracking error $\varepsilon_i(t)$ are defined by $\varrho_i(t)$ and κ . Given the following exponential performance function

$$\varrho_i(t) = (\varrho_{i0} - \varrho_{i\infty}) \varepsilon^{-l_i t} + \varrho_{i\infty}, \quad i = 1, 2, 3, \quad (5)$$

where $\varrho_{i0} = \varrho_i(0)$ and $\varrho_{i\infty} = \lim_{t \rightarrow \infty} \varrho_i(t)$, the constant ϱ_{i0} in (5) should be set such that (4) is satisfied at $t = 0$. l_i determines how fast $\varrho_i(t)$ decreases. Moreover, it is implied that the overshoot is required to be no larger than $\kappa \varrho_i(0)$.

The idea in [6, 7] of transforming the tracking error into a new function is borrowed here, by which the error constraints can be incorporated into the function analytically so that the controller optimization can be solved more efficiently. The error transformation is defined by

$$\xi_i = T_i \left(\frac{\varepsilon_i}{\varrho_i} \right), \quad i = 1, 2, 3, \quad (6)$$

where $T_i(\cdot)$, $i = 1, 2, 3$ are smooth, strictly increasing functions which define objective mappings:

$$T_i : (-\kappa, 1) \longrightarrow (-\infty, \infty) \quad \text{for } \varepsilon_i(0) \geq 0 \quad (7)$$

$$T_i : (-1, \kappa) \longrightarrow (-\infty, \infty) \quad \text{for } \varepsilon_i(0) \leq 0$$

for $i = 1, 2, 3$. Defining $\xi(t) = [\xi_1(t), \xi_2(t), \xi_3(t)]^T$, after differentiation we have

$$\dot{\xi} = T_\partial \left(\dot{\varepsilon} - \dot{\varrho} \frac{\varepsilon}{\varrho} \right), \quad (8)$$

where $T_\partial = \text{diag}([(dT_1/d(\varepsilon_1/\varrho_1))(1/\varrho_1) \cdots (dT_3/d(\varepsilon_3/\varrho_3))(1/\varrho_3)])$ and $\dot{\varrho}(\varepsilon/\varrho) = [\dot{\varrho}_1(\varepsilon_1/\varrho_1) \cdots \dot{\varrho}_3(\varepsilon_3/\varrho_3)]^T$.

According to [6], if $\xi_i(t)$ is able to be kept bounded $\forall t \geq 0$ with the transformation function $T_i(\cdot)$, $i = 1, 2, 3$, then (4) is guaranteed.

3. Adaptive Tracking Control Design

In this section, we will develop an adaptive tracking controller by combining the backstepping design technique and a RBF NN. The main advantage of RBF is local approximation and fast learning, which has been widely used in researches [14, 15].

Assume that full state information v and η of the vessel is available. Given the reference trajectory $\eta_r = [\eta_{xr}, \eta_{yr}, \eta_{\psi r}]^T$ and the tracking error

$$\varepsilon(t) = \eta(t) - \eta_r(t), \quad (9)$$

we can get

$$\dot{\varepsilon}(t) = J(\eta)v - \dot{\eta}_r(t). \quad (10)$$

The error is transformed based on (6) and (8) as follows:

$$\begin{aligned}\dot{\xi} &= T_{\partial} \left(\dot{\varepsilon} - \dot{\varrho} \frac{\varepsilon}{\varrho} \right) = T_{\partial} \left(\dot{\eta}(t) - \dot{\eta}_r(t) - \dot{\varrho} \frac{\varepsilon}{\varrho} \right) \\ &= T_{\partial} \dot{\eta}(t) - T_{\partial} r = T_{\partial} J v - T_{\partial} r,\end{aligned}\quad (11)$$

where $r = \dot{\eta}_r(t) + \dot{\varrho}(\varepsilon/\varrho)$.

According to [6, 13], the boundedness of the solution of (11) guarantees the predefined tracking performance of $\varepsilon(t)$ introduced via (4) for all $t \geq 0$. We introduce a virtual control based on the transformed error

$$\phi_1 = J^T r - K_1 (T_{\partial} J)^T \xi \quad (12)$$

with $K_1 = K_1^T > 0$ and $\dot{\phi}_1 = (\partial\phi_1/\partial\eta)\dot{\eta} + (\partial\phi_1/\partial\dot{\eta}_r)\dot{\eta}_r + (\partial\phi_1/\partial\xi)\dot{\xi}$. Define a second error variable in the transformed space as

$$\xi_a = v(t) - \phi_1(t). \quad (13)$$

Assume that exogenous effects and uncertainties have finite energy; we have $|\tau_{di}(\eta, v, t)| \leq m_i(\eta, v) + n_i(t)$, $i = 1, 2, 3$, where $m_i(\eta, v) : R^3 \times R^3 \rightarrow R^+$ is a positive, smooth, and nondecreasing function and $n_i(t)$ is a positive scalar. Consider the following desired control law:

$$\begin{aligned}\bar{\tau} &= -(T_{\partial} J)^T \xi - K_2 \xi_a + C(v) v + D(v) v + g(\eta) \\ &\quad + M\dot{\phi}_1 - \text{SGN}(\xi_a)(m(\eta, v) + n_i),\end{aligned}\quad (14)$$

where $\text{SGN}(\xi_a)$ is the diagonal matrix of signum functions $\text{sgn}(\xi_{a,i})$, $i = 1, 2, 3$, $K_2 = K_2^T > 0$, and $K_2 = \text{diag}(k_{ii}) \in R^{3 \times 3}$.

Since the dynamics and disturbance of the vessel are completely unknown, a RBF NN $\widehat{W}^T S(\xi)$ is used to approximate the vessel model and effect of the disturbance given by

$$\begin{aligned}\widehat{W}^T S(\xi) &= M\dot{\phi}_1 + C(v) v + D(v) v + g(\eta) \\ &\quad - \text{SGN}(\xi_a)(m(\eta, v) + n_i) - \varepsilon(\xi),\end{aligned}\quad (15)$$

where $\widehat{W} := \text{blockdiag}[\widehat{W}_1^T, \widehat{W}_2^T, \widehat{W}_3^T]$ are the estimation of the desired weight W^* , $S(\xi) = [S_1^T(\xi), S_2^T(\xi), S_3^T(\xi)]^T$ are the basis functions, $\xi = [\eta^T, v^T, \phi_1^T, \dot{\phi}_1^T]^T$ are the inputs of the NN, and $\varepsilon(\xi)$ is the approximation error. Thus, the control law and weight update law are defined as follows:

$$\tau = -(T_{\partial} J)^T \xi - K_2 \xi_a + \widehat{W}^T S(Z) \quad (16)$$

$$\dot{\widehat{W}}_i = -\Upsilon_i (S_i(\xi) \xi_{a,i} + \mu_i \widehat{W}_i), \quad (17)$$

where Υ_i are constant matrix and $\mu_i > 0$, $i = 1, 2, 3$, and $Z = [\eta^T, v^T, \xi^T, \dot{\xi}^T]^T$ are the inputs of NN.

Theorem 1. Consider the system consisting of vessel model (2), desired bounded trajectory η_d , the control law (16), and NN weight adaptation law (17). Given the performance functions $\varrho_i(t)$, $i = 1, 2, 3$, and constant κ satisfying $0 \leq \kappa \leq 1$,

TABLE 1: Parameters of the target vessel.

Parameter	Description	Value
m	Total mass	23.8 kg
L_v	Length of vessel	1.255 m
B_v	Breadth of vessel	0.290 m
z_g	Position along Z of the center of gravity	0.046 m
I_{vz}	Moments of inertia along Z	1.76 kg·m ²
ρ_{air}	Density of air	1.29 kg/m ³
ρ_{water}	Density of water	1025 kg/m ³
g_v	Gravitational acceleration	9.8 m/s ²

$i = 1, 2, 3$, which incorporate desired performance bounds on tracking errors $\varepsilon_i(t)$, $i = 1, 2, 3$, if initial conditions satisfy

$$\begin{aligned}|\varepsilon_i(0)| &< \varrho_i(0) \quad \text{if } \varepsilon_i(0) \neq 0 \\ \kappa &\neq 0, \quad \text{if } \varepsilon_i(0) = 0\end{aligned}\quad (18)$$

with transformation functions

$$\begin{aligned}T_i \left(\frac{\varepsilon_i(t)}{\varrho_i(t)} \right) &= \begin{cases} a_i \ln \left(\frac{\kappa + \varepsilon_i(t)/\varrho_i(t)}{1 - \varepsilon_i(t)/\varrho_i(t)} \right), & \text{for } \varepsilon_i(0) \geq 0 \\ a_i \ln \left(\frac{1 + \varepsilon_i(t)/\varrho_i(t)}{\kappa - \varepsilon_i(t)/\varrho_i(t)} \right), & \text{for } \varepsilon_i(0) \leq 0, \end{cases}\end{aligned}\quad (19)$$

where $a_i > 0$, $i = 1, 2, 3$, are positive design constants, then the following properties hold:

- (1) The tracking error of the vessel is uniformly bounded.
- (2) The predefined tracking performance of the vessel in the sense of (4) and (5) is guaranteed.

Proof. See the appendix. \square

4. Simulations

In this section, the parameters of a supply vessel from Norwegian University of Science and Technology [16] are adopted for simulation with a 1:70 scaling to test the performance of the proposed adaptive NN controller. The vessel parameters used are given in Table 1.

The objective of this work is to make the vessel track the given smooth, bounded reference trajectory with predefined bounds for tracking errors. In this section, two simulation scenarios with different desired trajectories are tested. The tracking performance of the proposed controller is also evaluated under two conditions: with the predefined performance bounds and without the predefined performance bounds.

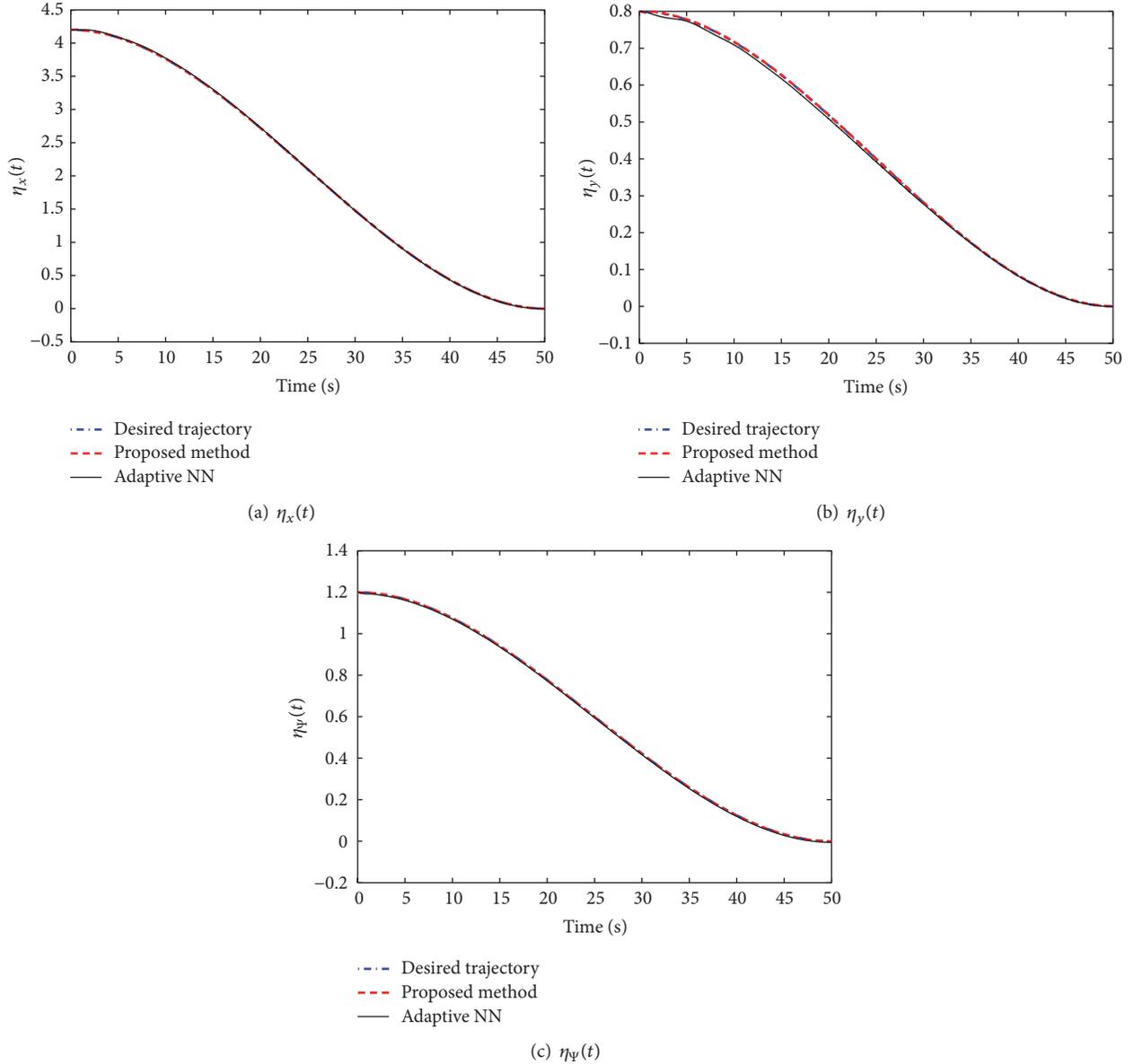


FIGURE 1: Comparison of tracking performance.

4.1. Scenario I. In this scenario, the reference trajectory is given as follows:

$$\eta_r(t, t_r) = \eta_0 + \left(-2.0 \frac{t^3}{t_r^3} + 3.0 \frac{t^2}{t_r^2} \right) (\eta_f - \eta_0), \quad (20)$$

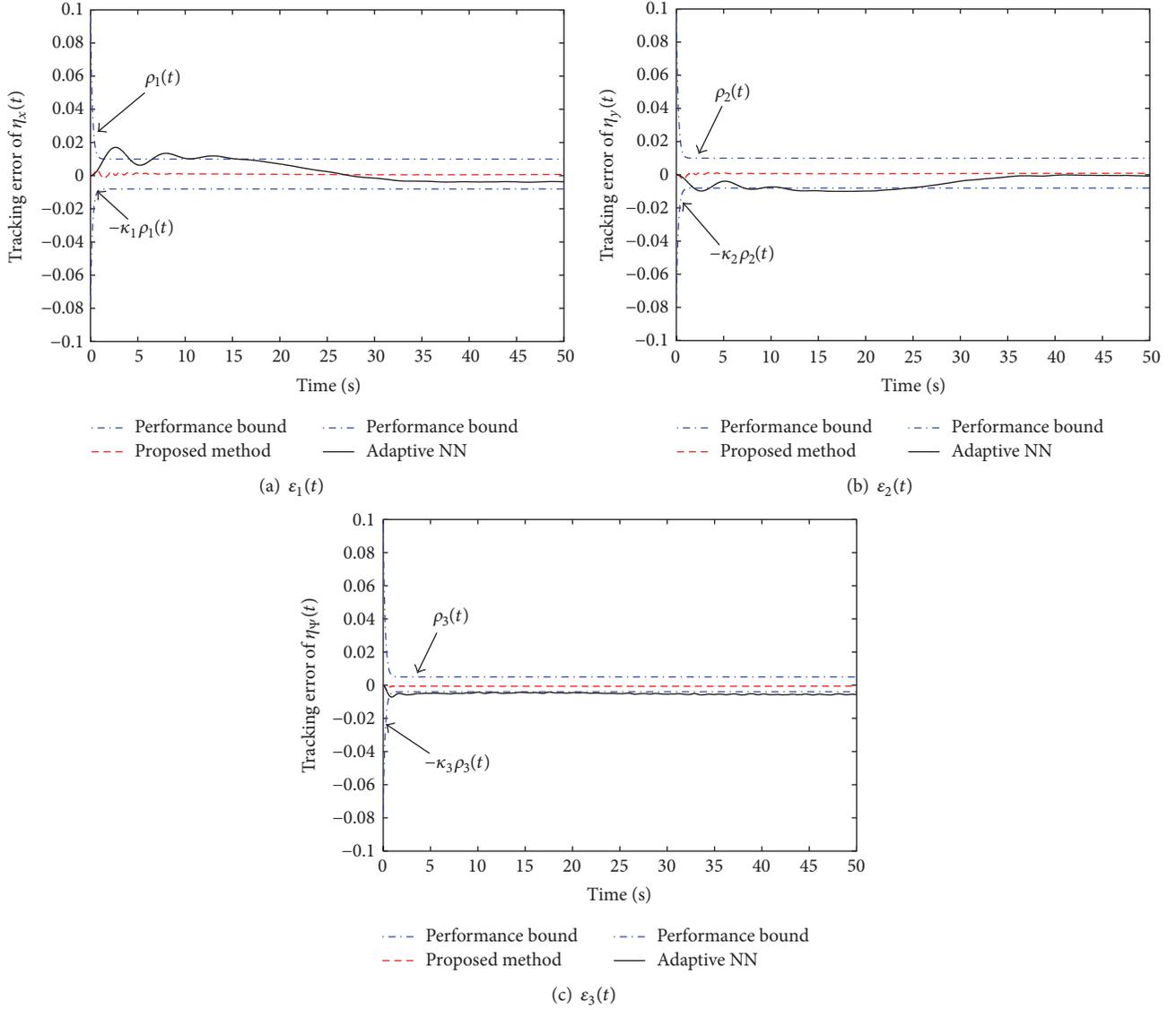
where $\eta_0 = [4.2, 0.8, 1.2]^T$ is the initial of the vessel, $\eta_f = [0, 0, 0]^T$ is the final position, and $t_r = 50$ s represents the stopping time.

The parameters κ , $i = 1, 2, 3$ appearing in (4) are set to be 0.8. The parameters in exponential performance functions (5) are $q_{i0} = 0.1$, $q_{i\infty} = 0.01$, and $l_i = 5$, $i = 1, 2, 3$. Furthermore, the parameters in error transformations function (19) are chosen as $a_i = 0.008$, $i = 1, 2, 3$.

The inputs of NNs are $Z = [\eta^T, v^T, \xi^T, \dot{\xi}^T]$. The number of nodes is $l = 512$. The centers of the nodes are

evenly distributed in $[-2, 2]$, $[-2, 2]$, $[-\pi, \pi]$, $[-0.1, 0.1]$, $[-3, 3]$, $[-3, 3]$. The variance of center of Gaussian radial basis function is 5. The initial weights are zeros. The control gain matrices in (17) are $Y = \text{diag}[0.001, 0.001, 0.0001]$ and $\mu = \text{diag}[0.001, 0.001, 0.0001]$. The Lyapunov gain matrices are tuned as $K_1 = K_2 = \text{diag}[20, 20, 20]$.

For comparative studies, we compare our method with the adaptive NN control without prescribed performance bound. Figure 1 shows the positions and heading evolution. The tracking errors are illustrated in Figure 2. It is observed that the proposed controller has much better transient and steady-state performance while the tracking errors are within the predefined bounds as compared to the one without considering the predefined tracking error bounds. The norm of the tracking error is shown in Figure 3(a). From Figure 3(b), we can see that, during transient stage, the control


 FIGURE 2: Tracking errors $\varepsilon(t)$ with respect to performance envelopes.

input of the controller without predefined error bounds oscillates seriously and requires a larger control effort, which results in larger overshoots and oscillations of the tracking performances as seen in Figure 2, whereas the control input of our proposed method is smooth. In addition, the proposed controller with predefined error bounds converges faster and the tracking errors stay strictly within the predefined bounds, while the one without considering the performance bounds cannot guarantee this.

4.2. Scenario II. In this scenario, the desired trajectory η_r is given by

$$\eta_r = \begin{bmatrix} 3 \cos(0.06t + 0.63) \\ 2 \sin(0.09t) \\ \pi + \pi \sin(0.13\pi t - 0.52) \end{bmatrix}, \quad (21)$$

where $\eta_0 = [2, 3, 2]^T$ is the payload initial position and $t_r = 200$ s represents the stopping time.

The parameters κ , $i = 1, 2, 3$ appearing in (4) are 0.7. The parameters in exponential performance functions (5) are $\varrho_0 = [2.2, 1.2, 0.63]^T$, $\varrho_{i\infty} = 0.01$, and $l_i = 0.45$, $i = 1, 2, 3$. Furthermore, the parameters in error transformations function (19) are chosen as $a_i = 0.01$, $i = 1, 2, 3$.

The inputs of NNs are $Z = [\eta^T, v^T, \xi^T, \dot{\xi}^T]$. The number of nodes is $l = 512$. The centers of the nodes are distributed in $[-2, 2]$, $[-2, 2]$, $[-\pi, \pi]$, $[-0.1, 0.1]$, $[-3, 3]$, $[-3, 3]$. The variance of center of Gaussian radial basis function is 5. The initial weights are zeros. The control gain matrices Υ and μ in (17) are chosen as $\Upsilon = \text{diag}[0.001, 0.001, 0.0001]$ and $\mu = \text{diag}[0.001, 0.001, 0.0001]$. The Lyapunov gain matrices are tuned as $K_1 = K_2 = \text{diag}[50, 50, 50]$.

For comparative studies, we compare our controller to the one without considering the performance bounds.

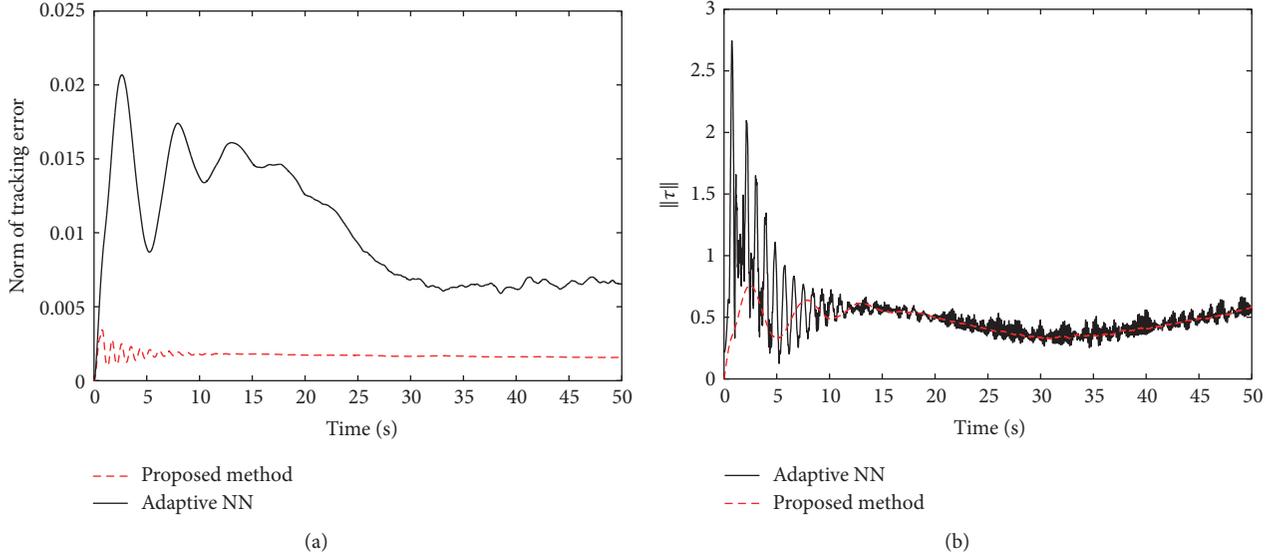
FIGURE 3: (a) $\|\varepsilon(t)\|$ and (b) $\|\tau\|$.

Figure 4 shows the positions and heading evolution; Figure 5 shows the tracking errors and Figure 6(a) shows the norm of the error vector. It can be seen that the proposed controller guarantees the tracking errors which are within the predefined bounds and produces better transient and steady-state performance as compared to the one without considering the performance bounds. From Figure 6(b), we can see that, during transient stage, the control input of the adaptive controller without considering the performance bounds oscillates seriously and requires a larger control effort compared with our proposed method, which results in larger overshoots and oscillations of the tracking performances as seen in Figure 5. Therefore, the proposed controller performs better than that without considering the performance bounds in both scenarios.

5. Conclusions

In this paper, the tracking control of the marine surface vessel is studied and a novel tracking controller with online modeling and predefined performance bounds was developed. The proposed control scheme can work with completely unknown vessel dynamic and unknown disturbances. Moreover, the tracking error was proven to be uniformly bounded. Simulation shows that the resulting tracking error stays strictly in the predefined bounds and the proposed controller performs better than that one without considering the predefined performance bounds.

Appendix

Proof of Theorem 1

Choose a Lyapunov function candidate with quadratic ξ_1 ,

$$V_1 = \frac{1}{2} \xi^T \xi. \quad (\text{A.1})$$

According to (1), (11), (13), and $J(\eta)J^T(\eta) = I$, we obtain

$$\dot{\xi} = T_\partial J (\xi_a + \phi_1(t)) - T_\partial r, \quad (\text{A.2})$$

$$\dot{V}_1 = \xi^T (T_\partial J \phi_1 - T_\partial r) + \xi^T T_\partial J \xi_a. \quad (\text{A.3})$$

Consider the first term on the right-hand side of (A.3) and multiply both sides of (12) by J :

$$\begin{aligned} \xi^T (T_\partial J \phi_1 - T_\partial r) &= \xi^T T_\partial (J \phi_1 - r) \\ &= -\xi^T T_\partial J K_1 (T_\partial J)^T \xi. \end{aligned} \quad (\text{A.4})$$

Let $\omega = (T_\partial J)^T \xi$; the above equation can be written as $\xi^T (T_\partial J \phi_1 - T_\partial r) = -\omega^T K_1 \omega \leq 0$. And (A.3) can be rewritten as

$$\dot{V}_1 = -\omega^T K_1 \omega + \xi^T T_\partial J \xi_a. \quad (\text{A.5})$$

Since $T_\partial J$ is invertible, $\omega = (T_\partial J)^T \xi = 0$ implies that $\omega = 0$. Therefore, $\xi^T (T_\partial J \phi_1 - T_\partial r) = 0$ if and only if $\omega = 0$.

According to (13) and (2), differentiating ξ_a , we have

$$\begin{aligned} \dot{\xi}_a &= M^{-1} (-C(v)v - D(v)v - g(\eta) + \tau + \tau_d(\eta, v, t)) \\ &\quad - \dot{\phi}_1. \end{aligned} \quad (\text{A.6})$$

Consider the following Lyapunov function candidate:

$$\bar{V} = V_1 + \frac{1}{2} \xi_a^T M \xi_a. \quad (\text{A.7})$$

From (A.5) and (A.6), we have

$$\begin{aligned} \dot{\bar{V}} &\leq -\omega^T K_1 \omega + \xi^T T_\partial J \xi_a \\ &\quad + \xi_a^T (-C(v)v - D(v)v - g(\eta) - M\dot{\phi}_1 + \tau) \\ &\quad + \sum_{i=1}^3 |\xi_{2,i}| (m_i(\eta, v) + n_i). \end{aligned} \quad (\text{A.8})$$

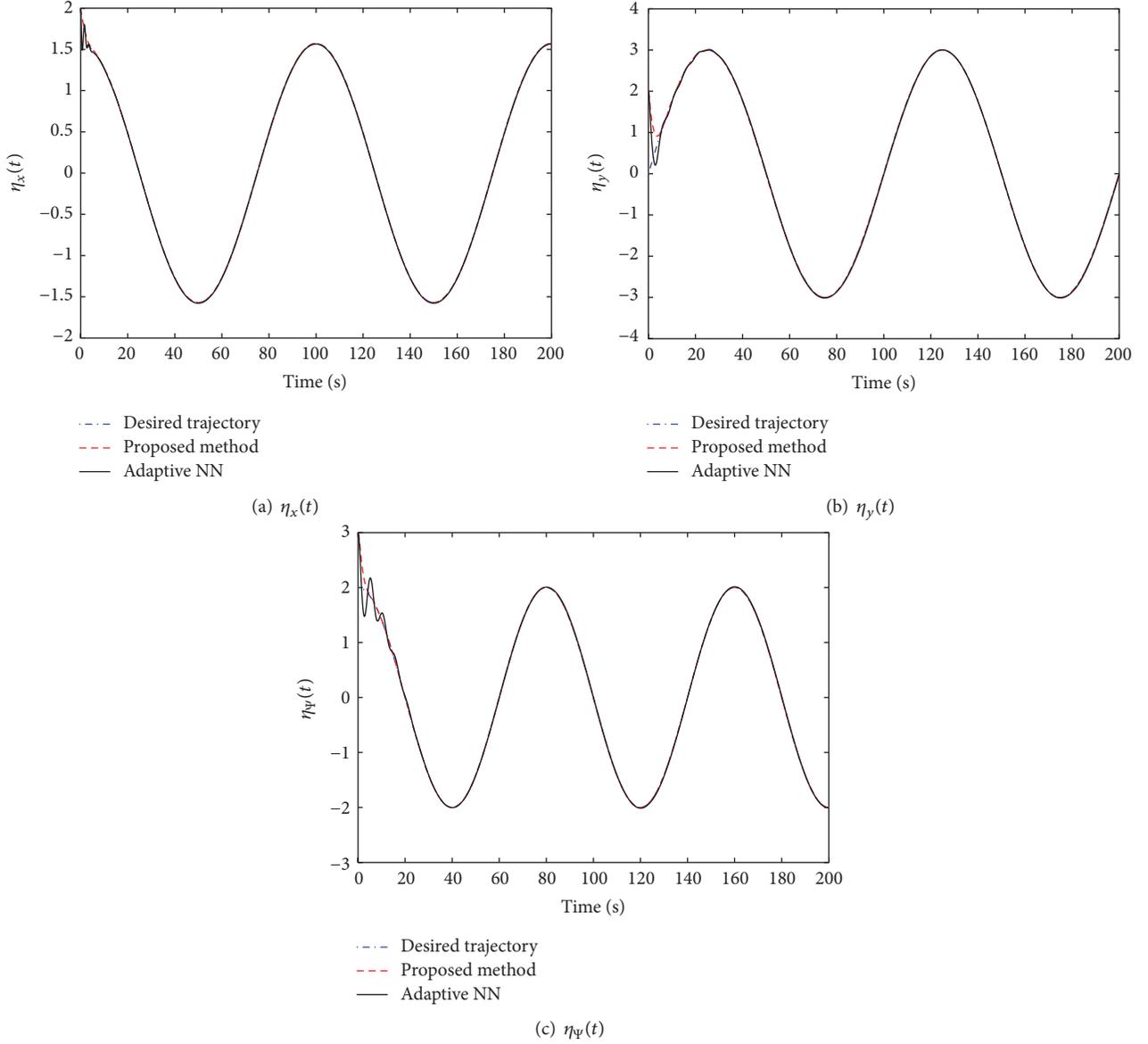


FIGURE 4: Comparison of tracking performance.

Substituting (14) into (A.8), the latter can be rewritten as $\dot{\bar{V}} \leq -\omega^T k \omega - \xi_a^T K_2 \xi_a$. Consider the augmented Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2} \xi_a^T M \xi_a + \sum_{i=1}^3 \bar{W}_i^T \Upsilon_i^T \bar{W}_i^T, \quad (\text{A.9})$$

where $\bar{W}_i = W_i - W_i^*$, W_i is NN estimated weight, W_i^* is the desired weight, and \bar{W}_i is weight estimation error. Differentiating (A.9), we obtain

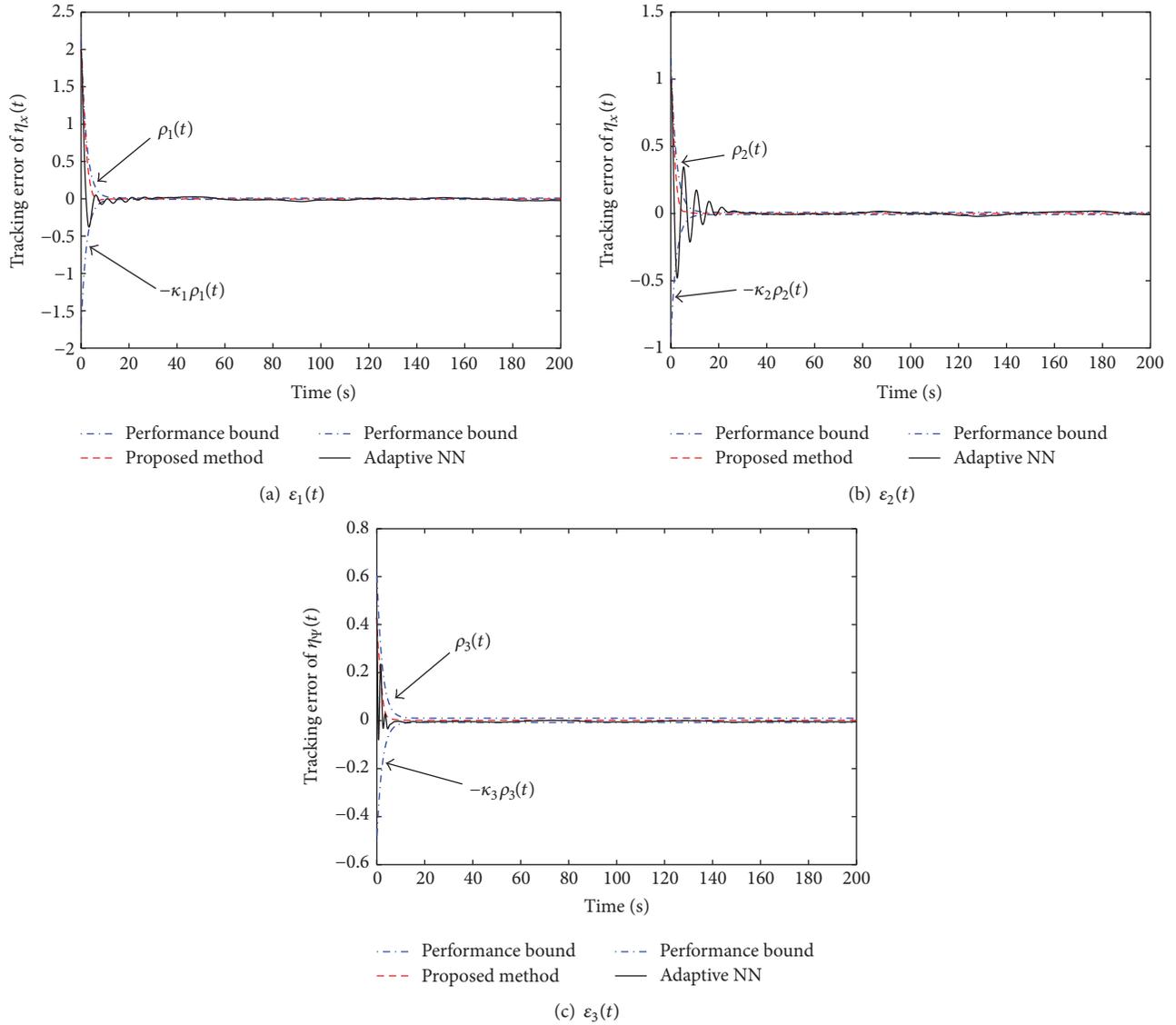
$$\begin{aligned} \dot{V}_2 \leq & -\omega^T K_1 \omega + \xi^T T_\partial J \xi_a + \xi_a^T \left(-C(v) v - D(v) v \right. \\ & \left. + \sum_{i=1}^3 \bar{W}_i^T \Upsilon_i^T \bar{W}_i^T - g(\eta) - M \phi_1 + \tau \right) \\ & + \sum_{i=1}^3 |\xi_{2,i}| (p_i(\eta, v) + q_i) + \sum_{i=1}^3 \mu_i \bar{W}_i^T \bar{W}_i. \end{aligned} \quad (\text{A.10})$$

Using approximation (16), we obtain

$$\begin{aligned} \dot{V}_2 \leq & -\omega^T K_1 \omega + \xi^T T_\partial J \xi_a \\ & + \xi_a^T \left[-W^{*T} S(Z) - \epsilon(Z) + \tau \right] \\ & + \sum_{i=1}^3 \bar{W}_i^T \Upsilon_i^{-1} \dot{\bar{W}}_i. \end{aligned} \quad (\text{A.11})$$

Substituting the control law (16) and the weight update law (17), (A.11) is rewritten as

$$\begin{aligned} \dot{V}_2 \leq & -\omega^T K_1 \omega - \xi_a^T K_2 \xi_a + \frac{1}{2} \xi_a^T \xi_a + \frac{1}{2} \|\epsilon\|^2 \\ & - \sum_{i=1}^3 \mu_i \bar{W}_i^T \bar{W}_i \end{aligned}$$

FIGURE 5: Tracking errors $\varepsilon(t)$ with respect to performance envelopes.

$$\begin{aligned} &\leq -\xi^T (T_\partial J) K_1 (T_\partial J)^T \xi - \xi_a^T K_2 \xi_a + \frac{1}{2} \xi_a^T \xi_a \\ &\quad + \frac{1}{2} \|\varepsilon\|^2 - \sum_{i=1}^3 \frac{\mu_i}{2} \|\widehat{W}_i\|^2 + \sum_{i=1}^3 \frac{\mu_i}{2} \|W_i^*\|^2, \end{aligned} \quad (\text{A.12})$$

$$\dot{V}_2 < -\beta V_2 + \delta, \quad (\text{A.13})$$

where

$$\begin{aligned} \beta &= \left\{ 2\lambda_{\min}((T_\partial J) K_1 (T_\partial J)^T), \right. \\ &\quad \left. \frac{2\lambda_{\min}(K_2 - (1/2)I_{3 \times 3})}{\lambda_{\max}(M)}, \min_{i=1,2,3} \left(\frac{\mu_i}{\lambda_{\max}(Y_i^{-1})} \right) \right\} \quad (\text{A.14}) \end{aligned}$$

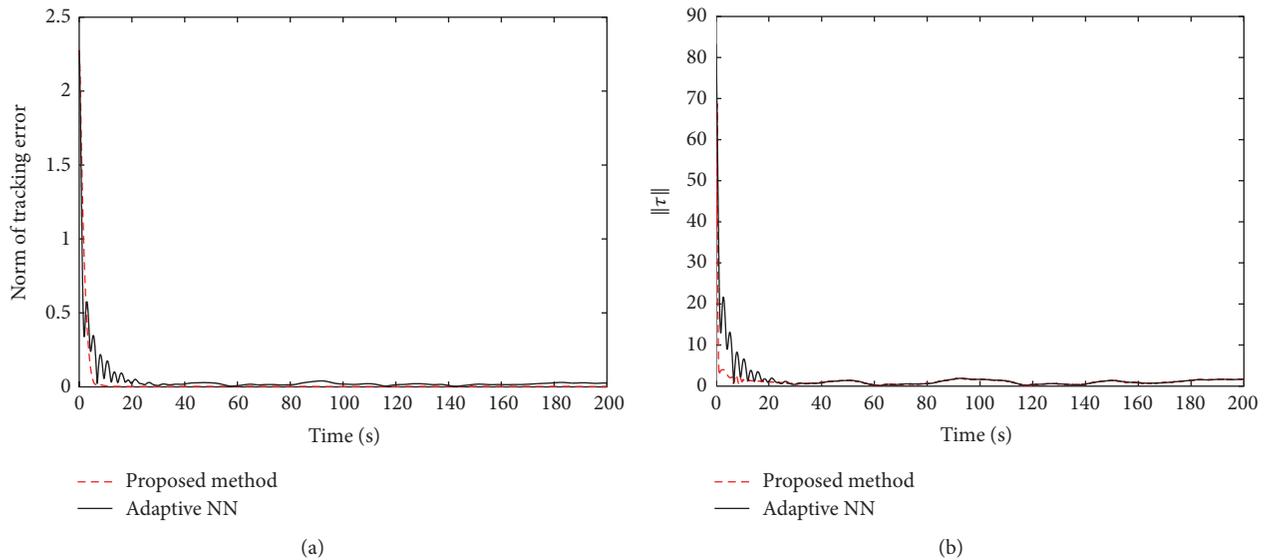
$$\delta = \frac{1}{2} \|\varepsilon\|^2 + \sum_{i=1}^3 \frac{\mu_i}{2} \|W_i^*\|^2,$$

where $\lambda_{\min}(\cdot)$ represents the minimum eigenvalue and $\lambda_{\max}(\cdot)$ is the maximum eigenvalue. The minimum eigenvalues of K_1 and $(K_2 - (1/2)I_{3 \times 3})$ should be positive in order to make $\beta > 0$.

According to (A.13) and Lemma 1.2 in [17], it is straightforward to show that the signals ξ_1 , ξ_2 , \widehat{W}_1 , \widehat{W}_2 , and \widehat{W}_3 are uniformly bounded. Since W_i^* is a constant, \widehat{W}_i is also bounded, for $i = 1, 2, 3$. T_∂ is bounded by construction. Consequently, the control signal τ in (16) is bounded. The transformed error ξ_1 is proven to be bounded. The prescribed performance (4) of the tracking control system (2) is achieved. The boundedness of the solution of (8) guarantees the predefined performance bounds for the tracking error $e(t)$ introduced via (4) for all $t \geq 0$ [6, 13].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

FIGURE 6: (a) $\|\varepsilon(t)\|$ and (b) $\|\tau\|$.

References

- [1] W. He, Z. Yin, and C. Sun, "Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function," *IEEE Transactions on Cybernetics*, 2016.
- [2] R. Hernández-Alvarado, L. G. García-Valdovinos, and T. Salgado-Jiménez, "Neural network-based self-tuning PID control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, 2016.
- [3] C.-Z. Pan, X.-Z. Lai, S. X. Yang, and M. Wu, "An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1629–1635, 2013.
- [4] C.-Z. Pan, X.-Z. Lai, S. X. Yang, and M. Wu, "A biologically inspired approach to tracking control of underactuated surface vessels subject to unknown dynamics," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2153–2161, 2015.
- [5] N. Wang and M. J. Er, "Direct adaptive fuzzy tracking control of marine vehicles with fully unknown parametric dynamics and uncertainties," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1845–1852, 2016.
- [6] C. P. Bechlioulis and G. A. Rovithakis, "Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems," *Automatica*, vol. 45, no. 2, pp. 532–538, 2009.
- [7] C. P. Bechlioulis and G. A. Rovithakis, "Prescribed performance adaptive control for multi-input multi-output affine in the control nonlinear systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 55, no. 5, pp. 1220–1226, 2010.
- [8] B. W. Kim and B. S. Park, "Robust control for the Segway with unknown control coefficient and model uncertainties," *Sensors*, vol. 16, no. 7, article no. 1000, 2016.
- [9] J. Na, Q. Chen, X. Ren, and Y. Guo, "Adaptive prescribed performance motion control of servo mechanisms with friction compensation," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 486–494, 2014.
- [10] E. Psomopoulou, A. Theodorakopoulos, Z. Doulgeri, and G. A. Rovithakis, "Prescribed Performance Tracking of a Variable Stiffness Actuated Robot," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1914–1926, 2015.
- [11] X. Li and C. C. Cheah, "Adaptive neural network control of robot based on a unified objective bound," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1032–1043, 2014.
- [12] S.-L. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1717–1727, 2016.
- [13] C. P. Bechlioulis, Z. Doulgeri, and G. A. Rovithakis, "Guaranteeing prescribed performance and contact maintenance via an approximation free robot force/position controller," *Automatica*, vol. 48, no. 2, pp. 360–365, 2012.
- [14] Z. Zhang, C. Ma, and R. Zhu, "Self-tuning fully-connected PID neural network system for distributed temperature sensing and control of instrument with multi-modules," *Sensors*, vol. 16, no. 10, article 1709, 2016.
- [15] W. He, S. S. Ge, Y. Li, E. Chew, and Y. S. Ng, "Neural network control of a rehabilitation robot by state and output feedback," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 15–31, 2015.
- [16] T. I. Fossen and J. P. Strand, "Adaptive maneuvering with experiments for a model ship in a marine control laboratory," *Automatica*, vol. 41, no. 2, pp. 289–298, 2005.
- [17] S. S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 15, no. 3, pp. 674–692, 2004.

Research Article

The Application of Group Theory in Communication Operation Pipeline System

Jianqing Zhang,¹ Feng Xiong ,¹ and Jinsheng Kang ²

¹School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200070, China

²College of Engineering, Design and Physical Sciences, Brunel University London, London UB8 3PH, UK

Correspondence should be addressed to Feng Xiong; xfeng@mail.shu.edu.cn and Jinsheng Kang; jinsheng.kang@brunel.ac.uk

Received 27 October 2017; Accepted 22 January 2018; Published 5 March 2018

Academic Editor: Qian Zhang

Copyright © 2018 Jianqing Zhang 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.

To resolve the “pipeline” crisis for telecom operators, this study pioneers the application of Group theory in communication operation pipeline system. The pipeline entity group model was built for information transmission in the pipeline system to analyze operation of pipeline entities. The equations of pipeline system network traffic were established according to the flux conservation principle and matrix of pipeline network. Based on pipeline entity group model, dimensionality of the matrix was reduced. The solution scheme of the flow state transition relationship of the pipeline system is obtained, which will be very useful for the telecom operators to construct high-level mobile e-commerce application model and architecture.

1. Introduction

With the advancement of the subversive power of mobile Internet, the new generation of information technology, such as cloud computing and Internet of things, inspired the imagination of enormous participants on the industry chain and led to a new round of information revolution in recent years. Particularly, Over-the-Top (OTT) [1–3] impacts telecom industry as a main participant of the information technology reform. Over-the-Top (OTT) refers to the phenomenon that downstream third parties of mobile Internet business chain offer a variety of value-added telecom applications and services as an alternative to basic telecom service providers in the mobile Internet era.

To avoid the risk of marginalization and channelization, telecom service providers have to improve and guarantee the Quality of Service (QoS) in key business by constructing fine and delicacy management network and upgrading operation platform [4]. Quality of Service (QoS) can be often quantitatively measured by error rates, throughput, transmission delay, and so on [5, 6]. Some research have been conducted to tackle this issue; for example, smart pipeline with visual, sustainable returns was proposed as a key to the telecommunication business of efficient and safe data

transmission capacity [7, 8]. Unfortunately, the proposed smart pipeline only redefined the system from two aspects of data transmission and data analysis without optimizing data traffic in distribution and transmission mechanism. Kashif et al. proposed a pipeline resource model associated with stage-level analysis to estimate the worst-case communication latency [9], and it was only for the improvement of scheduling based on real-time analysis. Markov Decision Process (MDP) model was also used to optimize the network and improve QoS in time domain dynamically [10–13], but the established model often meets “state space explosion” problems; namely, the object management problems caused by the exponential growth of object type and quantity in the computer system and computer network make Markov Decision Process (MDP) model not able to complete the calculation accurately.

Group theory has been used widely in the field of physics and chemistry as a kind of abstract mathematical tools [14, 15]. Group theory has also been used in engineering application, for example, to reduce the multiplicative computational complexity in the creation of digital holograms [16]. In the medical image processing field, the computation time is reduced considerably by exploiting symmetry presented in breast models using Group theory in order to detect breast cancer [17].

Although Group theory has been used in many fields, it has never been applied to communication network before. Based on our previous work on the three-dimensional pipeline system [18] of a telecom provider in Shanghai, this study pioneers the application of Group theory on the one-dimensional pipeline content, which is the core of service quality, namely, Quality of Service (QoS). First of all, the pipeline entity of one-dimensional pipeline system is abstracted to combine the business development needs of the Internet application of the communication industry. Secondly, the pipeline system is modeled and analyzed by Group theory. Group operation of pipeline entity state is established by discrete and symmetric transformation model. Finally, the flow state transition relationship of the pipeline system is established, and the transfer process of the system state in the group operation transformation is described. The solution scheme of the flow state transition relationship of the pipeline system is given, which will be helpful for the telecom operators to construct high-level mobile e-commerce application model and architecture.

2. Pipeline System Model Based on Group Theory

2.1. Introduction of Pipeline Entity. In the software application level in the pipeline system, pipeline entity is a platform system of business support service, including Undertaking pipeline, Service pipeline, Payment channel, Logistics pipeline, and Data pipeline. In physics and system level, pipeline entity can connect, transport, and package information. The pipeline entity can complete all kinds of connections between objects, including payment and logistics in application and physical connection nodes. One-dimensional pipe in the pipeline system has the effects of connection, support, and transmission. It is a connected entity object and transmission of information channel. When the pipeline entity constructed to scale, pipeline system will become a platform to support higher operation output.

2.2. Pipeline Entity Group and Pipeline Entity Object

2.2.1. Definition of Pipeline Entity Group. Group is a set, represented by the symbol G . A set G equipped with a binary operation \bullet is said to form a group (G, \bullet) , if the following conditions hold:

- (1) $\forall a, b \in G, a \bullet b \in G$.
- (2) $a \bullet (b \bullet c) = (a \bullet b) \bullet c$.
- (3) There exists an element $e \in G$, such that $\forall a \in G, a \bullet e = e \bullet a = a$.
- (4) $\forall a \in G$, there exists $a^{-1} \in G$, such that $a \bullet a^{-1} = a^{-1} \bullet a = e$ [19].

The definitions of pipeline group are as follows.

Definition 1. Pipeline entity group is the set of the symmetry transformation of pipeline entity.

Definition 2. The pipeline entity element is one kind of abstract operation unit which made the pipeline status transformed, acting on the communication pipeline which is estimative and logical.

In the specified information plane, a two-dimensional plane pipeline status expressed by the pipeline and its transformation is projected to the three-dimensional information space. Similarly, the pipeline status in the information space is also projected to the pipeline entity in the real space. There is a one-to-one correlation between these transformed pipeline objects in the real space and information space. Therefore, symmetric operation and translation operation of pipeline entity are EA and CP; status change operations are listed in Table 1, for example, OP, OM, PK, PB, PS, TP, TN, UM, UD, and UC. According to the definition of group, the transformations above can be judged as a part of group elements of pipeline entity group which is 12-order. Based on the two-dimensional nature, symmetry principle, and the symmetry transformation operations of 3-order components of information plane, the transformations can be classified into 6 types. These transformations can be defined as characteristic transformation matrixes and satisfy closure, distributive law, associative law, and inverse transformation of group. Characteristic transformation matrixes of group elements are listed in Table 1.

In the pipeline group G , any element G_i or G_j must satisfy that the result of calculation $G_i \bullet G_j$ is included in the group G ; the symbol " \bullet " is one kind of specific calculation, which is similar to the placeholder of integer addition. The equation of the characteristic matrix of group element can be described as follows:

$$a_1 \{G1\} + a_2 \{G2\} + a_3 \{G3, G7, G9\} + a_4 \{G4, G8, G10\} + a_5 \{G5, G11\} + a_6 \{G6, G12\} = 0. \quad (1)$$

Only when $a_1 = a_2 = a_3 = a_4 = a_5 = a_6$ can (1) be satisfied.

Choose $G1' = \{G1\}$, $G2' = \{G2\}$, $G3' = \{G3, G7, G9\}$, $G4' = \{G4, G8, G10\}$, $G5' = \{G5, G11\}$, and $G6' = \{G6, G12\}$, so $G1', G2', G3', G4', G5', G6'$ are linearly independent in the pipeline entity group; then group G can be characterized by 6-order basic tensor; that is, other group elements and pipeline objects can be expressed with $G1', G2', G3', G4', G5', G6'$. In addition, the transformations we mentioned above can only change the state of the pipeline entities, but not be related to the change of the substance of the pipeline entity.

2.2.2. Pipeline Entity Object Status

Definition 3. Information model of pipeline objects, $OM = O\{R1, R2, B, \dots\}$, abbreviated as O .

Pipeline object O is a vector and it is superposition state quantities of R_1, R_2, B, \dots , so the operation of the pipeline entity is equivalent to the treatment of the state quantities R_1, R_2, B, \dots . In the pipeline system, the pipeline state model is shown in Figure 1.

The state of A at different times of the transformation is A' ; that is, $A' = gA$, $g \in G$, where G is a pipeline entity group.

TABLE 1: Description of group elements.

Definition of group element	Transformation (concept of element) and symbol	Transformation matrix (representation)	Description of pipeline object
G1: Part Keep	PK: Part Keep	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	Pipeline hardware: keep an identical pipeline entity
G2: Negative	TN: Trend Negative	$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$	Pipeline hardware: an equivalent pipeline entity in a reverse flow direction
G3: Measure	UM: Uniform Measure	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$	Shop merchant: compared with another comparable pipeline object
G4: Plus	OP: Object Plus	$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	Shop merchant: add an addible pipeline entity Commodity payment data: add an addible pipeline entity
G5: Part Big	PB: Part Big	$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$	Pipeline hardware: enlarge a flow-increased pipeline entity
G6: Conflict	UC: Uniform Conflict	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	Shop merchant: conflict with another pipeline object

TABLE I: Continued.

Definition of group element	Transformation (concept of element) and symbol	Transformation matrix (representation)	Description of pipeline object
G7: Entity Abstract	EA: Entity Abstract	$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	Pipeline hardware: mapped as an information object Mobile APP: mapped as an information object Shop merchant: mapped as an information object Commodity payment data: mapped as an information object
G8: Positive	TP: Trend Positive	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	Pipeline hardware: an equivalent pipeline entity in the same flow direction
G9: Concept Project	CP: Concept Project	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	Pipeline hardware: projected as a pipeline entity Mobile phone APP: projected as a mobile phone APP entity Shop merchant: projected as a merchant entity Commodity payment data: projected as a data entity
G10: Object Minus	OM: Object Minus	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$	Pipeline hardware: delete a delectable pipeline entity Mobile phone APP: delete a delectable pipeline entity
G11: Part Small	PS: Part Small	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	Pipeline hardware: narrow a flow-decreased pipeline entity
G12: Double	UD: Uniform Double	$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	Shop merchant: match another pipeline object

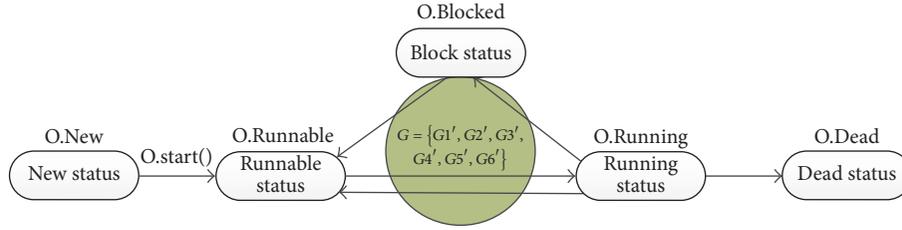


FIGURE 1: Pipeline state model.

TABLE 2: Description of state of pipeline object.

Pipeline object	State of an object	State symbol	Associated with traffic flow	Description of the role of pipeline system
Undertaking pipeline	Shop building	C: construct	Generate information flow	Router, linker, 2G/3G/4G/WiFi
	Goods bearing	L: load	Generate information flow	
	Business connection	U: union	Pass information flow	
Service pipeline	Service consulting	C: consult	Generate information flow	Mobile phone app, mall application software
	Service marketing	S: sale	Push information flow	
Payment channel	Commodity quotation	O: offer	Generate information flow	Mobile shopping mall
	Integral payment	P: pay	Generate information flow	
Logistics pipeline	Commodity logistics access	A: add	Pass information flow	Logistics business
	Logistics integration	L: link	Pass information flow	Commodity storage
Data pipeline	Data acquisition	G: gather	Generate information flow	Customers, order, delivery
	Data summary	T: total	Pass information flow	Mall business, chart
	Data mining and transportation	D: dig	Pass information flow	Excavating tools

The state of the pipeline object can be divided into ready state, running state, and blocking state in the system in which the pipeline object of the running state has an independent state. As shown in Table 2, g is a group element of G of the pipeline entity group, which can be expressed by the base vector as defined above:

$$g = \sum_{i=1}^6 G_i' \lambda_i = (G_1', G_2', G_3', G_4', G_5', G_6') \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_6 \end{pmatrix}. \quad (2)$$

2.3. Pipeline Entity Object State Transition Formula. In a realistic communication pipeline system, the state of the object is changed by the transformation operations that act on the object. It is measured in the Hilbert space of the transformation operation; in other words the state of an object is in a measurable state, which is a faithful reflection of the nature of the object. According to Definition 3 of the pipeline entity state, pipeline object in the Hilbert space is measured by the state and it is a space vector, respectively, R_1, R_2, R_3, \dots . It complies with the principle of superposition

in Hilbert space. According to $A' = gA$, $g \in G$, and the base vector representation method, from (2) pipeline group in 6-dimensional space can be described as

$$A' = \sum g_i A_i = \sum (G_1', G_2', G_3', G_4', G_5', G_6') \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_6 \end{pmatrix}_i A_i. \quad (3)$$

Therefore, similarly expanding to the pipeline object O , it satisfies the following formula:

$$O' = G * O. \quad (4)$$

For the pipeline entity object, O represents the initial state of the pipeline object and O' represents the intrinsic vector of the object of the pipeline, which is the object state vector of the pipeline entity under the action of the transformation. From formula (3) and formula (4), the entity state transition formula can be obtained:

$$G * O_{\text{start}} = E * O_{\text{end}}. \quad (5)$$

At this point, the pipeline object state formula is established. O_{start} is the initial state space expression of the pipeline system

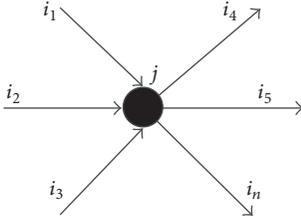


FIGURE 2: Model of node traffic.

and O_{end} is the terminal state space expression. In the state transition formula (5), the entity pipeline is treated as a whole system. Its O state changes from initial state to terminal state which is determined by the function of G . Among them, the quantity of flow in the transmission pipeline will not change with the change of state of pipe length, wideness, bending, and so on. The quantity of flow will not be generated or destroyed for no reason, and it stays unchanged within the pipeline transmission. This is the effect of applying Group theory on flow conservation, corresponding to reflecting the system structure of the pipeline itself. Due to the complexity of the traditional telecommunication system, the existing technology cannot provide enough storage space and reduce the throughput of the network system when the state space is too large. There are too many objects in the pipeline, which leads to the exponential increase in time complexity and the increase of time delay.

3. The Solution of Pipeline Network Flow State

3.1. Equations of Pipeline Network Flow. The whole pipeline network system can be seen as a system with producers and consumers; producers correspond to the service main body, providing the flow of relatively limited resources and consumers correspond to the users, utilizing producers' distribution of traffic resources. In the pipeline network model, all nodes are associated with several pipeline sections. According to the flow conservation law, the production flow is equal to the consumer flow resources of pipeline system. For any j node in the model, all traffic flowing into the node should be equal to the sum of all traffic flowing out of the node, as shown in Figure 2.

There are a number of pipeline sections connected at each stage; the sections and node flow equation can be obtained as

$$\sum_{i \in S_j} (\pm q_i) + Q_j = 0 \quad j = 1, 2, 3, \dots, N, \quad (6)$$

where q_i is the flow of section i ; Q_j is the flow of node j ; S_j is correlation sets of node j ; and N is the total number of the nodes in the network model. $\sum_{i \in S_j} (\pm q_i)$ means the sum of nodes associated with focus section, taking a minus sign when section direction is pointing to the node and the plus sign indicating departing from the node. When the traffic section flows from node, it gets plus sign. When the traffic flows into node, it gets minus sign. Simultaneous equations of all N nodes in the network model can constitute node

flow equations. From (6), the node flow equations in pipeline network model can be expressed as follows:

$$A\bar{q} + \bar{Q} = \bar{0}, \quad (7)$$

where A is the incidence matrix of network graph; $\bar{q} = [q_1 \ q_2 \ q_3 \ \dots \ q_n]^T$ is the column vector of section flow; and $\bar{Q} = [Q_1 \ Q_2 \ Q_3 \ \dots \ Q_n]^T$ is the column vector of node flow.

Because a huge number of the communication pipeline system objects and connections between object complex variables influence each other, these lead to the object "state space explosion." The structure of the correlation matrix A is so large that state variable cannot be solved directly. In this paper, the symmetry of pipeline entity was applied. From pipeline network information flow under the condition of conservation, pipeline network structure characteristics under the operation of transformation are studied. Therefore, equations are symmetrically simplified so that solutions of the state status can be achieved.

In the pipeline system, the total traffic inflows and outflows are equal, namely, the total flow conservation. In the process of information sending and receiving, the flow input rate is equal to the flow output rate of pipeline system. Therefore, flow is conserved at any moment; namely, the flow in the pipeline transmission balances, as shown below.

The traffic flow is assumed as a pipe object vector. According to the flow conservation equations (7) and the object state equation (5),

$$\sum_N q + \sum_M Q = \text{Flow} = \lambda * \text{Flow}, \quad (8)$$

where q is the output flow rate of any section (if it is in the opposite direction from the pipeline flow, it takes a negative value); Q is the input flow rate of any node (if it is in the opposite direction from the pipeline flow, it takes a negative value); N is the number of all sections; and M is the number of all nodes.

3.2. The Model and Solution in Flow Transmission of Pipeline Network. Pipeline network consists of M nodes and one node connected to i pipelines with one pipeline connecting two nodes. Therefore the largest network connection status of pipeline network is formed by M nodes and $M * (M - 1) / 2$ pipeline sections. In the information plane, after mapping and projection operations, the model properties of the pipeline entities are as follows.

- (1) Pipeline network system consists of N pipeline sections and M nodes.
- (2) One node can be connected to multiple pipeline sections and one pipeline section can connect only two nodes.
- (3) There is no intersection between the pipeline sections.
- (4) As an object of pipeline transmission, information flow meets conservation law.

Correlation matrix A of pipeline network flow formula describes the $N * M$ pipeline network connection form and it can be expressed generally as follows:

$$A = \begin{bmatrix} 1 & \cdots & k_{1j} & \cdots & k_{1M} \\ k_{21} & \cdots & k_{2j} & \cdots & k_{2M} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ k_{i1} & \cdots & k_{ij} & \cdots & k_{iM} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{N1} & \cdots & k_{Nj} & \cdots & 1 \end{bmatrix}_{N * M} \quad (9)$$

Element k_{ij} indicates whether nodes i and j are connected or not; when $k_{ij} = 0$, nodes i and j are connected; when $k_{ij} = 1$, nodes i and j are not connected. Particularly when $i = j$, $k_{ij} = 0$. For the multipoint crossover pipeline network, the properties of node and pipeline section are different, so it is difficult to define the symmetrical characteristic of the network in the two-dimensional plane. It is traditionally difficult to solve large volume pipeline network system status problems; therefore the Group theory solving steps are designed as follows.

Step 1. For pipeline network formed by nodes and pipeline sections, under pipeline entity G group symmetry transformation operation, flow of pipeline network where M nodes are connected to each other is transformed. According to demand response model of pipeline network in (8), pipeline network traffic incidence matrix is built and pipeline sections and nodes flow expressions are as follows.

For one pipeline section,

$$\text{Flow}_l = \sum_3 t_l^i Bps * \text{Flow} = t Bps * (\text{vector}) \text{Flow}. \quad (10)$$

For N pipeline sections,

$$\sum_N q = \sum_N \text{Flow}_l = \sum_N \sum_3 t_l^i Bps^i * (\text{vector}) \text{Flow}_l. \quad (11)$$

For one node,

$$\text{Flow}_n = \frac{P_n}{I_n} * \text{Flow}. \quad (12)$$

For M nodes,

$$\sum_M Q = \sum_M \text{Flow}_n = \sum_M \frac{P_n}{I_n} * \text{Flow}. \quad (13)$$

According to (10), (11), (12), and (13), the initial flow state formula of pipeline network is as follows:

$$\begin{aligned} (\text{initial}) \text{Flow} &= \sum_M \frac{P_n}{I_n} * \text{Flow}_n, \\ t_l^i &= 0. \end{aligned} \quad (14)$$

Final flow state of pipeline network is as follows:

$$(\text{final}) \text{Flow} = \sum_N t_l^i Bps^i * \text{Flow}_l + \sum_M \frac{P_n}{I_n} * \text{Flow}_n. \quad (15)$$

When pipeline network system is in the initial and the final state, according to (14) and (15), traffic balance formula can be obtained:

$$\sum_N t_l^i Bps^i * \text{Flow}_l + \sum_M \frac{(P_n^{1i} - P_n^{0i})}{I_n} * \text{Flow}_n = 0, \quad (16)$$

where Bps^i Intensity _{n} are the nature of the pipeline section and the node constant of pipeline network and t_l^i , P_n^{1i} , and P_n^{0i} are variables of pipeline network transmission. Generally, only part of information potential of pipeline network P_n^{1i} , P_n^{0i} is known.

Step 2. According to (16), the matrix form of the traffic demand response mode is calculated as

$$\begin{aligned} & \begin{pmatrix} 0 & t_1^i Bps_1^i, \dots, t_1^j Bps_1^j & t_1^m Bps_1^m \\ \vdots & \ddots & \vdots \\ t_1^m Bps_1^m & t_1^j Bps_1^j, \dots, t_1^i Bps_1^i & 0 \end{pmatrix}_{M * M} \begin{pmatrix} \text{Flow}_{\text{pipeline section}} \\ \vdots \\ \text{Flow}_{\text{pipeline section}} \end{pmatrix}_M \\ & + \begin{pmatrix} \sum_M \frac{(P_1^1 - P_1^0)}{I_1} & 0 \cdots 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 \cdots 0 & \sum_M \frac{(P_M^1 - P_M^0)}{I_M} \end{pmatrix}_{M * M} \begin{pmatrix} \text{Flow}_{\text{node}} \\ \vdots \\ \text{Flow}_{\text{node}} \end{pmatrix}_M = 0. \end{aligned} \quad (17)$$

Equation (17) can be simplified as follows:

$$\begin{aligned} & \begin{pmatrix} 0 & t_1^i Bps_1^i I_1, \dots, t_1^j Bps_1^j I_1 & t_1^m Bps_1^m I_1 \\ \vdots & \ddots & \vdots \\ t_1^m Bps_1^m I_M & t_1^j Bps_1^j I_M, \dots, t_1^i Bps_1^i I_M & 0 \end{pmatrix}_{M \times M} \\ & * \overline{\text{Flow}_{\text{pipeline section}}} \\ & = \begin{pmatrix} P_1 * K & 0 \dots 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 \dots 0 & P_M * K \end{pmatrix}_{M \times M} * \overline{\text{Flow}_{\text{node}}}, \end{aligned} \quad (18)$$

where t_1^i is pipeline section's effective transmission time; Bps_1^i is pipeline section's rate of information transmission; I_1 is the first node capacity; P_1 is the first throughput traffic; K is ratio of pipeline section to node flow direction; and $\overline{\text{Flow}_{\text{pipeline section}}}$ and $\overline{\text{Flow}_{\text{node}}}$ are pipeline sections and node vector units.

Step 3. Transformation matrix of pipeline network G_M is as follows:

$$\begin{aligned} & G_M \\ & = \begin{pmatrix} 0 & t_1^i Bps_1^i I_1, \dots, t_1^j Bps_1^j I_1 & t_1^m Bps_1^m I_1 \\ \vdots & \ddots & \vdots \\ t_1^m Bps_1^m I_M & t_1^j Bps_1^j I_M, \dots, t_1^i Bps_1^i I_M & 0 \end{pmatrix}_{M \times M}, \end{aligned} \quad (19)$$

$M - 1 \leq N \leq M(M - 1)/2$, where N is the number of pipeline sections connected to a node. According to traffic balance formula, due to the fact that transmission time and node of the potential difference are a variation of transmission pipeline system, the information transfer rate, information potential strength of pipeline structure characteristics, and entity group G are known, which makes the pipeline network form G_M matrix; suppose that M nodes and N pipeline sections consist of pipeline network. Under G group operation, pipeline sections and nodes connect to a network, and G_M matrix is constructed to represent pipeline network.

M nodes connected to a pipeline network are represented by the matrix G_M above. Generally, according to (19), the actual pipeline network system can be described by G_M

$$\begin{aligned} & G_M^{\text{general}} \\ & = \begin{pmatrix} 0 & t_1^i Bps_1^i I_1, \dots, t_1^j Bps_1^j I_1 & t_1^m Bps_1^m I_1 \\ \vdots & \ddots & \vdots \\ t_1^m Bps_1^m I_M & t_1^j Bps_1^j I_M, \dots, t_1^i Bps_1^i I_M & 0 \end{pmatrix}_{M \times M}. \end{aligned} \quad (20)$$

According to group theory irreducible representation method, under the condition of pipeline network flow conservation, $\{G_M\}$ group homomorphism in pipeline

entity group G group: $\{g_i\}$, $\{G_M\} \sim G$ group. So the network structure has nothing to do with the number of nodes. Network structure depends upon three inherent characteristics: information transformation rate Bps , potential strength (information capacity) intensity (I), and transfer duty cycle time (t).

Therefore, G_M can be expressed in a linear combination of the 6-dimensional spaces largely. G_M matrix is defined by node flow, $\text{Flow}_1, \text{Flow}_2, \dots, \text{Flow}_M$, while G group is defined by group transformation, $f_1, f_2, f_3, f_4, f_5, f_6$. Transformation matrix homomorphism in pipeline entity group: G group: $\{g_i\}$.

Step 4. Six base vectors of pipeline entity G group are as follows:

$$\begin{aligned} G1' &= \{G1\}, \\ G2' &= \{G2\}, \\ G3' &= \{G3, G7, G9\}, \\ G4' &= \{G4, G8, G10\}, \\ G5' &= \{G5, G11\}, \\ G6' &= \{G6, G12\}. \end{aligned} \quad (21)$$

In the pipeline entity group, $G1', G2', G3', G4', G5', G6'$ are linearly independent; according to characterization of group elements matrix, there is a set of real numbers, $a_1, a_2, a_3, a_4, a_5, a_6$, which makes the equation below hold:

$$\begin{aligned} G_M &= a_1 G1' + a_2 G2' + a_3 G3' + a_4 G4' + a_5 G5' \\ &+ a_6 G6'. \end{aligned} \quad (22)$$

According to Step 1, $a_1, a_2, a_3, a_4, a_5, a_6$ can be calculated; according to $G1', G2', G3', G4', G5', G6'$, $t_1, t_2, t_3, \dots, t_n$ of G_M are worked out.

Above is the process of applying Group theory to the information pipeline system. Regardless of the number of nodes and the number of pipeline sections, the G_M matrix can be simplified to a 6-order matrix. Thus "state space explosion" problem of pipeline network can be solved and it is a universal solution.

4. The Application of Pipeline Network System

In pipeline network system under the application of pipeline entity group, when the flow changes in pipeline entity, changing flow behavior of the pipeline system can be described as the Application Layer, Business Object Layer, and the Network Node Layer, as shown in Figure 3.

Application Layer. A functional unit in the Application Layer is the performance behavior of pipeline entity, which is the initiator of flow changing behavior. When a certain task is

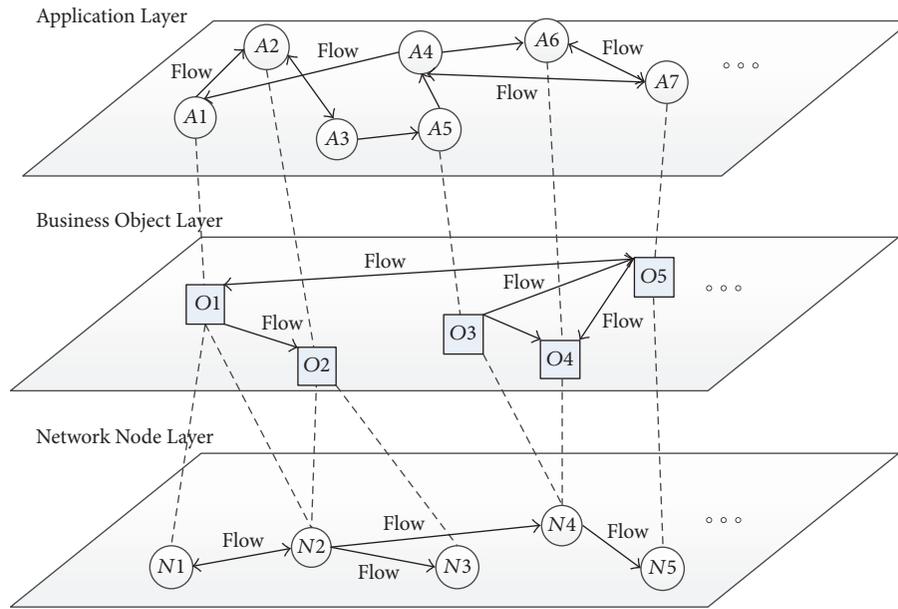


FIGURE 3: Architecture of pipeline system.

executed, traffic flows from units A_i to A_j , to complete the business process of information transmission and exchange.

Business Object Layer. The object unit of the Business Object Layer is the abstract components of pipeline entity. When the Application Layer initiates some functional behavior, business object unit is packaging and transporting the flow, which is designated as O_i receives and sends information to O_j .

Network Node Layer. Network Node Layer node unit consists of the physical nodes in pipeline system. When the Application Layer initiates some functional behaviors, Business Object Layer sends and receives information. In the meantime, Network Node Layer finds the optimal solution about the critical path and completes the distribution and reception of information flow.

5. Conclusion

In order to support the increasingly complex service quality of telecom operation system, this paper sets up the pipeline entity operation group and pipeline entity object state formula according to the current situation of mobile Internet. According to the law of flow conservation of pipeline system, the flow equation of pipe network, and pipeline object state transfer matrix, the time variables of the pipe network matrix G_M are calculated through the linear combination of six basis vectors of the operation group. This paper proposes a new optimization method for the general information network system, which provides a new idea for the future information network system optimization.

Conflicts of Interest

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

References

- [1] L. Ware, "Over-the-top video delivery and consumption in 2012 and beyond," *SMPTE Motion Imaging Journal*, vol. 121, no. 6, pp. 73–76, 2012.
- [2] J. Rückert, J. Blendin, and D. Hausheer, "Software-Defined Multicast for Over-the-Top and Overlay-based Live Streaming in ISP Networks," *Journal of Network and Systems Management*, vol. 23, no. 2, pp. 280–308, 2015.
- [3] N. Bouten, S. Latré, W. Van De Meerssche et al., "A multicast-enabled delivery framework for QoE assurance of over-the-top services in multimedia access networks," *Journal of Network and Systems Management*, vol. 21, no. 4, pp. 677–706, 2013.
- [4] C.-G. Yang, S. Trimi, and S.-G. Lee, "Effects of telecom service providers' strategic investments on business performance: a comparative study of US-Korea," *Industrial Management & Data Systems*, vol. 116, no. 5, pp. 960–977, 2016.
- [5] S.-L. Su, Y.-C. Tsai, and Y.-H. Yang, "A Quality-of-Service Routing Protocol with Supplementary Cooperation for Wireless Ad Hoc Networks," *Wireless Personal Communications*, vol. 84, no. 3, pp. 1627–1645, 2015.
- [6] N. Banerjee, S. Nagar, and S. Mukherjea, "Service control layer (SCL): Enabling rule-based control and enrichment in next-generation telecom service delivery," *Journal of Network and Systems Management*, vol. 21, no. 1, pp. 65–98, 2013.
- [7] K. Kimbler and M. Taylor, "Value added mobile broadband services Innovation driven transformation of the 'smart pipe,'" in *Proceedings of the 2012 16th International Conference on Intelligence in Next Generation Networks, ICIN 2012*, pp. 30–34, Germany, October 2012.
- [8] Z. Yang and Z. Ma, "Analysis of communication operators transformation on smart pipe," in *Proceedings of the 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2013*, pp. 131–134, China, August 2013.

- [9] H. Kashif, S. Gholamian, and H. Patel, "SLA: A stage-level latency analysis for real-time communication in a pipelined resource model," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 1177–1190, 2015.
- [10] S. A. Sharna, M. R. Amin, and M. Murshed, "An enhanced-MDP based vertical handoff algorithm for QoS support over heterogeneous wireless networks," in *Proceedings of the 10th IEEE International Symposium on Network Computing and Applications, NCA 2011*, pp. 289–293, USA, August 2011.
- [11] A. Munir and A. Gordon-Ross, "An MDP-based dynamic optimization methodology for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 616–625, 2012.
- [12] C. Lin, Y. Li, and J.-X. Wan, "Optimization approaches for QoS in computer networks: a survey," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 34, no. 1, pp. 1–14, 2011.
- [13] C. Lin, J. X. Wan, X. D. Xiang, K. Meng, and Y. Z. Wang, "Dynamic optimization in computer systems and computer networks: models, solutions, and applications," *Chinese Journal of Computers. Jisuanji Xuebao*, vol. 35, no. 7, pp. 1339–1357, 2012.
- [14] H. Osborn, "Symmetry relationships between crystal structures: applications of crystallographic group theory in crystal chemistry," *Contemporary Physics*, vol. 6, no. 1, pp. 97–98, 2015.
- [15] J. Laane and E. J. Ocola, "Applications of symmetry and group theory for the investigation of molecular vibrations," *Acta Applicandae Mathematicae*, vol. 118, no. 1, pp. 3–24, 2012.
- [16] A. Pérez-Ramírez, J. Guerrero Juk, R. Sanchez-Lara, J. A. Trejo-Sánchez, and L. de la Cruz-May, "Application of Mathematical Symmetrical Group Theory in the Creation Process of Digital Holograms," *Mathematical Problems in Engineering*, vol. 2017, 7 pages, 2017.
- [17] H. N. Patel and D. K. Ghodgaonkar, "Application of group theory for computation reduction in microwave imaging of human breast model at 500 MHz," *Progress in Electromagnetics Research M*, vol. 51, pp. 165–173, 2016.
- [18] J. Zhang, P. Wu, T. Yu et al., "Research of 3D pipeline model based on telecom operators," in *Proceedings of 2015 International Conference on Mechanical, Electronic and Information Technology Engineering*, pp. 681–682, 2015.
- [19] T. Ayush, "Semester project report-topics in finite group theory," *Diritto Marittimo*, vol. 83, pp. 364–365, 2015.

Research Article

Modeling and Simulation of Gas Emission Based on Recursive Modified Elman Neural Network

Lin Wei ¹, Yongqing Wu,¹ Hua Fu,² and Yuping Yin²

¹Department of Basic Education, Liaoning Technical University, Huludao, China

²School of Electrical and Control Engineering, Liaoning Technical University, Huludao, China

Correspondence should be addressed to Lin Wei; 29766164@qq.com

Received 13 October 2017; Revised 3 January 2018; Accepted 22 January 2018; Published 20 February 2018

Academic Editor: Qian Zhang

Copyright © 2018 Lin Wei 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.

For the purpose of achieving more effective prediction of the absolute gas emission quantity, this paper puts forward a new model based on the hidden recurrent feedback Elman. The recursive part of classic Elman cannot be adjusted because it is fixed. To a certain extent, this drawback affects the approximation ability of the Elman, so this paper adds the correction factors in recursive part and uses the error feedback to determine the parameters. The stability of the recursive modified Elman neural network is proved in the sense of Lyapunov stability theory, and the optimal learning rate is given. With the historical data of mine actual monitoring to experiment and analysis, the results show that the recursive modified Elman neural network model can effectively predict the gas emission and improve the accuracy and efficiency of prediction compared with the classic Elman prediction model.

1. Introduction

In the daily management of mine safety, an effective method of prevention and control of mine gas disasters is the scientific analysis of the gas emission data provided by the monitoring system [1]. The gas is one of the most important factors threatening the safety production of mine [2]. Most recent work mainly focuses on different methods for improving the prediction performance of the absolute gas emission quantity, such as Grey theory [3], principal component regression analysis method [4], partial least squares support vector machine [5], virtual state variables and Kalman filter [6], BP neural network [7], and RBF neural network [8].

In recent years, intelligent computing methods have been rapidly developed in dynamic system identification [9, 10], time series prediction [11, 12], and other fields. In fact, there are many factors influencing the absolute gas emission quantity, such as coal seam gas content, burying depth, and coal seam thickness [13, 14]. That means the gas emission prediction model is a multidimensional complex dynamic system, and it is difficult to accurately predict gas emission quantity, since recurrent neural network is a highly nonlinear dynamical system that exhibits complex behaviors and good ability of processing dynamic information [15]. As is well

known, the recurrent neural network has wide applications in various areas [16, 17]. It is expected that recurrent neural network possesses better performance than feedforward neural network (such as BP and RBF) in modeling and predicting gas emission quantity. In particular, Elman neural network (ENN) has been proved successful in gas emission prediction [18, 19]. Some works on improving the performance of gas emission prediction using ENN can be found in, for example, [20–22]. However, a common drawback of the above gas emission prediction models based on classic Elman neural network is that the recursive part of the hidden layer cannot be adjusted because it is fixed. This drawback affects the nonlinear approximation ability of classic Elman neural network.

From the above observation, this paper proposes a novel strategy of adding correction factors in recursive part of ENN, resulting in a new model called recursive modified Elman neural network (RMENN). The stability and convergence of RMENN model are theoretically proved, and some meaningful results are obtained in this paper. In practice, through the analysis of the main factors affecting coal gas emission, this paper puts forward the gas emission prediction model based on recursive modified Elman neural network.

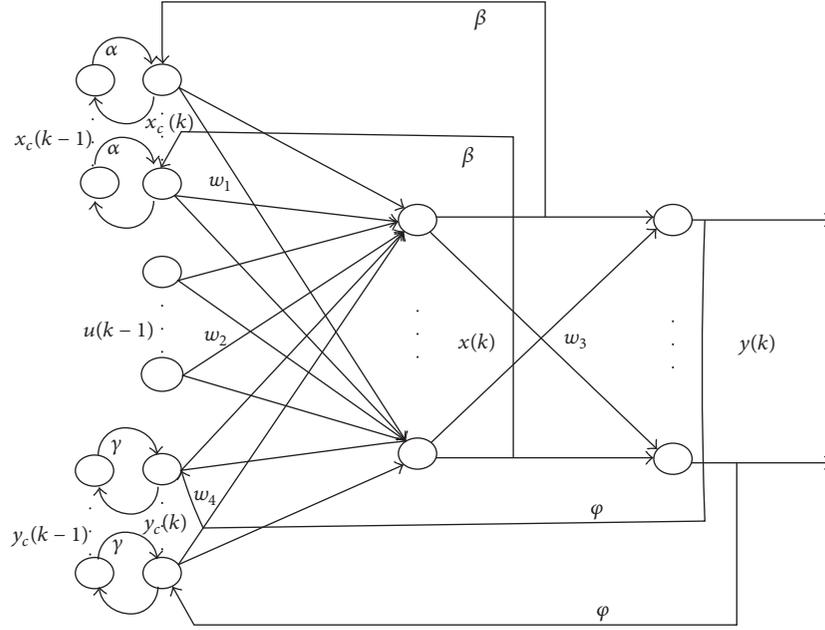


FIGURE 1: Topology of the RMENN.

The rest of this paper is organized as follows. The establishment of RMENN model is described in Section 2. The learning algorithms of RMENN model are described in Section 3. The performance analysis and flowchart of RMENN model are described in Section 4. Experiment analysis results on the gas emission prediction are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Establishment of RMENN Model

As discussed in Section 1, we aim to propose a specific architecture to overcome the aforementioned drawback of a fixed structure and improve the nonlinear approximation ability of ENN. In Figure 1, this paper adds correction factors in the context layer and the output layer to adjust the values of the recursive parts. Let $u(k) \in R^r$ and $y(k) \in R^m$ denote the network input and output vectors at the discrete time k , respectively. Let $w^1 \in R^{n \times n}$, $w^2 \in R^{n \times r}$, $w^3 \in R^{m \times n}$, $w^4 \in R^{n \times m}$ denote weight matrices of context-hidden, input-hidden, hidden-output, and output-hidden, respectively. Let $x_c(k) \in R^n$ and $x(k) \in R^n$ denote the output vectors of the context layer and the hidden layer at the discrete time k , respectively. Let $\beta x(k-1)$ be the correction part of the hidden context layer. Let $\gamma y(k-1)$ be the correction part of the output context layer. $f(\cdot)$ and $g(\cdot)$ are the activation functions, respectively. In a general way, $f(\cdot)$ is the sigmoid function and $g(\cdot)$ is the linear function. Let $y_c(k) \in R^m$ be the output vector of the output context layer at the discrete time k .

With this feature, the new model called RMENN is able to improve update power of the classic ENN and exhibit rapid convergence and high prediction accuracy. The relationship between input and output of RMENN can be expressed as

$$\begin{aligned} x(k) &= f(w^1 x_c(k) + w^2 u(k-1) + w^4 y_c(k)) \\ x_c(k) &= \alpha x_c(k-1) + \beta x(k-1) \\ y_c(k) &= \gamma y_c(k-1) + \varphi y(k-1) \\ y(k) &= g(w^3 x(k)), \end{aligned} \quad (1)$$

where α, β ($0 \leq \alpha < 1, 0 < \beta \leq 1$) are, respectively, feedback factor and correction factor of the context layer. γ, φ ($0 \leq \gamma < 1, 0 \leq \varphi \leq 1$) are, respectively, feedback factor and correction factor of the output layer. In particular, when $\alpha = 0, \beta = 1, \gamma = 0, \varphi = 0$, the special model is the classic Elman neural network [11].

The topology of the RMENN is shown in Figure 1.

3. Learning Algorithm for RMENN

The main objective of learning algorithm is to minimize a predefined energy function by adaptively adjusting the vector of network parameters based on a given set of input-output pairs. The particular energy function used in the RMENN is as follows:

$$e(k) = \frac{1}{2} (y_{qw}(k) - y(k))^T (y_{qw}(k) - y(k)), \quad (2)$$

where $y_{qw}(k) \in R^m$ is the desired output associated with the input pattern and $y(k) \in R^m$ is the inferred output at the discrete time k .

The weights of the RMENN are updated by the negative gradient of the energy function; those are

$$\Delta w_{ij}^3 = \eta_3(k) \delta_i^o x_j(k), \quad (3)$$

$$i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n;$$

$$\Delta w_{jq}^2 = \eta_2(k) \delta_j^h u_q(k-1), \quad (4)$$

$$j = 1, 2, \dots, n; \quad q = 1, 2, \dots, r;$$

$$\Delta w_{jl}^1 = \eta_1(k) \sum_{i=1}^m (\delta_i^o w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1}, \quad (5)$$

$$j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n;$$

$$\Delta w_{js}^4 = \eta_4(k) \sum_{i=1}^m (\delta_i^o w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{js}^4}, \quad (6)$$

$$s = 1, 2, \dots, m; \quad j = 1, 2, \dots, n,$$

where $\eta_1(k)$, $\eta_2(k)$, $\eta_3(k)$, $\eta_4(k)$ are the learning rate of w^1 , w^2 , w^3 , w^4 , respectively. With the learning-rate parameters $\eta_1(k)$, $\eta_2(k)$, $\eta_3(k)$, $\eta_4(k)$, the terms δ_i^o and δ_j^h are calculated as follows:

$$\begin{aligned} \delta_i^o &= y_{q_{w,i}}(k) - y_i(k) \\ \delta_j^h &= \sum_{i=1}^m (\delta_i^o w_{ij}^3) f_j'(\cdot), \end{aligned} \quad (7)$$

where $\partial x_j(k)/\partial w_{jl}^1$ and $\partial x_j(k)/\partial w_{js}^4$ are calculated as follows:

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = \alpha \frac{\partial x_j(k-1)}{\partial w_{jl}^1} + \beta f_j'(\cdot) x_l(k-1) \quad (8)$$

$$\frac{\partial x_j(k)}{\partial w_{js}^4} = \gamma \frac{\partial x_j(k-1)}{\partial w_{js}^4} + \varphi f_j'(\cdot) y_s(k-1). \quad (9)$$

4. Performance Analysis

4.1. Convergence and Stability. The appropriate learning rate can make the learning algorithm converge at a faster speed. According to the Lyapunov stability theory, $\eta_1(k)$, $\eta_4(k)$ can be detailedly proved, and the proof of $\eta_2(k)$, $\eta_3(k)$ is similar to $\eta_1(k)$, $\eta_4(k)$.

Theorem 1. Let the weights of RMENN be updated by (3)–(9).

(1) If $0 < \eta_1(k) < 32(1-\alpha)^2 \beta^{-2} (n^2 (\max_{ij} w_{ij}^3(k))^2)^{-1}$, the iterative learning process of w^1 is a stable and convergent process by (5).

(2) If $0 < \eta_2(k) < 8(nr |\max_k u_k(k)| |\max_{ij} w_{ij}^3(k)|)^{-1}$, the iterative learning process of w^2 is a stable and convergent process by (4).

(3) If $0 < \eta_3(k) < 2/n$, the iterative learning process of w^3 is a stable and convergent process by (3).

(4) If $0 < \eta_4(k) < 32(1-\gamma)^2 (mn)^{-1} (\varphi M (\max_{ij} w_{ij}^3(k)))^{-2}$, the iterative learning process of w^4 is a stable and convergent process by (6).

Proof. (1) Let the energy function be described by (2).

Since

$$\Delta e(k) = e(k+1) - e(k) = \frac{1}{2} \sum_{i=1}^m [e_i^2(k+1) - e_i^2(k)], \quad (10)$$

where

$$\begin{aligned} e_i(k+1) &= e_i(k) + \sum_{j=1}^n \frac{\partial e_i(k)}{\partial w_{ij}^1} \Delta w_{ij}^1 \\ &= e_i(k) - \sum_{j=1}^n \frac{\partial y_i(k)}{\partial w_{ij}^1} \Delta w_{ij}^1, \end{aligned} \quad (11)$$

then

$$\begin{aligned} \Delta e(k) &= \frac{1}{2} \sum_{i=1}^m e_i^2(k) \\ &\cdot \left(\left(1 - \eta_1(k) \left(\frac{\partial y_i(k)}{\partial w^1} \right)^T \left(\frac{\partial y_i(k)}{\partial w^1} \right) \right)^2 - 1 \right) = \frac{1}{2} \quad (12) \end{aligned}$$

$$\cdot \sum_{i=1}^m e_i^2(k) \left(\left(1 - \eta_1(k) \left\| \frac{\partial y_i(k)}{\partial w^1} \right\|^2 \right)^2 - 1 \right),$$

where w^1 is a $n \times n$ matrix and $\|\cdot\|$ is 2-norm.

Since

$$\left| \frac{\partial y_i(k)}{\partial w_{jl}^1} \right| = \left| \frac{\partial y_i(k)}{\partial x_j(k)} \frac{\partial x_j(k)}{\partial w_{jl}^1} \right| = |w_{ij}^3(k)| \left| \frac{\partial x_j(k)}{\partial w_{jl}^1} \right|, \quad (13)$$

$$i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n,$$

according to (8) with the initial condition

$$\frac{\partial x_j(0)}{\partial w_{jl}^1} = 0, \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n, \quad (14)$$

we can get the following equation:

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} \approx \beta \sum_{t=1}^k \alpha^{t-1} f_j'(k-t+1) x_l(k-t+1), \quad (15)$$

$$j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n,$$

Since $0 < f_j'(\cdot) \leq 1/4$, $0 \leq \alpha < 1$, $0 < \beta \leq 1$, $0 < x_l(k-t+1) < 1$, we can get the following conclusions:

$$\begin{aligned} \left| \frac{\partial x_j(k)}{\partial w_{jl}^1} \right| &\leq \beta \sum_{t=1}^k \alpha^{t-1} |f_j'(k-t+1)| |x_l(k-t+1)| \\ &< \frac{\beta}{4(1-\alpha)}, \end{aligned} \quad (16)$$

$$j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n.$$

Therefore

$$\begin{aligned} \left| \frac{\partial y_i(k)}{\partial w_{jl}^1} \right| &= \left| w_{ij}^3(k) \right| \left| \frac{\partial x_j(k)}{\partial w_{jl}^1} \right| \\ &< \frac{\beta}{4(1-\alpha)} \left| \max_{i,j} w_{ij}^3(k) \right|, \end{aligned} \quad (17)$$

$j = 1, 2, \dots, n; l = 1, 2, \dots, n;$

$$\left\| \frac{\partial y(k)}{\partial w^1} \right\| < \frac{n\beta}{4(1-\alpha)} \left| \max_{i,j} w_{ij}^3(k) \right|.$$

Since $0 < \eta_1(k) < 32(1-\alpha)^2/n^2\beta^2(\max_{i,j} w_{ij}^3(k))^2$, for $e(k) \geq 0$ and $\Delta e(k) = e(k+1) - e(k) < 0$, we can ensure that Δw_{jl}^1 is a stable and convergent process.

Proof (4) It is similar to the proof (1).

Since

$$\Delta e = \frac{1}{2} \sum_{i=1}^m e_i^2(k) \left(\left(1 - \eta_4(k) \left\| \frac{\partial y_i(k)}{\partial w^4} \right\|^2 \right)^2 - 1 \right) \quad (18)$$

according to (9) with the initial condition

$$\frac{\partial x_j(0)}{\partial w_{js}^4} = 0, \quad j = 1, 2, \dots, n; s = 1, 2, \dots, n, \quad (19)$$

we can get the following equation:

$$\begin{aligned} \frac{\partial x_j(k)}{\partial w_{js}^4} &\approx \varphi \sum_{t=1}^k \gamma^{t-1} f_j'(k-t+1) y_s(k-t+1), \\ &j = 1, 2, \dots, n; s = 1, 2, \dots, n. \end{aligned} \quad (20)$$

Let $M = \max_k |y_s(k)|$.

Then $0 \leq \gamma < 1$ and $|\partial x_j(k)/\partial w_{js}^4| < \varphi M/4(1-\gamma)$.

Hence,

$$\begin{aligned} \left| \frac{\partial y_i(k)}{\partial w_{js}^4} \right| &= \left| w_{ij}^3(k) \right| \left| \frac{\partial x_j(k)}{\partial w_{js}^4} \right| \\ &< \frac{\varphi M}{4(1-\gamma)} \left| \max_{i,j} w_{ij}^3(k) \right|, \end{aligned} \quad (21)$$

$$\left\| \frac{\partial y(k)}{\partial w^4} \right\| < \frac{\varphi M \sqrt{mn}}{4(1-\gamma)} \left| \max_{i,j} w_{ij}^3(k) \right|.$$

Since $0 < \eta_4(k) < 32(1-\gamma)^2(mn)^{-1}(\varphi M(\max_{i,j} w_{ij}^3(k)))^{-2}$, for $e(k) \geq 0$ and $\Delta e(k) < 0$, we can ensure that Δw_{js}^4 is a stable and convergent process.

The proof of $\eta_2(k)$, $\eta_3(k)$ is similar to $\eta_1(k)$, $\eta_4(k)$.

The proof of theorem is completed. \square

4.2. Adaptive Learning Rate of RMENN. As explained before, we can get the optimal learning rate as follows.

Let $1 - \eta_1(k) \|\partial y_i(k)/\partial w^1\|^2 = 0$, $\Delta e(k)$ is the minimum negative value, and the convergence speed of RMENN is the fastest.

Let $\|\partial y(k)/\partial w^1\| \approx (n\beta/4(1-\alpha))|\max_{i,j} w_{ij}^3(k)|$; we can get the optimal learning rate as follows:

$$\eta_1^*(k) = 16(1-\alpha)^2 \left(n\beta \left(\max_{i,j} w_{ij}^3(k) \right) \right)^{-2}. \quad (22)$$

Similarly,

$$\begin{aligned} \eta_2^*(k) &= 4 \left(nr \left| \max_k u_k(k) \right| \left| \max_{ij} w_{ij}^3(k) \right| \right)^{-1} \\ \eta_3^*(k) &= \frac{1}{n} \end{aligned} \quad (23)$$

$$\eta_4^*(k) = 16(1-\gamma)^2 (mn)^{-1} \left(\varphi M \left(\max_{ij} w_{ij}^3(k) \right) \right)^{-2},$$

where $\eta_1^*(k)$, $\eta_2^*(k)$, $\eta_3^*(k)$, $\eta_4^*(k)$ are the optimal adaptive learning rate of w^1 , w^2 , w^3 , w^4 , respectively.

The training algorithm procedures of RMENN are shown in Figure 2.

5. Model Test

5.1. Data Selection and Preliminary Analysis. China is the larger coal consumer among the developing countries. China's secure producing situation of the coalmine is very grim, especially the accident of gas disasters, which would result in a large quantity of casualties and property losses, and has absorbed high attention of the government. The precise prediction of gas emission is important for the mineral production safety in China. Gas emission prediction model based on small sample has always been a significant subject in coal mine gas study field [1].

This paper selects Kailuan mining group money mining camp in May 2007 to December 2008 working face of absolute gas emission quantity [23]. The main factors are shown in Table 1 such as coal seam gas content (x_1), burying depth (x_2), coal seam thickness (x_3), coal seam dip angle (x_4), mining height (x_5), daily work progress (x_6), working face length (x_7), production rate (x_8), adjacent layer gas content (x_9), adjacent layer thickness (x_{10}), adjacent layer spacing (x_{11}), mining intensity (x_{12}), interlayer lithology (x_{13}), and gas emission quantity (y).

In order to reduce the influence of the different dimension, the experimental data are conducted by generating a value between lower and upper limits of each factor by using the formula $x^* = (x - x_{\min})/(x_{\max} - x_{\min})$. x is the experimental data as shown in Table 1. x_{\min} and x_{\max} describe the lower limit and upper limit of the experimental data, respectively. Let x^* be the standardized data. Then this formula $x = (x_{\max} - x_{\min})x^* + x_{\min}$ is used to restore data. In particular, we use the first 16 data for training and the remaining 4 for validation. Through some experiments, the topological structure of the classic ENN is optimal in the form of the 13-16-1. So the RMENN has the same topological structure to compare with ENN, and let $\alpha = 0.4$, $\beta = 0.6$, $\gamma = 0.1$, $\varphi = 0.3$. The training error is set to 0.01.

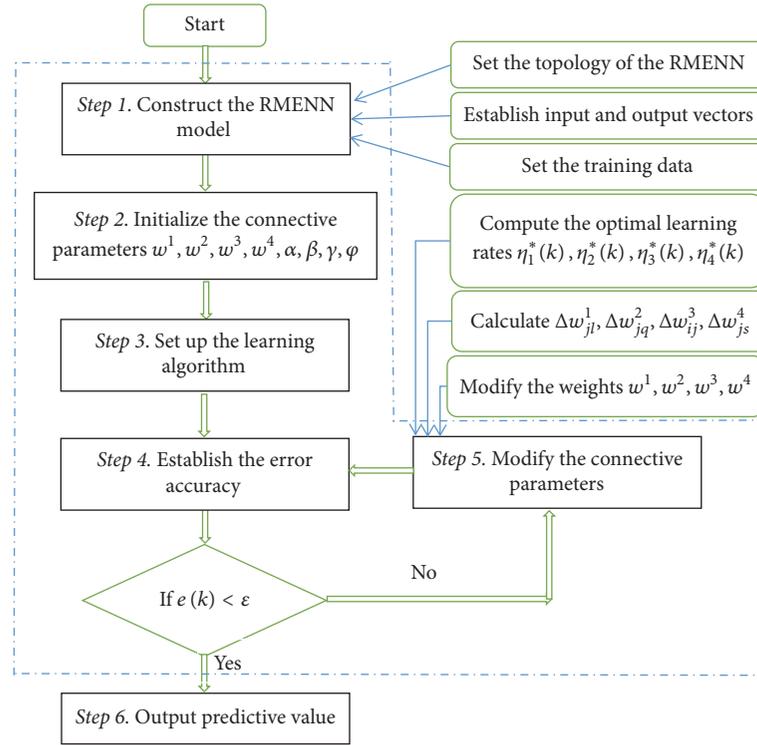


FIGURE 2: Training algorithm procedures of RMENN.

5.2. *Training Results.* After 50 times independent simulations, we compare the training effects of the two models. From the point of view of training error traces, Figures 3(a)–3(c) show that the RMENN has stronger update power than the classic ENN. However, the classic ENN obviously lacks update power; even it is not sufficiently close to the target error as shown in Figure 3(c). It is because the fact that recursive parts of the classic ENN cannot be adjusted, but the recursive parts of RMENN can be adjusted and the learning rate can be dynamically adjusted to improve update power.

From the point of view learning speed, the RMENN has superior convergence speed than the classic ENN. When it averagely reach 348 epochs, the training error of the RMENN meets the requirement, but the classic ENN do not meet the requirements in the best training error traces as shown in Figure 3(a) (mean square error of the classic ENN is 0.010193).

Figure 4 shows the state of relative error distribution in the training process. The maximum relative error, the minimum relative error, and the average relative error of the RMENN are 10.04%, 0.87%, and 3.54%, respectively. However the maximum relative error, the minimum relative error, and the average relative error of the classic ENN are 15.09%, 2.14%, and 5.21%, respectively. It demonstrates that the RMENN has higher training accuracy than the classic ENN.

Figure 5 shows that the RMENN has superior average approximation effect than the classic ENN in 50 times independent simulations.

5.3. *Comparison of Model Prediction Ability.* The mean squared error (MSE), median absolute error (MAE), and

mean absolute percentage error (MAPE) are used as the indicators to measure the prediction precision. These indicators are defined as follows:

$$\begin{aligned}
 \text{MSE} &= \frac{1}{T} \sum_{k=1}^T (y_k - \hat{y}_k)^2 \\
 \text{MAE} &= \frac{1}{T} \sum_{k=1}^T |y_k - \hat{y}_k| \\
 \text{MAPE} &= \frac{1}{T} \sum_{k=1}^T \left| \frac{y_k - \hat{y}_k}{y_k} \right| \times 100\%,
 \end{aligned}
 \tag{24}$$

where y_k and \hat{y}_k denote the real and predicted values at time k , respectively.

Table 2 shows the state of relative error distribution in the prediction process. The maximum relative error, the minimum relative error, and the average relative error of the RMENN are 5.48%, 0.32%, and 3.43%, respectively. However the maximum relative error, the minimum relative error, and the average relative error of the classic ENN are 9.12%, 2.48%, and 5.50%, respectively. The relative errors of the RMENN appear more smaller. The mean squared error (MSE), median absolute error (MAE), and mean absolute percentage error (MAPE) of the RMENN are 0.0620, 0.2181, and 3.43%, respectively. The mean squared error (MSE), median absolute error (MAE), and mean absolute percentage error (MAPE) of the classic ENN are 0.1280, 0.3385, and 5.50%, respectively. It demonstrates that the proposed RMENN model

TABLE I: The statistical data of coalface gas emission and influencing factors.

Number	$x_1/(m^3 \cdot t^{-1})$	x_2/m	x_3/m	$x_4/(o)$	x_5/m	$x_6/(m \cdot d^{-1})$	x_7/m	$x_8\%$	$x_9/(m^3 \cdot t^{-1})$	x_{10}/m	x_{11}/m	$x_{12}/(t \cdot d^{-1})$	x_{13}	$y/(m^3 \cdot min^{-1})$
(1)	1.92	408	2.0	10	2.0	4.42	155	96	2.02	1.50	20	1	5.03	3.34
(2)	2.14	421	1.8	11	1.8	4.13	145	95	2.64	1.62	19	1	4.75	3.56
(3)	2.58	450	2.3	10	2.3	4.67	150	95	2.41	1.48	18	2	4.91	3.67
(4)	2.40	456	2.2	15	2.2	4.51	160	94	2.55	1.75	20	2	4.63	4.17
(5)	3.22	516	2.8	13	2.8	3.45	180	93	2.21	1.72	12	2	4.78	4.60
(6)	2.80	527	2.5	17	2.5	3.28	180	94	2.81	1.81	11	1	4.51	4.92
(7)	3.23	517	2.8	13	2.8	3.46	180	93	2.23	1.71	12	2	4.76	4.61
(8)	3.35	531	2.9	9	2.9	3.68	165	93	1.88	1.42	13	2	1.82	4.78
(9)	3.61	550	2.9	12	2.9	4.02	155	92	2.12	1.60	14	2	4.83	5.23
(10)	3.71	573	3.2	11	3.2	2.92	175	91	3.11	1.46	13	2	4.63	5.62
(11)	4.21	590	5.9	8	5.9	2.85	170	79	3.40	1.50	18	3	4.77	7.24
(12)	4.03	604	6.2	9	6.2	2.64	180	81	3.15	1.80	16	3	4.70	7.80
(13)	4.80	630	6.5	9	6.16	2.77	165	78	3.02	1.74	17	3	4.62	7.68
(14)	4.67	640	6.3	11	3	2.75	175	80	2.56	1.75	15	3	4.60	7.95
(15)	2.43	450	2.7	11	2.7	4.32	165	93	2.35	1.85	16	2	4.58	5.06
(16)	3.16	544	2.7	17	2.7	3.81	165	93	2.81	1.79	13	2	4.90	4.93
(17)	4.62	629	6.4	13	6.4	2.80	170	80	3.35	1.61	19	3	4.63	8.04
(18)	4.53	635	6.2	9	6.2	2.73	160	72	2.94	1.73	17	3	4.61	7.56
(19)	3.87	580	3.9	11	3.9	2.85	170	92	3.02	1.39	14	2	4.72	5.82
(20)	3.24	509	2.5	14	2.5	4.40	160	93	2.79	1.72	13	2	4.65	4.36

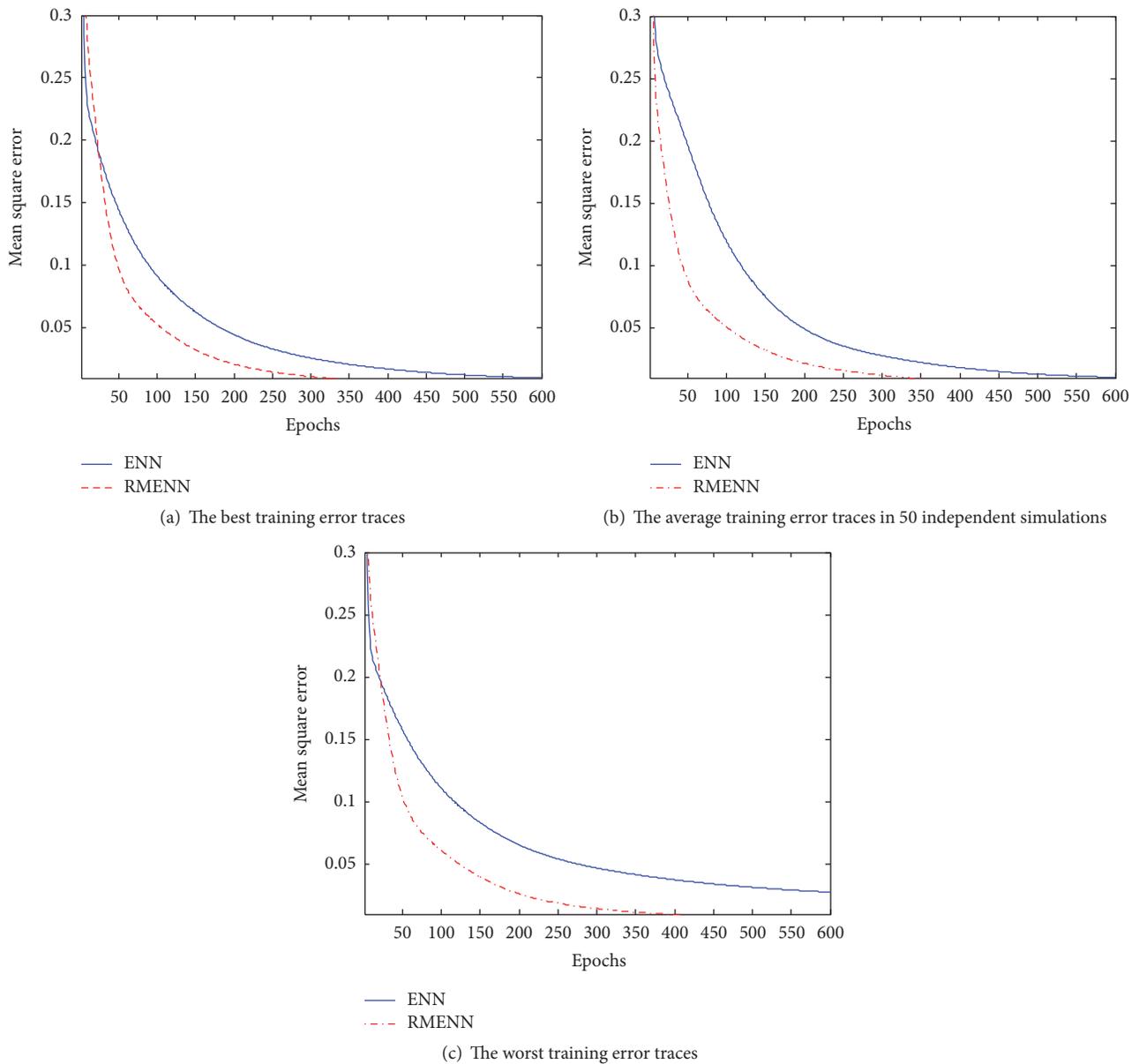


FIGURE 3: Comparison of convergence from the classic ENN and the RMENN.

TABLE 2: Results of the error analysis based on MSE, MAE, and MAPE.

Number	Actual data	ENN		RMENN	
		Output results	Relative error/%	Output results	Relative error/%
(17)	8.04	7.5926	5.56	7.7611	3.47
(18)	7.56	7.9246	4.82	7.8958	4.44
(19)	5.82	5.6757	2.48	5.8015	0.32
(20)	4.36	4.7576	9.12	4.5990	5.48
MSE ($\text{m}^3 \cdot \text{min}^{-1}$)		0.1280		0.0620	
MAE ($\text{m}^3 \cdot \text{min}^{-1}$)		0.3385		0.2181	
MAPE (%)		5.50		3.43	

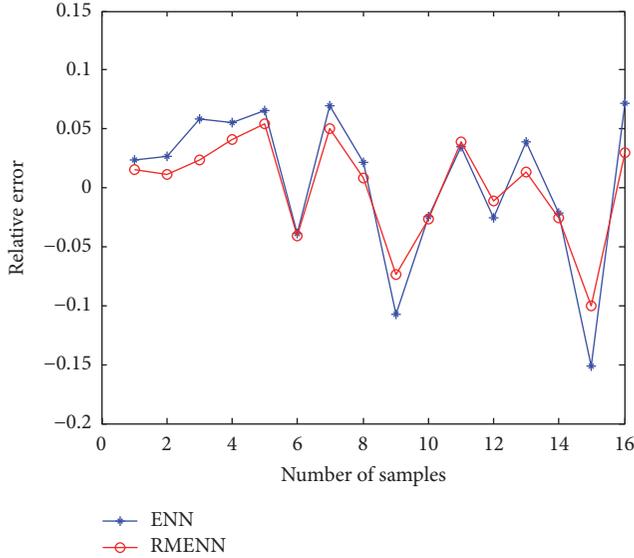


FIGURE 4: Comparison of the relative error from the classic ENN and the RMENN.

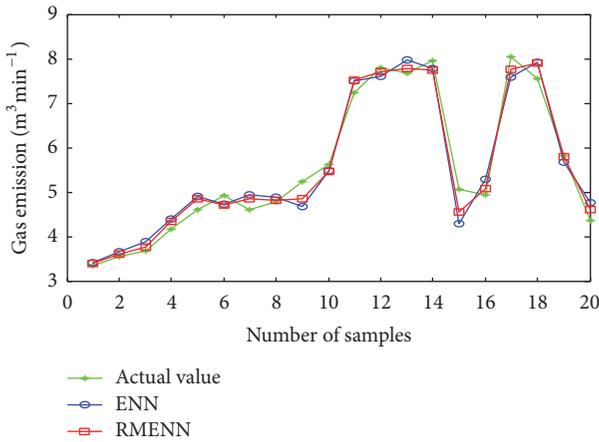


FIGURE 5: The contrast of gas emission average prediction in 50 times independent simulations.

has the better performance as evaluated by MSE, MAE, and MAPE.

To comprehensively evaluate the performance and differences significant of the two prediction models, Diebold-Mariano (DM) test and three loss functions are adopted, including MSE, MAE, and MAPE. DM test is a comparison test that focuses on the predictive accuracy and can be used to evaluate the prediction performance of the proposed hybrid model and other comparing models. The details of DM test are given as follows:

$$DM = \frac{\sum_{i=1}^T (\text{Loss}(\varepsilon_i^1) - \text{Loss}(\varepsilon_i^2)) / T}{\sqrt{S^2 / T}}, \quad (25)$$

TABLE 3: DM values based on MSE, MAE, and MAPE.

	MSE	MAE	MAPE
DM	2.1383*	3.2718***	2.6912**

* is 5% significance level; ** is 1% significance level; *** is 0.5% significance level.

where $\text{Loss}(\cdot)$ is the loss function. ε_i^1 and ε_i^2 are the prediction errors from two models. S^2 is an estimator of the variance of $\text{Loss}(\varepsilon_i^1) - \text{Loss}(\varepsilon_i^2)$. The hypothesis test is defined as

$$\begin{aligned} H_0: E(\text{Loss}(\varepsilon_i^1) - \text{Loss}(\varepsilon_i^2)) &= 0; \\ H_1: E(\text{Loss}(\varepsilon_i^1) - \text{Loss}(\varepsilon_i^2)) &\neq 0; \end{aligned} \quad (26)$$

the null hypothesis is that the two models have the same accuracy. Under the null hypothesis, the test statistics DM are asymptotically $N(0, 1)$ distributed. If $|DM| > z_{\alpha/2}$, the null hypothesis will be rejected, the two models are significantly different.

Table 3 shows that the DM value as evaluated by MSE is larger than the upper limit at the 5% significance level. The DM value as evaluated by MAE is larger than the upper limit at the 0.5% significance level. The DM value as evaluated by MAPE is larger than the upper limit at the 1% significance level.

In order to further verify the validity of the RMENN model, four sets of sample data are randomly selected for validation and other data are selected for training. Table 4 shows the error analysis of prediction results. The relative error of the ENN is larger than that of the RMENN except the tenth sample, and the errors of the ENN based on MSE, MAE, and MAPE are larger than that of the RMENN. Table 5 shows that the DM value as evaluated by MSE is not larger than the upper limit at the 10% significance level. The DM values as evaluated by MAE and MAPE are larger than the upper limits. Overall conclusions based on all the tests indicate that the RMENN model is significantly better than the ENN model.

6. Conclusion

In this paper, we analyze the drawback of the classic ENN. A novel type of network architecture called RMENN has been proposed for gas emission prediction. In theory, the convergence and stability of learning algorithm of RMENN are proved, and the approximate optimal learning rate is given. In practice, experiment analysis results on the gas emission prediction have demonstrated that RMENN has better performance in convergence rate and prediction accuracy than the class ENN, at the cost of slightly heavier structure (correction factors). Therefore, the RMENN has certain application value and the prospect.

Conflicts of Interest

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

TABLE 4: Results of the error analysis based on MSE, MAE, and MAPE.

Number	Actual data	ENN		RMENN	
		Output results	Relative error/%	Output results	Relative error/%
(2)	3.56	3.6555	2.68	3.6006	1.14
(3)	3.67	3.8832	5.81	3.7566	2.36
(10)	5.62	5.4835	2.43	5.4687	2.69
(13)	7.68	7.9757	3.85	7.7862	1.38
MSE ($\text{m}^3 \cdot \text{min}^{-1}$)		0.0402		0.0108	
MAE ($\text{m}^3 \cdot \text{min}^{-1}$)		0.1852		0.0962	
MAPE (%)		0.0369		0.0189	

TABLE 5: DM values based on MSE, MAE, and MAPE.

	MSE	MAE	MAPE
DM	1.4139	1.7445*	1.9719**

* is 10% significance level; ** is 5% significance level.

Acknowledgments

This research was supported by Foundation of Liaoning Educational Committee (Grant no. LJ2017QL021) and the National Natural Science Foundation of China (Grant no. 61304173).

References

- [1] H. Fu and L. S. Shao, *The Characteristics of Mining Gas Disaster in Coal Mine and The Fusion Prediction*, Science Press, Beijing, China, 2011.
- [2] M. C. He, X. L. Ren, and W. L. Gong, "Experimental analysis of mine pressure influence on gas emission and control," *Journal of China Coal Society*, vol. 41, no. 1, pp. 7–13, 2016.
- [3] G. Z. Li, X. J. Li, and Z. J. Meng, "Prediction on gas emission value from unmined block of mine base on grey theory," *Coal Engineering*, vol. 29, no. 9, pp. 85–87, 2010.
- [4] F. Lü, B. Liang, W.-J. Sun, and Y. Wang, "Gas emission quantity prediction of working face based on principal component regression analysis method," *Journal of the China Coal Society*, vol. 37, no. 1, pp. 113–116, 2012.
- [5] L. Wei, T. L. Bai, and H. Fu, "New gas concentration dynamic prediction model based on the EMD-LSSVM," *Journal of Safety and Environment*, vol. 16, no. 2, pp. 119–123, 2016.
- [6] X. L. Wang, J. Liu, and J. J. Lu, "Gas emission quantity forecasting based on virtual state variables and Kalmanfilter," *Journal of China Coal Society*, vol. 36, no. 1, pp. 80–85, 2011.
- [7] T. Wang, Y. Wang, C. Guo, and J. Zhang, "Application of QGA-RBF for predicting the amount of mine gas emission," *Chinese Journal of Sensors and Actuators*, vol. 25, no. 1, pp. 119–123, 2012.
- [8] B. B. Gao and J. Y. Pan, "Based on PLS associated with BP neural network for different-source gas emission prediction model of working face," *Journal of Hunan University of Science & Technology (Natural Science Edition)*, vol. 30, no. 4, pp. 14–20, 2015.
- [9] Z. Zhang, Z. Tang, S. Gao, and G. Yang, "An algorithm of chaotic dynamic adaptive local search method for Elman neural network," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 2, pp. 647–656, 2011.
- [10] Z. Zhang, Z. Tang, S. Gao, and G. Yang, "Training elman neural network for dynamic system identification using an adaptive local search algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 5, pp. 2233–2243, 2010.
- [11] C. Vairappan, H. Tamura, S. Gao, and Z. Tang, "Batch type local search-based adaptive neuro-fuzzy inference system (ANFIS) with self-feedbacks for time-series prediction," *Neurocomputing*, vol. 72, no. 7-9, pp. 1870–1877, 2009.
- [12] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowledge-Based Systems*, vol. 105, pp. 214–224, 2016.
- [13] H. W. Cui, *Site measurement study on influence factors of gas emission value from mechanized longwall coal mining face*, *Coal Science and Technology*, vol. 39, no. 11, pp. 70–72, 2011.
- [14] L. W. He, X. B. Li, and H. Lin, "Modeling and simulation of gas emission system based on CML model," *Journal of Central South University (Science and Technology)*, vol. 43, no. 12, pp. 4801–4806, 2012.
- [15] D. T. Pham and X. Liu, "Training of Elman networks and dynamic system modelling," *International Journal of Systems Science*, vol. 27, no. 2, pp. 221–226, 1996.
- [16] S. Jiang, G. Fu, L. Huang, Y. Tang, P. Cai, and H. Peng, "Forecasting on ash fusion temperatures of bituminous coal and biomass co-firing based on Elman neural network," *Journal of Central South University (Science and Technology)*, vol. 47, no. 12, pp. 4240–4247, 2016.
- [17] P.-H. Li, Y. Chai, and Q.-Y. Xiong, "Quantum gate Elman neural network and its quantized extended gradient back-propagation training algorithm," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 39, no. 9, pp. 1511–1522, 2013.
- [18] H. Fu, S. Xie, Y.-S. Xu, and Z.-C. Chen, "Gas emission dynamic prediction model of coal mine based on ACC-ENN algorithm," *Journal of the China Coal Society*, vol. 39, no. 7, pp. 1296–1301, 2014.
- [19] R. Q. Li, S. L. Shi, and A. Wu, "Research on coal mining workplace gas emission prediction method based on EMD-Elman and its application," *China Safety Science Journal*, vol. 24, no. 6, pp. 51–56, 2014.
- [20] H. Fu and W. Dai, "Dynamic prediction of Gas Emission based on LLE and BA-ENN," *Chinese Journal of Sensors and Actuators*, vol. 29, no. 9, pp. 1383–1388, 2016.
- [21] W. H. Chen, X. H. Yan, and H. Fu, "On the innovated Elman neural network for forecasting the mining gas emission," *Journal of Safety and Environment*, vol. 15, no. 3, pp. 19–24, 2015.
- [22] L. Wei, H. Fu, and Y. P. Yin, "Gas emission prediction model of coal mine based on hidden recurrent feedback Elman," *China Safety Science Journal*, vol. 26, pp. 42–46, 2016.

- [23] D. Xie, T. Feng, and C. Zhu, "Attributed measurement prediction model of entropy value for coal face gas emission and its application," *Journal of Central South University (Science and Technology)*, vol. 44, no. 6, pp. 2482–2487, 2013.

Research Article

A Systematic Optimization Design Method for Complex Mechatronic Products Design and Development

Jie Jiang,¹ Guofu Ding,¹ Jian Zhang ,¹ Yisheng Zou,¹ and Shengfeng Qin ²

¹*School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China*

²*Department of Design, Northumbria University, Newcastle upon Tyne NE1 8ST, UK*

Correspondence should be addressed to Jian Zhang; jerrysmail@263.net
and Shengfeng Qin; sheng-feng.qin@northumbria.ac.uk

Received 11 October 2017; Accepted 24 December 2017; Published 11 February 2018

Academic Editor: Qian Zhang

Copyright © 2018 Jie Jiang 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.

Designing a complex mechatronic product involves multiple design variables, objectives, constraints, and evaluation criteria as well as their nonlinearly coupled relationships. The design space can be very big consisting of many functional design parameters, structural design parameters, and behavioral design (or running performances) parameters. Given a big design space and inexplicit relations among them, how to design a product optimally in an optimization design process is a challenging research problem. In this paper, we propose a systematic optimization design method based on design space reduction and surrogate modelling techniques. This method firstly identifies key design parameters from a very big design space to reduce the design space, secondly uses the identified key design parameters to establish a system surrogate model based on data-driven modelling principles for optimization design, and thirdly utilizes the multiobjective optimization techniques to achieve an optimal design of a product in the reduced design space. This method has been tested with a high-speed train design. With comparison to others, the research results show that this method is practical and useful for optimally designing complex mechatronic products.

1. Introduction

It is very difficult to optimally design a complex mechatronic product for many reasons. First, there are a big number of design parameters, usually greater than 100. Second, there are many key performance indicators as either goals or constraints. Third, these parameters are multiple-disciplines related and their determination needs multidisciplinary collaborative efforts. Furthermore, the coupled relations among these parameters and performance indicators are highly nonlinear and vague. In a word, optimally designing a complex mechatronic product is very challenging in a big design space. Therefore, the optimization efficiency is low and it is difficult to obtain a satisfactory solution. In addition, complex mechatronic products are usually composed of many subsystems having different parameters, and their performances are correlated. It is difficult to have an effective system model to describe the relationships between parameters, subsystem performances, and the whole system performances. Thus, Data-Driven Modelling techniques such as artificial neural

networks- (ANNs-) based surrogate modelling provide alternative solutions to this problem.

Designing a complex mechatronic product optimally requires considering numerous design parameters and ways of identifying a set of best design variables and obtaining a best design solution effectively under major constraints such as safety/security and stability. The challenges are threefold: (1) the number of design variables or design space is very big; (2) these design parameters have complex system coupling relationships, and it is difficult to take all design parameters in optimization, needing a design space reduction; and (3) a practical model for predicting the coupled vast system performances involves several subsystem performances. Taking a high-speed train as an example, its dynamics performances are related to high-speed train dynamics, high-speed pantograph-catenary dynamics, high-speed train aerodynamics, and high-speed train-track coupling dynamics. Therefore, even with changes in a small set of design variables in practice, the evaluation of the system performances is not straightforward because of their coupled relationships. If each

subsystem is modelled as a component, it can be seen that some variables may affect several subsystem models and others may strongly influence only one subsystem. These design variables are coupled and affect different subsystem models. There is a need to construct an overall design performance evaluation model or a goal function for optimal design because so far there is no one established. The ultimate challenge is to develop a systematic optimization design method to solve the above challenges properly and support optimal design of complex mechatronic products.

This paper presents a systematic optimization design method for designing complex mechatronic products. At the beginning, it uses each subsystem dynamics model to conduct design parameter sensitivity analysis and identify key design parameters for design space reduction. Then, a neural network-based surrogate model is established between the identified key design parameters and key performance indicators to describe the whole system performance. Upon the neural network surrogate model, an optimization design model is developed, and finally, an optimal design computing is realized with an improved optimization algorithm for better quality and efficiency. In the current design practice, designing a typical complex mechatronic product such as a high-speed train is mainly by the trial and error method. It lacks a systematic design optimization method for its design. Therefore, we take the optimal design of a high-speed train as a case study to verify the effectiveness of the proposed method.

The paper is structured as follows. Section 2 reviews related work and Section 3 introduces the proposed systematic optimization design method for developing complex mechatronic products. Section 4 shows the case study results, followed by conclusions in Section 5.

2. Related Work

Multiobjective and multidisciplinary optimization in engineering is closely related to our research problems. Many scholars or engineers have conducted a lot of research on optimization frameworks and algorithms. Fabio et al. [1] proposed the use of design optimization techniques to find the ideal truncated full-scale design considering the dynamic effects. Wei et al. [2] proposed a comprehensive framework including a multiobjective interval optimization model and evidential reasoning approach to solve the unit-sizing problem of small-scale integrated energy systems. Lei et al. [3] built a constrained multiphysics model of a motor wheel for an electric vehicle and then optimized it. Liu et al. [4] established a multihierarchical integrated product design data model supporting the multidisciplinary design optimization (MDO) in the Web environment and a Web services-based framework considering uncertainties was proposed. Zheng and Liao [5] improved particle swarm algorithms, which could be applied to many other parameter identification and optimization problems. Zhang et al. [6] presented a modified multiobjective evolutionary algorithm based on the decomposition approach to solve an optimal power flow problem with multiple and competing objectives. Lee et al. [7] proposed a Web services-based MDO framework that enabled

the synthesis of available disciplinary and cross-disciplinary resources for MDO via the Globus Toolkit. Gong et al. [8] demonstrated a design sensitivity analysis (DSA) method. Some scholars also applied multidisciplinary optimization to deal with engineering problems in mechanical systems and multibody systems. Kuzmanovic et al. [9] considered damping optimization in a mechanical system excited by an external force. He and McPhee [10] presented a methodology for the design optimization of multibody systems by using genetic algorithms. Hosder et al. [11] considered surrogate functions as an important tool in multidisciplinary design optimization to deal with noisy functions, high computational cost, and the practical difficulty of integrating legacy disciplinary computer codes.

The above methods usually need to establish a large number of equations and formulas for derivation. Thus, their efficiency is low and the solution is not guaranteed. Some scholars proposed using surrogate models to solve optimization problems. Wang and Shan [12] described the meta-modelling techniques in support of engineering design optimization. Golovidov and Kodiyalam [13] described the ideas and methods of how to use an approximate model to do multidiscipline optimizations. Jiang et al. [14] used a neural network model to realize a multiobjective optimization involving process parameters. Kim et al. [15] combined differential and genetic methods for a suspension system design. Yuan et al. [16] promised a methodology for the optimal design of complex mechatronic systems. James and Azad [17] presented two case studies on the use of agent-based modelling in the design of complex systems. Xu et al. [18] proposed two improved strategies for supporting system design optimization. Adaptive meta-model approaches were also proposed. Yi and Malkawi [19] utilized a neural network model for energy simulation. Cheng and Lee [20] explored an efficient back-propagation neural network-based meta-model for approximating optimal solutions. Cheng and Yang studied the optimal design of suspension system parameters for high-speed trains with Kriging model [21].

However, the optimization design of complex mechatronic products is a systematic problem, which involves parameter identification, design space reduction, and optimization strategies. Some scholars studied some of the related problems. Wang [22] considered optimization strategies including sensitivity analysis, surrogate models, and searing algorithms to enable global engineering optimization. Cai et al. [23] presented a general multiagent control methodology for an energy system optimization in a “plug-and-play” manner. Park et al. [24] described the properties of sensitivity analysis between some of the suspension characteristics of the Korean high-speed train as the design variables and the dynamic performance as the response variables. Li et al. [25] presented a new meta-model-based global optimization method using fuzzy clustering for design space reduction. Forrester and Keane [26] reviewed the work on constructing surrogate models and their use in optimization strategies, while Shyy et al. [27] reviewed the fundamental issues arising in surrogate-based analysis and optimization. Zhang et al. [28] presented a new method to identify the key design variables based on the sensitivity analysis for high-speed train

design. Kim et al. [29] used a separate meta-model for each performance indicator, requiring multiple meta-models for a multiple-objective optimization. Ma et al. [30] proposed a global sensitivity analysis method by dividing variables in groups. Queipo et al. [31] applied a robust optimization design method to a real complex nonlinear system design. Coello [32] provided a comprehensive discussion on the fundamental issues arising from the use of surrogate-based analysis and optimization.

In summary, the surrogate model technology is useful for the optimization of complex electromechanical systems, and it has a successful application in the optimization of some products. There are a body of work on surrogate modelling and multiobjective optimization. However, for optimally designing complex mechatronic products, it still lacks a systematic approach to guide and guarantee the optimization design of such complex systems. Therefore, this paper proposes a systematic optimization method based on integral design space reduction and system surrogate modelling techniques for designing complex mechatronic products optimally.

3. Systematic Optimization Design Method

In order to improve the design efficiency and reduce the difficulty of performance evaluation (simulation) calculation, this paper puts forward a new systematic optimization design method based on the surrogate model and intelligent multiobjective optimization techniques for designing complex mechatronic products. It includes five stages. In Stage 1, according to the topology structure, design parameters, and boundary conditions of a complex system, the design parameters of the complex system are extracted. In Stage 2, the design parameter space is reduced by expert knowledge or by the parameter sensitivity analysis with each subsystem evaluation (simulation) model. In Stage 3, the key design parameters affecting the whole system design objectives (or running performances) are obtained, forming a reduced design space. In Stage 4, a surrogate model describing the relationship between the whole system performances and the key design parameters is established, and then a corresponding optimization model is developed based on the surrogate model. In Stage 5, intelligent optimization algorithms are used for the optimization design of a complex mechatronic product.

The system optimization design flow is shown in Figure 1. It includes (1) specifying design parameters and objectives, (2) conducting design space reduction, (3) setting up a system surrogate model, (4) setting up an optimization model, and (5) conducting optimization computing.

3.1. Specifying Design Parameters and Objectives. Figure 2 shows the optimal design problem space of a complex mechatronic product related to product structure, design parameters, and performance indicators. For a complex mechatronic product, there are many design parameters associated with a number of subsystems. The design parameters are divided into structural design parameters and performance design parameters. The number of parameters usually is very big and

the data range is very wide. At the same time, the running performances of the complex mechatronic product are synthesized in different aspects, so there are a lot of performance indicators (or design objectives).

In order to solve the optimization design problem for a complex mechatronic product, it is necessary to define what the design parameters (variables) are and what the objective indicators are. The objectives of the optimization are defined by considering the comprehensive requirements of the performances. Usually setting up a set of the design objectives is very important, while the design goal (comprehensive objective) function is usually formed by summing up all of the weighted objectives. Weights for each individual objective are determined based on their relative importance and previous design knowledge. So the system optimization is transformed into solving the maximum or minimum value of the comprehensive objective function.

3.2. Design Space Reduction. Within a huge design space, there are many parameters and the coupled relationships between these design parameters and objectives are very complex and usually nonlinear. If m is used to represent the number of design parameters and n for objectives, the design space is a $m \times n$ multidimensional problem. Because the design space is large and high dimensional, it is necessary but difficult to identify the key design variables in the optimal design.

To solve this problem, a method for design space reduction is proposed with two rounds. In round 1, important parameters are chosen by experts with related domain knowledge. In round 2, the important parameters will be used to establish surrogate models against each performance indicator, and then sensitivity analysis is conducted based on the established surrogate models. Finally, according to the sensitivity analysis results, the key design parameters are determined based on a predefined sensitivity threshold. The specific process is shown in Figure 3 and the details can be found in [28].

3.3. Setting Up a System Surrogate Model. For a complex mechatronic product, usually, there is no practical model for describing whole system performances, and instead, there are several subsystem performance (simulation) models within multidisciplinary fields. The whole system performances are nonlinearly coupled with the subsystem performances. Thus, the direct use of subsystem performance simulation models in the optimal design process is difficult because it requires a coupled system simulation method with spatiotemporal synchronization process control over subsystem simulation computing [33]. One big problem with this kind of coupled system simulation is being time consuming and requiring huge computing resources. Especially for the whole system optimal design, this requires persistently iterative simulation and optimization. The cost of computing becomes much higher. For this reason, a whole system surrogate model is proposed as a very good alternative model, to reduce the difficulty of optimization and improve the efficiency of optimization.

The surrogate model is a mathematical model for fitting discrete data using an approximation approach, which can

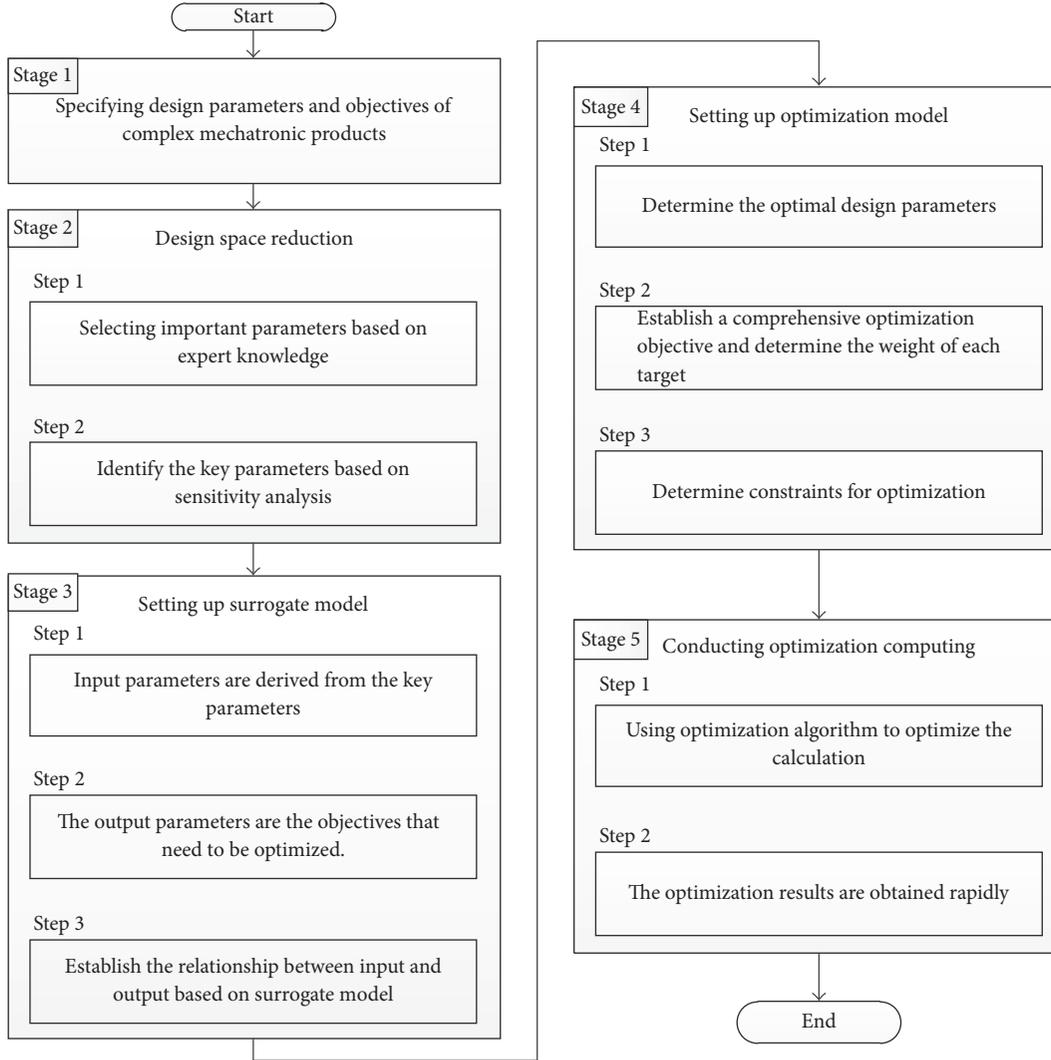


FIGURE 1: Optimization design flow.

determine key parameters and their value ranges with less training samples and advanced trial design methods. [27, 32] Back-propagation neural network (BPN) is one of these effective surrogate models, and its structure is shown in Figure 4.

The design performances of a complex product are considered at the same time with different performance indexes. When performing an optimal design, it is necessary to meet all these indexes at the same time and thus synthesize these indexes into a comprehensive goal function. Lastly, a multiobjective optimization model can be established based on the surrogate model.

Another aspect is to establish a radial basis function network for the parameters. The establishment of the back-propagation neural network (BPN) parameters includes the accuracy of the model and diffusion factor. The mean square error of the same model can also affect the adjustment of the BPN surrogate model.

For having a surrogate model with high accuracy, RRMSE (Relative Root Mean Square Error) error criterion is used in training. RRMSE of the model is defined as follows:

$$\text{RRMSE} = \sqrt{\frac{\sum_{k=1}^m (y(x^k) - \hat{y}(x^k))^2}{\sum_{k=1}^m (y(x^k) - \bar{y}(x^k))^2}}. \quad (1)$$

Among them, $y(x^k)$ is the actual values with response to the test points x^k ; $\hat{y}(x^k)$ is the actual values, with response to the test points x^k ; m is the number of the test points; x^k represents test samples set. $\bar{y}(x^k)$ refers to the mean value of the actual responses.

3.4. Setting Up the Optimization Model. The optimization model can be described with the surrogate model as follows:

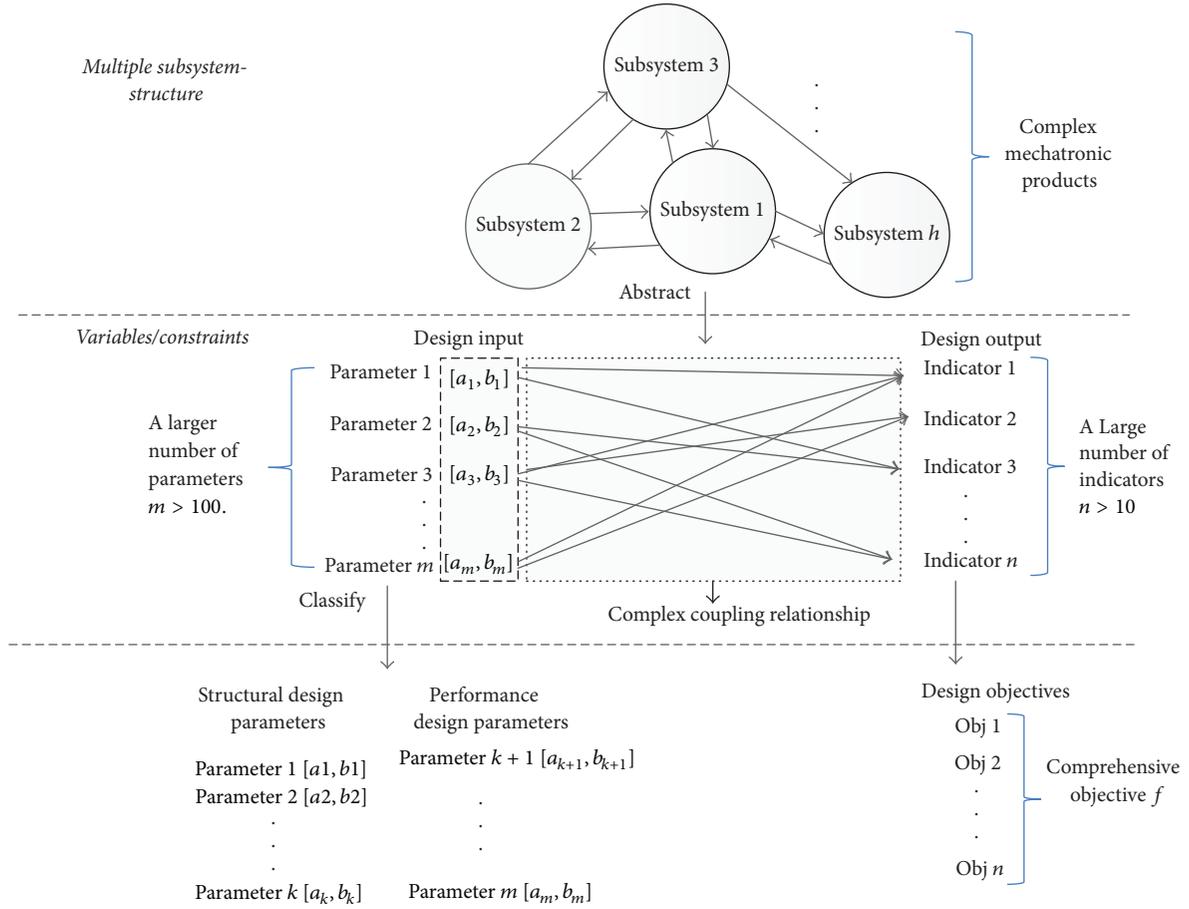


FIGURE 2: Optimal design problem space of a complex mechatronic product.

- (1) Design variables (key design parameters): $x_1, x_2, x_3, \dots, x_m$
- (2) Optimization goal function with subperformance functions (indicators): $f_1, f_2, f_3, \dots, f_n$

When constructing the optimization goal function F , we need to obtain the best overall performance indicator with a set of design parameters within their constraints. In the optimal design, the target is to obtain the minimum of F . The F can be represented in (2) with subfunctions. For some subfunctions such as $f_{m+1}, \dots, f_{n-1}, f_n$, they are transformed into the minimum value of F with their reciprocal substitutions because they are expected to achieve the biggest values in real term

$$F = \alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_m f_m + \alpha_{m+1} \frac{1}{f_{m+1}} + \dots + \alpha_{n-1} \frac{1}{f_{n-1}} + \alpha_n \frac{1}{f_n} = \sum_{i=1}^m \alpha_i f_i + \sum_{j=m+1}^n \alpha_j f_j^{-1}, \quad (2)$$

where $f_1, f_2, \dots, f_m, f_{m+1}, \dots, f_n$ represent objective 1, objective 2, ..., objective m , objective $m + 1$, and objective n , and $\alpha_1, \alpha_2, \dots, \alpha_n$ are the corresponding weight coefficients.

The multiobjective optimization is to find the minimum F solution. Thus, the goal is to obtain

$$\min (F) = \min \left(\sum_{i=1}^m \alpha_i f_i + \sum_{j=m+1}^n \alpha_j f_j^{-1} \right). \quad (3)$$

3.5. Conducting Optimization Computing. The optimization of complex mechatronic products is a very complex problem, involving many parameters, indicators, and boundary conditions. Meanwhile, it also requires considering the efficiency and accuracy of optimization. Intelligent optimization algorithm is a good method for speeding up the optimization. Intelligent optimization algorithms include genetic algorithm, differential evolution algorithm, and particle swarm algorithm. Cai and Aref [34] developed a genetic algorithm-(GA-) based optimization procedure. Zheng and Liao [5] realized parameter identification of nonlinear dynamic systems using an improved particle swarm optimization. Qin et al. [35] utilized the differential evolution algorithm (DE) for global numerical optimization.

Each algorithm has its own characteristics. DE (differential evolution) is a parallel optimization algorithm evolved from GA (genetic algorithm), and it has excellent characteristics for global optimization. We select DE in our application

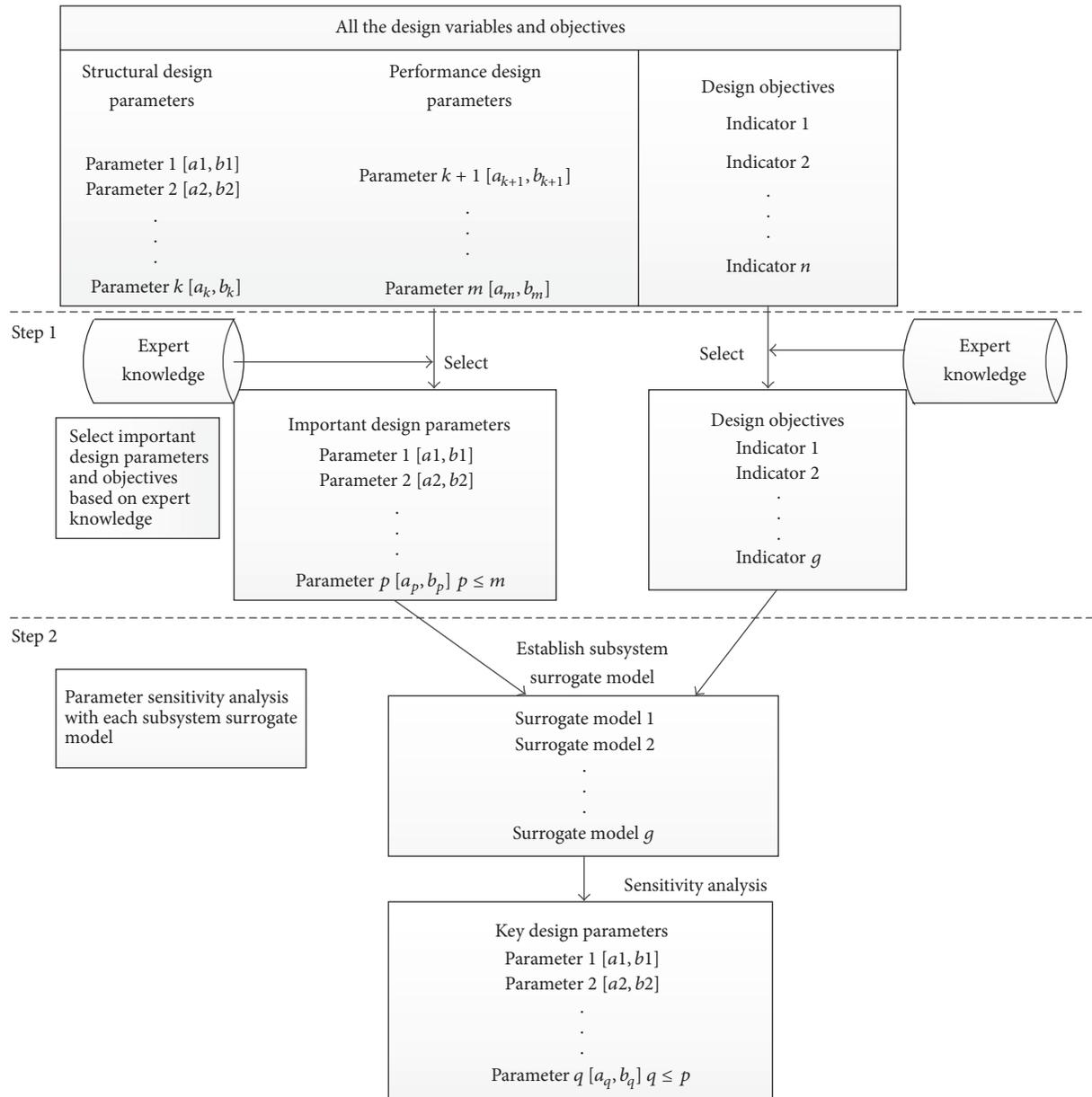


FIGURE 3: The flow of design space reduction.

because DE reportedly has a more effective evolutionary strategy [35] to generate new individuals, which makes it an efficient and powerful population-based stochastic search technique for solving optimization problems over continuous space and the implementation of DE is relatively easy.

The novelty of this systematic optimization design method has twofold. Firstly, it can effectively couple design space reduction and the system surrogate modelling techniques into the system optimization modelling. In the existing literatures, these two techniques are discussed separately; thus, when applying the surrogate modelling technique to develop a surrogate model for describing complex system relationships, there is a general difficulty in determining what parameters in both inputs and outputs should be chosen

to establish an effective surrogate model for use in the optimization. Secondly, it demonstrates that the identified key variables from the space reduction are well qualified as the input variables for establishing the corresponding system surrogate model and the optimization model, thus, providing a general form of the system optimization modelling and solving techniques for complex mechatronic product optimal design.

4. Case Study

Here, we demonstrate the proposed optimal design method with a high-speed train design. We validate the system optimization from the following three aspects. The first is to

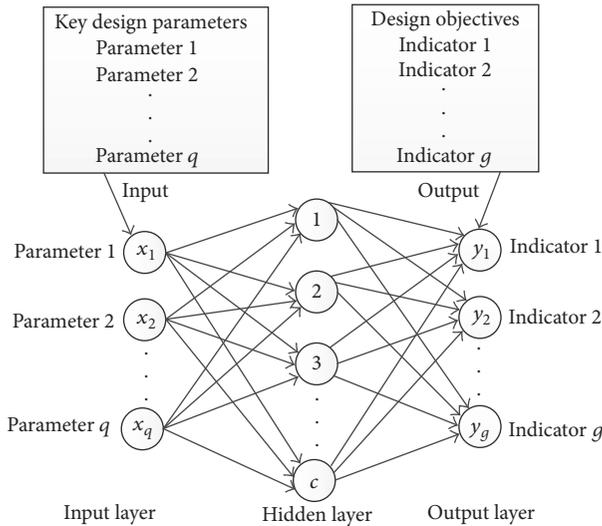


FIGURE 4: The framework of the BPN.

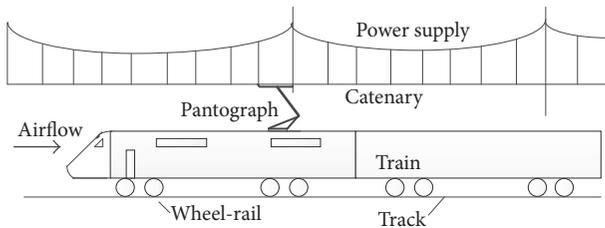


FIGURE 5: High-speed train and its running environment.

approve that the proposed systematic method is necessary and solution-guaranteed by the comparison of a direct surrogate model without design space reduction and the one with this operation. The second is to compare the convergence speed and accuracy of the surrogate model under different design parameters. This indicates that, without a design space reduction operation, the resultant surrogate model will perform quite differently. The last is to prove the effectiveness of the system method in terms of its efficiency and accuracy. We overall demonstrate the usefulness of the method by comparing three different tests.

4.1. Specifying Design Parameters and Objectives of High-Speed Train. High-speed train is a typical complex mechatronic product. As a means of rapid transportation, it has been widely appreciated and vigorously developed. However, design and development of a high-speed train need to evaluate its dynamics behaviors and performances under various running environments to meet its safety and running performance requirements. It has several subsystems (shown in Figure 5), and its total number of design parameters are very big (more than 100), involving parameters related to structure design, mechanical design, bogie design, dynamics performance design, and so on.

Therefore, here, we only take dynamics performance-related design as an example because it is the top level of

design for high-speed trains. The running performances of a high-speed train include 7 performance indicators as our design objectives; the number of initial design parameters possibly affecting the performance indicators is more than 100 (shown in Table 1). In addition, some performance indicators might be affected and coupled with other indicators. For instance, its safety indicator is mainly influenced and affected by coupled system dynamics among the train, the track, the catenary, and the airflow. The 7 performance indicators are lateral stability f_1 , vertical stability f_2 , derailment coefficient f_3 , the ratio of wheel load reduction f_4 , lateral wheelset force f_5 , overturning coefficient f_6 , and critical speed f_7 .

Due to the complexity of mechatronic products, the optimal design of a mechatronic product is a very complex problem. Some papers suggest that the surrogate modelling technique can be applied to this problem [12–15]. When focusing on high-speed train, we choose the typical neural network surrogate model based on design space reduction and intelligent optimization algorithm to optimize its design.

The number of initial design parameters of high-speed train is more than 100. The relationship between design parameters is very complicated and highly nonlinearly coupled in the wheel/rail contact model. The construction of a surrogate model of a complex electromechanical product requires big enough samples to train the model. Therefore, in order to get enough good samples, a Railway System Dynamics simulation Package SIMPACK Rail is utilized to generate a set of corresponding data between a set of inputs of design variables and a set of performance indicators. When we prepare the design variable values for simulation, there is a difficulty in knowing the right value range of each design parameter. Thus, we use experts' guessed values as our references.

Figure 6 illustrates a high-speed train topological structure and its dynamics model. The software Package SIMPACK Rail is an “add-on” module for use with SIMPACK to simulate rail system dynamics (http://www.simpack.com/uploads/media/datasheet_wheel-rail.pdf). In the SIMPACK software, the design parameters are the inputs to the system and then SIMPACK uses embedded wheel/rail contact model to calculate and output performance indicators such as Ride Comfort indicator and Derail coefficient. It is worth noting that using such system dynamics analysis software is quite time consuming, but it is still doable and cheaper comparing with real tests; thus, we use SIMPACK Rail to generate our sampling data.

Next, we take a unified approach to determine the range of each parameter. Based on the initial range value of the parameter, the upper and lower 10% are used as the initial range value of each parameter. We use this range to carry out the experimental design to get hundreds of test data sets. We use the experimental data as input and use the SIMPACK software to simulate performances of each sample design. Due to the mismatch of parameters, only 58 sets of 100 can produce simulation results. For the remaining 42 sets, we cannot get simulation results; thus, as a result, there are not enough samples to train the surrogate model with only 58% of the calculation results.

TABLE 1: The all design parameters of high-speed train.

Parameter	Name	Unit
Parameter 1	Nominal wheel radius	mm
Parameter 2	Distance between backs of wheel flanges	mm
Parameter 3	Wheelset roll moment of inertia	Kg·m ²
Parameter 4	Wheelset yaw moment of inertia	Kg·m ²
Parameter 5	Longitudinal stiffness of primary suspension per axle side	KN/m
Parameter 6	Vertical damping of primary suspension per axle side	KN·s/m
Parameter 7	Longitudinal stiffness of axle box tumbler joint per axle side	MN/m
Parameter 8	Yaw damper lateral span	mm
Parameter 9	Lateral stiffness of secondary suspension per bogie side	KN/m
Parameter 10	Vertical stiffness of secondary suspension per bogie side	KN/m
Parameter 11	Secondary vertical damper	KN·s/m
Parameter 12	Secondary lateral damper	KN·s/m
Parameter 13	Wheelset mass	Kg
Parameter 14	Lateral stiffness of axle box tumbler joint per axle side	MN/m
Parameter 15	Longitudinal stiffness of secondary suspension per bogie side	KN/m
Parameter 16	Longitudinal stiffness of Yaw damper joint per bogie side	MN/m
Parameter 17	Carbody roll moment of inertia	Kg·m ²
Parameter 18	Lateral distance between the secondary suspension of the two sides of the bogie	mm
Parameter 19	Carbody mass	Kg
Parameter 20	Lateral stiffness of primary suspension per axle side	KN/m
Parameter 21	Longitudinal distance between bogie centers	mm
Parameter 18	Carbody yaw moment of inertia	Kg·m ²
Parameter 19	Vertical distance from the rail surface to the center of gravity	mm
Parameter 20	Wheelbase	mm
Parameter 21	Vertical stiffness of primary suspension per axle side	KN/m
Parameter 22	Carbody pitch moment of inertia	Kg·m ²
Parameter 23	Framework mass	Kg
Parameter 24	Wheelset pitch moment of inertia	Kg·m ²
Parameter 25	Vertical damping joint stiffness per axle side	MN/m
Parameter 26	Nominal wheel radius	mm
Parameter 27	Distance between backs of wheel flanges	mm
Parameter 28	Wheelset roll moment of inertia	Kg·m ²
Parameter 29	Air spring vertical stiffness (per spring)	KN/m
⋮	⋮	⋮
Parameter 100	Lateral damper joint stiffness of secondary suspension per bogie side	MN/m
Parameter 101	Lateral stop clearance of secondary suspension	mm
Parameter 102	Vertical damping transverse span of secondary suspension	mm
Parameter 103	Yaw damper lateral span	mm
Parameter 104	Traction drawbar mass	Kg
Parameter 105	Swing stiffness of traction joint of secondary suspension	Nm/rad

Therefore, it is not feasible to use the full parameters in the original design space to establish a surrogate model. In order to solve this problem, we need a system method to have a guaranteed solution. Thus, in this paper we propose a method with design space reduction as a key step.

4.2. Design Space Reduction. The proposed design space reduction method has two rounds. In round 1, important parameters are selected by experts with related domain knowledge. In round 2, the important parameters will be used to establish surrogate models against each performance

TABLE 2: The design parameters sorted as the importance.

Parameter	Name	Unit	Section
x_1	Nominal wheel radius	mm	395–430
x_2	Distance between backs of wheel flanges	mm	1,351–1,355
x_3	Wheelset roll moment of inertia	Kg·m ²	500–750
x_4	Wheelset yaw moment of inertia	Kg·m ²	500–800
x_5	Longitudinal stiffness of primary suspension per axle side	KN/m	800–1,150
x_6	Vertical damping of primary suspension per axle side	KN-s/m	10–30
x_7	Longitudinal stiffness of axle box tumbler joint per axle side	MN/m	5–10
x_8	Yaw damper lateral span	mm	2,400–2,800
x_9	Lateral stiffness of secondary suspension per bogie side	KN/m	100–200
x_{10}	Vertical stiffness of secondary suspension per bogie side	KN/m	120–300
x_{11}	Secondary vertical damper	KN-s/m	20–60
x_{12}	Secondary lateral damper	KN-s/m	30–50
x_{13}	Wheelset mass	Kg	1,800–2,200
x_{14}	Lateral stiffness of axle box tumbler joint per axle side	MN/m	4–10
x_{15}	Longitudinal stiffness of secondary suspension per bogie side	KN/m	100–200
x_{16}	Longitudinal stiffness of Yaw damper joint per bogie side	MN/m	5–13
x_{17}	Carbody roll moment of inertia	Kg·m ²	70,000–120,000
x_{18}	Lateral distance between the secondary suspension of the two sides of the bogie	mm	2,400–2,500
x_{19}	Carbody mass	Kg	28,000–40,000
x_{20}	Lateral stiffness of primary suspension per axle side	KN/m	800–1,200
x_{21}	Longitudinal distance between bogie centers	mm	17,000–18,000
x_{22}	Carbody yaw moment of inertia	Kg·m ²	1,100,000–1,500,000
x_{23}	Vertical distance from the rail surface to the center of gravity	mm	1,400–1,600
x_{24}	Wheelbase	mm	2,400–2,600
x_{25}	Vertical stiffness of primary suspension per axle side	KN/m	1,000–1,500
x_{26}	Carbody pitch moment of inertia	Kg·m ²	1,200,000–1,700,000
x_{27}	Framework mass	Kg	2,100–3,100
x_{28}	Wheelset pitch moment of inertia	Kg·m ²	65–100
x_{29}	Vertical damping joint stiffness per axle side	MN/m	3–6

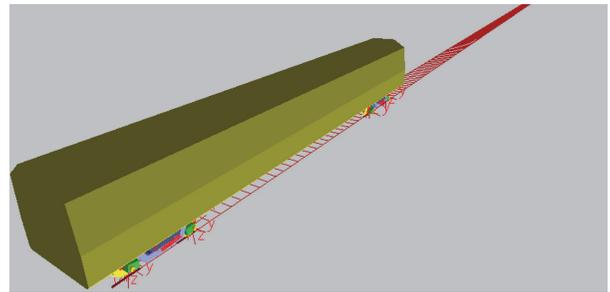
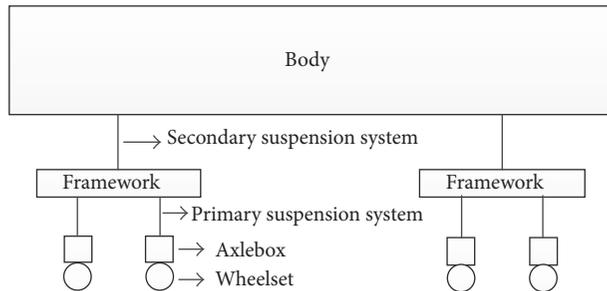


FIGURE 6: Topological structure and dynamics model in SIMPACK software of high-speed train.

indicator, and then sensitivity analysis is conducted based on established surrogate models.

The details of the design space reduction technique have been reported in [28] with some domain expert involvement. As a result, in round 1 reduction, the 29 important design parameters (shown in Table 2) are selected by experts from more than 100 parameters and in round 2 reduction, the 16

key design variables are finally identified from the sensitivity analysis based on individual performance indicator models [28]. According to the sensitivity analysis, the parameters are sorted according to their importance (shown in Table 2).

4.3. *Setting Up the System Surrogate Model.* Based on the results obtained from the design space reduction, the system

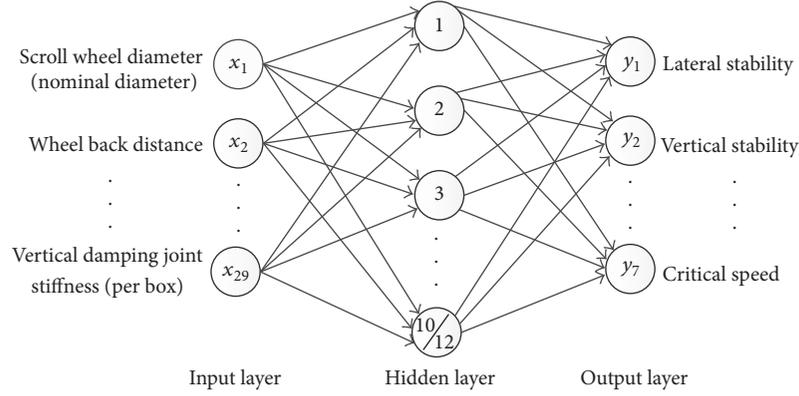


FIGURE 7: The surrogate model structure of high-speed train based on different parameter groups.

TABLE 3: Six surrogate models.

	BPN1	BPN2	BPN3	BPN4	BPN5	BPN6
Data set	U	V	W	X	Y	Z
The number of parameters	12	16	18	20	25	29
The number of iterations	68	93	126	195	256	291

surrogate model with the 16 key design parameters and performance indicators was established. In order to prove the system approach and justify the reason for choosing the 16 key design parameters in the system surrogate model (named as BPN2), we construct other 5 surrogate models for comparison. The result comparisons with the 6 surrogate models are detailed in the next section.

We use the same structure (see Figure 7) to construct the 6 surrogate models, namely, BPN1, BPN2, BPN3, BPN4, BPN5, and BPN6, with the numbers of design parameters: 12, 16, 18, 20, 25, and 29, respectively. When a parameter in Table 2 is not chosen as a design variable, its value is fixed to the middle of its range values. In this way, the sampling design of a design available is based on Latin hypercube sample design method and the sample data are produced with SIMPACK software. The details associated with the six surrogate models BPN1 to BPN6 are shown in Table 3.

Figure 7 shows the structure of the neural network surrogate model. For example, when n equals 29, the 29 design variables are in the surrogate model. The 29 variables first generate 100 samples, and then these samples are inputted into SIMPACK simulation software to generate the corresponding performance indicator values. Because these parameters range differently, all parameters in the input layer and the output layer are then normalized for training a model with 95 samples and testing the model with the other 5 samples in its establishment process. In this way, we establish the six surrogate models.

The convergence rates of these surrogate models are shown in Figure 8. The numbers of iterations for each model are shown in Table 3 (last row). It is clear that the number of iterations increases as the sampling parameter increases. The maximum number of times is 5 times more than the

minimum number. Therefore, the more the sampling parameters are, the slower the convergence of the model is, and the more resources and time it takes.

4.4. Setting Up the Optimization Model. Design variables are corresponding to the 6 BPN models, and they are a subset of $\{x_1, x_2, x_3, \dots, x_{29}\}$ for each model (shown in Table 2), where $0 \leq x_j \leq 1$, $j = 1, 2, \dots, 29$. According to the design requirements of the high-speed train, the main design objectives are the 7 performance indicators: f_1, f_2, \dots, f_7 .

According to the design requirements and specifications of high-speed train in China, the range of the 7 indexes can be obtained. In this way, we regard the performance requirements as the boundary conditions of the optimization design. Constraints are

$$\begin{aligned}
 0 &< f_1 < 2.5, \\
 0 &< f_2 < 2.5, \\
 0 &< f_3 < 0.8, \\
 0 &< f_4 < 0.8, \\
 f_5 &> 0, \\
 0 &< f_6 < 0.8, \\
 f_7 &> 0;
 \end{aligned} \tag{4}$$

with these boundary conditions for f_1 to f_6 , the smaller the better regarding the performance of high-speed train. However, to critical speed f_7 , the higher the better regarding performance of high-speed train. With the reciprocal of f_7 into the optimization function, we get the goal function.

The function is

$$\begin{aligned}
 \min \quad &(F) \\
 &= \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4 + \alpha_5 f_5 + \alpha_6 f_6 \\
 &\quad + \alpha_7 \frac{1}{f_7},
 \end{aligned} \tag{5}$$

where α is the weight coefficient. For high-speed trains, we think the 7 indicators are of the same importance. In

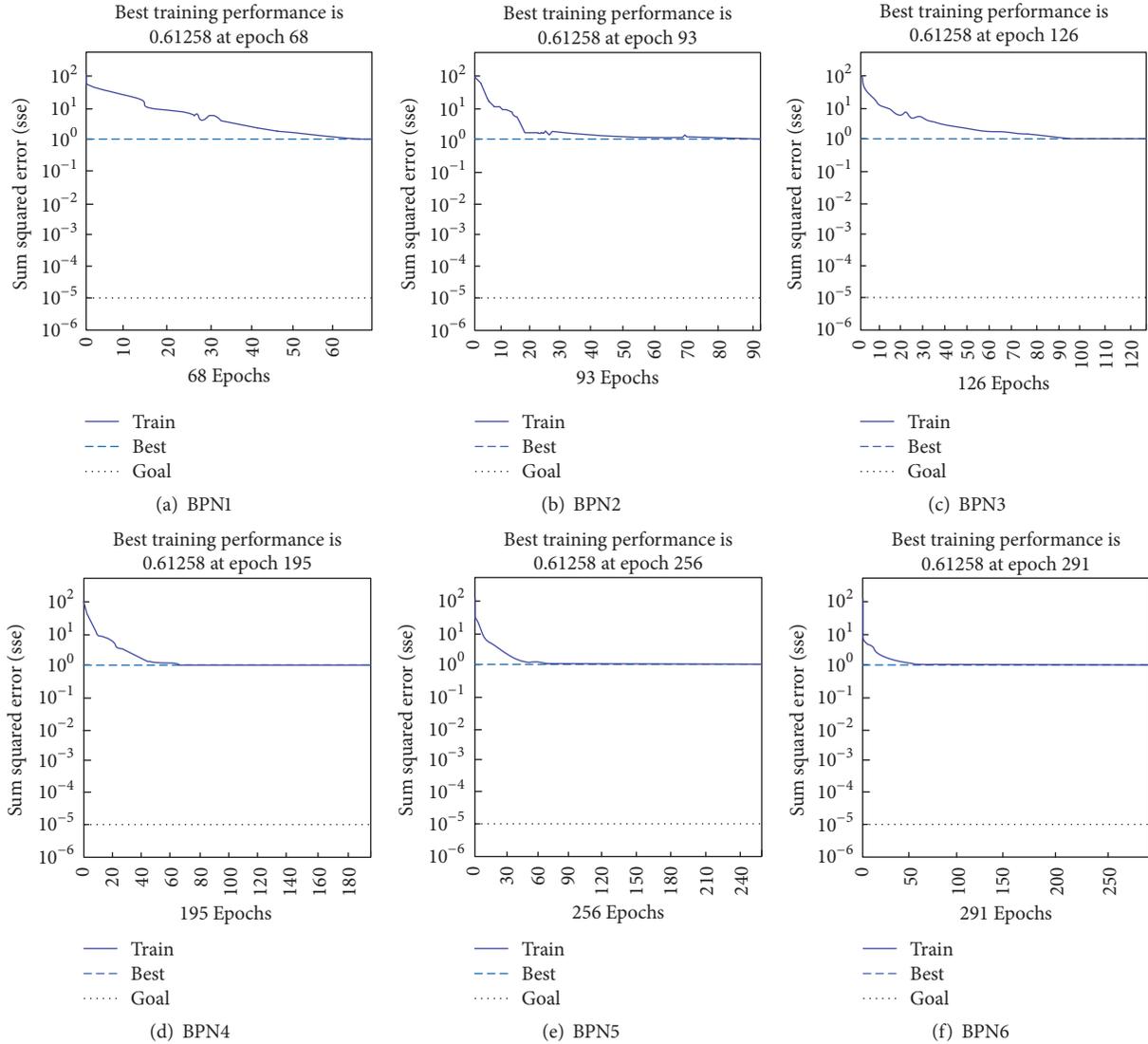


FIGURE 8: The convergence rate of BPN surrogate models.

practice, because the different units of the variables need to be normalized into dimensionless variables, the weight coefficient can be set to a specific value:

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = \alpha_7 = \frac{1}{7}. \quad (6)$$

The optimization function is simplified as

$$\min (F) = f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + \frac{1}{f_7}. \quad (7)$$

4.5. Conducting Optimization Computing of High-Speed Train. In order to compare the optimization results, we use each of the 6 surrogate models as an optimization analyzer to get the corresponding results between the input variables and 7 key indicator outcomes; in this way, we establish 6 optimization models corresponding to the BPN1, BPN2, BPN3, BPN4, BPN5, and BPN6 surrogate models. We use the same

optimization algorithm (differential evolution algorithm) to solve the 6-optimization models. The optimization algorithm is implemented within MATLAB.

The optimization times for each case are shown in Figure 9. For example, it takes 2.2 minutes to obtain the optimization result from BPN1 with 12 variable parameters. Figure 9 shows that the optimization time with BPN6 is 14 times of that with BPN1 and 10 times of that with BPN2. Therefore, the number of design variables involved in optimization affects the optimization time greatly and it is necessary to identify a suitable number of parameters in optimization with acceptable quality of the results.

The optimization results are shown in Table 4 with varied performance parameters in the goal function. From Table 4, it can be seen that the optimization method is with good accuracy and effectiveness. In comparison with the initial performance parameters, the corresponding optimization results with the performance improvement percentages are shown in Table 5.

TABLE 4: Comparison table of optimization results.

	Lateral stability	Vertical stability	Derailment coefficient * 10	The ratio of wheel load reduction * 10	Lateral wheelset force/10 (KN)	Overturning coefficient * 10	Critical speed/200 (km/h)
Initial performances	2.3804	2.0245	1.497	1.978	1.2232	1.98	2.98
BPN1	2.2162	2.0235	1.568	2.104	1.12513	2.054	3.03
BPN2	2.1935	2.0021	1.272	1.828	1.02541	1.939	3.43
BPN3	2.1925	2.001	1.253	1.82	1.02465	1.915	3.40
BPN4	2.192	2.0003	1.248	1.806	1.01683	1.867	3.44
BPN5	2.178	1.9986	1.232	1.79	1.01485	1.819	3.35
BPN6	2.1715	1.9927	1.211	1.781	1.01312	1.804	3.39

TABLE 5: The table of performance improvement percentage (%).

	Lateral stability	Vertical stability	Derailment coefficient	The ratio of wheel load reduction	Lateral wheel force	Overturning coefficient	Critical speed
BPN1	6.90%	0.05%	-4.74%	-6.37%	8.02%	-3.74%	1.68%
BPN2	7.85%	1.11%	15.03%	7.58%	16.17%	2.07%	15.22%
BPN3	7.89%	1.11%	16.30%	7.58%	16.23%	3.28%	14.28%
BPN4	7.91%	1.19%	16.63%	8.70%	16.87%	5.70%	15.69%
BPN5	8.50%	1.28%	17.70%	9.50%	16.87%	8.13%	12.62%
BPN6	8.78%	1.36%	19.11%	9.96%	17.17%	8.89%	14.30%

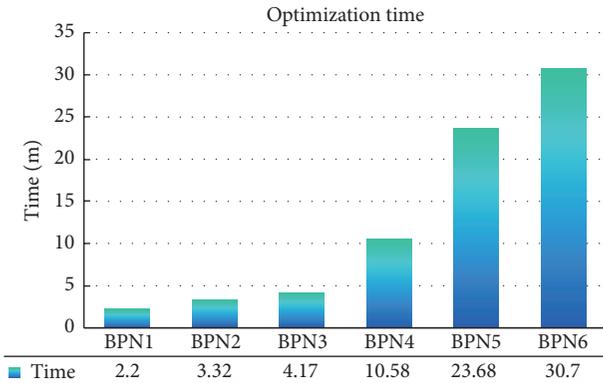


FIGURE 9: The time histogram required for obtaining optimal results based on six surrogate models.

In order to show the optimization results more intuitively, the optimized performance results in Table 4 are shown in Figure 10, while the performance improvement percentages in Table 5 are illustrated in Figure 11.

Figures 9, 10, and 11 together lead to three new understandings of the big system optimization. First, when the number of design variables is inadequate comparing to the number of the key variables, the optimization results are quite poor. Therefore, we need to avoid this kind of case. Second, when the number of the design variables is bigger than that of key variables, the optimization results are better. Third, when the number of the design variables is much bigger than the number of key variables, the optimization result is just slightly better but the optimization times are far worse. For example, the results from BPN5 with 25 variables are not significantly

improved when comparing with BPN2, but the computing time and resources are greatly increased.

In summary, using just the key variables can give a system surrogate model with a better balance between the quality of optimization results and computational costs. This is a balanced system solution. In our case study, the surrogate model BPN 2 with 16 key variables is identified as the system surrogate model and supports the system optimization very well with a good balance between the optimization quality and time. It can be concluded that the system optimization method based on the system surrogate model with only key design parameters is more effective in terms of optimization accuracy and computational cost. The importance of the design space reduction and the effectiveness of the system optimization method for complex mechatronic products are proved.

5. Conclusion

Due to the complexity of complicated mechatronic products, the multiplicity of parameters, and the intricate relationship between design parameters and performance indicators, the optimal design of such a product is very sophisticated with hard problems. Therefore, it is almost impossible to ensure that all parameters are optimal. Because of the complex relationship among subsystems, the whole system simulation model is difficult to build, and the coupled simulation time is too long and the cost is huge, it is necessary to provide a data-driven modelling and optimization design method for complex mechatronic product design from a systemic perspective to balance the system executing time and effectiveness.

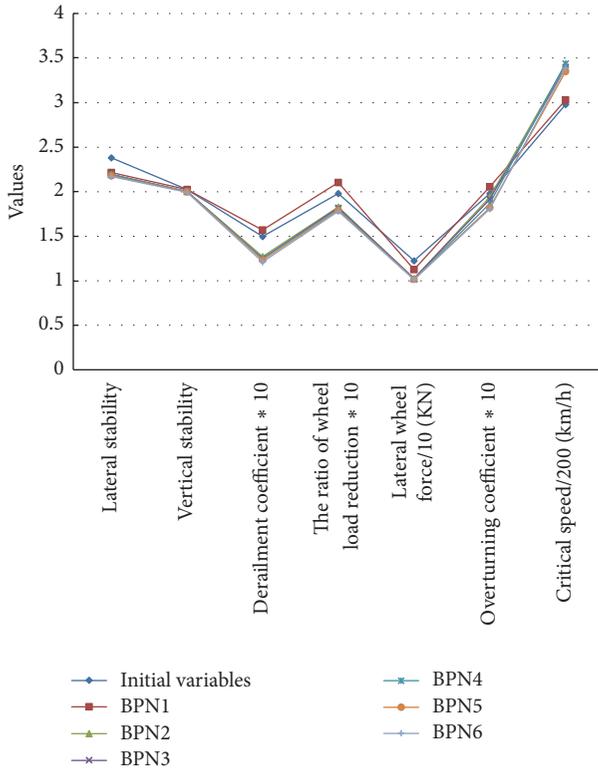


FIGURE 10: Comparison chart of optimization results.

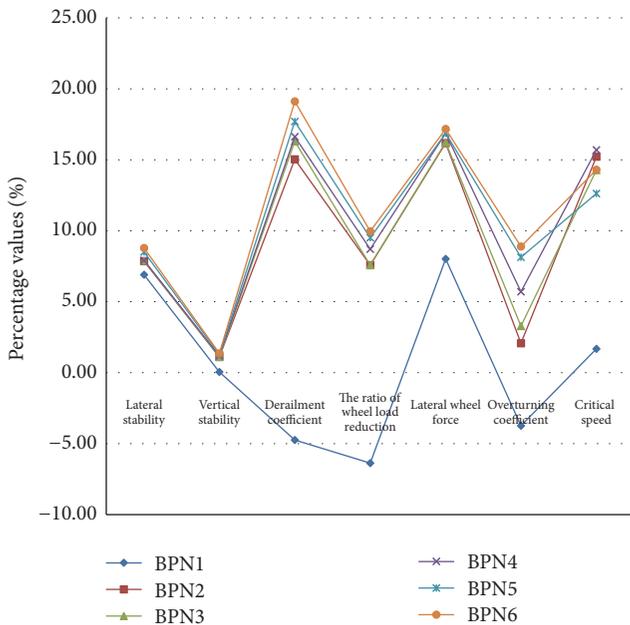


FIGURE 11: The chart of performance improvement percentage.

This paper has proposed a systematic design optimization method for complex mechatronic products from the identification of initial design parameters and objectives, to design space reduction for key variable identification, setting up a system surrogate model with just the key variables,

establishing a system optimization model, and optimization computing.

The implementation of this method has been demonstrated through a case study of China high-speed train design with 6 different surrogate models. From the comparison study, it can be seen that the appropriate design space reducing is very important, which leads to not only the identification of key variables but also the establishment of the system surrogate and optimization models. The system optimization method based on the system surrogate model established with just key design parameters is more effective in terms of optimization accuracy and computational cost. The ability to produce a better design solution with the proposed method has been validated and demonstrated.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Fund of Intelligent Mapping Research for Requirement Driven Fine Design Process of High-Speed Railway Vehicle (no. 51575461), China.

References

- [1] M. G. Fabio, N. L. Eduardo, and M. B. Silvana, "Dynamic design optimization of an equivalent truncated mooring system," *Ocean Engineering*, vol. 122, pp. 186–201, 2016.
- [2] F. Wei, Q. H. Wu, Z. X. Jing, J. J. Chen, and X. X. Zhou, "Optimal unit sizing for small-scale integrated energy systems using multi-objective interval optimization and evidential reasoning approach," *Energy*, vol. 111, pp. 933–946, 2016.
- [3] F. Lei, B. Du, X. Liu, X. Xie, and T. Chai, "Optimization of an implicit constrained multi-physics system for motor wheels of electric vehicle," *Energy*, vol. 113, pp. 980–990, 2016.
- [4] C. W. Liu, X. X. Jin, and L. S. Li, "A web services-based multi-disciplinary design optimization framework for complex engineering systems with uncertainties," *Computers in Industry*, vol. 65, no. 4, pp. 585–597, 2014.
- [5] Y.-X. Zheng and Y. Liao, "Parameter identification of nonlinear dynamic systems using an improved particle swarm optimization," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 19, pp. 7865–7874, 2016.
- [6] J. Zhang, Q. Tang, P. Li, D. Deng, and Y. Chen, "A modified MOEA/D approach to the solution of multi-objective optimal power flow problem," *Applied Soft Computing*, vol. 47, pp. 494–514, 2016.
- [7] H.-J. Lee, J.-W. Lee, and J.-O. Lee, "Development of Web services-based Multidisciplinary Design Optimization framework," *Advances in Engineering Software*, vol. 40, no. 3, pp. 176–183, 2009.
- [8] S. G. Gong, G. L. Xie, J. P. Zhang, S. H. Nie, and Y. M. Li, "Sensitivity analysis and shape optimization based on FE-EFG coupled method," *Research in Engineering Design*, vol. 20, no. 2, pp. 117–128, 2009.
- [9] I. Kuzmanovic, Z. Tomljanovic, and N. Truhar, "Damping optimization over the arbitrary time of the excited mechanical

- system,” *Journal of Computational and Applied Mathematics*, vol. 304, pp. 120–129, 2016.
- [10] Y. P. He and J. McPhee, “Multidisciplinary optimization of multibody systems with application to the design of rail vehicles,” *Multibody System Dynamics*, vol. 14, no. 2, pp. 111–135, 2015.
- [11] S. Hosder, L. T. Watson, and B. Grossman, “Polynomial Response Surface Approximations for the Multidisciplinary Design Optimization of a High Speed Civil Transport,” *Optimization and Engineering*, vol. 2, pp. 431–452, 2001.
- [12] G. G. Wang and S. Shan, “Review of metamodeling techniques in support of engineering design optimization,” *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380, 2007.
- [13] O. Golovidov and S. Kodiyalam, “Flexible Object-based Implementation of Approximation Models in an MDO Framework,” in *Int. J. Product Process Improvement*, pp. 388–404, 4, 388–0, 1999.
- [14] Z. Jiang, Q. Wang, P. Zhao, and C. Wu, “Optimization of high-pressure die-casting process parameters using artificial neural network,” *The International Journal of Advanced Manufacturing Technology*, vol. 44, no. 7–8, pp. 667–674, 2009.
- [15] J. Kim, K. Park, S. Yoon et al., “Design and optimization of suspension with optical flying head using integrated optimization frame,” *The International Journal of Advanced Manufacturing Technology*, vol. 1, no. 2, pp. 161–168, 2005.
- [16] W. Q. Yuan, Y. S. Liu, J. J. Zhao, and H. Wang, “Pattern-based integration of system optimization in mechatronic system design,” *Advances in Engineering Software*, vol. 98, pp. 23–37, 2016.
- [17] H. James and M. Azad, “Integrate agent-based modeling and optimization in complex systems analysis,” in *Proceedings of the Conference on Systems Engineering Research*, Redondo Beach, CA, 2014.
- [18] W. L. Xu, A. Chong, O. T. Karaguzel, and K. P. Lam, “Improving evolutionary algorithm performance for integer type multi-objective building system design optimization,” *Energy and Buildings*, vol. 127, pp. 714–729, 2016.
- [19] Y. K. Yi and A. M. Malkawi, “Integrating neural network models with computational fluid dynamics (CFD) for site-specific wind condition,” *Building Simulation*, vol. 4, no. 3, pp. 245–254, 2011.
- [20] Y. C. Cheng and C. K. Lee, “Integration of uniform design and Kriging interpolation to the optimization of suspension parameters of a high speed railway vehicle,” in *Proceedings of the 2010 International Conference on Modeling, Identification and Control*, 19, 17 pages, Okayama, Japan, 2010.
- [21] B. Cheng and S. Yang, “Multi-sources distinguishing of vector transducer via differential evolution,” *Journal of Marine Science and Application*, vol. 5, no. 4, pp. 48–53, 2006.
- [22] C. Wang, “Insights from developing a multidisciplinary design and analysis environment,” *Computers in Industry*, vol. 65, no. 4, pp. 786–795, 2014.
- [23] J. Cai, D. Kim, R. Jaramillo, J. E. Braun, and J. Hu, “A general multi-agent control approach for building energy system optimization,” *Energy and Buildings*, vol. 127, pp. 337–351, 2016.
- [24] C. Park, Y. Kim, and D. Bae, “Sensitivity analysis of suspension characteristics for Korean high speed train,” *Journal of Mechanical Science and Technology*, vol. 23, no. 4, pp. 938–941, 2009.
- [25] Y. Li, L. Liu, T. Long, and W. Dong, “Metamodel-based global optimization using fuzzy clustering for design space reduction,” *Chinese Journal of Mechanical Engineering*, vol. 26, no. 5, pp. 928–939, 2013.
- [26] A. I. J. Forrester and A. J. Keane, “Recent advances in surrogate-based optimization,” *Progress in Aerospace Sciences*, vol. 45, no. 1–3, pp. 50–79, 2009.
- [27] W. Shyy, Y.-C. Cho, W. Du, A. Gupta, C.-C. Tseng, and A. M. Sastry, “Surrogate-based modeling and dimension reduction techniques for multi-scale mechanics problems,” *Acta Mechanica Sinica*, vol. 27, no. 6, pp. 845–865, 2011.
- [28] J. Zhang, G. F. Ding, Y. S. Zhou, J. Jiang, X. Ying, and S. F. Qin, “Identification of key design parameters of high-speed train for optimal design,” *The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 1–4, pp. 251–265, 2014.
- [29] Y.-G. Kim, C.-K. Park, H.-S. Hwang, and T.-W. Park, “Design optimization for suspension system of high speed train using neural network,” *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 46, no. 2, pp. 727–735, 2003.
- [30] T. Ma, W. Zhang, Y. Zhang, and T. Tang, “Multi-parameter sensitivity analysis and application research in the robust optimization design for complex nonlinear system,” *Chinese Journal of Mechanical Engineering*, vol. 28, no. 1, pp. 55–62, 2015.
- [31] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker, “Surrogate-based analysis and optimization,” *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.
- [32] C. A. C. Coello, “A comprehensive survey of evolutionary-based multiobjective optimization techniques,” *Knowledge and Information Systems*, vol. 1, no. 3, pp. 269–308, 1999.
- [33] Y. Zou, G. Ding, W. Zhang, J. Zhang, S. Qin, and J. K. Tan, “Collaborative simulation method with spatiotemporal synchronization process control,” *Chinese Journal of Mechanical Engineering*, vol. 29, no. 6, pp. 1074–1082, 2016.
- [34] H. Cai and A. J. Aref, “A genetic algorithm-based multi-objective optimization for hybrid fiber reinforced polymeric deck and cable system of cable-stayed bridges,” *Structural and Multidisciplinary Optimization*, vol. 52, no. 3, pp. 583–594, 2015.
- [35] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.