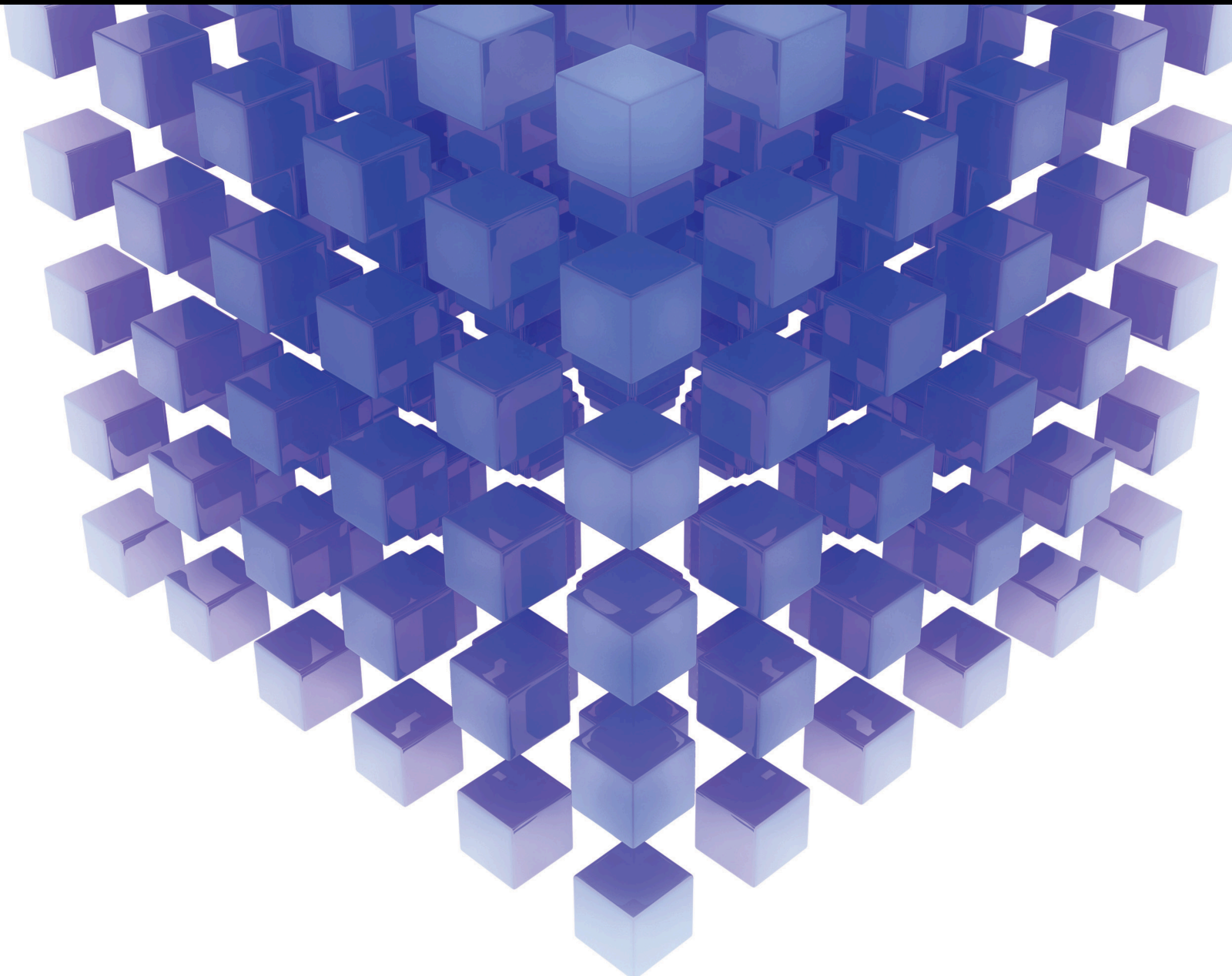


Mathematical Problems in Engineering

# Computational Intelligence in Data-Driven Modelling and Its Engineering Applications 2020

Lead Guest Editor: Qian Zhang

Guest Editors: Jun Chen and Trung Thanh Nguyen





---

**Computational Intelligence in Data-Driven  
Modelling and Its Engineering Applications  
2020**

Mathematical Problems in Engineering

---

**Computational Intelligence in Data-  
Driven Modelling and Its Engineering  
Applications 2020**

Lead Guest Editor: Qian Zhang

Guest Editors: Jun Chen and Trung Thanh Nguyen



---

Copyright © 2022 Hindawi Limited. All rights reserved.

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



# Chief Editor

Guangming Xie, China

## Editorial Board

Mohamed Abd El Aziz, Egypt  
Ahmed A. Abd El-Latif, Egypt  
Mahmoud Abdel-Aty, Egypt  
Mohammad Yaghoub Abdollahzadeh  
Jamalabadi, Republic of Korea  
Rahib Abiyev, Turkey  
Leonardo Acho, Spain  
José Ángel Acosta, Spain  
Daniela Addressi, Italy  
Paolo Addresso, Italy  
Claudia Adduce, Italy  
Ramesh Agarwal, USA  
Francesco Aggogeri, Italy  
Ricardo Aguilar-Lopez, Mexico  
Ali Ahmadian, Malaysia  
Tarek Ahmed-Ali, France  
Elias Aifantis, USA  
Akif Akgul, Turkey  
Guido Ala, Italy  
Andrea Alaimo, Italy  
Reza Alam, USA  
Nicholas Alexander, United Kingdom  
Salvatore Alfonzetti, Italy  
Nouman Ali, Pakistan  
Mohammad D. Aliyu, Canada  
Juan A. Almendral, Spain  
Watheq Al-Mudhafar, Iraq  
Mohammad Alomari, Jordan  
Ali Saleh Alshomrani, Saudi Arabia  
José Domingo Álvarez, Spain  
Cláudio Alves, Portugal  
Juan P. Amezquita-Sanchez, Mexico  
Lionel Amodeo, France  
Sebastian Anita, Romania  
Renata Archetti, Italy  
Muhammad Arif, Pakistan  
Sabri Arik, Turkey  
Francesco Aristodemo, Italy  
Fausto Arpino, Italy  
Alessandro Arsie, USA  
Edoardo Artioli, Italy  
Rashad Asharabi, Saudi Arabia  
Fumihiko Ashida, Japan  
Farhad Aslani, Australia

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

Guido Bolognesi, United Kingdom  
Rodolfo Bontempo, Italy  
Alberto Borboni, Italy  
Marco Bortolini, Italy  
Paolo Boscariol, Italy  
Daniela Boso, Italy  
Guillermo Botella-Juan, Spain  
Abdesselem Boulkroune, Algeria  
Boulaïd Boulkroune, Belgium  
Fabio Bovenga, Italy  
Francesco Braghin, Italy  
Ricardo Branco, Portugal  
Maurizio Brocchini, Italy  
Julien Bruchon, France  
Matteo Bruggi, Italy  
Michele Brun, Italy  
Maria Elena Bruni, Italy  
Vasilis Burganos, Greece  
Maria Angela Butturi, Italy  
Raquel Caballero-Águila, Spain  
Guillermo Cabrera-Guerrero, Chile  
Filippo Cacace, Italy  
Pierfrancesco Cacciola, United Kingdom  
Salvatore Caddemi, Italy  
zuowei cai, China  
Roberto Caldelli, Italy  
Alberto Campagnolo, Italy  
Eric Campos, Mexico  
Salvatore Cannella, Italy  
Francesco Cannizzaro, Italy  
Maosen Cao, China  
Javier Cara, Spain  
Raffaele Carli, Italy  
Ana Carpio, Spain  
Rodrigo Carvajal, Chile  
Caterina Casavola, Italy  
Sara Casciati, Italy  
Federica Caselli, Italy  
Carmen Castillo, Spain  
Inmaculada T. Castro, Spain  
Miguel Castro, Portugal  
Giuseppe Catalanotti, United Kingdom  
Nicola Caterino, Italy  
Alberto Cavallo, Italy  
Gabriele Cazzulani, Italy  
Luis Cea, Spain  
Fatih Vehbi Celebi, Turkey

Song Cen, China  
Miguel Cerrolaza, Venezuela  
M. Chadli, France  
Gregory Chagnon, France  
Ludovic Chamoin, France  
Xiaoheng Chang, China  
Qing Chang, USA  
Ching-Ter Chang, Taiwan  
Kuei-Lun Chang, Taiwan  
Dr. Prasenjit Chatterjee, India  
Kacem Chehdi, France  
Peter N. Cheimets, USA  
Shyi-Ming Chen, Taiwan  
Chih-Chiang Chen, Taiwan  
Xinkai Chen, Japan  
Xizhong Chen, Ireland  
Kebing Chen, China  
Xue-Bo Chen, China  
Zhiwen Chen, China  
Chien-Ming Chen, China  
Xiao Chen, China  
He Chen, China  
Zeyang Cheng, China  
Qiang Cheng, USA  
Luca Chiapponi, Italy  
Ryoichi Chiba, Japan  
Francisco Chicano, Spain  
Nicholas Chileshe, Australia  
Tirivanhu Chinyoka, South Africa  
Adrian Chmielewski, Poland  
Seongim Choi, USA  
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  
Francesco Conte, Italy  
Enrico Conte, Italy  
Alessandro Contento, USA  
Mario Cools, Belgium  
Gino Cortellessa, Italy  
Juan Carlos Cortés, Spain

Carlo Cosentino, Italy  
Paolo Crippa, Italy  
Erik Cuevas, Mexico  
Guozeng Cui, China  
Maria C. Cunha, Portugal  
Mehmet Cunkas, Turkey  
Peter Dabnichki, Australia  
Luca D'Acerno, Italy  
Pei Dai, China  
Zhifeng Dai, China  
Weizhong Dai, USA  
Purushothaman Damodaran, USA  
Bhabani S. Dandapat, India  
Giuseppe D'Aniello, Italy  
Sergey Dashkovskiy, Germany  
Adiel T. de Almeida-Filho, Brazil  
Fabio De Angelis, Italy  
Samuele De Bartolo, Italy  
Abilio De Jesus, Portugal  
Pietro De Lellis, Italy  
Alessandro De Luca, Italy  
Stefano de Miranda, Italy  
Filippo de Monte, Italy  
José António Fonseca de Oliveira Correia, Portugal  
Jose Renato de Sousa, Brazil  
Michael Defoort, France  
Alessandro Della Corte, Italy  
Laurent Dewasme, Belgium  
Sanku Dey, India  
Gianpaolo Di Bona, Italy  
Angelo Di Egidio, Italy  
Roberta Di Pace, Italy  
Francesca Di Puccio, Italy  
Ramón I. Diego, Spain  
Yannis Dimakopoulos, Greece  
Rossana Dimitri, Italy  
Alexandre B. Dolgui, France  
José M. Domínguez, Spain  
Georgios Dounias, Greece  
Z. Du, China  
Bo Du, China  
George S. Dulikravich, USA  
Emil Dumic, Croatia  
Bogdan Dumitrescu, Romania  
Saeed Eftekhar Azam, USA  
Antonio Elipe, Spain

R. Emre Erkmen, Canada  
John Escobar, Colombia  
Francisco Periago Esparza, Spain  
Gilberto Espinosa-Paredes, Mexico  
Leandro F. F. Miguel, Brazil  
Andrea L. Facci, Italy  
Giovanni Falsone, Italy  
Hua Fan, China  
Nicholas Fantuzzi, Italy  
Muhammad Shahid Farid, Pakistan  
Hamed Faroqi, Iran  
Mohammad Fattahi, Iran  
Yann Favennec, France  
Fiorenzo A. Fazzolari, United Kingdom  
Roberto Fedele, Italy  
Giuseppe Fedele, Italy  
Zhongyang Fei, China  
Mohammad Ferdows, Bangladesh  
Arturo J. Fernández, Spain  
Jesus M. Fernandez Oro, Spain  
Massimiliano Ferraioli, Italy  
Massimiliano Ferrara, Italy  
Francesco Ferrise, Italy  
Constantin Fetecau, Romania  
Eric Feulvarch, France  
Iztok Fister Jr., Slovenia  
Thierry Floquet, France  
Eric Florentin, France  
Gerardo Flores, Mexico  
Alessandro Formisano, Italy  
FRANCESCO FOTI, Italy  
Francesco Franco, Italy  
Elisa Francomano, Italy  
Juan Frausto-Solis, Mexico  
Shujun Fu, China  
Juan C. G. Prada, Spain  
Matteo Gaeta, Italy  
Mauro Gaggero, Italy  
Zoran Gajic, USA  
Jaime Gallardo-Alvarado, Mexico  
Mosè Gallo, Italy  
Akemi Gálvez, Spain  
Rita Gamberini, Italy  
Maria L. Gandarias, Spain  
Zhong-Ke Gao, China  
Yan Gao, China  
Xingbao Gao, China

Hao Gao, Hong Kong  
Shangce Gao, Japan  
Zhiwei Gao, United Kingdom  
Giovanni Garcea, Italy  
José García, Chile  
Luis Rodolfo Garcia Carrillo, USA  
Jose M. Garcia-Aznar, Spain  
Akhil Garg, China  
Harish Garg, India  
Alessandro Gasparetto, Italy  
Gianluca Gatti, Italy  
Oleg V. Gendelman, Israel  
Stylios Georgantzinis, Greece  
Fotios Georgiades, India  
Parviz Ghadimi, Iran  
Georgios I. Giannopoulos, Greece  
Agathoklis Giaralis, United Kingdom  
Pablo Gil, Spain  
Anna M. Gil-Lafuente, Spain  
Ivan Giorgio, Italy  
Gaetano Giunta, Luxembourg  
Alessio Gizzi, Italy  
Jefferson L.M.A. Gomes, United Kingdom  
HECTOR GOMEZ, Chile  
José Francisco Gómez Aguilar, Mexico  
Emilio Gómez-Déniz, Spain  
Antonio M. Gonçalves de Lima, Brazil  
Chris Goodrich, USA  
Rama S. R. Gorla, USA  
Veena Goswami, India  
Xunjie Gou, Spain  
Jakub Grabski, Poland  
Antoine Grall, France  
George A. Gravvanis, Greece  
Fabrizio Greco, Italy  
David Greiner, Spain  
Jason Gu, Canada  
Federico Guarracino, Italy  
Michele Guida, Italy  
Muhammet Gul, Turkey  
Hu Guo, China  
Dong-Sheng Guo, China  
Zhaoxia Guo, China  
Jian-Ping Guo, China  
Quang Phuc Ha, Australia  
Li Haitao, China  
Petr Hájek, Czech Republic

Muhammad Hamid, United Kingdom  
Shigeyuki Hamori, Japan  
Renke Han, United Kingdom  
Weimin Han, USA  
Zhen-Lai Han, China  
Xingsi Han, China  
Thomas Hanne, Switzerland  
Xinan Hao, China  
Mohammad A. Hariri-Ardebili, USA  
Khalid Hattaf, Morocco  
Defeng He, China  
Fu-Qiang He, China  
Yu-Ling He, China  
Xiao-Qiao He, China  
salim HEDDAM, Algeria  
Ramdane Hedjar, Saudi Arabia  
Jude Hemanth, India  
Reza Hemmati, Iran  
Nicolae Herisanu, Romania  
Alfredo G. Hernández-Diaz, Spain  
M.I. Herreros, Spain  
Eckhard Hitzer, Japan  
Paul Honeine, France  
Jaromir Horacek, Czech Republic  
S. Hassan Hosseinnia, The Netherlands  
Yingkun Hou, China  
Xiaorong Hou, China  
Lei Hou, China  
Yunfeng Hu, China  
Gordon Huang, Canada  
Can Huang, China  
Sajid Hussain, Canada  
Asier Ibeas, Spain  
Wubshet Ibrahim, Ethiopia  
Orest V. Iftime, The Netherlands  
Przemyslaw Ignaciuk, Poland  
Muhammad Imran, Pakistan  
Giacomo Innocenti, Italy  
Emilio Insfran Pelozo, Spain  
Alessio Ishizaka, France  
Nazrul Islam, USA  
Benoit Iung, France  
Benjamin Ivorra, Spain  
Breno Jacob, Brazil  
Tushar Jain, India  
Amin Jajarmi, Iran  
Payman Jalali, Finland

Mahdi Jalili, Australia  
Prashant Kumar Jamwal, Kazakhstan  
Chiranjibe Jana, India  
Łukasz Jankowski, Poland  
Fahd Jarad, Turkey  
Samuel N. Jator, USA  
Juan C. Jauregui-Correa, Mexico  
Kandasamy Jayakrishna, India  
Reza Jazar, Australia  
Khalide Jbilou, France  
Isabel S. Jesus, Portugal  
Chao Ji, China  
Linni Jian, China  
Bin Jiang, China  
Qing-Chao Jiang, China., China  
Peng-fei Jiao, China  
Ricardo Fabricio Escobar Jiménez, Mexico  
Emilio Jiménez Macías, Spain  
Xiaoliang Jin, Canada  
Zhuo Jin, Australia  
Maolin Jin, Republic of Korea  
Dylan F. Jones, United Kingdom  
Viacheslav Kalashnikov, Mexico  
Mathiyalagan Kalidass, India  
Tamas Kalmar-Nagy, Hungary  
Zhao Kang, China  
Tomasz Kapitaniak, Poland  
Julius Kaplunov, United Kingdom  
Konstantinos Karamanos, Belgium  
Michal Kawulok, Poland  
Irfan Kaymaz, Turkey  
Vahid Kayvanfar, Iran  
Krzysztof Kecik, Poland  
Chaudry M. Khaliq, South Africa  
Mukhtaj Khan, Pakistan  
Abdul Qadeer Khan, Pakistan  
Mostafa M. A. Khater, Egypt  
MOHAMMAD REZA KHEDMATI, Iran  
Kwangki Kim, Republic of Korea  
Nam-Il Kim, Republic of Korea  
Philipp V. Kiryukhantsev-Korneev, Russia  
P.V.V Kishore, India  
Jan Koci, Czech Republic  
Ioannis Kostavelis, Greece  
Sotiris B. Kotsiantis, Greece  
Frederic Kratz, France  
Vamsi Krishna, India  
Kamalanand Krishnamurthy, India  
Petr Krysl, USA  
Edyta Kucharska, Poland  
Krzysztof S. Kulpa, Poland  
Kamal Kumar, India  
Michal Kunicki, Poland  
Cedrick A. K. Kwuimy, USA  
Kyandoghere Kyamakya, Austria  
Ivan Kyrchei, Ukraine  
Davide La Torre, Italy  
Márcio J. Lacerda, Brazil  
Risto Lahdelma, Finland  
Giovanni Lancioni, Italy  
Jaroslaw Latalski, Poland  
Antonino Laudani, Italy  
Hervé Laurent, France  
Agostino Lauria, Italy  
Aimé Lay-Ekuakille, Italy  
Nicolas J. Leconte, France  
Kun-Chou Lee, Taiwan  
Dimitri Lefebvre, France  
Eric Lefevre, France  
Marek Lefik, Poland  
Gang Lei, Saudi Arabia  
Yaguo Lei, China  
Kauko Leiviskä, Finland  
Thibault Lemaire, France  
Ervin Lenzi, Brazil  
Roman Lewandowski, Poland  
Jian Li, USA  
Yuxing Li, China  
Jun Li, China  
Yueyang Li, China  
ChenFeng Li, China  
Yang Li, China  
Zhen Li, China  
Zhiyun Lin, China  
Jian Lin, China  
Mingwei Lin, China  
Yao-Jin Lin, China  
En-Qiang Lin, USA  
Yuanchang Liu, United Kingdom  
Jianxu Liu, Thailand  
Heng Liu, China  
Bo Liu, China  
Lei Liu, China  
Yu Liu, China

Sixin Liu, China  
Wanquan Liu, China  
Bin Liu, China  
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  
Cesar S. Lopez-Monsalvo, Mexico  
Luis M. López-Ochoa, Spain  
Ezequiel López-Rubio, Spain  
Vassilios C. Loukopoulos, Greece  
Jose A. Lozano-Galant, Spain  
Gabriele Maria Lozito, Italy  
Songtao Lu, USA  
Rongxing Lu, Canada  
Zhiguo Luo, China  
Gabriel Luque, Spain  
Valentin Lychagin, Norway  
Junhai Ma, China  
Dazhong Ma, China  
Antonio Madeo, Italy  
Alessandro Magnani, Belgium  
Toqeer Mahmood, Pakistan  
Fazal M. Mahomed, South Africa  
Arunava Majumder, India  
Paolo Manfredi, Italy  
Adnan Maqsood, Pakistan  
Giuseppe Carlo Marano, Italy  
Damijan Markovic, France  
Filipe J. Marques, Portugal  
Luca Martinelli, Italy  
Rodrigo Martinez-Bejar, Spain  
Guiomar Martín-Herrán, Spain  
Denizar Cruz Martins, Brazil  
Francisco J. Martos, Spain  
Elio Masciari, Italy  
Franck Massa, France  
Paolo Massioni, France  
Alessandro Mauro, Italy  
Jonathan Mayo-Maldonado, Mexico  
Fabio Mazza, Italy  
Pier Luigi Mazzeo, Italy

Laura Mazzola, Italy  
Driss Mehdi, France  
Dr. Zahid Mehmood, Pakistan  
YUE MEI, China  
Roderick Melnik, Canada  
Debiao Meng, China  
Xiangyu Meng, USA  
Jose Merodio, Spain  
Alessio Merola, Italy  
Mahmoud Mesbah, Iran  
Luciano Mescia, Italy  
Laurent Mevel, France  
Constantine Michailides, Cyprus  
Mariusz Michta, Poland  
Prankul Middha, Norway  
Aki Mikkola, Finland  
Giovanni Minafò, Italy  
Hiroyuki Mino, Japan  
Dimitrios Mitsotakis, New Zealand  
saleh mobayen, Taiwan, R.O.C., Iran  
Nikunja Mohan Modak, India  
Sara Montagna, Italy  
Roberto Montanini, Italy  
Francisco J. Montáns, Spain  
Gisele Mophou, France  
Rafael Morales, Spain  
Marco Morandini, Italy  
Javier Moreno-Valenzuela, Mexico  
Simone Morganti, Italy  
Caroline Mota, Brazil  
Aziz Moukrim, France  
Shen Mouquan, China  
Dimitris Mourtzis, Greece  
Emiliano Mucchi, Italy  
Taseer Muhammad, Saudi Arabia  
Josefa Mula, Spain  
Jose J. Muñoz, Spain  
Giuseppe Muscolino, Italy  
Dino Musmarra, Italy  
Marco Mussetta, Italy  
Ghulam Mustafa, Pakistan  
Hariharan Muthusamy, India  
Hakim Naceur, France  
Alessandro Naddeo, Italy  
Benedek Nagy, Turkey  
Omar Naifar, Tunisia  
Mariko Nakano-Miyatake, Mexico

Keivan Navaie, United Kingdom  
Adrian Neagu, USA  
Erivelton Geraldo Nepomuceno, Brazil  
Luís C. Neves, United Kingdom  
AMA Neves, Portugal  
Dong Ngoduy, New Zealand  
Nhon Nguyen-Thanh, Singapore  
Papakostas Nikolaos, Ireland  
Jelena Nikolic, Serbia  
Tatsushi Nishi, Japan  
Shanzhou Niu, China  
Xesús Nogueira, Spain  
Ben T. Nohara, Japan  
Mohammed Nouari, France  
Mustapha Nourelfath, Canada  
Kazem Nouri, Iran  
Ciro Núñez-Gutiérrez, Mexico  
Włodzimierz Ogryczak, Poland  
Roger Ohayon, France  
Krzysztof Okarma, Poland  
Mitsuhiro Okayasu, Japan  
Diego Oliva, Mexico  
Alberto Olivares, Spain  
Enrique Onieva, Spain  
Calogero Orlando, Italy  
Sergio Ortobelli, Italy  
Naohisa Otsuka, Japan  
Taoreed Owolabi, Nigeria  
Cenap Özel, Turkey  
Pawel Packo, Poland  
Arturo Pagano, Italy  
Roberto Palma, Spain  
Alessandro Palmeri, United Kingdom  
Pasquale Palumbo, Italy  
Li Pan, China  
Weifeng Pan, China  
Chandan Pandey, India  
K. M. Pandey, India  
Jürgen Pannek, Germany  
Elena Panteley, France  
Achille Paolone, Italy  
George A. Papakostas, Greece  
Xosé M. Pardo, Spain  
You-Jin Park, Taiwan  
Manuel Pastor, Spain  
Petr Páta, Czech Republic  
Pubudu N. Pathirana, Australia  
Surajit Kumar Paul, India  
Sitek Paweł, Poland  
Luis Payá, Spain  
Alexander Paz, Australia  
Igor Pažanin, Croatia  
Libor Pekař, Czech Republic  
Francesco Pellicano, Italy  
Marcello Pellicciari, Italy  
Zhi-ke Peng, China  
Mingshu Peng, China  
Haipeng Peng, China  
Yuexing Peng, China  
Zhengbiao Peng, Australia  
Bo Peng, China  
Jian Peng, China  
Xindong Peng, China  
Marzio Pennisi, Italy  
Maria Patrizia Pera, Italy  
Matjaz Perc, Slovenia  
A. M. Bastos Pereira, Portugal  
Ricardo Perera, Spain  
F. Javier Pérez-Pinal, Mexico  
Michele Perrella, Italy  
Francesco Pesavento, Italy  
Ivo Petras, Slovakia  
Francesco Petrini, Italy  
EUGENIA PETROPOULOU, Greece  
Hoang Vu Phan, Republic of Korea  
Lukasz Pieczonka, Poland  
Dario Piga, Switzerland  
Antonina Pirrotta, Italy  
Marco Pizzarelli, Italy  
Javier Plaza, Spain  
Goutam Pohit, India  
Kemal Polat, Turkey  
Dragan Poljak, Croatia  
Jorge Pomares, Spain  
Hiram Ponce, Mexico  
Sébastien Poncet, Canada  
Volodymyr Ponomaryov, Mexico  
Jean-Christophe Ponsart, France  
Mauro Pontani, Italy  
Cornelio Posadas-Castillo, Mexico  
Francesc Pozo, Spain  
Aditya Rio Prabowo, Indonesia  
Anchasa Pramuanjaroenkij, Thailand  
Christopher Pretty, New Zealand

Leonardo Primavera, Italy  
Luca Pugi, Italy  
Krzysztof Puszynski, Poland  
Goran D. Putnik, Portugal  
Chuan Qin, China  
Jianlong Qiu, China  
Giuseppe Quaranta, Italy  
Vitomir Racic, Italy  
Ahmed G. Radwan, Egypt  
Hamid Rahman, Pakistan  
Carlo Rainieri, Italy  
Kumbakonam Ramamani Rajagopal, USA  
Venkatesan Rajinikanth, India  
Ali Ramazani, USA  
Higinio Ramos, Spain  
Angel Manuel Ramos, Spain  
Muhammad Afzal Rana, Pakistan  
Amer Rasheed, Pakistan  
Muhammad Rashid, Saudi Arabia  
Manoj Rastogi, India  
Alessandro Rasulo, Italy  
S.S. Ravindran, USA  
Abdolrahman Razani, Iran  
Alessandro Reali, Italy  
Jose A. Reinoso, Spain  
Oscar Reinoso, Spain  
Haijun Ren, China  
X. W. Ren, China  
Carlo Renno, Italy  
Fabrizio Renno, Italy  
Shahram Rezapour, Iran  
Ricardo Riaza, Spain  
Francesco Riganti-Fulginei, Italy  
Gerasimos Rigatos, Greece  
Francesco Ripamonti, Italy  
Marcelo Raúl Risk, Argentina  
Jorge Rivera, Mexico  
Eugenio Roanes-Lozano, Spain  
Bruno G. M. Robert, France  
Ana Maria A. C. Rocha, Portugal  
Luigi Rodino, Italy  
Francisco Rodríguez, Spain  
Rosana Rodríguez López, Spain  
Alessandra Romolo, Italy  
Abdolreza Roshani, Italy  
Francisco Rossomando, Argentina  
Jose de Jesus Rubio, Mexico

Weiguo Rui, China  
Rubén Ruiz, Spain  
Ivan D. Rukhlenko, Australia  
Chaman Lal Sabharwal, USA  
Kishin Sadarangani, Spain  
Andrés Sáez, Spain  
Bekir Sahin, Turkey  
John S. Sakellariou, Greece  
Michael Sakellariou, Greece  
Salvatore Salamone, USA  
Jose Vicente Salcedo, Spain  
Alejandro Salcido, Mexico  
Alejandro Salcido, Mexico  
Salman saleem, Pakistan  
Ahmed Salem, Saudi Arabia  
Nunzio Salerno, Italy  
Rohit Salgotra, India  
Miguel A. Salido, Spain  
Zabidin Salleh, Malaysia  
Roque J. Saltarén, Spain  
Alessandro Salvini, Italy  
Abdus Samad, India  
Nikolaos Samaras, Greece  
Sylwester Samborski, Poland  
Ramon Sancibrian, Spain  
Giuseppe Sanfilippo, Italy  
Omar-Jacobo Santos, Mexico  
J Santos-Reyes, Mexico  
José A. Sanz-Herrera, Spain  
Evangelos J. Sapountzakis, Greece  
Musavarah Sarwar, Pakistan  
Marcelo A. Savi, Brazil  
Andrey V. Savkin, Australia  
Tadeusz Sawik, Poland  
Roberta Sburlati, Italy  
Gustavo Scaglia, Argentina  
Thomas Schuster, Germany  
Tapan Senapati, China  
Lotfi Senhadji, France  
Junwon Seo, USA  
Michele Serpilli, Italy  
Joan Serra-Sagrasta, Spain  
Silvestar Šesnić, Croatia  
Erhan Set, Turkey  
Gerardo Severino, Italy  
Ruben Sevilla, United Kingdom  
Stefano Sfarra, Italy



Mohamed Shaat, United Arab Emirates  
Mostafa S. Shadloo, France  
Kamal Shah, Pakistan  
Dr. Zahir Shah, Pakistan  
Leonid Shaikhet, Israel  
Xingling Shao, China  
Xin Pu Shen, China  
Bo Shen, Germany  
hang shen, China  
Hao Shen, China  
Dimitri O. Shepelsky, Ukraine  
Weichao SHI, United Kingdom  
Jian Shi, China  
Suzanne M. Shontz, USA  
Babak Shotorban, USA  
Zhan Shu, Canada  
Angelo Sifaleras, Greece  
Nuno Simões, Portugal  
Mehakpreet Singh, Ireland  
Harendra Singh, India  
Thanin Sitthiwirattam, Thailand  
Seralthan Sivamani, India  
S. Sivasankaran, Malaysia  
Christos H. Skiadas, Greece  
Konstantina Skouri, Greece  
Neale R. Smith, Mexico  
Bogdan Smolka, Poland  
Delfim Soares Jr., Brazil  
Alba Sofi, Italy  
Francesco Soldovieri, Italy  
Raffaele Solimene, Italy  
Bosheng Song, China  
Yang Song, Norway  
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  
Miladin Stefanović, Serbia  
Florin Stoican, Romania  
Salvatore Strano, Italy  
Yakov Strelniker, Israel  
Qiuye Sun, China  
Qiuqin Sun, China  
Shuaishuai Sun, Australia  
Zong-Yao Sun, China  
Xiaodong Sun, China  
Suroso Suroso, Indonesia  
Sergey A. Suslov, Australia  
Nasser Hassen Sweilam, Egypt  
Andrzej Swierniak, Poland  
M Syed Ali, India  
Andras Szekrenyes, Hungary  
Kumar K. Tamma, USA  
Yong (Aaron) Tan, United Kingdom  
Marco Antonio Taneco-Hernández, Mexico  
Hafez Tari, USA  
Alessandro Tasora, Italy  
Sergio Teggi, Italy  
Adriana del Carmen Téllez-Anguiano, Mexico  
Ana C. Teodoro, Portugal  
Efsthathios E. Theotokoglou, Greece  
Jing-Feng Tian, China  
Alexander Timokha, Norway  
Stefania Tomasiello, Italy  
Gisella Tomasini, Italy  
Isabella Torcicollo, Italy  
Francesco Tornabene, Italy  
Javier Martinez Torres, Spain  
Mariano Torrisi, Italy  
Thang nguyen Trung, Vietnam  
Sang-Bing Tsai, China  
George Tsiatas, Greece  
Antonios Tsourdos, United Kingdom  
Le Anh Tuan, Vietnam  
Federica Tubino, Italy  
Nerio Tullini, Italy  
Emilio Turco, Italy  
Ilhan Tuzcu, USA  
Efstratios Tzirtzilakis, Greece  
Filippo Ubertini, Italy  
Mohammad Uddin, Australia  
Marjan Uddin, Pakistan  
Serdar Ulubeyli, Turkey  
FRANCISCO UREÑA, Spain  
Panayiotis Vafeas, Greece  
Giuseppe Vairo, Italy  
Jesus Valdez-Resendiz, Mexico  
Eusebio Valero, Spain

Stefano Valvano, Italy  
Marcello Vasta, Italy  
Carlos-Renato Vázquez, Mexico  
Miguel E. Vázquez-Méndez, Spain  
Martin Velasco Villa, Mexico  
Kalyana C. Veluvolu, Republic of Korea  
Franck J. Vernerey, USA  
Georgios Veronis, USA  
Vincenzo Vespri, Italy  
Renato Vidoni, Italy  
Venkatesh Vijayaraghavan, Australia  
Anna Vila, Spain  
Francisco R. Villatoro, Spain  
Francesca Vipiana, Italy  
Stanislav Vítek, Czech Republic  
Jan Vorel, Czech Republic  
Michael Vynnycky, Sweden  
Fu-Kwun Wang, Taiwan  
C. H. Wang, Taiwan  
Yung-Chung Wang, Taiwan  
Hao Wang, USA  
Zhenbo Wang, USA  
Zenghui Wang, South Africa  
Yongqi Wang, Germany  
Yong Wang, China  
Dagang Wang, China  
Bingchang Wang, China  
Ji Wang, China  
Guoqiang Wang, China  
Shuo Wang, China  
Xinyu Wang, China  
Weiwei Wang, China  
Kang-Jia Wang, China  
Qingling Wang, China  
Hui Wang, China  
J.G. Wang, China  
Zhibo Wang, China  
Qiang Wang, China University of Petroleum  
(East China), Qingdao, Shandong, 266580,  
People's Republic of  
China, China  
Roman Wan-Wendner, Austria  
Fangqing Wen, China  
P.H. Wen, United Kingdom  
Waldemar T. Wójcik, Poland  
Wai Lok Woo, United Kingdom  
Xianyi Wu, China

Qihong Wu, China  
Zhibin Wu, China  
Zhizheng Wu, China  
Changzhi Wu, China  
Yuqiang Wu, China  
Michalis Xenos, Greece  
hao xiao, China  
Xiao Ping Xie, China  
Xue-Jun Xie, China  
Hang Xu, China  
Zeshui Xu, China  
Lei Xu, China  
Qingzheng Xu, China  
Lingwei Xu, China  
Qilong Xue, China  
Joseph J. Yame, France  
Chuanliang Yan, China  
Zhiguo Yan, China  
Xinggang Yan, United Kingdom  
Mijia Yang, USA  
Ray-Yeng Yang, Taiwan  
Jixiang Yang, China  
Weilin Yang, China  
Zhihong Yao, China  
Min Ye, China  
Jun Ye, China  
Luis J. Yebra, Spain  
Peng-Yeng Yin, Taiwan  
Muhammad Haroon Yousaf, Pakistan  
Yuan Yuan, United Kingdom  
Qin Yuming, China  
Abdullahi Yusuf, Nigeria  
Akbar Zada, Pakistan  
Elena Zaitseva, Slovakia  
Arkadiusz Zak, Poland  
Daniel Zaldivar, Mexico  
Ernesto Zambrano-Serrano, Mexico  
Francesco Zammori, Italy  
Vittorio Zampoli, Italy  
Rafal Zdunek, Poland  
Ahmad Zeeshan, Pakistan  
Ibrahim Zeid, USA  
Bo Zeng, China  
Nianyin Zeng, China  
Junyong Zhai, China  
Tianwei Zhang, China  
Jian Zhang, China



---

Yinyan Zhang, China  
Xianming Zhang, Australia  
Yong Zhang, China  
Qian Zhang, China  
Tongqian Zhang, China  
Xiaofei Zhang, China  
Wenyu Zhang, China  
Lingfan Zhang, China  
Kai Zhang, China  
Hao Zhang, China  
Xuping Zhang, Denmark  
Haopeng Zhang, USA  
Yifan Zhao, United Kingdom  
Yongmin Zhong, Australia  
Zhe Zhou, China  
Zebo Zhou, China  
Jian G. Zhou, United Kingdom  
Debao Zhou, USA  
Wu-Le Zhu, China  
Quanxin Zhu, China  
Gaetano Zizzo, Italy  
Zhixiang Zou, China


# Contents

## **Computational Intelligence in Data-Driven Modelling and Its Engineering Applications 2020**

Qian Zhang , Jun Chen , and Trung Thanh Nguyen 


Editorial (2 pages), Article ID 9845906, Volume 2022 (2022)

## **Modeling a Thermochemical Reactor of a Solar Refrigerator by BaCl<sub>2</sub>-NH<sub>3</sub> Sorption Using Artificial Neural Networks and Mathematical Symmetry Groups**

Onesimo Meza-Cruz, Isaac Pilatowsky, Agustín Pérez-Ramírez , Carlos Rivera-Blanco, Youness El Hamzaoui, Miguel Perez-Ramirez, and Mauricio A. Sanchez


Research Article (11 pages), Article ID 9098709, Volume 2020 (2020)

## **A Machine-Learning Based Nonintrusive Smart Home Appliance Status Recognition**

Liston Matindife, Yanxia Sun, and Zenghui Wang 

Research Article (21 pages), Article ID 9356165, Volume 2020 (2020)

## **Analysing and Data-Driven Modelling the Handling Performance of Rugby Balls under Wet Conditions**

Jiaojiao Liu, Yunan Liu, Ya Wang, and Chao Liu 


Research Article (11 pages), Article ID 2946980, Volume 2020 (2020)

## **Data-Driven Model for Rockburst Prediction**

Hongbo Zhao  and Bingrui Chen


Research Article (14 pages), Article ID 5735496, Volume 2020 (2020)

## **A Novel Hybrid CNN-LSTM Scheme for Nitrogen Oxide Emission Prediction in FCC Unit**

Wei He, Jufeng Li, Zhihe Tang, Beng Wu, Hui Luan, Chong Chen , and Huaqing Liang 

Research Article (12 pages), Article ID 8071810, Volume 2020 (2020)

## **Financial Trading Strategy System Based on Machine Learning**

Yanjun Chen, Kun Liu, Yuantao Xie , and Mingyu Hu



Research Article (13 pages), Article ID 3589198, Volume 2020 (2020)

## **Health Assessment of High-Speed Train Running Gear System under Complex Working Conditions Based on Data-Driven Model**

Chao Cheng , Ming Liu , Bangcheng Zhang , Xiaojing Yin, Caixin Fu, and Wanxiu Teng




Research Article (13 pages), Article ID 9863936, Volume 2020 (2020)

## **A Scene Text Detector for Text with Arbitrary Shapes**

Weijia Wu, Jici Xing, Cheng Yang, Yuxing Wang , and Hong Zhou 

Research Article (11 pages), Article ID 8916028, Volume 2020 (2020)

## **Texts as Lines: Text Detection with Weak Supervision**

Weijia Wu, Jici Xing , Cheng Yang, Yuxing Wang , and Hong Zhou 

Research Article (12 pages), Article ID 3871897, Volume 2020 (2020)

## Editorial

# Computational Intelligence in Data-Driven Modelling and Its Engineering Applications 2020

**Qian Zhang <sup>1</sup>, Jun Chen <sup>2</sup>, and Trung Thanh Nguyen <sup>1</sup>**

<sup>1</sup>Liverpool John Moores University, Liverpool, UK

<sup>2</sup>Queen Mary University of London, London, UK

Correspondence should be addressed to Qian Zhang; [q.zhang@ljmu.ac.uk](mailto:q.zhang@ljmu.ac.uk)

Received 5 January 2022; Accepted 5 January 2022; Published 11 February 2022

Copyright © 2022 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 by them. In many cases, conventional mathematical models, such as differential equations that can accurately describe the complex systems and can be exploited in real-life applications, do not exist. However, with the fast development of advanced sensing, measurement, and data collection technologies, a large amount 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 generalization ability, which can be viewed as regression problems in mathematics. It analyses the data that characterize a system to find relationships among the system state variables (input, internal, and output variables) without taking into account explicit knowledge about physical behaviors. Many paradigms utilized 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, while evolutionary algorithms (EAs), swarm intelligence (SI), and machine learning (ML) methods provide learning and optimization abilities for calibrating and improving the intelligent or statistical models. In recent years, DDM has found widespread applications, ranging across machinery, manufacturing, materials, power and energy systems, transport, and so on.

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

This special issue has received 27 manuscripts and 9 high-quality papers have been accepted and published (33% 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. A brief introduction for each contribution is provided in the following paragraphs.

O. Meza-Cruz et al. applied the techniques of ANNs and mathematical symmetry groups in modelling a thermochemical reactor of a solid-gas cooling system, where barium chloride ( $\text{BaCl}_2$ ) is the solid and ammonia ( $\text{NH}_3$ ) is the refrigerant. It was found that using an alternating group of mathematical symmetry in the input data of the ANN helped improve modelling precision, and using the permutations of the mathematical symmetry group in the input data helped improve the convergence speed of the training algorithm.

L. Matindife et al. designed a deep learning-based approach for a smart home application, i.e., classification of appliances, specifically for some equal or very close power specification electronic appliances (EVPSAs). They evaluated three deep learning methods for nonintrusive load monitoring (NILM) disaggregation, including the multiple parallel structures convolutional neural networks (MPS-CNNs), the recurrent neural network (RNN) with parallel dense layers for a shared input, and the hybrid convolutional

recurrent neural network (CNN-RNN). Then, CNN and long short-term memory (LSTM) based networks were proposed for classification.

J. Liu et al. applied the ANN and neuro-fuzzy models into a sport engineering problem, which relates to the modelling the rugby players' performance under different moisture conditions. The developed intelligent models showed good performance in accuracy though using only a small number of training data. It was anticipated that the models would help the design of training programmes and the better preparation for rugby games with wet conditions.

In W. He et al.'s work, predictive models for nitrogen oxide emission were constructed and validated. In their models, CNN was employed to extract features among multidimensional data, in which the LSTM network was used to approximate the relationships among different time steps. The combination of CNN and LSTM showed better efficiency and accuracy than the baseline models. The developed models would be beneficial for providing reliable information for  $\text{NO}_x$  risk assessment and management.

In H. Zhao and B. Chen's work, a complex phenomenon of rockburst was studied. A data-driven method using CNN was proposed to predict the potential of rockburst. The method has been compared with the conventional ANNs and shown better performance. It was assumed that such a model would help evaluate the potential of rockburst for rock underground excavation.

Y. Chen et al. tackled the problem of predicting the stock price using data-driven models. They employed the light gradient boosting machine (LightGBM) algorithm and constructed the minimum variance portfolio of mean-variance model with conditional value at risk (CVaR) constraint. The proposed method was validated using China's stock market data between 2008 and 2018 and showed good accuracy.

C. Cheng et al. employed the data-driven models into assessing the working condition of the running gear of high-speed trains, which is complicated due to the existence of random noise in the monitoring data. Their method was developed based on a slow feature analysis-support tensor machine (SFA-STM). It was shown the developed technique could accurately anticipate the actual health status of the running gear system and outperformed the other four types of traditional data-driven models.

In the last two papers, W. Wu et al. conducted two pieces of studies for text detection. They first proposed a pixelwise technique using instance segmentation for scene text detection. The proposed method showed good performance in the common text benchmark problems and did well in the cases including text instance with irregular shapes. In their second work, they proposed a new text detector based on weakly supervised learning. The validation results showed that the proposed method works well in scene text detection, especially for the curved texts.

## Conflicts of Interest

The editors 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 contribution to this special issue. The lead guest editor would also like to express the deep gratitude to other coeditors for their great support and cooperation throughout the development of the special issue.

*Qian Zhang  
Jun Chen  
Trung Thanh Nguyen*

## Research Article

# Modeling a Thermochemical Reactor of a Solar Refrigerator by $\text{BaCl}_2\text{-NH}_3$ Sorption Using Artificial Neural Networks and Mathematical Symmetry Groups

Onesimo Meza-Cruz,<sup>1,2</sup> Isaac Pilatowsky,<sup>2</sup> Agustín Pérez-Ramírez <sup>1</sup>,  
Carlos Rivera-Blanco,<sup>2</sup> Youness El Hamzaoui,<sup>1</sup> Miguel Perez-Ramirez,<sup>3</sup>  
and Mauricio A. Sanchez<sup>4</sup>

<sup>1</sup>Universidad Autónoma Del Carmen, 24039 Cd Del Carmen Campeche, Mexico

<sup>2</sup>Instituto de Energías Renovables, Universidad Nacional Autónoma de México, 62580 Temixco, Morelos, Mexico

<sup>3</sup>Instituto Nacional de Electricidad y Energías Limpias, Reforma 113, Col. Palmira, C. P. 62490 Cuernavaca, Morelos, Mexico

<sup>4</sup>Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California, Tijuana, Mexico

Correspondence should be addressed to Agustín Pérez-Ramírez; nitsug@hotmail.com

Received 26 April 2020; Revised 26 July 2020; Accepted 10 August 2020; Published 30 September 2020

Guest Editor: Qian Zhang

Copyright © 2020 Onesimo Meza-Cruz et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The aim of this work is to present a model for heat transfer, desorbed refrigerant, and pressure of an intermittent solar cooling system's thermochemical reactor based on backpropagation neural networks and mathematical symmetry groups. In order to achieve this, a reactor was designed and built based on the reaction of  $\text{BaCl}_2\text{-NH}_3$ . Experimental data from this reactor were collected, where barium chloride was used as a solid absorbent and ammonia as a refrigerant. The neural network was trained using the Levenberg–Marquardt algorithm. The correlation coefficient between experimental data and data simulated by the neural network was  $r=0.9957$ . In the neural network's sensitivity analysis, it was found that the inputs, reactor's heating temperature and sorption time, influence neural network's learning by 35% and 20%, respectively. It was also found that, by applying permutations to experimental data and using multibase mathematical symmetry groups, the neural network training algorithm converges faster.

## 1. Introduction

It is estimated that 15% of the electrical energy in the world is used for cooling [1]. One way to decrease this percentage is to use energy from alternative sources such as geothermal or solar energy, among others [2]. In [3], sorption cooling is proposed (adsorption) which uses low-intensity sources such as solar energy. Sorption cooling systems use thermal energy and apply thermodynamic equilibrium: liquid-vapor and solid-vapor. The best known liquid-vapor works are mentioned in [4, 5]. The solid-vapor system is divided into physical sorption and chemisorption. The main physical sorption works are mentioned in [6, 7].

The solids frequently used are metal salts, in particular, alkaline metals and alkaline earth metal halides, which, in anhydrous state, can absorb large amounts of ammonia or other refrigerants. The salts that are most used as sorbents are calcium chloride ( $\text{CaCl}_2$ ) [8–10], strontium chloride ( $\text{SrCl}_2$ ) [11], and barium chloride ( $\text{BaCl}_2$ ) [12, 13]. In [14], 36 solid-gas reactions are collected, mainly chlorides reacting with ammonia. On the contrary, composite materials are a mixture between the aforementioned salts and inert solids such as expanded graphite [15], vermiculite [16, 17], carbon Sibunit, and  $\text{Al}_2\text{O}_3$  [17]. Composite materials have also been used as solid sorbents as they increase thermal conductivity and prevent agglomeration and swelling. The disadvantage of these composite materials is the low solid-gas ratio and

the reactor's increased volume due to the addition of new materials, compared to the use of pure salts. On the contrary, the predominant refrigerant in solid-gas systems is ammonia due to its vaporization properties, high availability, and minimal environmental impact.

A thermodynamic solid-gas sorption cycle is composed primarily of four elements: an evaporator, an absorber/generator, a condenser, and an expansion device. This cycle works intermittently; when the absorber/generator generates, it cannot absorb and vice versa. The reactor is the device where the phenomena of heat transfer occur by mass and kinetic reactions. In [18], the phenomena of heat and mass transfer are modeled, as well as the kinetic reaction by means of differential equation systems, in order to have a better knowledge in solid-gas reactors.

Mathematical models for studying solid-gas reactors are classified as local, global, and analytical. Local models deal with the phenomena of heat and mass transfer and are numerically solved. Global models deal with thermal conductivity, specific heat, and permeability averages. Analytical models consider reaction times and geometrical configurations [19]. There are also approaches where phenomena within the reactor have been analyzed at different scales [20] and different materials are used as sorbents [21]. It is also reported that the kinetic reaction in thermochemical reactors is complicated and has not been precisely modeled yet [6].

As it has been mentioned before, very complete studies on the phenomena of heat and mass transfer in solid-gas systems in numerical and analytical modalities with different salts have been carried out. There are also reports on intermittent cooling systems that use artificial neural networks (ANNs) to determine the performance in thermochemical reactors during the solid-gas sorption process [22, 23] and show that they are a promising technique in this area [24]. In this work, artificial neural networks and mathematical symmetry groups were used to model a thermochemical reactor of a solid-gas cooling system, where the solid is barium chloride ( $\text{BaCl}_2$ ) and the refrigerant is ammonia ( $\text{NH}_3$ ).

A thermochemical reactor working with the  $\text{BaCl}_2\text{-NH}_3$  pair was designed and built. Experimental data were collected from this reactor in order to obtain a mathematical prediction based on neural networks that model such a reactor. This provides an artificial intelligence tool that contributes to the design of reactors used in solid-gas sorption cooling.

The rest of this work is organized as follows: Section 2 describes the experimental setup and the influence of the heating fluid on the thermochemical reactor. Section 3 shows the modeling of the thermochemical reactor using neural networks and mathematical symmetry groups. Section 4 shows the obtained results. Section 5 shows the most relevant variables in ANN learning. Then, Section 6 shows the results of the uncertainty analysis followed by the main conclusions and a list of references.

## 2. System's Description and Experiments

Figure 1 shows the intermittent solid-gas sorption refrigeration cycle used in this work; this is composed as follows:

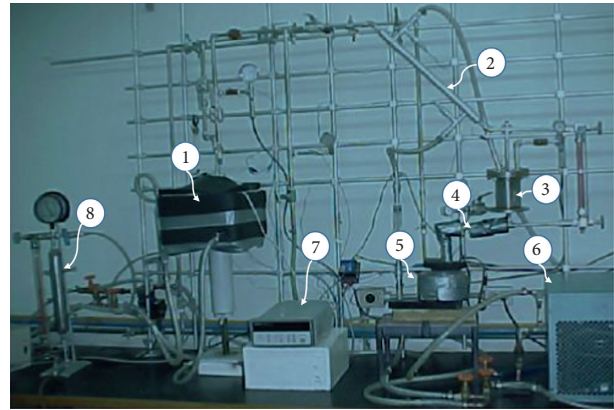


FIGURE 1: Thermochemical refrigerator system: 1, generator/absorber, 2, condenser, 3, ammonia condensate tank, 4, expansion valve, 5, evaporator, 6, auxiliary systems for heat and cold water supply, 7, data acquisition system, and 8, ammonia supply tank [12].

the thermochemical reactor that acts as a generator/absorber, the condenser, the refrigerant storage tank, an expansion valve, and the evaporator. The system was built in stainless steel [12].

In addition, the following subsystems were built: (1) water supply to heat/cool the thermochemical reactor and to remove heat from the condenser. Both temperatures are controlled by this subsystem consisting of a thermal bath (heat), a vapor compression cooler (cool), and a water pumping system (remove heat); (2) ammonia supply to the thermodynamic cycle; and (3) a data acquisition system.

Figure 2 depicts an intermittent cooling system. The system works in two stages: (a) generation-condensation and (b) evaporation-sorption. In the former, the  $V_A$  and  $V_C$  ball valves are closed, and the  $V_B$  ball valve is open. Subsequently, the generator/absorber containing the  $\text{BaCl}_2\text{-NH}_3$  pair is heated ( $Q_G$ ) to a temperature where the refrigerant ( $\text{NH}_3$ ) is desorbed. The refrigerant in the vapor state passes to the condenser where heat ( $Q_C$ ) is extracted, becomes a liquid, and is stored in the condensate tank.

In the latter,  $V_B$  closes, and  $V_A$  and  $V_C$  open. Subsequently, the expansion valve opens and regulates the refrigerant to enter the evaporator where it absorbs heat from the medium to be cooled ( $Q_E$ ). The refrigerant then returns to the generator/absorber where heat ( $Q_A$ ) is removed during the absorption process.

Three J-type thermocouples with different radii were installed in the generator/absorber, taking the reactor centre as reference:  $r_1 = 0$  cm,  $r_2 = 2$  cm, and  $r_3 = 4$  cm. This was done in order to obtain the temperatures  $Tr_1$ ,  $Tr_2$ , and  $Tr_3$  at the respective radii, when heat ( $Q_G$ ) is added to the reactor by heating water at various temperatures ( $Th_f$ ). Thermocouples were calibrated with high-precision Ametek brand electronic equipment built by Jofra Instruments model D55SE and had a temperature range of 0 to 150°C. A Cole-Parmer pressure sensor with an operating range of 0 to 20.68 bar (g) and 4 to 20 mA output was also installed to measure the generated pressure ( $P_g$ ). To measure the volume of condensed or desorbed ammonia ( $V_d$ ), an external glass



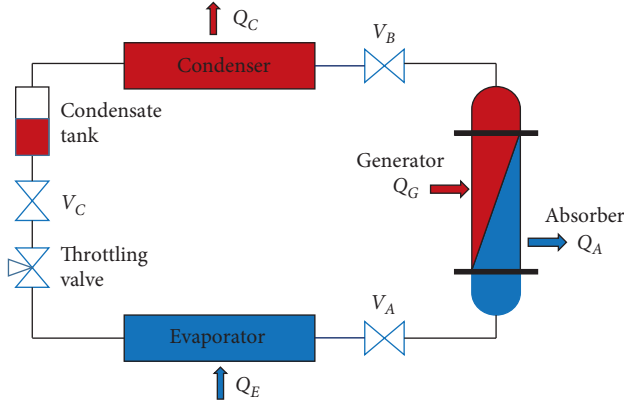


FIGURE 2: Solid-gas sorption refrigeration cycle.

meter of 38.1 mm (1½ ") diameter was installed, coupled to their respective valves as can be seen in Figure 1 (3). To measure the amount of desorbed ammonia (accumulated volume), readings were taken every 5 minutes from the start of the condensation until the desorption process was completed.

**2.1. Influence of the Heating Fluid on the Reactor.** The desorption process was carried out in order to determine the behavior of the temperature distribution within the reactor and quantify the volume of the desorbed refrigerant. To perform the desorption, the reactor was heated with water at different Thf temperatures (70°C, 80°C, and 95°C) and at a condensation temperature of 23°C. The variables recorded were temperatures Tr1, Tr2, and Tr3, generated pressure Pg, and the volume of desorbed ammonia in its liquid state (Vd).

Temperature distributions within the reactor (Tr1, Tr2, and Tr3) were obtained; however, only the result of Tr2 is shown (Figure 3(a)) as an illustrative case of the influence of Thf on the reactor's radial temperatures. A linear behavior can be observed in Figure 3, that is, the higher the temperature of Thf, the higher the temperature of Tr2, and the same happens with the variable Vg (Figure 3(b)). The pressure Pg (Figure 3(c)) does not show an increasing behavior; this is because the pressure during the desorption process (phase change) must behave as an isobaric process.

### 3. Neural Network Proposal for Modeling the Thermochemical Reactor

An artificial neural network (ANN) is a set of interconnected nodes, trying to mimic the functioning of the human brain. An ANN is a function  $f: R^n \rightarrow R^m$ . Each of the nodes has an input function and an output function. Once the ANN is designed and trained, then given an input dataset, the ANN can predict the values of the output variables. ANNs are used especially to model the behavior of nonlinear systems [24]. To train the ANN, the Levenberg–Marquardt algorithm [25, 26] was used because of its convergence speed and precision.

Figures 4 and 5 show the network structure and its input and output variables, which was implemented in MATLAB. Table 1 shows the definition of the neural network's input

TABLE 1: Working range of the thermochemical reactor by barium chloride-ammonia and variables used for the ANN training.

Parameter	Nomenclature	Minimum	Maximum	Unit
<b>Input</b>				
Radius 1 inside the reactor	$r1$	0	0	cm
Radius 2 inside the reactor	$r2$	2	2	cm
Radius 3 inside the reactor	$r3$	4	4	cm
Heating fluid's temperature	Thf	22.512	95.065	°C
Volume of the solid (BaCl <sub>2</sub> )	Vs	0.0002	0.0002	m <sup>3</sup>
Refrigerant volume (NH <sub>3</sub> )	Vr	0.0003	0.0003	m <sup>3</sup>
Desorption time	$t$	0	12.4375	hr
Reactor height	$H$	0.1702	0.1702	m
Reactor nominal diameter	Dn	0.0732	0.0732	m
<b>Output</b>				
Temperature at radius 1	Tr1	23.162	75.309	°C
Temperature at radius 2	Tr2	22.866	73.992	°C
Temperature at radius 3	Tr3	22.772	75.437	°C
Volume of the refrigerant	Vg	0	110	ml
Pressure into the reactor	Pg	2.572	11.834	bar

and output variables. The ANN's precision was determined by using the root-mean-square error (RMSE).

For the ANN training, 2 databases were used. The first database has 7637 experimental data and was generated with the Thf fluid at 70°C. The second database has 8956 experimental data and was generated with the Thf fluid at 95°C. The neural network was trained with these two merged databases. For a better network training, the databases were merged obtaining a single database of 16,593 records, and an alternating group  $A_n$  was applied to this database [27, 28]. Then, tuples or permutations of the base 2 multibase mathematical symmetry group (obtained from the multibase symmetry group) were applied to the resulting database. The multibase mathematical symmetry groups are obtained as follows [29].

Let  $G$  be a finite group of order  $N$ . Rename the elements of  $G$  by the set  $I = \{0, 1, 2, 3, \dots, N - 1\}$ . Let  $m_i$  be a number that will indicate a numerical base (for example, base 10 and base 2). Then,  $G$  can be represented as  $G = Z_{m_1} \times Z_{m_2} \times \dots \times Z_{m_k}$ , where  $Z_{m_i}$  is a cyclic group of order  $m_i$  and  $N = m_1 \times m_2 \times \dots \times m_k$ .

On the contrary, each number in the set  $I$  can be represented as

$$t \in I \Rightarrow t = \sum_{i=1}^k t_i \left( \prod_{j=0}^{i-1} m_j \right), \quad (1)$$

where  $t_i \in \{0, 1, \dots, m_i - 1\}$  y  $m_0 = 1$ .

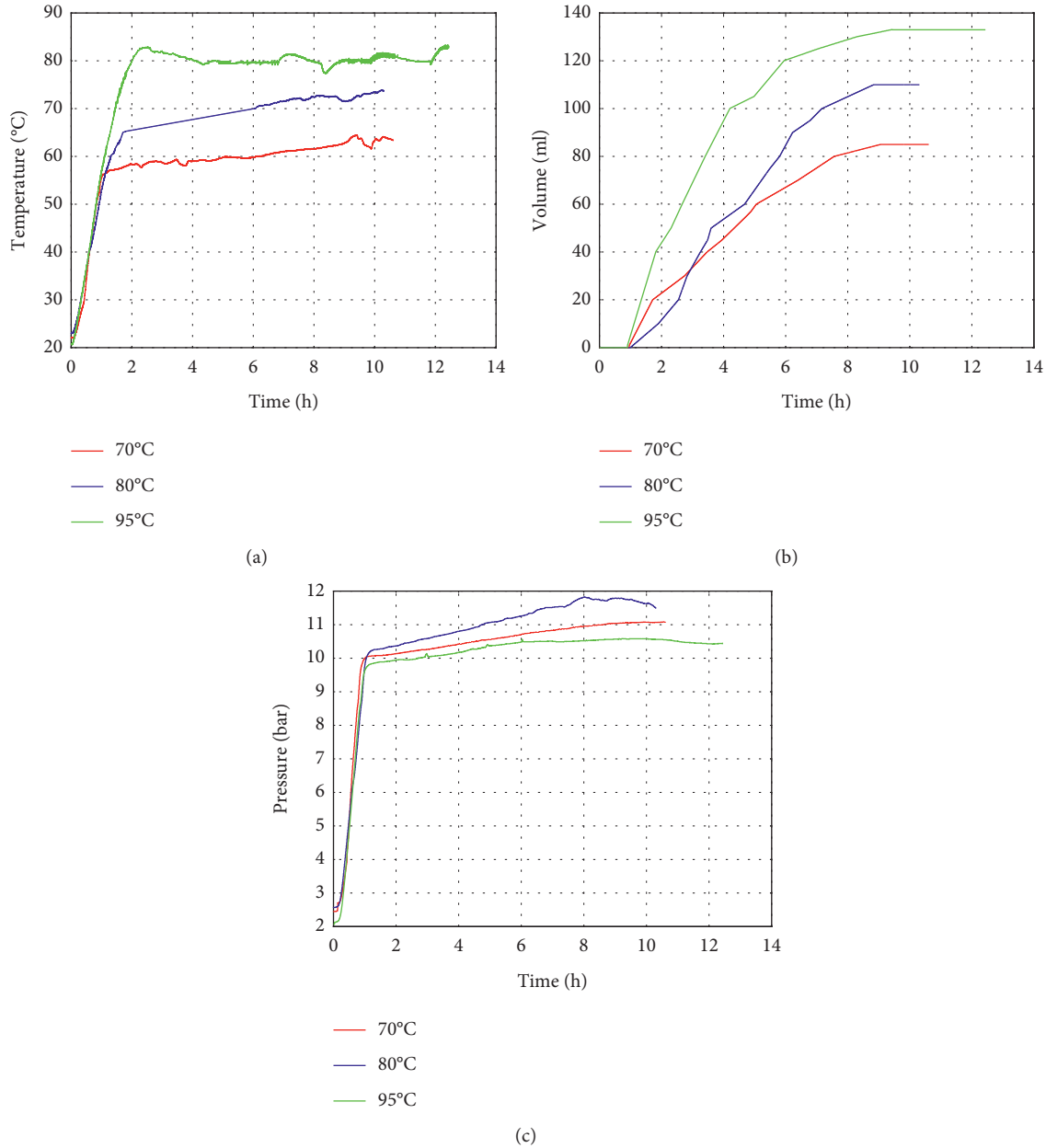


FIGURE 3: Influence of the heating fluid on the reactor parameters. (a) Influence of Thf (70°C, 80°C, and 95°C) on Tr2. (b) Influence of Thf (70°C, 80°C, and 95°C) on Vg. (c) Influence of Thf (70°C, 80°C, and 95°C) on Pg.

Each element  $t$  in  $G$  is given by  $t = (t_1, t_2, t_3, \dots, t_k)$  (vector form of the number), where  $t_i \in Z_{m_i}^{m_i}$ .

Let  $\circ$  be the operation of group  $G$  and  $\oplus$  the operation of the cyclic group  $Z_{m_i}$ . If  $t, g \in G$ , then

$$t \circ g = \left( t_1 \oplus g_1, t_2 \oplus g_2, t_3 \oplus g_3, \dots, t_k \oplus g_k \right), \quad (2)$$

where the cyclical group operation is represented as

$$t_i \oplus g_i = \begin{cases} t_i + g_i, & \text{if } t_i + g_i < m_i, \\ t_i + g_i - m_i, & \text{if } t_i + g_i \geq m_i. \end{cases} \quad (3)$$

The inverse  $g^{-1}$  of the element  $g \in G$  is defined as

$$g^{-1} = \begin{cases} 0, & \text{if } g_i = 0, \\ m_i - g_i, & \text{if } g_i > m_i. \end{cases} \quad (4)$$

With the previous group operations, the matrix  $V_{N \times N}$  is formed, where  $N$  is the number of columns or records in the database used to train the ANN, and each row of this matrix represents a permutation of this database. The matrix  $V_{N \times N}$  is defined as follows:

$$V(k, i) = k \circ i. \quad (5)$$

On the contrary, to determine the optimal ANN architecture, a set of various network configurations was

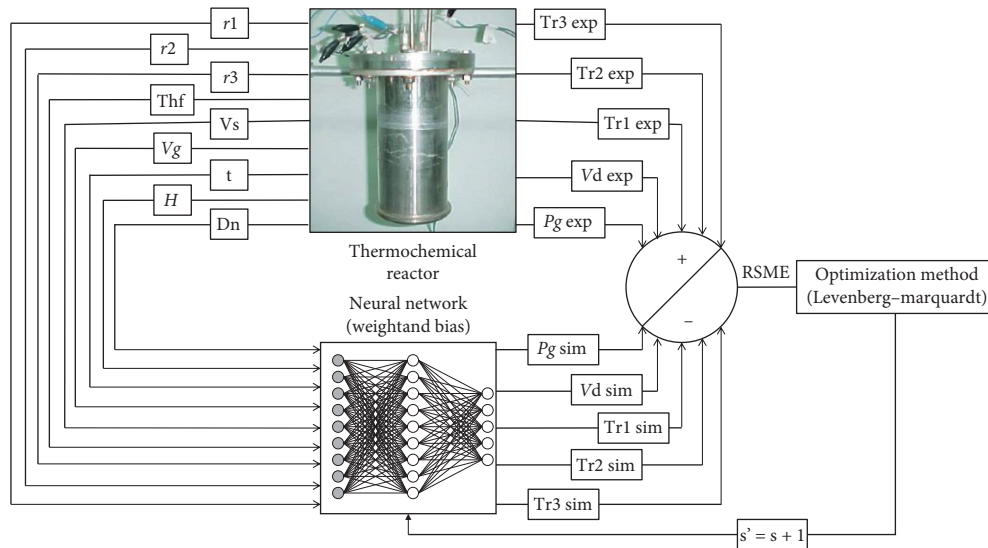


FIGURE 4: Numerical procedure used in the ANN learning process and the iterative architecture used by the ANN model to estimate temperatures, pressures, and ammonia production, respectively.

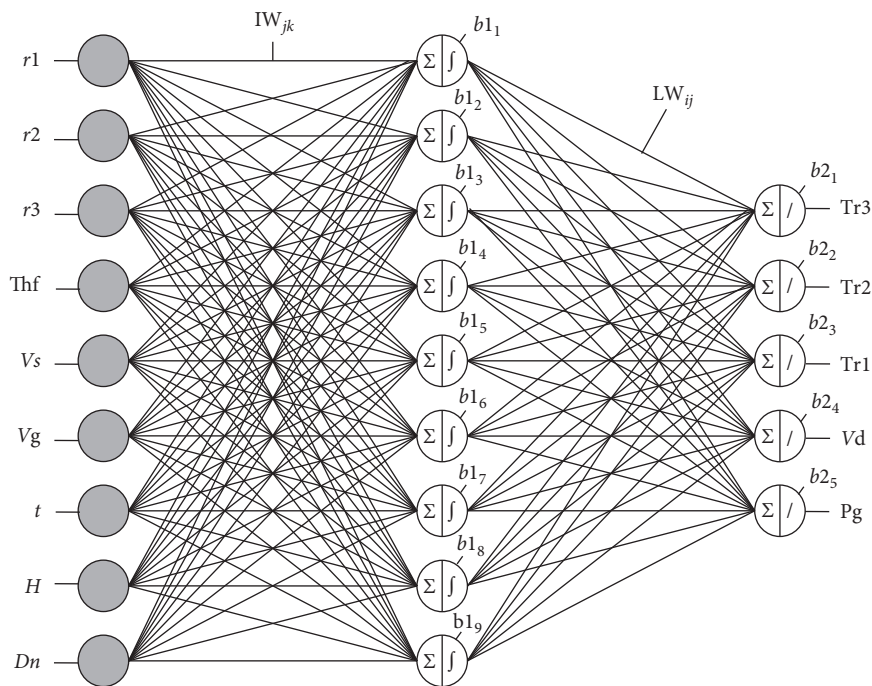


FIGURE 5: Optimized architecture of the ANN model.

tested, each one with different numbers of hidden layers and different numbers of neurons in each layer. It was found that the simplest and most accurate network model was 9-9-5 (nine neurons in the hidden layer and 5 neurons in the output layer) as shown in Figure 6. The transfer functions used in the network training were tansig function (hidden layer) and pureline function (output layer).

According to Figure 6, it was concluded that the best ANN model was achieved with 09 neurons on the hidden layer. However, the performance in the training set is good

with more than 09 neurons, but the performance of the test set is significantly worse, which could indicate an overfit. Therefore, in order to improve the results, the number of neurons in the hidden layer has been reduced to 09 neurons [30]. Consequently, the optimal network configuration was approximately 09-09-05. This model presents smaller values of RMSE (0.4%) and higher  $R^2$  (99.97%).

Once the ANN architecture was defined, each output of the neural network is calculated by the following equation:

$$Q_i = \sum_{j=1}^N \left[ LW_{ij} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_i. \quad (6)$$

Explicitly, each output of the neural network is calculated by the following equations (7)–(11):

$$Q_1 = Tr_3 = \sum_{j=1}^N \left[ LW_{1j} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_1, \quad (7)$$

$$Q_2 = Tr_2 = \sum_{j=1}^N \left[ LW_{2j} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_2, \quad (8)$$

$$Q_3 = Tr_1 = \sum_{j=1}^N \left[ LW_{3j} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_3, \quad (9)$$

$$Q_4 = V_D = \sum_{j=1}^N \left[ LW_{4j} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_4, \quad (10)$$

$$Q_5 = P_G = \sum_{j=1}^N \left[ LW_{5j} \left( \frac{2}{1 + \exp(-2(\sum_{k=1}^M (IW_{jk} In_k)) + b1_j)} - 1 \right) \right] + b2_5, \quad (11)$$

where  $N$  = total neurons of the hidden layer,  $M$  = total input,  $LW_{ij}$  = weights of the output layer from neuron  $j$  of the hidden layer to the output,  $IW_{jk}$  = weights of the input layer from input  $k$  to neuron  $j$  of the hidden layer,  $In_k = k$ -th input,  $b1_j$  = bias of the hidden layer of the  $j$ -th neuron,  $b2_i$  = bias of the output layer at the  $i$ -th output, and  $I = i$ -th output.

#### 4. Results and Discussion

After training the neural network, the results were obtained as follows (see Table 2). The ANN was trained with 75% of data, and the other 25% of data were used for validation and testing (see Section 3).

In order to verify learning of the network, a database with 7,420 experimental data was used. These data were the heating time ( $t$ ) from 0 to 10.31 h, the heating temperature (Thf) from 24.7°C to an average maximum temperature of 81.2°C, and the reactor geometry values which were constant, as well as the amounts of  $NH_3$  and  $BaCl_2$  (see Table 1).

By training the network with the appended database and verifying its learning, a correlation coefficient of  $r = 0.9671$  was obtained (see Section 3). Applying an alternating group to the ANN training database and checking its learning with this new database, a correlation coefficient of  $r = 0.9920$  was obtained. Finally, applying an alternating group and then

applying elements from the multibase mathematical symmetry group to the ANN training database and verifying their learning with the new database, a correlation coefficient of  $r = 0.9957$  was obtained. For a fair comparison, in all ANN trainings with different permutations (alternating group and multibase mathematical symmetry group) from the training database, the network weights were initialized with a random seed of 1230.

Figure 7(a) shows that the behavior of simulated temperature  $Tr_3$  during the addition of heat to  $BaCl_2-NH_3$  (sensible heat), against the experimental data, coincides with a correlation coefficient of  $r = 0.9996$ . During the phase change or desorption, the difference of the simulated data against the experimental data becomes minimal.

In the  $Tr_2$  case (Figure 7(b)), it is observed that the simulated and experimental  $Tr_2$  temperatures are consistent and have a correlation coefficient of  $r = 0.9924$ . It should be noted that, during desorption, there are some minimal differences between simulated and experimental values; however, from the experimental point of view, this is acceptable. The evolution of the  $Tr_1$  simulated temperature (Figure 7(c)) coincides with a correlation coefficient of  $r = 0.9953$  which is acceptable from a practical point of view.

The simulated values of the refrigerant volume versus the experimental ones coincide with a correlation coefficient of  $r = 0.99$  (Figure 7(d)). It should be noted that the

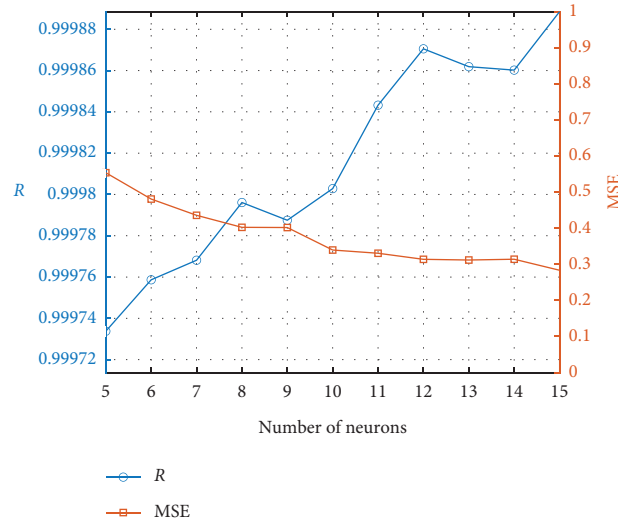


FIGURE 6: The effect of the neuron number of the hidden layer on the correlation coefficient “R” and the mean square error “MSE” for the ANN model.

TABLE 2: Weights and bias of the neural network.

	Input layer								
	<i>r</i> 1	<i>r</i> 2	<i>r</i> 3	Thf	Vs	Vg	<i>t</i>	<i>H</i>	Dn
Neuron 1	3.26E+00	1.63E+00	0.00E+00	4.81E−02	1.40E−04	2.26E−04	−9.14E−01	1.39E−01	5.96E−02
Neuron 2	−1.71E+00	−8.56E−01	0.00E+00	1.48E−01	−7.36E−05	−1.19E−04	7.18E−01	−7.29E−02	−3.13E−02
Neuron 3	4.29E−01	2.15E−01	0.00E+00	5.96E−02	1.85E−05	2.97E−05	5.72E−01	1.83E−02	7.85E−03
Neuron 4	3.90E+00	1.95E+00	0.00E+00	−1.75E−01	1.68E−04	2.70E−04	−1.39E−01	1.66E−01	7.13E−02
Neuron 5	−4.39E−01	−2.19E−01	0.00E+00	1.65E−02	−1.89E−05	−3.04E−05	−5.77E−03	−1.87E−02	−8.03E−03
Neuron 6	3.60E+00	1.80E+00	0.00E+00	−3.76E−01	1.55E−04	2.49E−04	1.59E+00	1.53E−01	6.58E−02
Neuron 7	3.03E−01	1.52E−01	0.00E+00	−5.32E−03	1.30E−05	2.10E−05	−1.18E−01	1.29E−02	5.55E−03
Neuron 8	−6.36E−02	−3.18E−02	0.00E+00	−8.44E−02	−2.74E−06	−4.41E−06	1.37E−01	−2.71E−03	−1.16E−03
Neuron 9	2.76E+00	1.38E+00	0.00E+00	−2.16E−01	1.19E−04	1.91E−04	−2.28E+00	1.17E−01	5.05E−02
	Hidden layer								
	Bias								
	1	2	3	4	5	6	7	8	9
<i>b</i> 1	−1.13E+01	−5.76E+00	−9.33E+00	−4.22E+00	9.58E−01	−5.05E+00	−1.03E+00	4.56E+00	4.26E+00
	Output layer								
	Inputs								
	1	2	3	4	5	6	7	8	9
Neuron 1	−5.86E−02	4.22E+00	−3.83E+00	9.03E−01	5.15E+01	6.45E−01	−4.80E+00	−2.78E+00	−1.64E+00
Neuron 2	−4.09E−01	3.27E+00	−3.77E+00	8.74E−01	5.41E+01	8.69E−01	−1.03E+01	−7.84E−01	2.38E−01
Neuron 3	−1.10E+01	1.05E+00	−9.41E−03	8.09E−01	−3.78E+00	−3.65E+00	−4.87E+00	−1.28E+01	−1.08E−01
Neuron 4	3.88E+00	−2.67E+01	3.95E+01	−3.04E+01	−5.10E+01	9.09E−02	−6.28E+01	−4.04E+00	−7.94E+00
Neuron 5	8.20E−02	4.34E−01	−2.80E−01	1.70E+00	5.02E+00	3.54E−01	−2.67E+00	−2.41E+00	2.14E−01
	Output layer								
	Bias								
<i>b</i> 2	6.19E+01	6.08E+01	4.83E+01	3.76E+01	6.93E+00				

experimental data were visually recorded every five minutes. However, it has an approximate value to the recorded final volume at the experimental desorption.

The simulated pressure inside the reactor (Figure 7(e)) is consistent with the sensible heating process, as it occurred with temperatures at different radii, and it has a correlation coefficient of  $r=0.9503$ .

During desorption, the pressure must have a constant behavior; however, in the experimental stage, there is a gradual increase. In the case of simulated pressure, the same increase occurs, with a maximum difference of 1 bar; it must be taken into account that the pressure values in Figure 3(c) do not show an ascending or descending pattern when increasing Thf.

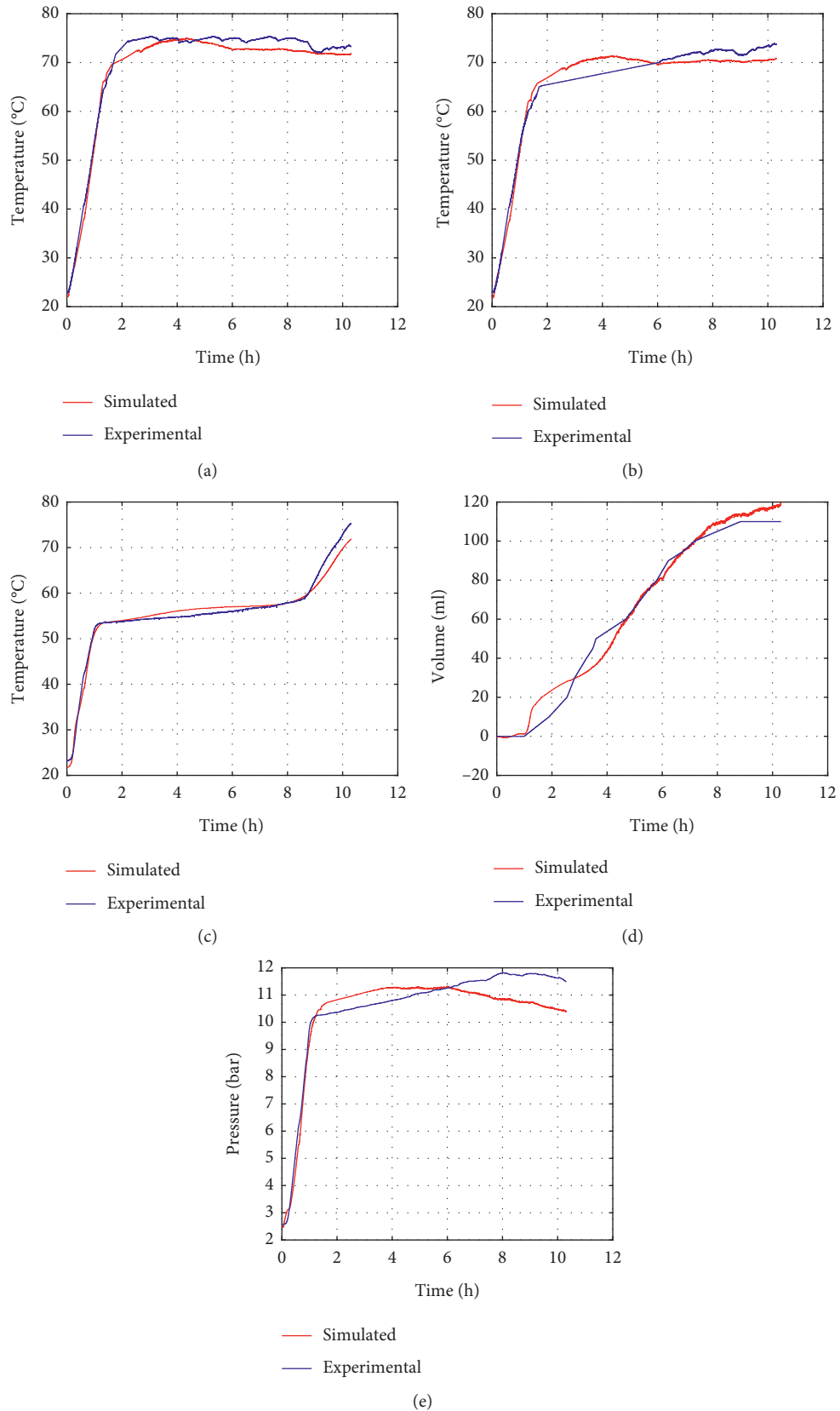


FIGURE 7: Testing the neural network (correlation coefficient  $r = 0.9957$ ): (a) temperature near the reactor wall (Tr3); correlation coefficient  $r = 0.9996$ , (b) temperature at the midpoint between the centre and the reactor wall (Tr2); correlation coefficient  $r = 0.9924$ , (c) temperature at the reactor's centre (Tr1); correlation coefficient  $r = 0.9953$ , (d) ammonia produced; desorption volume ( $V_g$ ); correlation coefficient  $r = 0.99$ , and (e) pressure into the reactor ( $P_g$ ); correlation coefficient  $r = 0.9503$ .



TABLE 3: Result of sensitivity analysis of the input parameters of the neural network (all data are multiplied by  $1e+57$ ).

Output	Input								
	r1	r2	r3	Thf	Vs	Vg	t	H	Dn
Tr3	0	0	0	0.01442	0	0	0.000000000000006	0	0
Tr2	0	0	0	1.971712	0	0	0.000000000000185	0	0
Tr1	0	0	0	0.163830	0	0	0.000000000000321	0	0
Vd	0	0	0	0.576825	0	0	0.00000000001319	0	0
Pg	0	0	0	0.202679	0	0	0.000000000000003	0	0

## 5. Sensitivity Analysis

In order to determine the impact of each input variable on the outputs estimated by the neural network, a sensitivity analysis was developed using a genetic programming algorithm executed in Eureqa [31]. The sensitivity analysis showed that the most used characteristic is the heating fluid's temperature (35%) followed by the desorption time (20%), that is, these variables are the most important in the ANN's learning process.

The sensitivity analysis for each of the input variables was also performed by generalizing the equations of Dimopoulos et al. [32] for  $M$  outputs, which is given as follows:

$$SSD_q = \sum_{p=1}^N d_q^{p^2}, \quad (12)$$

where  $SSD_q$  is the sensitivity of the  $q$ -th output of the neural network, for  $q = 1, 2, \dots, M$ ,  $N$  is the total input data to the neural network, and  $M$  is the total number of outputs of the network.

$$d_q^p = (d_{1q}^p, d_{2q}^p, \dots, d_{nq}^p), \quad (13)$$

where  $n$  is the total input parameters to the neural network.

$$d_{eq}^p = [f_q^o(x_q^o)] \sum_{i=1}^{ni} W_{qi}^o [f_i^h(x_i^h)] W_{ei}^h, \quad (14)$$

where  $ni$  is the number of neurons of the hidden layer,  $e = 1, 2, \dots, n$ ,  $f_q^o$  is the  $q$ -th activation function of the output layer,  $f_i^h$  is the  $i$ -th activation function of the hidden layer,  $W^o$  are the weights of the output layer,  $W^h$  are the weights of the hidden layer,  $W_{ei}^h$  is the weight from the entrance to the hidden layer, from the  $i$ -th network entrance to the  $e$ -th neuron of the hidden layer,  $W_{qi}^o$  is the weight that goes from the hidden layer to the output layer, from the  $i$ -th neuron of the hidden layer to the  $q$ -th neuron of the output layer, and  $[f_q^o(x_q^o)]$  and  $[f_i^h(x_i^h)]$  are those derived from the activation functions of the neurons of the output layer and the hidden layer, respectively.

In equation (14), the domains of the activation functions are defined as follows:

$$x_i^h = \sum_{j=1}^n W_{ij}^p x_j^p, \quad (15)$$

where  $x_j^p$  is the  $p$ -th input to train or simulate the network.

$$x_q^o = \sum_{i=1}^{ni} W_{qi}^o x_i^h, \quad (16)$$

where  $x_q^o$  is the domain of the activation function of the  $q$ -th neuron of the output layer.

Equations (12)–(16) were implemented in MATLAB yielding Table 3, as in Eureqa; it is concluded that the most important variables for the learning process of the neural network are Thf and  $t$ .

## 6. Uncertainty Analysis

In order to verify the accuracy improvement of the ANN, the uncertainties of the temperature sensors, pressure sensor, and desorption were analyzed. Some methodologies and standards for obtaining uncertainties are mentioned in [33–35]. All the uncertainties calculated for the sensors used in this research were based on evaluation of measurements described in [35], and a probability  $p = 0.99$  and a coverage factor  $k = 3$  were proposed. The expanded uncertainty of the J-type class 2 temperature sensor of Ametek brand [36] is  $U = 2.9^\circ\text{C}$ . The expanded uncertainty of the pressure sensor of Cole-Parmer brand [37] is  $U = 0.18$  bar. On the contrary, the desorption of the refrigerant was recorded manually, so the fuzzy set type 2 technique [34] and the central limit theorem were used to obtain the expanded uncertainty, obtaining  $U = 4.47$  ml. Using these uncertainties, the correlation coefficient of  $r = 0.9957$  was obtained between the experimental data (those never used for the network training) and the data simulated by the neural network.

## 7. Conclusions

The proposed artificial neural network allowed modeling the behavior of a  $\text{BaCl}_2\text{-NH}_3$  reactor globally with a precision of 0.9957 without using complex systems of differential equations. However, it is still necessary to improve the learning precision and the architecture of this neural network to model the behavior of these reactors. For example, it is necessary to improve the output of the neural network that simulates pressure, which has a correlation coefficient of  $r = 0.9503$ .

Applying an alternating group  $A_n$  of mathematical symmetry to the input data of the neural network allowed obtaining better precision in the learning process than only appending the  $70^\circ\text{C}$  and  $95^\circ\text{C}$  databases. However, when applying the permutations of the multibase mathematical symmetry groups to the input data of the neural network, a better precision ( $r = 0.9957$ ) was obtained in the mentioned

learning process than applying the alternating symmetry group  $A_n$ .

By applying the permutations of the mathematical symmetry group to the neural network input database, the training algorithm (Levenberg–Marquardt) converges faster and with greater precision compared to the appended database. However, given the high computational complexity of the permutations generated for the mathematical symmetry group, not all permutations were applied to the input data of the neural network.

As shown by the sensitivity analysis, the most important variables for the neural network training are heating fluid temperature ( $T_{hf}$ ) and desorption time ( $t$ ). The analysis of the uncertainties of experimental measurements allowed obtaining a higher learning precision of the neural network ( $r = 0.9957$ ).

### Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

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

### Acknowledgments

Special thanks are due to the Institute of Renewable Energies (IER) for their valuable help in the development of this work and to the Secretary of Public Education (SEP) for granting the National Conventional Scholarship for doctoral studies PROMEP/103.5/09/4363.

### References

- [1] L. Wang, X. Bu, and W. Ma, "Experimental study of an adsorption refrigeration test unit," *Energy Procedia*, vol. 152, pp. 895–903, 2018.
- [2] H. Z. Hassan, "Energy analysis and performance evaluation of the adsorption refrigeration system," *ISRN Mechanical Engineering*, vol. 2013, Article ID 704340, 14 pages, 2013.
- [3] F. Martínez-Tejeda, I. Pilatowsky, R. Best et al., "Experimental barium chloride-ammonia cooling cycle study at low generation temperatures," *Applied Thermal Engineering*, vol. 141, pp. 751–761, 2018.
- [4] K. R. Ullah, R. Saidur, H. W. Ping, R. K. Akikur, and N. H. Shuvo, "A review of solar thermal refrigeration and cooling methods," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 499–513, 2013.
- [5] R. Novella, V. Dolz, J. Martín, and L. Royo-Pascual, "Thermodynamic analysis of an absorption refrigeration system used to cool down the intake air in an Internal Combustion Engine," *Applied Thermal Engineering*, vol. 111, pp. 257–270, 2017.
- [6] G. L. An, L. W. Wang, J. Gao, and R. Z. Wang, "A review on the solid sorption mechanism and kinetic models of metal halide-ammonia working pairs," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 783–792, 2018.
- [7] F. Meunier, "Adsorption heat powered heat pumps," *Applied Thermal Engineering*, vol. 61, no. 2, pp. 830–836, 2013.
- [8] S. O. Enibe and O. C. Iloejei, "Transient analysis and performance prediction of a solid absorption solar refrigerator," *Journal of Solar Energy Engineering*, vol. 61, no. 1, pp. 43–59, 1997.
- [9] L. L. Vasiliev, D. A. Mishkinis, A. A. Antukh, and L. L. Vasiliev, "Solar-gas sorption refrigerator," *Adsorption*, vol. 7, no. 2, pp. 149–161, 2001.
- [10] R. G. Oliveira and R. Z. Wang, "A consolidated calcium chloride-expanded graphite compound for use in sorption refrigeration systems," *Carbon*, vol. 45, no. 2, pp. 390–396, 2007.
- [11] K. NagaMalleswara Rao, M. Ram Gopal, and S. Bhattacharyya, "Analysis of a SrCl<sub>2</sub>-NH<sub>3</sub> solid sorption refrigeration system," *International Journal of Low-Carbon Technologies*, vol. 10, no. 4, pp. 365–373, 2013.
- [12] C. Rivera, I. Pilatowsky, E. Méndez, and W. Rivera, "Experimental study of a thermo-chemical refrigerator using the barium chloride-ammonia reaction," *International Journal of Hydrogen Energy*, vol. 32, no. 15, pp. 3154–3158, 2007.
- [13] C. Dueñas, I. Pilatowsky, R. J. Romero, A. Oskam, and P. A. Finck, "Dynamic study of the thermal behaviour of solar thermochemical refrigerator: barium chloride-ammonia for ice production," *Solar Energy Materials and Solar Cells*, vol. 70, no. 3, pp. 401–413, 2001.
- [14] P. Neveu and J. Castaing, "Solid-gas chemical heat pumps: field of application and performance of the internal heat of reaction recovery process," *Heat Recovery Systems and CHP*, vol. 13, no. 3, pp. 233–251, 1993.
- [15] T. X. Li, R. Z. Wang, L. W. Wang, and J. K. Kiplagat, "Performance study of a consolidated manganese chloride-expanded graphite compound for sorption deep-freezing processes," *Journal Applied Energy*, vol. 86, no. 7–8, pp. 1201–1209, 2009.
- [16] Y. Zhong, R. E. Critoph, R. N. Thorpe et al., "Isothermal sorption characteristics of the BaCl<sub>2</sub>-NH<sub>3</sub> pair in a vermiculite host matrix," *Journal of Thermal Science and Engineering*, vol. 27, no. 14–15, pp. 2455–2462, 2007.
- [17] J. V. Veselovskaya, M. M. Tokarev, and Y. I. Aristov, "Novel ammonia sorbents porous matrix modified by active salt for adsorptive heat transformation 1. Barium chloride in various matrices," *Journal of Thermal Science and Engineering*, vol. 30, no. 6–7, pp. 584–589, 2010.
- [18] H.-B. Lu, N. Mazet, and B. Spinner, "Modelling of gas-solid reaction-coupling of heat and mass transfer with chemical reaction," *Chemical Engineering Science*, vol. 51, no. 15, pp. 3829–3845, 1996.
- [19] D. Stitou and G. Crozat, "Dimensioning nomograms for the design of fixed-bed solid-gas thermochemical reactors with various geometrical configurations," *Chemical Engineering and Processing: Process Intensification*, vol. 36, no. 1, pp. 45–58, 1997.
- [20] P. Neveu and J. Castaing-Lasvignottes, "Development of a numerical sizing tool applied to a solid-gas thermochemical transformer-II. Influence of external couplings on the dynamic behaviour of a solid-gas thermochemical transformer," *Journal of Thermal Science and Engineering*, vol. 17, no. 6, pp. 519–536, 1997.
- [21] H.-B. Lu and N. Mazet, "Mass-transfer parameters in gas-solid reactive media to identify permeability of IMPEX," *AIChE Journal*, vol. 45, no. 11, pp. 2444–2453, 1999.
- [22] B. A. Escobedo-Trujillo, D. Colorado, W. Rivera, and F. A. Alaffita-Hernández, "Neural network and polynomial



- model to improve the coefficient of performance prediction for solar intermittent refrigeration system,” *Solar Energy*, vol. 129, pp. 28–37, 2016.
- [23] V. Baiju and C. Muraleedharan, “Exergy assessment of single stage solar adsorption refrigeration system using ANN,” *ISRN Mechanical Engineering*, vol. 2012, Article ID 915154, 2012.
- [24] Y. Liu, N. Dinh, Y. Sato, and B. Niceno, “Data-driven modeling for boiling heat transfer: using deep neural networks and high-fidelity simulation results,” *Applied Thermal Engineering*, vol. 144, pp. 305–320, 2018.
- [25] R. Roohia, M. Jafarib, E. Jahantab et al., “Application of artificial neural network model for the identification the effect of municipal waste compost and biochar on phytoremediation of contaminated soils,” *Journal of Geochemical Exploration*, vol. 208, Article ID 106399, 2019.
- [26] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [27] Oriol Vinyals, S. Bengio, and M. Kudlur, “Order matters: sequence to sequence for sets,” Published as a conference paper at ICLR, 2016.
- [28] A. G. Kurosh, *The Theory of Group*, Chelsea Publishing Company, New York, NY, USA, 2nd edition, 1960.
- [29] E. Shelomov, J. Seguel, A. Ortiz, J. Ruz, and G. Shelomova, “Fourier analysis in commutative finite groups with matlab,” in *Proceedings of the Memories of the VII Conference of Electrical Engineering CINVESTAV*, pp. 90–116, Mexico city, Mexico, November 2002.
- [30] K. Gnana Sheela and S. N. Deepa, “Review on methods to fix number of hidden neurons in neural networks,” *J. Math. Probl. Eng.*, vol. 2013, Article ID 425740, 11 pages, 2013.
- [31] *Eureqa 1.24.0*, Nutonian Inc., Cambridge, MA, USA, 2019, <http://www.nutonian.com..>
- [32] Y. Dimopoulos, P. Bourret, and S. Lek, “Use of some sensitivity criteria for choosing networks with good generalization ability,” *Neural Processing Letters*, vol. 2, no. 6, pp. 1–4, 1995.
- [33] Y. Liu, N. T. Dinh, R. C. Smith, and X. Sun, “Uncertainty Quantification of Two-Phase flow and boiling heat transfer simulations through a data-driven modular Bayesian approach,” *International Journal of Heat and Mass Transfer*, vol. 138, pp. 1096–1116, 2019.
- [34] J. E. Moreno, M. A. Sanchez, O. Mendoza et al., “Design of an interval Type-2 fuzzy model with justifiable uncertainty,” *Information Sciences*, vol. 513, pp. 206–221, 2020.
- [35] *Guide to the Expression of Uncertainty in Measurement*, ISO, Geneva, Switzerland <https://www.bipm.org/en/publications/guides/gum.html>.
- [36] M. T. Thermocouple Assembly, *Data Sheet 1500. Class 2 Type-J*, Ametek, Berwyn, PA, USA, 2020, [https://www.ametekcalibration.cn/-/media/ametekcalibration/download\\_links/temperature%20sensors/fp%20sensors/1500%20and%201600%20series/temperature-sensors-1500-series-datasheet-us.pdf](https://www.ametekcalibration.cn/-/media/ametekcalibration/download_links/temperature%20sensors/fp%20sensors/1500%20and%201600%20series/temperature-sensors-1500-series-datasheet-us.pdf).
- [37] Cole-Parmer, *High-Accuracy Pressure Transducers* Cole-Parmer, Vernon Hills, IL, USA, 2020, <https://www.coleparmer.com/p/cole-parmer-high-accuracy-pressure-transducers/15143.visited>.

## Research Article

# A Machine-Learning Based Nonintrusive Smart Home Appliance Status Recognition

Liston Matindife,<sup>1</sup> Yanxia Sun,<sup>1</sup> and Zenghui Wang <sup>2</sup>

<sup>1</sup>Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park 2006, South Africa

<sup>2</sup>Department of Electrical and Mining Engineering, University of South Africa, Florida 1710, South Africa

Correspondence should be addressed to Zenghui Wang; wangzengh@gmail.com

Received 18 June 2020; Revised 1 September 2020; Accepted 7 September 2020; Published 18 September 2020

Academic Editor: Qian Zhang

Copyright © 2020 Liston Matindife et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In a smart home, the nonintrusive load monitoring recognition scheme normally achieves high appliance recognition performance in the case where the appliance signals have widely varying power levels and signature characteristics. However, it becomes more difficult to recognize appliances with equal or very close power specifications, often with almost identical signature characteristics. In literature, complex methods based on transient event detection and multiple classifiers that operate on different hand crafted features of the signal have been proposed to tackle this issue. In this paper, we propose a deep learning approach that dispenses with the complex transient event detection and hand crafting of signal features to provide high performance recognition of close tolerance appliances. The appliance classification is premised on the deep multilayer perceptron having three appliance signal parameters as input to increase the number of trainable samples and hence accuracy. In the case where we have limited data, we implement a transfer learning-based appliance classification strategy. With the view of obtaining an appropriate high performing disaggregation deep learning network for the said problem, we explore individually three deep learning disaggregation algorithms based on the multiple parallel structure convolutional neural networks, the recurrent neural network with parallel dense layers for a shared input, and the hybrid convolutional recurrent neural network. We disaggregate a total of three signal parameters per appliance in each case. To evaluate the performance of the proposed method, some simulations and comparisons have been carried out, and the results show that the proposed method can achieve promising performance.

## 1. Introduction

*1.1. Background and Motivations.* It is now common today to remotely monitor and control various appliances in the smart-home [1, 2]. The monitoring system is often integrated into the Internet of Things (IoT). In addition to standalone appliances, the smart home is composed of security, air-conditioning personalised medical equipment, and plug-in-electrical-vehicles (PEVs) [3, 4] monitoring. In the smart home, a convenient way to automatically establish the on/off operational status and identity of an appliance is through the nonintrusive load monitoring (NILM) recognition method which was firstly proposed by Hart in 1992 [5–7]. The NILM method establishes the identity of an appliance through the intelligent extraction of that

appliance's specific load signal information from an aggregate load profile acquired through a single signal sampling unit on the main power cable into the building. In contrast, sensors dedicated to each appliance define the intrusive load monitoring (ILM) [5] system. However, the ILM method involves a large number of sensors and extensive cabling in the house. Another recognition scheme known as the semi-intrusive load monitoring (SILM) [8] system only obtains part samples of the aggregate energy and guesses the remainder. SILM cannot give accurate specific load disaggregation but is appropriate for aggregate energy forecasting and needs some sensors and cabling.

The main thrust of the NILM systems is smart-home demand side energy management, whether it is based on single appliance or system based. Hence, we need to know

which and when the appliance/system is switched on or off. Load signal extraction and identification is achieved with high performance when the appliance component signals are due to large power appliances such as electric car charging that have widely varying power differences and whose signatures are very different from each other. The electric car charging in the smart home is now a prominent feature requiring consideration in the NILM recognition system design. The authors in [9] showed that the electric car charging can successfully be implemented into the NILM system using data from the Pecan Street Inc. Dataport. There are a number of challenges facing NILM recognition systems for achieving high recognition performance and they include the follows: (1) the system includes some equal or very close power specification electronic appliances (EVPSAs) during steady state operation and having basically identical signature characteristics, (2) the system has low power appliances that are difficult to recognize and are often interpreted as noise when the aggregate is composed of low and high power appliances (LHPAs), (3) the system includes continuously variable operating states' (CVOS) appliances, and (4) the same power appliances are switched on/off at the same time [5–7, 10]. However, in this paper, motivated by the need to differentiate and monitor the ever increasing array of EVPSAs in the smart home, we limit our research only to challenge (1) above. When summed up a large number of same specification laptops, televisions, refrigerators, light-emitting diode (LED) lamps, etc. will contribute significantly to the energy used in the smart home, and it becomes necessary to identify the operational status for each appliance through a deep learning NILM recognition system. Also, a high number of appliances in the house results in a higher overlap of their respective individual signals and switching events. A few studies often with complex detection algorithms [11, 12] have actively been involved in the NILM recognition of EVPSAs. In this paper, we fill the gap in the established literature by introducing less complex new deep learning model configurations with enhanced computation time and high accuracy for the NILM recognition of EVPSAs. By proposing three deep learning disaggregation algorithms, based on the multiple parallel structures convolutional neural networks (MPS-CNNs), the recurrent neural network (RNN) with parallel dense layers for a shared input, and the hybrid convolutional recurrent neural network (CNN-RNN), we aim to achieve a considerable improvement in the NILM recognition of EVPSAs. In this study, we propose to use in-house generated data from similar low power appliances such as light-emitting diode (LED) main lamps as opposed to the high energy consumption of the electric car charging since they are more difficult to be recognized.

*1.2. Literature Survey.* In the literature, we identify three approaches to detecting similar appliance signals in the NILM recognition systems. These are (1) event detection [13–15], (2) machine learning with hand crafted features, multiple classifiers, and complex algorithms [11, 12, 16–19], and (3) deep neural networks [3, 4, 10, 20–23]. Event

detection algorithms are premised on being able to extract a large number of unique signature characteristics at the beginning, end, and during the transient period. The CUSUM and genetic algorithm have been implemented in solving the recognition challenge due to appliance disaggregated signals that are similar to each other [13]. With reference to the NILM system, the CUSUM adaptive filter is based on adaptive threshold (difference between maximum and minimum value of the parameter being measured within the transient period and the starting and ending of the transient detection [13]). By doing so the filter is capable of extracting the signal information during fast and slow transients. The Genetic Algorithm (GA) on the other hand obtains a fitness function that converges to zero for successful appliance signal recognition [13]. However, although it is capable of extracting a large number of appliance signatures, both the CUSUM adaptive filter and GA are complex requiring involved design. The authors in [14] proposed a high accuracy event detection algorithm (High Accuracy NILM Detector (HAND)) characterized by low complexity and better computation speeds. The HAND monitors the standard deviation of the current signal through the transient period and is capable of detecting unique signal magnitudes within the transient. However, this algorithm suffers suppressed recall value and the precision is sensitive to noise [14]. In [15], an unsupervised clustering event detection algorithm is proposed, which functions on noting the original signal state before and after an event. The approach in [15] is incapable of high recognition at low frequencies. Hence, requiring extra consideration of a large count of high frequency features adds to the complexity and cost of data acquisition.

Machine learning with hand crafted features, multiple classifiers, and complex algorithms seeks to avail a large number of signal features for discrimination between similar appliance signals often through carefully designed feature extraction algorithms for processing through various machine learning models. To date a large number of NILM systems have been developed around Hidden Markov models (HMMs), as HMMs achieve enhanced recognition and reduced computational capabilities. However, HMMs have limited discrete value modeling capability and the algorithms are complex [6, 16]. An emerging method, the NILM Graph Spectral Clustering aggregate energy forecasting method, mentioned in [17] assumes prior knowledge of the appliances' on/off states to provide future disaggregated signal duration of each appliance. This method has a deficiency in the conventional NILM system design as it assumes that appliance will in future always operate as in the past. In reality, appliances are randomly switched on/off at times for varying periods spanning from their minimum operational activation times to up to many hours, days, or weeks depending. Hence, it becomes difficult to implement the design for constantly changing on/off appliance states. The method in [17] is applicable where we have data acquisition of appliances' operating states over very long past periods, unlike in our case where we have limited data as it is the norm in many NILM systems. To this end, the authors in [17] acknowledge the need to enhance the forecasting

capability of this system. In [18], the authors proposed the disaggregation and classification of high power resistive and reactive appliances. They consider step change in implementing their disaggregation to include true and reactive powers of appliances with widely varying signatures. However, the NILM recognition system in [18] is incapable of disaggregating or classifying similar signatures due to its reliance on differentiating between active and reactive powers and on an appreciable level difference between like powers.

Still under machine learning with hand crafted features, multiple classifiers, and complex algorithms, the authors in [19] proposed to improve on the recognition of similar appliances from previous work based only on true power parameter level change by adding more features extracted in total from the true power, reactive power, and power factor of the respective signal. The authors in [19] went on further to propose the MinMaxSteady-State algorithm that constitutes hand crafting of the steady state features from the power and power factor signals. By hand crafting the steady state feature extraction, we increase the complexity of the system and at the same time we limit the system performance since it is difficult by trial and error to determine exactly the number of features required to provide absolute recognition of the appliance signals. In [18, 19], the performance of various classification algorithms that include the decision tree, 5-nearest neighbour, discriminant analysis, and support vector machine was investigated. The decision tree algorithm provided the highest identification rate of appliances for the said classifiers.

In [11], the generalized NILM algorithm provides a considerable improvement in the recognition of similar appliances here given by detecting between iron and kettle. In this algorithm, any machine learning classifier can be used in the recognition. However, different classifiers are assigned to a limited number of features out of the whole set of features under consideration. As in [18], the authors in [11] also consider a step change in the initiation of their disaggregation part of the NILM system. In the finality of the disaggregation, they consider an elaborate design to select an optimal number features out of possible nine features. In [11], the selected features are mean current, DC component, mean power, and for the first sixteen harmonics (active power, reactive power, real and imaginary current components, and conductance and susceptance values). Although the method in [11] gives good discrimination, among the various appliance signals, under consideration, the overall performance of the classifier on the identification between similar appliances requires further improvement as alluded to by the same authors in their conclusions. Furthermore, the number of hand-crafted features under consideration is very high, requiring a complex feature selection and extraction algorithm. In [12], the hierarchical support vector machine (HSVM) classifier is proposed for the classification of the disaggregated signals. However, the HSVM burdens the computational resources of the system. As in [11, 18], the authors in [12] also consider a step change in the formulation of their NILM disaggregation comprising a host of hand-crafted features that include average, peak

value, root mean square, standard deviation, crest factor, and form factor for analysis per appliance. In addition to formulation of hand crafted event detection and hand crafted feature extraction, in [12], we observe a slightly suppressed average classification accuracy of 98.8% due to the HSVM.

The advent of deep learning algorithms has allowed for an accelerated increase in the development and performance of NILM recognition systems. In [20], the authors propose the following three deep learning neural networks for the NILM recognition: (1) recurrent neural network and (2) denoising autoencoder, and a model based on considering the steady state operation value and appliance activation start and end times. The experiments in [20] are performed using high power appliances that have widely varying signatures and result in acceptable average  $F$ -measures ( $F1$  scores) that are however less than unity. The appliances considered in [20] are kettle, dish washer, fridge, microwave oven, and washing machine. The research here [20] forms one of the basis for application of deep learning to the NILM recognition, and as such requires further improvement as alluded to by the authors in their conclusions. In [20], networks (2) and (3) performed reasonably well for recognition of unseen appliance data, whilst network (1) did not perform well on unseen data. However, all the networks in [20] still need considerable improvement. In [21], the authors propose to predict the extent to which Parkinson's disease is manifest from gait generated data. Just like in NILM recognition, the system in [21] tries to infer an outcome from a composite input of gait information. An averaged output from the result of a parallel combination of a long-short-term memory (LSTM) network and convolutional neural network (CNN) model is obtained. The good results in [21] show that both LSTM and CNN models can be adopted for use in the NILM recognition system as the formats of the power series signals are the same in both cases.

Still under deep learning algorithms in [22] the authors propose a CNN NILM system based on differential input, with the aim of achieving higher performance than systems based on "raw" data. This is somewhat a form of signal preprocessing obtained by differentiating the raw data into power change signals. An auxiliary raw data feed is then applied in parallel to the differential input to provide additional mean, standard deviation, and max and min signal information. However, a well-constructed deep CNN network is capable of high performance internal signal differentiation and feature selection without the need for preprocessing the signals. Furthermore, the authors in [22] used a standard dataset that includes a dishwasher, fridge, and microwave oven without articulating the similar appliances signal issue. In [23], the authors propose a deep learning autoencoder-based NILM recognition system. Applying the concept of noise removal from speech signal, the authors in [23] are able to disaggregate the unique appliance signals from the aggregate with very high performance. However, in [23], the authors experiment on appliances that do not have similar signatures, and these are washing machine, desktop computer, electric stove, and electric heater. In [10], the authors approach the NILM



recognition through a convolutional neural network (CNN) applied to appliance voltage-current (V-I) trajectories. The V-I trajectories are transformed to the image form for input to the CNN. The features in [10] are attributed to slope, encapsulated area etc. of the V-I trajectory. The authors in [10] consider data acquired from high frequency measurements and does not sufficiently address low frequency (1 Hz) data acquisition. In [10], the authors are able to recognize a large pool of appliances from the WHITED and PAID datasets with macroaverage F1 scores of 75.46% and 77.60%, respectively. Poor recognition between similar appliances is a contribution to the low F1 score.

Analogous to detecting similar appliance signals is the modeling of travel behavior patterns for designing a charging strategy for plug-in electric vehicle [3, 4]. In [4], Plug-in Electric Vehicles (PEVs) travel pattern prediction accuracies of up to 97.6% were obtained through a hybrid classification approach. Similar travel patterns are grouped together and assigned to a particular forecasting network. Using stored previous PEVs data (departure time, arrival time, and travelled distance), the approach in [4] first runs an unsupervised model to establish those masked travel-behaviour patterns and assigns them to a specific group. The grouped travel-behaviour patterns are then channelled to the respective supervised model for final recognition. The unsupervised and supervised operations are both performed by LSTM networks that are characterized by enhanced feature extraction capabilities. The results in [4] show that deep learning as opposed to legacy scenario-based demand modeling achieves very high performance in PEV systems. In [3], PEVs travel pattern prediction was obtained through the use of the Rough Artificial Neural Network (R-ANN) with reference to the recurrent neural network system. R-ANNs are capable of enhanced forecasting of the masked travel-behaviour patterns of PEVs. In [3], the Conventional Error Back Propagation (CEBP) and LevenbergeMarquardt training approach was used with the LevenbergeMarquardt achieving higher performance in training Plug-in Electric Vehicles-Travel Behaviour (PEVs-TB). The outcome of the research in [3] shows that the Recurrent Rough Artificial Neural Network (RR-ANN) approach allows for better PEV-TB and PEVs load forecasting than the reference Monte Carlo Simulation (MCS). The overall result in [3] is a substantial saving in the use of electricity by the PEVs. In context of our research, we extend the application of the LSTM model to the NILM disaggregation part.

*1.3. Paper Contribution.* In this paper, we address the deficiencies mentioned in [11–23] of the NLM disaggregation and classification of EVPSAs with similar signatures by improving the deep learning approach. Deep learning neural networks are good at mastering the complex nonlinear connection between the source aggregate signal and the output target appliance signal. The success of the NILM recognition depends in principle on the feature extraction capabilities of the designed system. Hence, we propose NILM models that will attempt to extract as much feature information as possible from the experimental signals.

Firstly, with the view of obtaining appropriate EVPSAs overall high performing disaggregation deep learning networks, we propose three deep learning disaggregation algorithms based on the multiple parallel structures convolutional neural networks (MPS-CNNs), the recurrent neural network (RNN) with parallel dense layers for a shared input, and the hybrid convolutional recurrent neural network (CNN-RNN). We then disaggregate a total of three signal parameters per appliance in each case for a limited number of similar signature appliances in the form of light-emitting diode (LED) main lamps. We propose CNN- and LSTM-based disaggregation networks. The CNN is a feed-forward neural network (FFNN) modelled on the naturally “vision perfect” biological visual cortex [24, 25] and has achieved extremely high levels of object recognition and classification. The LSTM network, on the other hand, which accurately models short and long term trends in the appliance signals [4], is characterized by enhanced feature extraction capabilities. Secondly, we propose an appliance classification strategy premised on the deep multilayer perceptron (MLP) having three appliance signal parameters as input to increase the number of trainable samples and hence accuracy. In the case where we have limited data, we implement a transfer learning- (TL-) based appliance classification strategy. In this paper, our first and second proposals attempt to fill the knowledge gap in the established literature by introducing less complex but powerful new deep learning model configurations with enhanced computation time and high accuracy for the NILM recognition of EVPSAs. The MLP feedforward neural network in its own right is an enhanced nonlinear problem solving deep neural network capable of high classification performance [26]. During data acquisition, we obtain three signal parameter values for both the aggregate and appliance target signals. We then perform a regression-based training of each disaggregation model based on the target parameters. Using the sliding window concept, we disaggregated the appliance signals through the trained disaggregation networks. We then use the mean summation of the part window disaggregated signals to obtain the overall disaggregated signals. We also train the classification network based on the three parameters of the ground truth signals and finally apply the disaggregated signal sums into the trained classification network for recognition. Our proposed NILM recognition system is tested on raw in-house generated data from similar LED main lamps. Disaggregation is carried out on all the appliances, and in the final analysis, we show the classification rates of all the appliances under test. To evaluate the performance of the proposed method, some simulations and comparisons are carried out. In summary, we make the following contributions in this study:

- (i) Incorporate an all-encompassing disaggregation feature extraction capability that includes step change, transient, and steady state values deep learning framework based on three separate deep learning disaggregation algorithms: the multiple parallel structure convolutional neural networks, the recurrent neural network with parallel dense layers

for a shared input, and the hybrid convolutional recurrent neural network to substantially increase the disaggregation performance of the NILM system

- (ii) Increase the classification accuracy by availing three parameters per signal into the classification network based on a simple deep learning multilayer-perceptron network

*1.4. Organization of the Paper.* The rest of this paper is structured as follows. Section 2 details the proposed methodology including the models, the proposed NILM recognition theory, aspects pertaining to data, performance metrics, verification of the proposed method performance to include proposed model description, pseudocode for proposed method, Keras model architectures, and the training framework and procedure. Section 3 gives a discussion of the experimental results, and Section 4 gives the conclusion.

## 2. Methodology

*2.1. The Proposed Models.* We propose our deep learning model structure based on the hybrid convolutional recurrent neural network (CNN-RNN). The CNN-RNN approach is referred to the GoogleNet model as done by the authors in [27]. However, we modify the concept and break it down into three possible networks for exploration in this paper. The first model in Figure 1 is premised on the multiple parallel structure convolutional neural networks (MPS-CNNs) disaggregation approach. In the GoogleNet model, we basically disaggregate one input parameter with a number of parallel feature extractors, whereas in our model, we disaggregate three independent input parameters, as shown in Figure 1. The second model in Figure 2 is a recurrent neural network in the form of an LSTM with parallel dense layers for a shared input for enhanced sequence prediction. The final model in Figure 3 is based on a hybrid convolutional recurrent neural network (RNN-CNN) that combines the enhanced feature extraction with ordered sequence prediction for CNN and RNN, respectively [28]. The authors in [29] use bidirectional LSTMs (BiLSTM or BLSTM) that preserve past and future information from combined hidden states for better interpretation of missing information. A BLSTM trained on the past and future information 12.17. . .12.175 will predict a 12.1725 instead of a likely 12.178 when trained on an LSTM. Notwithstanding the benefits of BLSTM, we will however base our LSTM models on forward pass ones only. Our models in this paper have three aggregate parameters separately disaggregated to give three individual mains lamp disaggregated signals. These three disaggregated signals become three (multivariate) signal inputs into the classification network with any one target signal of Watt,  $I_{rms}$ , or PF. Doing so may increase the appliance classification accuracy and improve on appliance generalization.

The idea for this research is to place a single measurement piece of equipment at the mains power cable input to the house, and to measure the current, power, and power factor parameters of four similar LED mains to find out

which LED is on or not. The recognition module can be housed in a separate meter box next to the original one, or in the house just after the mains circuit breaker, as shown in Figure 4.

This system is meant to recognize similar LED mains lamps effectively connected to an alternating current main power supply cable, either supplied through the power grid, standby generator, or photovoltaic inverter system, to determine which area of the building is illuminated. This project includes the hardware design, signal processing, and signal recognition. The software and hardware can be implemented on microchip or arduino microcontrollers. Besides, the smart-home proposed project can find application in commercial and industrial installations, where there is a large count of similar LED main lamps. The recognition project concept can be extended to other similar electronic appliances such as laptops in a school or company and similar televisions in a hotel. In Figure 4, the NILM unit can then be combined with Internet of Things (IoT) premised on industry 4.0 standard platform for remote access.

*2.2. The Proposed NILM Recognition Theory.* The typical NILM appliance identification process is made up of (1) acquisition of the composite load profile, (2) obtaining of appliance state transitions (events), (3) feature extraction, and (4) with reference to supervised and unsupervised learning obtaining the disaggregated appliance signal and its class [6]. In supervised learning, the input aggregate is trained against each appliance signature target. In unsupervised learning, there is no target training but an intermediate disaggregated signal is produced which is compared with a known signature databank for pairing; if no pairing is possible, then the intermediate signal is labelled as a new appliance signature. Acquisition of the composite signal can be carried out at high sampling frequencies of 1 kHz to 100 MHz [6]. However, 1 Hz low sampling frequencies are the norm as sampling integrated into smart meters requires simple hardware [8]. The data in our study has been sampled at this low 1 Hz frequency for ease of acquisition. The feature extraction and disaggregated and classification appliance signatures can either be taken as steady state or transient state [5, 6, 8, 30]. Switching transients for each appliance are of different amplitudes, contain unique settling times, and harmonics thereby defining a unique signature for each appliance. On the contrary, steady state features define the normal operational unique signatures of appliances. The mathematical expressions of the load signatures and composite profiles have conveniently been represented in [31]. In our study, the disaggregation problem stated in [31] is tackled by implementing the “pattern recognition” approach that allows us to use the deep learning algorithms that we have proposed.

*2.2.1. Deep Learning Algorithms.* Well-configured deep learning neural networks are capable of extracting a large number of different features that define an input signal, whereas some deep learning algorithms are better at

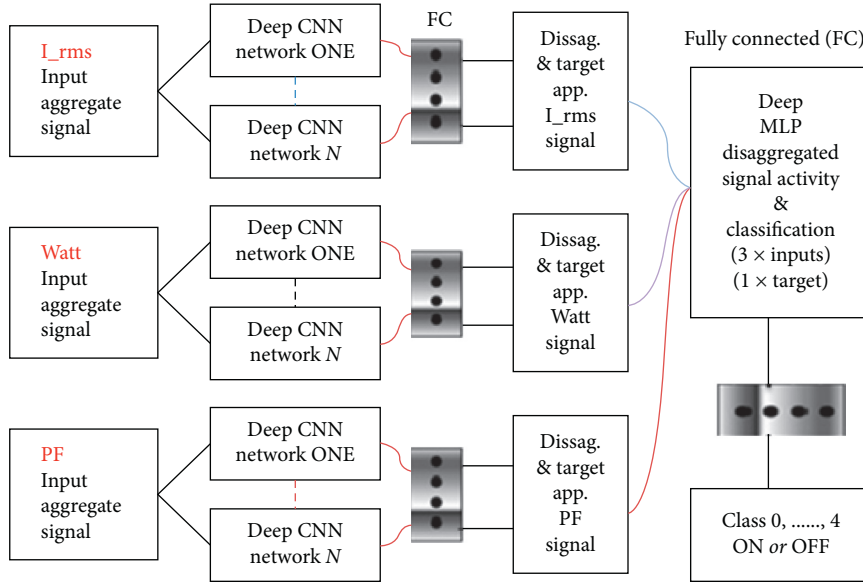


FIGURE 1: Multiple parallel structure convolutional neural networks. Incorporating the appliance disaggregation and classification.

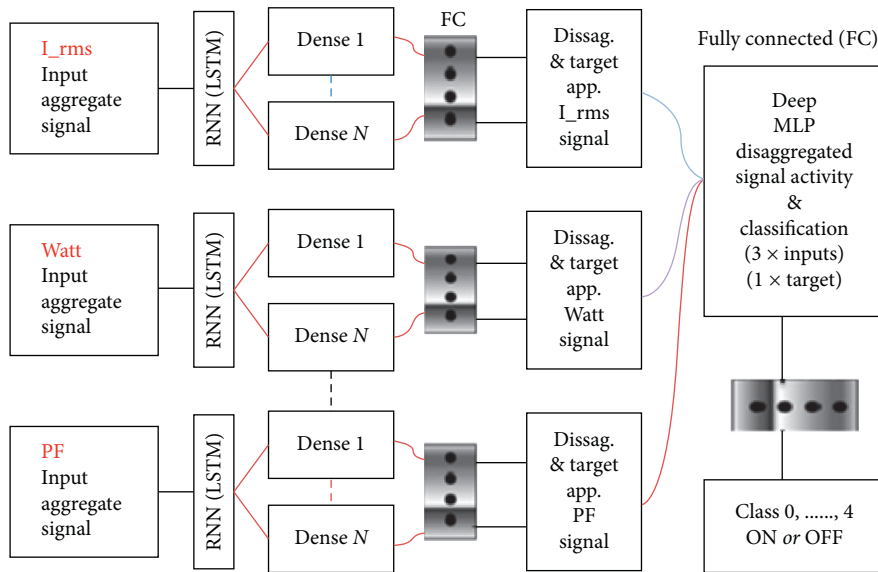


FIGURE 2: Recurrent neural network. Incorporating the appliance disaggregation and classification.

regression-based analysis some deep learning models such as the multilayer perceptron (MLP) feedforward neural network is more situated to classification [26]. However, the MLP normally forms the last stage of most CNN or RNN (LSTM) deep neural networks. According to [32], inputs bounded by convex polygon decision regions are sufficiently solved by two-layer feedforward networks where the inputs are continuous real and the outputs are discrete values. The underlying layers in a CNN are convolution, pooling or subsampling and fully connected or multilayer perceptron [24, 25]. The convolution through nonlinearity (ReLU) to pooling layers has feature extraction capabilities. Pooling effectively reduces the dimension of the preceding feature maps but maintaining all the important detail of the input, while the object recognition and classification is performed

through the backpropagation algorithm in the fully connected layer. CNNs also require little data preprocessing. The image can be a three (red, green, and blue) channel or single (greyscale) channel matrix with pixel values 0 to 255.

In this paper, the CNN is adapted to 1D aggregate appliance signal inputs and targets. A matrix (the filter, kernel, or feature detector) of smaller dimension than the input matrix is used as the feature detector. Different filter matrix entries will extract different features of the input image. In appliance classification, the number of outputs is required to be equal to the number of appliances under test [10, 29]. CNNs have recently been incorporated into Capsule Networks (CapsNets) for significantly improved feature extraction and recognition based on dynamic routing by agreement rather than max pooling of image-based datasets

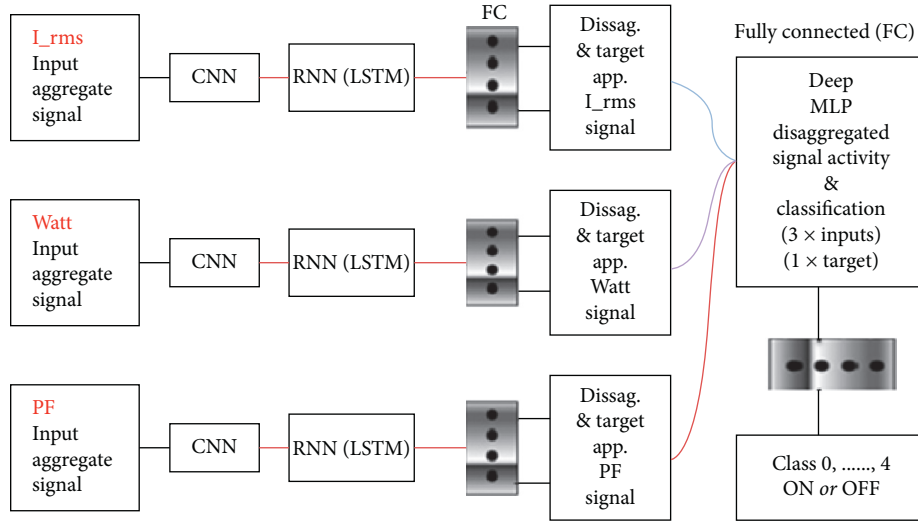


FIGURE 3: Hybrid convolutional recurrent neural network. Incorporating the appliance disaggregation and classification.

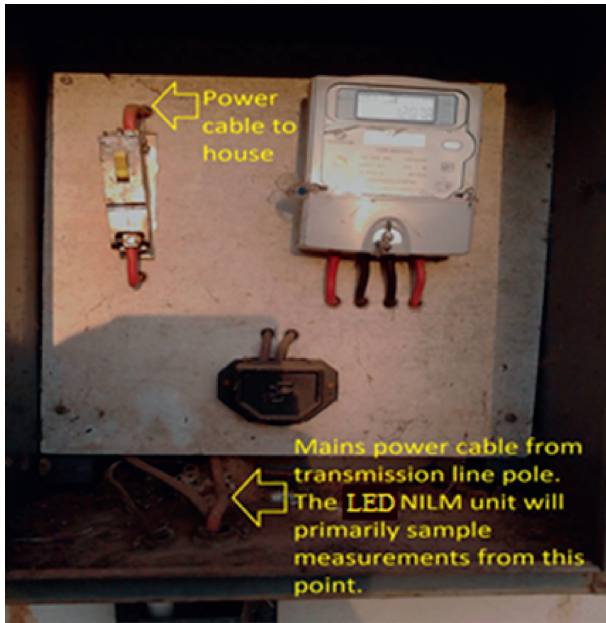


FIGURE 4: Power cables in the meter box. We show the provision for installing the LED main lamps NILM recognition module.

[33]. However, the application of CapsNets in the NILM scheme is not yet extensively documented and is not considered for application in this paper. Convolutional neural network training error can be significantly reduced by the use of a filter-based learning pooling (LEAP) CNN algorithm developed by the authors in [34]. However, in this paper, we use CNNs based on the traditional hand engineered average pooling scheme.

An RNN shown in Figure 5 is a neural network formulated to capture information from sequences and is based on considering immediate and just previous inputs in its calculations. As such the RNN has some memory attributes to easily enable it to decide the outcome of next input determined by the conditions of the stated present and just

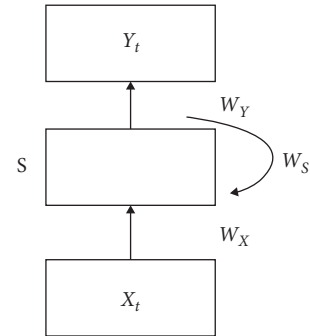


FIGURE 5: Recurrent neural network architecture. The outcome of next input is determined by the conditions of the stated present and just previous inputs.

previous inputs. A deep RNN is obtained by channelling consecutive  $S$  hidden layers from previous RNNs to subsequent RNN inputs. However, the RNN suffers a gradient problem which adversely affects model performance. To this end, the RNN-LSTM network is developed to solve the vanishing gradient issue by putting gating functions within its operation process [6, 10, 20, 35]. The RNN state expression is given in [1]

$$S_t = F_W(W_X X_t + W_S S_{t-1}), \quad (1)$$

where  $S_t$  is hidden state at time step  $t$ ;  $W_X$  is weights between hidden layer and input;  $W_S$  is weights between previous and current layers;  $X_t$  is input at time step  $t$ ;  $F_W$  is a recursive function (tanh or ReLU);  $W_Y$  is weights between hidden and output layers; and  $S_{t-1}$  is previous hidden state at time  $t - 1$ .

**2.2.2. Disaggregation.** As opposed to Hart’s disaggregation framework that emphasizes event detection rather than individual appliance disaggregation from the composite signal [27], in this paper, we focus on the latter technique. The authors in [22, 36] use a sliding window on the aggregate. Sliding windows that partially overlap with each



other have dimensions that depend on the appliance activation sizes. A median filter is then used to add the intermediate outputs to get the final output. Kelly and Knottenbelt [20] propose, on the contrary, the constitution of the intermediate outputs by considering their mean values. In this particular case, the output is recognized by the start, end, and mean values of the target appliance from the aggregate. While disaggregation considers on all the data points on the target appliance, classification is based on assigning a label value that relates the disaggregated signal to the ground truth appliance signature. The authors in [27] base their disaggregation scheme on the parallel connection of CNN/RNN layers with varying filter sizes of  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  as in the GoogleLeNet structure. These CNN/RNN layers are then concatenated together after having extracted a large number of useful signal features from the aggregate signal. In this paper, the training to validation datasets are split in the ratio 7 : 3, respectively.

**2.2.3. Transfer Learning-Based Classification.** The method of using a model trained on a larger dataset which is similar to the new smaller dataset is known as transfer-based learning. Transfer learning allows for the speedy development of new models on constrained datasets and allows the application of these models in more varied situations [37, 38]. Transfer learning is more compactly defined as follows [37].

*Definition 1.* Given a set of source domains  $DS = D_{s,1}, \dots, D_{s,n}$ , where  $n > 0$ , a target domain,  $D_t$ , a set of source tasks  $TS = T_{s,1}, \dots, T_{s,n}$ , where  $T_{s,i} \in TS$  corresponds with  $D_{s,i} \in DS$ , and a target task  $T_t$  which corresponds to  $D_t$ , transfer learning helps improve the learning of the target predictive function  $f_t$  in  $D_t$ , where  $D_t \notin DS$  and  $T_t \notin TS$ .

**2.3. Aspects Pertaining to Data.** We use a set of mains lighting lamps in the form of light-emitting diodes (LEDs) in our experiments. Three of the lamps are shown in Figure 6. The measurement setup is performed in the laboratory where we use the same length of extension cables from the mains to the lamps. Hence, we do not consider the effect cable length contribution to our collected data. We obtain three aggregate signal parameters sampled at 1 sec intervals per mains lighting lamp using a Tektronix PA1000 Power Analyser [39]. The parameters that we measured for each light-emitting diode lamp are voltage current ( $I_{rms}$ ), power (Watt), and power factor (PF). We create an appliance signature databank of all the individual mains lamps. These signals are our target data in the deep leaning training. We will not show the individual LED lamp signatures here, but in Section 3, when we compare these signatures (ground truths) with the reconstructed disaggregated signals as a way of accessing the performance of the disaggregation. Model simulation is performed in the Python 3.5 environment with Keras 2.2.2 TensorFlow 1.5.0 backend, Numpy, Pandas, and scikit-learn packages, on an Intel R CPU 1.60 GHz 4.00 GB Ram 64 bit HP laptop.

From the composite current ( $I_{rms}$ ) signal, as shown in Figure 7, a recognition strategy is developed for a set of three



FIGURE 6: Experiment LED lamps. We show three of the lamps with the fourth being LED1-2 having the same power specification as LED1-1 and LED2-1.

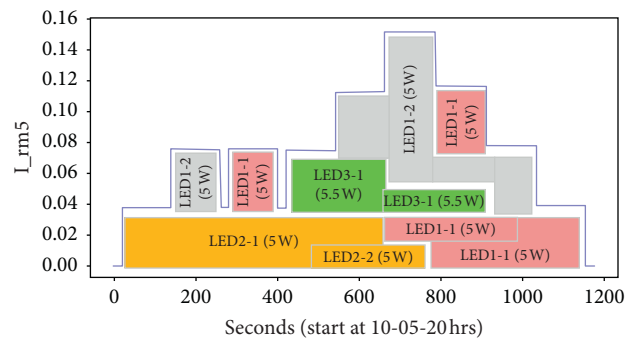


FIGURE 7: Equal power appliances (EPAs) aggregate current ( $I_{rms}$ ) signal acquired on 09 Nov. 2018, showing the various combinations of the main lamp activations.

5 W and one 5.5 W light-emitting diode (LED) lamps numbered as LED1-1 (Philips 5 W (60 W) 100 V–240 V), LED1-2 (Philips 5 W (60 W) 100 V–240 V), LED2-1 (Philips 5 W (60 W) 170–250 V), and LED3-1 (Radiant 5.5 W B22 Candle 230 V, 50 Hz, 5000 K). For example, we aim to disaggregate LED1-2 from LED1-2 and LED2-1 aggregate. The aggregate power (Watt) and power factor (PF) signals equally valid also are not shown. As can be seen in a 600 seconds window in Figure 8, from the dynamics of the four LEDs, there is an order of less than ten to the power minus 4 difference in current magnitude for three LEDs and very close relationships in the steady state profiles of all the LEDs. This shows close tolerance of the LED characteristics especially for LED1-2 and LED1-1 as expected from the specifications.

The aspects pertaining to the selection of the training signal points are

- (i) The overall length of the target series ( $T$ ) defines the input and output series lengths into and out of the network, respectively (regression training)
- (ii) The target series data should not be too long but enough to sufficiently define the ground truth signal
- (iii) The on/off points should be captured in the target and aggregate data, with the training period chosen to be longer than the appliance activation window that incorporates appliances' start and end

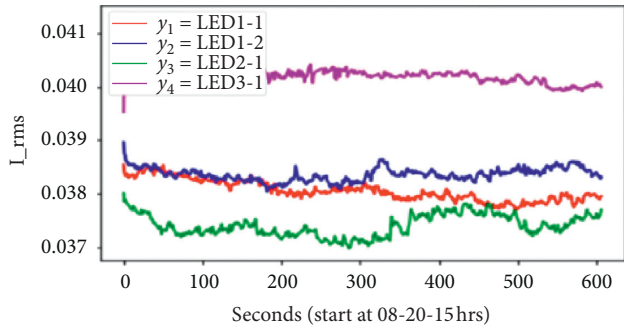


FIGURE 8:  $I_{rms}$  waveforms for the LED lamps (LED1-1, LED1-2, LED2-1, and LED3-1), showing the close tolerance of the parameter magnitude. For the first 300 seconds, LED1-1 and LED1-2 are hardly discernable from each other.

- (iv) The disaggregation algorithm should align the target and the point at which target becomes active in the aggregate signal

The overall length of the aggregate signal should contain all the information about the specific target appliance. We consider the shape of the aggregate data and accordingly reshape our input data into the DL network. We can generate artificial data where our raw data is too limited for deep learning. CNN and LSTM are both premised on a three-dimensional input whose shape is [number of samples, timesteps length, and number of features]. The hybrid CNN-LSTM system requires that we further obtain subsequences from each sample. The CNN works on the subsequences with the LSTM working, summarizing the CNN results from the subsequences.

The aggregate data is normalized and then standardized (zero mean and unit standard deviation) to improve on deep learning (DL) gradient convergence. DL algorithms require a large training dataset and as a result before the normalization and standardization the acquired dataset (only input training data) size is increased by considering all sections of the entire aggregate signal where the target appliance appears. For example, the input training set for LED1-2 is enlarged from 121 sample points to 614 (spanning 5 LED1-2 activations) sample points by considering the total aggregate data length covered by the grey areas in Figure 7. Likewise, for LED2-1, the total aggregate signal length is obtained by considering the orange areas, an increase from 119 sample points to 714 (spanning six LED2-1 activations) sample points. The further addition of artificially generated data as done by Kelly and Knottenbelt [20] in their 50:50 ratio of real aggregate data to artificially generated data will improve the ability of our network to generalize to “unfamiliar” appliances not involved in the training.

As in [20], we created additional artificial data by synthesizing random values between the maximum and minimum readings of the aggregate signal from the `RANDBETWEEN` function in excel. Although there is a further possibility of increasing the aggregate length by adding generated delayed versions of the total real aggregate

signal where that appliance appears, we experimented with only these increased real sample points plus synthesized samples to give respective total aggregate lengths of (614 real + 614 synthetic) for LED1-2 and (714 real + 714 synthetic) for LED2-1. The validation aggregate signal in Figure 9 is only real data without synthetic additions; however, this data is normalized and standardized. The validation dataset (containing the appliance activations) length is 441 samples in total with, for example, 121 to 363 samples for LED1-2 and 119 to 238 samples for LED2-1.

Data trains for Watt and PF are also available and applicable to the developed algorithm evaluation.

In this paper, using the prepared data, we first train the model in Figure 1 using only one network with varying filter sizes, and we obtain its performance, reconstruct the disaggregated signal, and compare it with the ground truth signal. We go on to add subsequent parallel networks and perform the overall networks’ performance evaluation until there is no more appreciable change as we add extra parallel arms. It is only after this do we employ the disaggregated signal for an absolute classification test. Like other researchers [22, 36], we also employ the sliding window shown in Figure 10 based on the appliances activation size in the disaggregation. During training and using data prepared from Figure 5, we go on to add another network to have a model with two parallel networks. For the second added network, we again vary the filter sizes and evaluate the resultant parallel networks’ performance and how good the reconstructed disaggregated signal is compared with the ground truth signal. In the second and final models, we gradually vary the RNN/LSTM memory cells while noting the performance.

We develop our recognition models in the random order of LED1-2, LED2-1, LED1-1, and LED3-1. For LED1-2, the actual target sample (divided by the largest value in that sample) length is 76 with four zeroes at start and end of series broken down as  $((68 \times 1) + (8 \times 1))$  features. The actual aggregate length is 1224 including four aggregate signal samples that have no information about LED1-2 at both ends of the series, broken down as  $((68 \times 18) + (8 \times 18))$  features. It should be noted that only one parameter is disaggregated at time, but three parameters are used in the classification.

The resultant disaggregated signal is obtained by finding the mean values of the window disaggregated parts. In some cases, the aggregate signals in Figures 7 and 9 span as little as 120 sample points with the disaggregated signal represented by as little as 68 sample points of data after the removal of redundancies. This represents limited data for use during the classification stage. Hence, we propose to use pretrained classification networks that use data spanning as much as 600 sample points for each ground truth signal obtained from an independent but related measurements, as shown in Figure 11. We then train the classification using this extended time series and implement transfer learning to test and classify the shorter disaggregated signals that are based on shorter initial target lengths. The disaggregation task is given by Pseudocode 1.

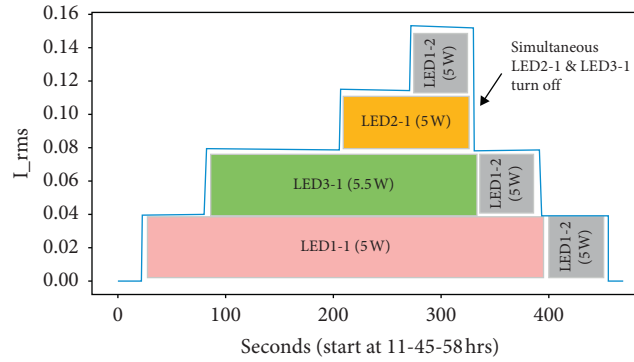


FIGURE 9: Disaggregation testing/validation data train acquired on 09 Nov. 2019 for the equal power appliances’ (EPAs) LEDs. This signal is input into the trained disaggregation models to extract the appliance activations and hence their signatures.

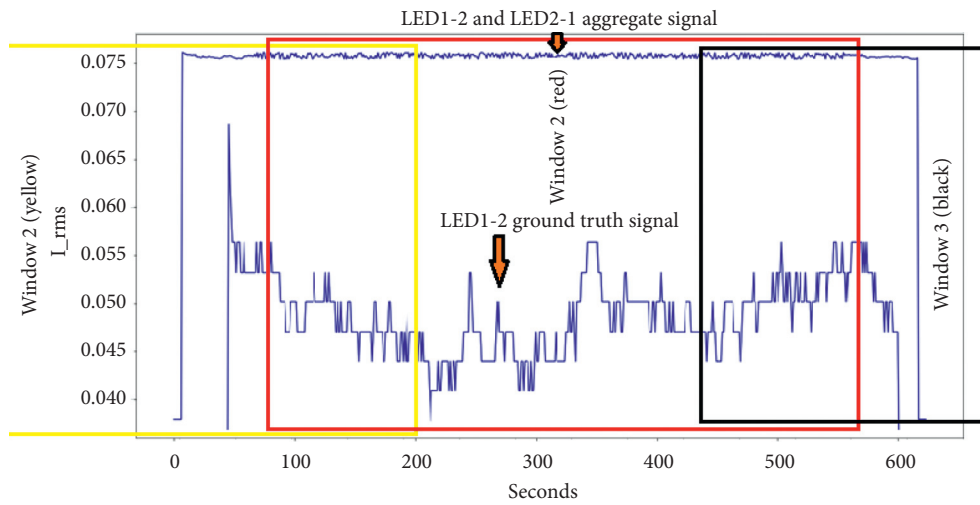


FIGURE 10: Sliding window in appliance disaggregation. The aggregate is a composite of LED1-2 and LED2-1. We disaggregate LED1-2 based on sliding window length equal to ground truth signal length.

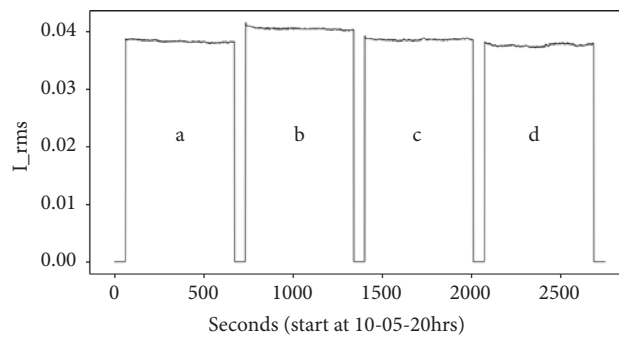


FIGURE 11: Independent classification data for LED lamps. This shows the LEDs series lengths for transfer learning acquired on 09 Nov. 2018. (a) LED1-1, (b) LED3-1, (c) LED1-2, and (d) LED2-1.

2.4. Performance Metrics. In this paper, for disaggregation performance, we consider the logcosh, root-mean-square-error (RMSE), mean\_squared\_error (MSE), and mean\_absolute\_error (MAE), and Coefficient of Determination (CD) ( $R^2$ ) for the model evaluation. To evaluate our regression models, the  $R^2$  shows the close relationships between the predicted and training values, with a good

$R^2 \rightarrow 1$ . Logcosh is not easily affected by spurious predictions. Whilst, we consider the accuracy (Acc), recall ( $R$ ), precision ( $P$ ),  $F$ -measure ( $f_1$ ), and confusion matrix for the classification [6, 7, 40]. We can also compare a plot of the reconstructed signal with the ground truth signal plot of each appliance through superimposition of these plots to physically see the relationship of these two signals:

- (1) Begin: obtain preprocessed, formatted, and transformed training input aggregate data of series length  $TT_a$  secs according to Figure 7
- (2) Obtain training target data of series length  $TT_t$  secs with redundancies removed
- (3) Train the network
- (4) Obtain preprocessed, formatted, and transformed validation/test input aggregate data of series length  $TV_a$  secs according to Figure 9
- (5) Specify disaggregation window,  $T_W = TT_t < TV_a < TT_a$
- (6) Slide trained network input through validation/test aggregate data by amount equal to disaggregation window
- (7) Repeat 6, until end of validation/test aggregate data series length
- (8) Use mean method sum up all results of disaggregation window movement to obtain disaggregated signal of series length  $TT_t, TT_t$  secs
- (9) Input 7 into trained classification network for appliance recognition
- (10) Repeat 1 to 9, until performance  $\rightarrow$  100%

PSEUDOCODE 1: The disaggregation task

$$\text{MSE} = \frac{1}{T} \sum_{i=1}^n |\bar{y} + \hat{y}|^2,$$

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{i=1}^n |\bar{y} + \hat{y}|^2},$$

$$\text{MAE} = \frac{1}{T} \sum_{i=1}^n |\bar{y} + \hat{y}|,$$

$$\text{Logcosh}(t) = \sum_{p \in P} \log(\cosh(p - t)),$$

$$R^2 = 1 - \frac{\sum (\text{Original} - \text{Predicted})^2}{(\text{Original} - \text{Originalmean})^2} \quad (2)$$

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

$$R = \frac{\text{TP}}{\text{TP} + \text{TN}},$$

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$f_1 = \frac{2 * P * R}{P + R},$$

where  $T$  is activation time (time-series) for each appliance,  $i = 1, \dots, n$  is number of appliances,  $\hat{y}$  is disaggregated power signal,  $\bar{y}$  is aggregate actual power at time  $t$ , Original is target signal and Predicted is the disaggregated signal, TP is true positives, FP is false positives, FN is false negatives, and TN is true negatives [6, 7].

The results and discussion may be presented separately, or in one combined section, and may optionally be divided into headed sections.

## 2.5. Verification of the Proposed Method Performance

### 2.5.1. Proposed Model Description.

Disaggregation is performed by using a sliding window on real test/validation

data. Training is performed by using a combination of real and synthesized data to improve on the recognition generalization of the NILM system. The disaggregation is performed on three parameters one at a time using the three proposed models separately. Each model goes through three training and disaggregation processes for the disaggregation part, excluding the classification part. Hence, we assign the three trained and disaggregating model outputs for model 1 in Figure 1 as *mdl1I\_rms*, *mdl1Watt*, and *mdl1PF*. Likewise for model 2 in Figure 2 and model 3 in Figure 3 we have *mdl2I\_rms*, *mdl2Watt*, *mdl2PF*, *mdl3I\_rms*, *mdl3Watt*, and *mdl3PF*, respectively. In summary, the number of disaggregating trained model outputs are nine (three per model), and the total number of disaggregating signals is nine. Of the three models, under consideration, we note the one with the higher or better disaggregation (regression performance plot) performance and exclude the results of the other models for further processing. This effectively leaves us with only three better disaggregated signals at any one time represented by *mdlbI\_rms*, *mdlbWatt*, and *mdlbPF*, where *mdlb* is model better output.

The classification model is trained based on tuning the MLP hyperparameters to provide the best performance on the ground truth signal parameters of I\_rms, Watt, and PF for four input LED similar signature appliances. The total number of parameters input into the classification network is twelve during the training stage. However, in the recognition stage, the total number of signal parameters input into the trained MLP is three, obtained from the best disaggregating model (that is, the *mdlb* model output). Due to the limited data for training the MLP deep network, we implement transfer-based learning where we train the classification network on a larger training dataset of the four LEDs than the one we have acquired that is directly related to the experiment.

### 2.5.2. Pseudocode for Proposed Method.

The proposed method evaluates the performance of the disaggregation algorithm on three models and carries out the classification only on one model. Although we have the same disaggregation task, we have in actual fact three disaggregation



algorithms due to the different model structures. Hence, we show the pseudocodes of the training of the three disaggregation algorithms one for each model as Pseudocodes 2–4. Pseudocode 1, which shows the actual sliding window disaggregation, is a common operation in the three different disaggregation algorithms. We then add Pseudocode 5 which shows how the classification is performed.

*2.5.3. Keras Model Architectures.* The architectures for the models we used in the disaggregation and classification are given as follows.

(1) *Disaggregation.* For Model 1 (MPS-CNN), the architecture we used is detailed as follows:

- (i) Input of length equal to  $T$  of target series.
- (ii) Three parallel double layer 1D convolutional networks filter sizes 64 and 128, 64 and 28, and 64 and 28 but having kernel size = 1, 3, and 7 each and activation = relu. Each network has a single MaxPooling1D(pool\_size = (2)) layer
- (iii) A merge layer.
- (iv) Three hidden dense layers with 50, 100, and 200 neurons, and activation = relu.
- (v) Output dense layer of length equal to  $T$  of target series.

For Model 2 (RNN), the architecture we used is detailed as follows:

- (i) Input of length equal to  $T$  of target series.
- (ii) An LSTM layer with 500 memory cells and two parallel dense layer networks, one with 1024 neurons and the other with three layers have
  - (i) LSTM(500)
  - (ii) Dense(1024, activation = "relu") first parallel dense network
  - (iii) Dense(500, activation = "relu") second parallel dense network in series with two dense layers comprising a Dense(1024, activation = "relu"), and a Dense(500, activation = "relu") layer
- (iii) A merge layer.
- (iv) An output dense layer of length equal to  $T$  of target series.

For Model 3 (CNN-RNN(LSTM)), the architecture we used is detailed as follows:

- (i) A TimeDistributed 1D convolutional network with 128 of filter sizes 1, followed by another 1D convolutional layer with 256 filters and filter size 1, activation = relu, and a single TimeDistributed(MaxPooling1D(pool\_size = 2)) layer
- (ii) A flatten layer
- (iii) Three hidden LSTM hidden layers with memory cells of lengths 1024, 4096, and 1024, respectively
- (iv) A hidden dense layers with 512 neurons, and activation = relu

- (v) Output dense layer of length equal to  $T$  of target series.

We experimented with learning rates of the Adam optimizer from 0.0000001 to 0.1 and found a good compromise for a value of 0.01. We used the logcosh to evaluate all regression-based experiments and also included and evaluated other regression metrics as given in the results.

(2) *Classification.* We have developed the classification algorithm using transfer learning and have adopted the weights from the large dataset given in Figure 11 to our constrained dataset. The MLP transfer learning model used is shown below. The CNN is more appropriate when the classification input dimension is very large. However, in our case, for training, we format the data as a matrix of three parameter values (multivariate time series of thirty columns (points) per parameter for current ( $I_{rms}$ ), power (Watt), and power factor (PF).

The MLP transfer learning-based classification architecture is

- (i) Input into Dense layer with 8 units, activation = "relu," and input\_dimension = 3
- (ii) A hidden Dense layer with 10 units and activation = "relu"
- (iii) A hidden Dense layer with 16 units and activation = "relu"
- (iv) An output Dense layer (Dense(3, activation = "softmax"))

The model used the Adam optimizers with a validation split = 0.3, one hot encoded labels, and only 50 epochs to achieve high performance. In the architecture shown, we use only 3 classes instead of 4, and the reason is explained in detail in Section 3. Although the classification model above achieved good performance, we are able to reach high validation accuracy faster by changing the input Dense layer to 500 units.

*2.5.4. Training Framework and Procedure.* The classification training framework is based on the Rectified Linear Unit (ReLU) activation function, the softmax function, selecting maximum number of epochs of 50, the Adam optimizer, and a validation split of 0.3. We initially provisionally include the training dropout regularization in the classification model. The ReLU shown in Figure 12 is an operation meant to introduce nonlinearity in the network, and it replaces all negative values with zero. Nonlinearity network characteristics are required to solve complex nonlinear situations. All the disaggregation networks are also based on this ReLU [24] activation function.

Furthermore, CNN networks inherently perform linear operations, and as such to consider nonlinearity, we incorporate the ReLU activation.

The basic training procedure of the MLP is defined by

$$w_{i+1} = w_i + \Delta w = w_i + \eta y_i (x_i - w_i y_i), \quad (3)$$

where  $\eta$  is the learning rate,  $x_i$  is an  $m$ -dimensional input vector (input neuron), and  $y_i = w_i^T x_i$ ,  $y_i = x_i^T w_i$  (output

- (i) Acquire the aggregate and ground truth signals for the selected parameter ( $p$ )
- (ii) Define: no. of parallel CNN networks ( $pn$ ), hidden layers in each  $pn$ , filters in each  $pn$ , dense layers ( $ds$ ), epochs, and early stopping condition
- (iii) Initialize all weights  
While parameter ( $p$ ) is True: **do**
  - (1) Epochs = 0 (train)
    - (i) Iterate through each  $pn$
    - (ii) Concatenate all  $pn$
    - (iii) Iterate through each  $ds$
    - (iv)  $mdl1$  converged  $\rightarrow$  stop
  - (2) Disaggregation
    - (i) Pseudocode 1
  - (3)  $mdl1$  not converged
    - (i) epochs = epochs + 1
    - (ii) Return [(1) (i)]
- Otherwise: **do**
  - (4)  $p = p + 1$  (next parameter): **do**  
(1) to (3).
  - (5) Train and disaggregate all parameters:
    - (i)  $p \leq 3$ : return [(4)]
    - (ii)  $p > 3$ : Exit

PSEUDOCODE 2: Training and disaggregation process of the MPS-CNN ( $mdl1$ )

- (i) Acquire the aggregate and ground truth signals for the selected parameter ( $p$ )
- (ii) Define: no. of LSTM memory cells, parallel MLP networks (MLPn) connected in series with LSTM layer, hidden layers in dense parallel layers ( $dsp$ ), dense layers ( $ds$ ) connected to series combination of LSTM + MLPn, epochs, and early stopping condition
- (iii) Initialize all weights  
While parameter ( $p$ ) is True: **do**
  - (1) Epochs = 0 (train)
    - (i) Iterate through LSTM
    - (ii) Iterate through LSTM + each MLPn arm
    - (iii) Concatenate all MLPn layers
    - (iv) Iterate through each  $ds$
    - (v)  $mdl2$  converged  $\rightarrow$  stop
  - (2) Disaggregation
    - (i) Pseudocode 1
  - (3)  $mdl2$  not converged
    - (i) epochs = epochs + 1
    - (ii) Return [(1) (i)]
- Otherwise: **do**
  - (4)  $p = p + 1$  (next parameter): **do**  
(2) to (3).
  - (5) Train and disaggregate all parameters:
    - (i)  $p \leq 3$ : return [(4)]
    - (ii)  $p > 3$ : Exit

PSEUDOCODE 3: Training and disaggregation process of the RNN (LSTM) ( $mdl2$ )

- (i) Acquire the aggregate and ground truth signals for the selected parameter ( $p$ )
- (ii) Define: no. of CNN networks including hidden (CNNn), filters, LSTM networks including hidden and memory cells (LSTMn), dense layers (ds), epochs, and early stopping condition
- (iii) Initialize all weights
  - While parameter ( $p$ ) is True: **do**
    - (1) Epochs = 0 (train)
      - (i) Iterate through each CNNn
      - (ii) Iterate through LSTMn
      - (iii) Iterate through each ds
      - (iv)  $mdl3$  converged  $\rightarrow$  stop
    - (2) Disaggregation
      - (i) Pseudocode 1
    - (3)  $mdl3$  not converged
      - (i) epochs = epochs + 1
      - (ii) Return [(1) (i)]
  - Otherwise: **do**
    - (4)  $p = p + 1$  (next parameter): **do**
      - (3) to (3).
    - (5) Train and disaggregate all parameters:
      - (i)  $p \leq 3$ : return [(4)]
      - (ii)  $p > 3$ : Exit

PSEUDOCODE 4: Training and disaggregation process of the CNN-RNN (LSTM ( $mdl3$ ))

- (i) Acquire four LED ground truth extended time series signals for transfer learning (TL)
- (ii) Acquire three disaggregated signals of  $mdlI\_rms$ ,  $mdlWatt$  and  $mdlPF$  from the best performing disaggregation model ( $mdl$ ), no. of neurons of TL model per dense layer, and early stopping condition
- (iii) Initialize all weights
  - (1) Epochs = 0 (train TL network)
    - (i) Iterate through each MLPn
    - (ii) TL model converged  $\rightarrow$  stop
  - (2) TL model not converged
    - (i) epochs = epochs + 1
    - (ii) Return [(1) (i)]
  - (3) Classification of disaggregated signals: **do**
    - (i) Input into TL
    - (ii) Recognize LED  $\rightarrow$  stop

PSEUDOCODE 5: Transfer learning-based deep MLP classification

neuron) is the output. The new and old synapse weights are  $w_{i+1}$  and  $w_i$ , respectively, and the weight change is given as  $\Delta w$  [41]. The backpropagation (Error-correction (ECL) (supervised learning is typically used to update the learning weights  $\Delta w$ )). The error change can be used to increase or decrease the magnitude of the weight update component given in equation (4). A change in the weight results in a change in the error. Minimum error point is achieved through gradient descent. However, gradient descent may converge to local instead of global minima. Hence, there is need to mitigate this shortcoming by continually randomly selecting the initial weights during the training process [42]:

$$\Delta w_{kj} = -\eta \frac{\partial E_k}{\partial w_{kj}}, \quad (4)$$

where  $E_k$  with respect to node  $k$  is the resultant sum of squares of errors (cost or loss function) between the target output ( $y_t$ ) and the network output ( $y_o$ ). For all the weights in the network, an update of these weights is achievable through the backward propagation (BP) of this error through the said network [42]. The backpropagation algorithm is more efficient than the normal feedforward algorithm. This is so because there are more passes to achieve significant weight change in a normal feedforward network

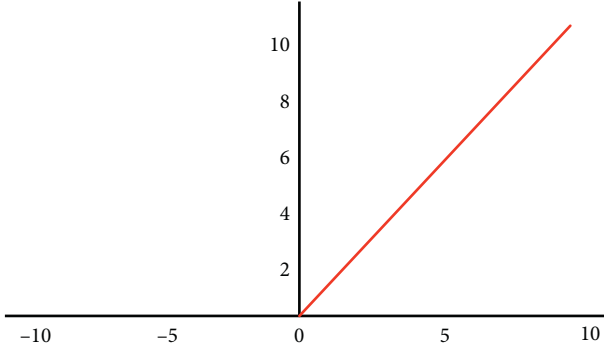


FIGURE 12: ReLU nonlinear activation function.

than there are in backpropagation. The algorithms of the standard and improved backpropagation methods referenced to Figure 13 are given below.

Let the same error backpropagated through the network be  $E = E_k$  and the activation function be the sigmoidal ( $\sigma(\cdot)$ ) one as given in

$$z_i = \sigma(q_i) = \frac{1}{1 + e^{-q_i}}. \quad (5)$$

Then, the standard backpropagation error derivative between nodes  $i$  and  $j$  is

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial q_j} \cdot \frac{\partial q_j}{\partial w_{ji}} = \delta_j \cdot z_i, \quad (6)$$

where

$$\delta_j = \frac{\partial E}{\partial q_j} = \sum_k \frac{\partial E}{\partial q_k} \cdot \frac{\partial q_k}{\partial q_j} = \sum_k \delta_k \cdot \frac{\partial q_k}{\partial q_j}, \quad (7)$$

$$\frac{\partial q_k}{\partial q_j} = \frac{\partial q_k}{\partial z_j} \cdot \frac{\partial z_j}{\partial q_j} = w_{kj} \cdot \sigma'(q_j).$$

Node 2 delta is given as

$$\delta_j = \sigma'(q_j) \cdot \sum_k \delta_k w_{kj}. \quad (8)$$

The error derivative between nodes  $j$  and  $k$  is

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial q_k} \cdot \frac{\partial q_k}{\partial w_{kj}} = \delta_k \cdot z_j. \quad (9)$$

Equation (9) can be written as

$$\begin{aligned} \frac{\partial E}{\partial w_{kj}} &= \frac{\partial E}{\partial (y_t - y_o)} \cdot \frac{\partial (y_t - y_o)}{\partial w_{kj}}, \\ \frac{\partial E}{\partial w_{kj}} &= \sum (y_t - y_o) \cdot \frac{\partial (\sigma(q_k))}{\partial w_{kj}}, \\ \frac{\partial E}{\partial w_{kj}} &= - \sum (y_t - y_o) \cdot y_o (1 - y_o) \cdot z_j. \end{aligned} \quad (10)$$

Comparing equations (9) and (10), the magnitude of  $\delta_k$  is given by  $\sum (y_t - y_o) \cdot y_o (1 - y_o)$ :

- (i) Node 1 layer error is given by  $\sum_j \delta_j w_{ji}$  (where this node is another hidden layer node in a two hidden layer network)
- (ii) Node 2 layer error is given by  $\sum_k \delta_k w_{kj}$
- (iii) Node 3 layer error is given by  $(y_{lt} - y_{lo})$

The standard backpropagation algorithm is given below:

- (1) Obtain initial values of weights and offsets.
- (2) Establish the input vector  $x$  and target output. Also, determine the number of hidden and output units.
- (3) Find the deltas ( $\delta$ ) for all the output nodes.
- (4) Backpropagate the deltas using  $\delta_j = \sigma'(q_j) \cdot \sum_k \delta_k w_{kj}$ .
- (5) Evaluate the derivatives  $(\partial E / \partial w_{kj}) = \delta_k \cdot z_j$  for all synapses.
- (6) Update the weights according to  $w_{kj}(\text{new}) \leftarrow w_{kj}(\text{old}) - \eta (\partial E / \partial w_{kj})$ .
- (7) Repeat (2).

In the recognition training, we experimented with various optimizers that included the Adam, rmsprop, and sgd. The sgd was set to optimizers SGD(lr = 0.000001 to 0.1), decay =  $1e - 6$ , momentum = 0.9, netrov = True). The Adam and rmsprop were set to a learning rate that varied between 0.000001 and 0.1. Both the Adam and sgd optimizers performed well with a learning rate of 0.01 and 0.001 for the disaggregation and classification algorithms, respectively. The categorical\_crossentropy cost function was used in the classification model training. We also experimented with various activation functions that included the tanh (sigmoid) (mainly used in artificial neural networks (ANNs) since its characteristics can accommodate both linear and nonlinear situations), relu, and the leaky\_relu (an improvement over the normal relu). We settled on the relu which achieved acceptable performance. In the output stages of the disaggregation and classification models, we implanted the linear and softmax activation functions, respectively. We also experimented with the  $l_1$  and dropout regularizers, but found out that due to the relatively simpler designed models the regularization did not affect the performance of the algorithms. Hence, there was no need to implement regularization in all the models. The choice of the number of hidden layers, neurons (units), number and size of CNN filters, and memory units in the LSTM was achieved through trial and error.

With respect to the CNN and LSTM disaggregation networks, we invoke the training procedure after specifying the Keras model architectures. The input aggregate power series of length ( $T$ ) is trained against another power series represented by the target series also on length ( $T$ ),  $X = \{x_1, x_2, x_3, \dots, x_T\}$ . The objective of the training procedure is to minimize the regression cost functions



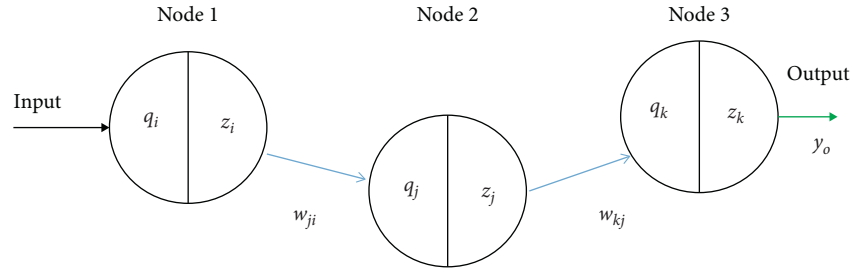


FIGURE 13: Cross-section of multilayer neural network.

represented by logcosh, root-mean-square-error (RMSE), mean\_squared\_error (MSE), and mean\_absolute\_error (MAE). However, another regression function, the Coefficient of Determination (CD) ( $R^2$ ) for the model evaluation is required to be high. We also evaluate the training computation times of the proposed models.

### 3. Results and Discussion

**3.1. Regression Training and Disaggregation.** We compare our proposed models to each other and only use the output from the most accurate model as input into the classifier. Although disaggregation was carried out on all the LEDs, we limit our analysis to one LED lamp; however, we show the classification rates of all the LEDs. If we can achieve good performance for one LED, then we can also achieve good performance for other LED lamps since the features and their relative magnitudes are almost similar. Figures 14–16 show the relative performance of the regression models for LED1-2 I<sub>rms</sub> signal using the data in Figure 7. The ground truth signal for this LED1-2 lamp is shown in Figure 17.

We did achieve comparable results for the power and PF signals. We experimented with different LSTM memory lengths and we found lengths above 500 provided good results. Furthermore, when we tried paralleling the LSTM networks by using the API Keras structure, we did not get an improvement in the LSTM model results. However, the network based on a single LSTM network provided acceptable results. The model based on the CNN-RNN also provided good regression results. It is, however, the MPS-CNN structure that achieved top disaggregation in this paper. The MPS-CNN structure allows us to capture a wide range of features and detail that include the on/off edge detection.

**3.2. Classification.** For the LED1-2 recognition, we apply three disaggregated input parameters into a deep MLP classification network. We first train the network on a larger dataset depicted in Figure 11 than the one obtained from the disaggregated signal in the transfer learning-based classification scheme. We fine tune the network on the larger dataset and when we have obtained satisfactory results, as shown by the training Figures 18 and 19, we apply the model on our disaggregated dataset. In Figure 18, we show that the model accuracy achieves high value early in the training of the TL model. From the training and validation loss characteristics in Figure 19, we show that our MLP TL model is

very stable and the characteristics converge well. We tried six different classification MLP models using the larger dataset and all models misclassified LED1-1 and LED1-2 that have exactly the same specifications and identical parameter values. Also, where LED1-1 and LED1-2 appear in the disaggregation algorithm, we were not able to separate the two from each other.

Hence, we eliminate one of the LEDs, LED1-1, in our analysis as there is no added useful recognition information. So, in the whole recognition process, LED1-1 is taken as LED1-2. This explains why the classification model under Keras Models' Architectures is based on three classes. In future, we can detect LED1-1 and LED1-2 by considering the actual cable lengths that are different from each other from the main supply in a typical building installation. In the laboratory measurement setup, we did not factor in this issue and we just measured the appliance parameters using the same extension cables from the mains distribution point. We can also use deeper learning which is not possible in our experimental CPU platform. In addition, recognition can be based on parameter phase change and some advanced event detection schemes. Due to the initial experimental results, we modify our recognition strategy to only consider LED1-2, LED3-1, and LED2-1. In this case, for LED1-2, the class is 0, for LED3-1 the class is 1, and for LED2-1, the class is 3.

Table 1 and Figure 20 show the classification report and the classification matrix, respectively, of the model trained using a larger dataset in Figure 11. We see that all the three achieve one hundred percent classification. In Figure 20, the history parameters are batch size-1, epochs-50, steps-None, samples-892, verbose-2, do\_validation-True, and metrics [loss, acc, val\_loss, val\_acc]. The classification model in Figure 20 achieved the following: Evaluation: loss-0.010676, accuracy-1.0, Test score-0.0173, and Test accuracy-1.0. We transfer this model without modification to the smaller disaggregated dataset in transfer learning, where we maintain the same class labels, as shown in the confusion matrix in Figure 21. Table 2 shows the classification report for Figure 21.

Table 3 gives the regression-based metrics during the training of the disaggregation algorithms.

It is necessary to evaluate the relative computation times of the models, especially those for the disaggregation algorithms. A fast computation time allows for fast turnaround of program development and indirectly implies less stress on the computation processor. The code for evaluating the computation time of each model is given as

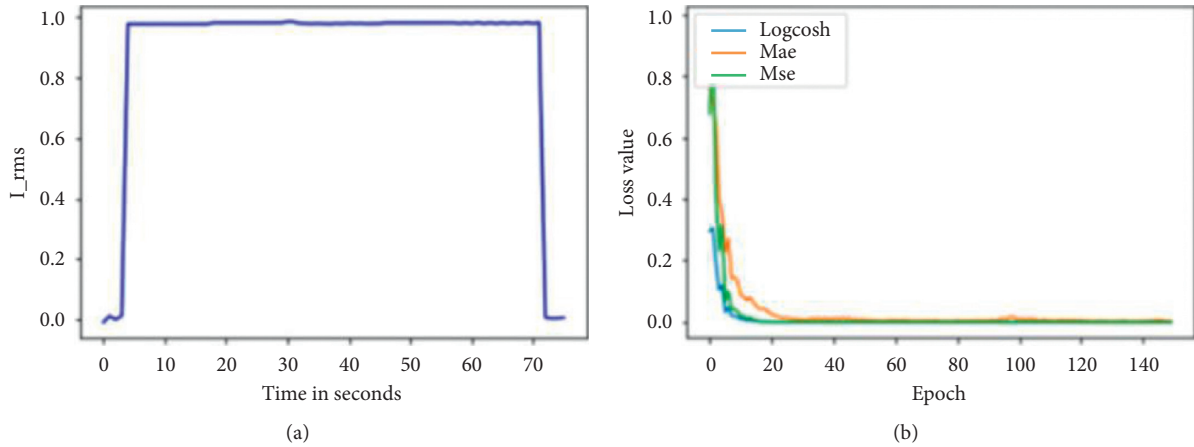


FIGURE 14: LSTM disaggregation model plots. (a) Reconstructed signal and (b) loss.

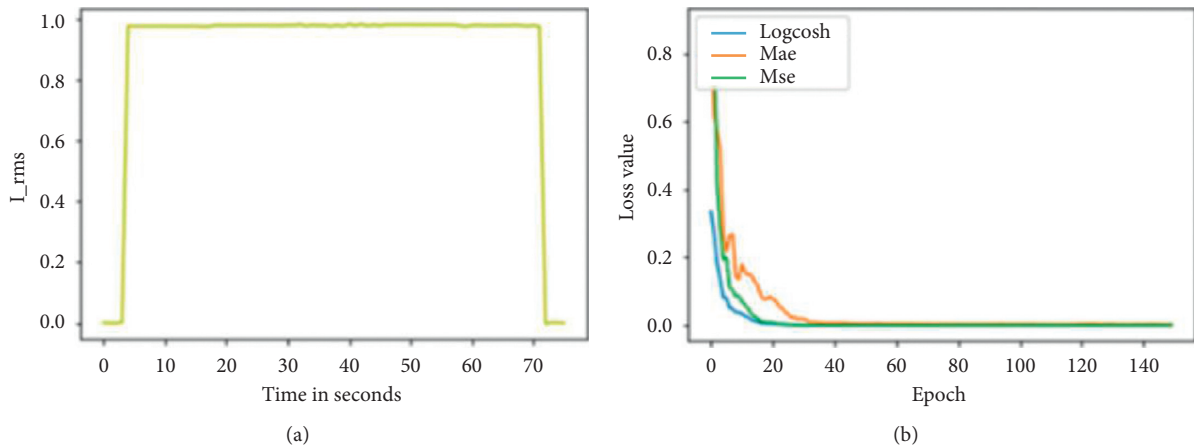


FIGURE 15: CNN-LSTM disaggregation model plots. (a) Reconstructed signal and (b) loss.

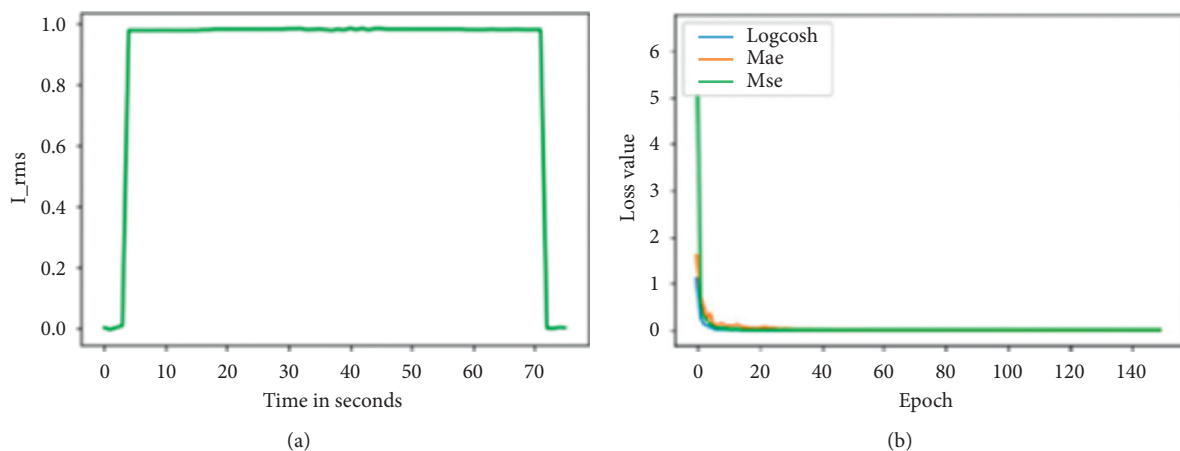


FIGURE 16: MPS-CNN disaggregation model. (a) Reconstructed signal and (b) loss.

```

from timeit import default_timer as timer
from datetime import timedelta
start = timer()
    
```

```

history = model.fit(X, Y, epochs = 150, verbose=, vali-
dation_split = 0.3) #Any model training.
end = timer()
    
```

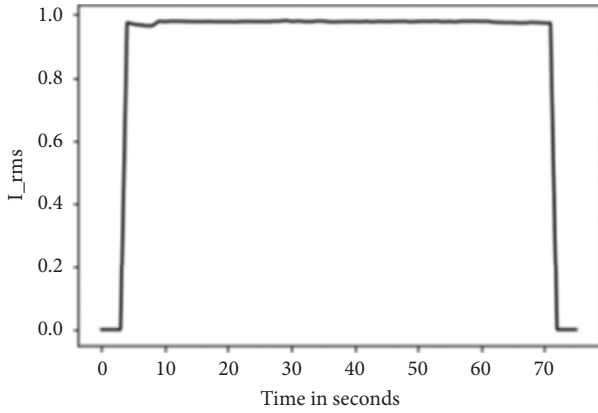


FIGURE 17: Ground truth signal for LED1-2.

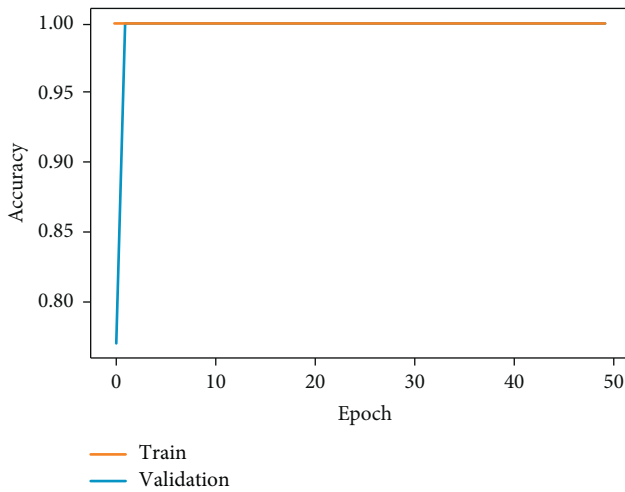


FIGURE 18: Model accuracy for the classification TL model training.

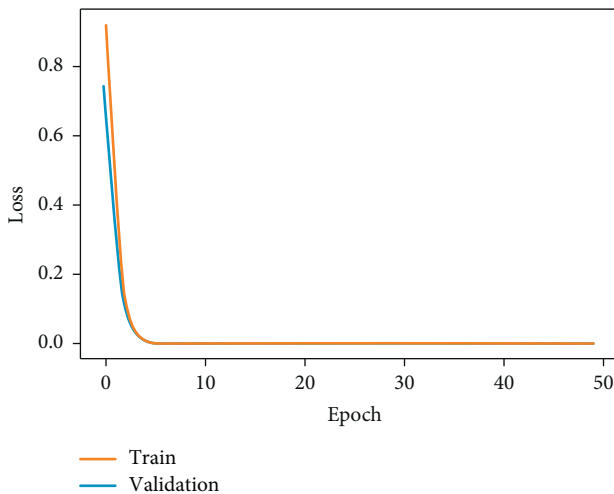


FIGURE 19: Model loss for the classification TL model training.

```
print(timedelta(seconds = end-start))
```

Table 4 shows the computation times of the models in relation to the total trainable parameters. The computation times of the models increase with an increase in the number

TABLE 1: Transfer learning model classification report.

Class	Precision	Recall	<i>F</i> -score	Support
0	1.00	1.00	1.00	175
1	1.00	1.00	1.00	175
2	1.00	1.00	1.00	175
Weighted avg	1.00	1.00	1.00	525

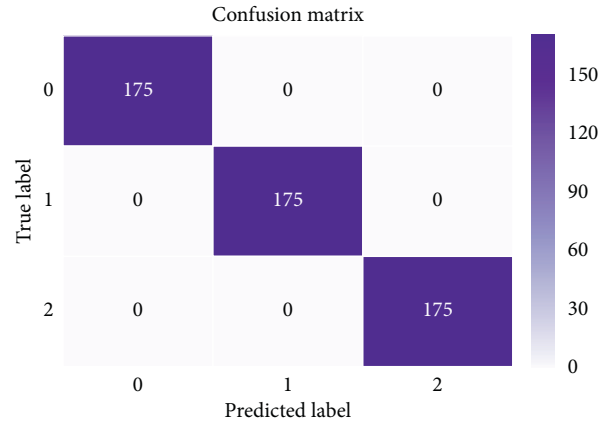


FIGURE 20: Classification matrix for the model trained on disaggregated dataset. This is the transfer learning stage. All base lamp signals are accurately classified.

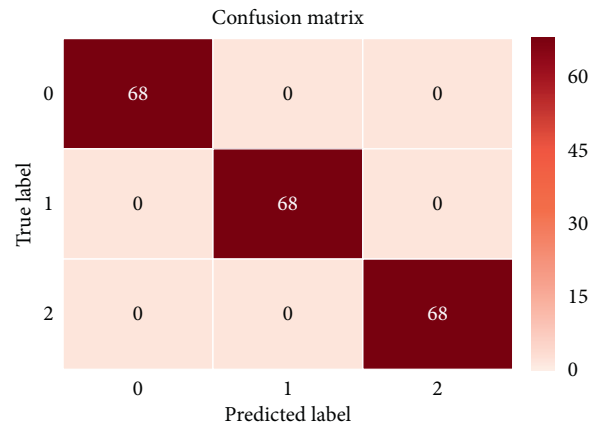


FIGURE 21: Classification matrix for the disaggregated lamp signals. All lamps are accurately classified.

TABLE 2: Disaggregated signal recognition classification report.

Class	Precision	Recall	<i>F</i> -score	Support
0	1.00	1.00	1.00	68
1	1.00	1.00	1.00	68
2	1.00	1.00	1.00	68
Weighted avg	1.00	1.00	1.00	204

of trainable parameters. The MLP-TL classification process is the fastest due to its simpler network structure and the fewer number of output labels required as compared to the

TABLE 3: Disaggregation regression-based model metrics.

Metric	MPS-CNN	LSTM	CNN-LSTM
MAE	0.0036	0.0025	0.0022
MSE	$2.275e-05$	$1.254e-05$	$9.420e-06$
RMSE	0.00477	0.00354	0.00307
CD	1.00	1.00	1.00

TABLE 4: Training computation times of the NILM models.

Model name	Number of trainable parameters	Number of epochs	Computation time (hh:mm:ss)
MLP TL	3,503	50	0:07:38.034314
MPS-CNN	238,543	150	0:00:57.302319
LSTM	2,794,573	150	01:6:46.786972
CNN-LSTM	8,429,057	150	0:38:21.626814

longer output power series signal samples required the disaggregation algorithms. The MPS-CNN model is faster than the LSTM-based models that have a larger number of total trainable parameters. LSTM RNN networks are adapted to capturing of information from power series data or sequences. However, LSTM RNNs suffer degraded performance [43] when the information is available in very long power series such as the ones we have in the NILM recognition. As such, this slows down their training computation times. Large LSTM RNN blocks also have a large number of gating functions which increases the number of trainable parameters, hence computation time.

The results show the ability of our proposed models to achieve high disaggregation and classification accuracy of the LED lamps in our experiment. It is important to take cognizance of the fact that state-of-the-art [20, 32] systems tested on a variety of widely deferring appliance specifications using more or less the same types of models might outperform our recognition in accurate classification of all test samples. In our case, we had to eliminate one highly misclassified LED1-1 in the final analysis. However, this paper is biased towards developing algorithms to recognize relatively low power appliances having the same specifications. Our argument has here been that if we can accurately classify and disaggregate low power same specification appliances, then naturally it should be a matter of fact to achieve the same for the widely varying power levels different specification appliances.

## 4. Conclusions

This paper evaluated three NILM disaggregation and one classification algorithm for equal power appliances with almost similar signatures, in the form of three 5 W and one 5.5 W mains LED lamps. We used the following labelled LED lamps in our experiments: LED1-1 (Philips 5 W (60 W)), LED1-2 (Philips 5 W (60 W)) and LED2-1 (Philips 5 W (60 W)), and LED3-1 (Radiant 5.5 W). We show that

same specification appliances can indeed be recognized from each other. However, we need a cautious and elaborate approach in developing a holistic NILM recognition for appliances that have identical specifications. In our study, we had to eliminate in the final analysis from our experiments LED1-1 as it grossly misclassified as LED1-2 since its characteristics were almost identical to those of LED1-2. The point of divergence from the normal approaches was the disaggregation and classification based on three appliance parameters to substantially increase the accuracy. This in itself did not cure the problem. As no two appliances are exactly the same from manufacture, developing deeper learning algorithms is one possible way of solving this problem; however, the CPU platform we operated from has limitations both in speed and processing power. The results also show that equal power specification appliances should have parameters measured whilst in the actual installation and not in laboratory to take advantage of such issues as contributions due to wiring where we can measure phase change, time lag, wiring resistance etc. from the sampling point. However, our NILM recognition strategy is promising as we did obtain accurate recognition for some of the lamps.

## Data Availability

All the data and codes used in this paper are available from the authors at the University of Johannesburg.

## Conflicts of Interest

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

## Acknowledgments

This research was supported partially by South African National Research Foundation Grants (nos. 112108 and 112142) and South African National Research Foundation Incentive Grants (nos. 95687 and 114911), Eskom Tertiary Education Support Programme Grants, Research grant from URC of University of Johannesburg.

## References

- [1] D. Ding, R. A. Cooper, P. F. Pasquina, and L. Fici-Pasquina, "Sensor technology for smart homes," *Maturitas*, vol. 69, no. 2, pp. 131–136, 2011.
- [2] V. Fabi, G. Spigliantini, and S. P. Corgnati, "Insights on smart home concept and occupants' interaction with building controls," *Energy Procedia*, vol. 111, pp. 759–769, 2017.
- [3] H. Jahangir, H. Tayarani, A. Ahmadian et al., "Charging demand of plug-in electric vehicles: forecasting travel behavior based on a novel rough artificial neural network approach," *Journal of Cleaner Production*, vol. 229, pp. 1029–1044, 2019.
- [4] H. Jahangir, S. Sadeghi Gougheri, B. Vatandoust, M. Aliakbar Golkar, A. Ahmadian, and A. Hajizadeh, "Plug-in electric vehicle behavior modeling in energy market: a novel deep learning-based approach with clustering technique," *IEEE Transactions on Smart Grid*, 2020.

- [5] N. F. Esa, M. P. Abdullah, and M. Y. Hassan, "A review disaggregation method in non-intrusive appliance load monitoring," *Renewable and Sustainable Energy Reviews*, vol. 66, pp. 163–173, 2016.
- [6] C. Nalmpantis and D. Vrakas, "Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparison," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 217–243, 2019.
- [7] R. Bonfigli, S. Squartini, M. Fagiani, and F. Piazza, "Unsupervised algorithms for non-intrusive load monitoring: an up-to-date overview," in *Proceedings of the IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, pp. 1175–1180, Rome, Italy, June 2015.
- [8] Q. Liu, K. M. Kamoto, X. Liu, M. Sun, and N. Linge, "Low-complexity non-intrusive load monitoring using unsupervised learning and generalized appliance models," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 28–37, 2019.
- [9] A. U. Rehman, L. T. Lie, B. Valles, and S. R. Tito, "low complexity event detection algorithm for nonintrusive load monitoring systems," in *Proceedings of the 2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, pp. 746–751, Singapore, May 2018.
- [10] L. De Baets, J. Ruysinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using VI trajectories and convolutional neural networks," *Energy and Buildings*, vol. 58, pp. 32–36, 2018.
- [11] P. Bilski and W. Winiecki, "Generalized algorithm for the non-intrusive identification of electrical appliances in the household," in *Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 730–735, Bucharest, Romania, September 2017.
- [12] G. Hoogsteen, J. O. Krist, V. Bakker, and G. J. M. Smit, "Non-intrusive appliance recognition," in *Proceedings of the 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pp. 1–7, Berlin, Germany, October 2012.
- [13] K. N. Trung, E. Dekneuvel, O. Zammil et al., "Event detection and disaggregation algorithms for NIALM system," in *Proceedings of the International Symposium on Industrial Electronics (ISIE)*, pp. 28–31, Taipei, Taiwan, May 2013.
- [14] M. N. Meziane, P. Ravier, G. Lamarque, J.-C. L. Bunetelt, and Y. Raingeaud, "High accuracy event detection for non-intrusive load monitoring," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2452–2456, New Orleans, LA, USA, March 2017.
- [15] T. Bernard and M. Marx, "Unsupervised learning algorithm using multiple electrical low and high frequency features for the task of load disaggregation," in *3rd International Workshop on Non-Intrusive Load Monitoring, Electric Power Research Institute—EPRI-, NDE Center NILM 2016*, pp. 1–5, Vancouver, Canada, May 2016.
- [16] A. S. Bouhouras, P. A. Gkaidatzis, E. Panagiotou, N. Poulakis, and G. C. Christoforidis, "A NILM algorithm with enhanced disaggregation scheme under harmonic current vectors," *Energy and Buildings*, vol. 183, pp. 392–407, 2019.
- [17] C. Dinesh, S. Makonin, and I. V. Bajić, "Residential power forecasting using load identification and graph spectral 25 clustering," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 6, no. 11, pp. 1900–1904, 2019.
- [18] O. Hamid, M. Barbarosou, P. Papageorgas, K. Prekas, and C. Salame, "Automatic recognition of electric loads analyzing the characteristic parameters of the consumed electric power through a non-intrusive monitoring methodology," *Energy Procedia*, vol. 119, pp. 724–751, 2017.
- [19] M. Figueiredo, A. de Almeida, and B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," *Neurocomputing*, vol. 96, pp. 66–73, 2012.
- [20] J. Kelly and W. Knottenbelt, "Neural NILM: deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd 31 ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 55–64, Seoul, South Korea, November 2015.
- [21] A. Zhao, L. Qi, J. Li, J. Dong, and H. Yu, "A hybrid spatio-temporal model for detection and severity rating of Parkinson's disease from gait data," *Neurocomputing*, vol. 315, pp. 1–8, 2018.
- [22] Y. Zhang, G. Yang, and S. Ma, "Non-intrusive load monitoring based on convolutional neural network with differential input," *Procedia CIRP*, vol. 83, pp. 670–674, 2019.
- [23] F. C. C. Garcia, C. M. C. Creayla, and E. Q. B. Macabebe, "Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders," *Procedia Computer Science*, vol. 105, pp. 248–255, 2017.
- [24] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [25] <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>, June 2020.
- [26] P. Lacko, "From perceptrons to deep neural networks," in *Proceedings of the IEEE 15th International Symposium on Applied Machine Intelligence and Informatics*, pp. 169–172, Herl'any, Slovakia, January 2017.
- [27] W. He and Y. Chai, "An empirical study on energy disaggregation via deep learning," in *Proceedings of the 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*, vol. 133, pp. 338–342, Advances in Intelligent Systems Research, Beijing, China, November 2016.
- [28] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, Brisbane, Australia, April 2015.
- [29] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [30] D. de Paiva Penha and A. R. C. Castro, "Home appliance identification for NILM systems based on deep neural network," *International Journal of Artificial Intelligence and Applications (IJAI)*, vol. 9, no. 2, pp. 69–80, 2018.
- [31] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—part I: basic concept, structure, and methodology," *IEEE Transactions on Power Delivery*, vol. 25, no. 2, pp. 551–560, 2010.
- [32] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acoustics Speech and Signal Processing Magazine*, vol. 4, no. 2, pp. 4–22, 1987.
- [33] S. Sabour, N. Frost, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing*, pp. 3856–3866, 2017.
- [34] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning pooling for convolutional neural network," *Neurocomputing*, vol. 224, p. 96, 2017.
- [35] H. Patel, A. Thakkar, M. Pandya, and K. Makwana, "Neural network with deep learning architectures," *Journal of*



- Information and Optimization Sciences*, vol. 39, no. 1, pp. 31–38, 2017.
- [36] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, “Denoising autoencoders for non-intrusive load monitoring: improvements and comparative evaluation,” *Energy and Buildings*, vol. 158, pp. 1461–1474, 2018.
- [37] D. Cook, K. D. Feuz, and N. C. Krishnan, “Transfer learning for activity recognition: a survey,” *Knowledge and Information Systems*, vol. 36, no. 3, pp. 537–556, 2013.
- [38] S. Al-Stouhi and C. K. Reddy, “Transfer learning for class imbalance problems with inadequate data,” *Knowledge and Information Systems*, vol. 48, no. 1, pp. 201–228, 2016.
- [39] PA1000, *Power Analyser User Manual*, Tektronix, Inc., Beaverton, OR, USA, 2018.
- [40] [https://www.tensorflow.org/opi\\_doc/python/tf/keras/losses/logcosh](https://www.tensorflow.org/opi_doc/python/tf/keras/losses/logcosh), June 2020.
- [41] K. Samiee, A. Iosifidis, and M. Gabbouj, “On the comparison of random and hebbian weights for the training of single-hidden layer feedforward neural networks,” *Expert Systems with Applications*, vol. 83, pp. 177–186, 2017.
- [42] M. Bascema and W. J. Tastle, *Intelligent Data Mining in Law Enforcement Analysis: New Neural Networks Applied to Real Problems*, Springer, Berlin, Germany, 2013.
- [43] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM fully convolutional networks for time series classification,” *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

## Research Article

# Analysing and Data-Driven Modelling the Handling Performance of Rugby Balls under Wet Conditions

Jiaojiao Liu,<sup>1</sup> Yunan Liu,<sup>2</sup> Ya Wang,<sup>3</sup> and Chao Liu <sup>2</sup>

<sup>1</sup>School of Sports and Physical Education, Shandong Sport University, Rizhao, Shandong 276800, China

<sup>2</sup>School of Martial Arts, Shandong Sport University, Rizhao, Shandong 276800, China

<sup>3</sup>Zaozhuang Sport School, Zaozhuang, Shandong 277000, China

Correspondence should be addressed to Chao Liu; chaoliu088@gmail.com

Received 24 July 2020; Revised 13 August 2020; Accepted 5 September 2020; Published 14 September 2020

Academic Editor: Jun Chen

Copyright © 2020 Jiaojiao Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies the effect of moisture on the handling performance of rugby balls, where different moisture conditions will cause a change in skin friction between finger pads and rugby balls. In this study, a set of rugby ball passing tests was performed under wet conditions to test the effect of skin hydration on the performance of ball handling. A comprehensive analysis was then conducted to assess skin frictional properties at various moisture levels. It was observed that skin moisture is strongly associated with the handling performance of rugby balls. Based on the collected experimental data, intelligent data-driven models were finally developed, including a neural network and a fuzzy rule-based system. These models will likely enhance the anticipation of the handling performance of rugby players, which helps in designing specific training programmes and better preparing for rugby games to cope with wet conditions.

## 1. Introduction

As an essential element in most sports games, sports equipment appears to be used very regularly in a variety of forms depending on the sport involved. Examples of the equipment could be balls like basketballs, rugby balls, and footballs, flying discs for freestyle and disc golf, bats for baseball and cricket, rackets for badminton and tennis, etc. In general, this equipment has been designed to offer a high quality of service lives and sports performance with a continuous exploration of new forms of design and materials. For instance, rugby balls were often made by stitching a pig's bladder in an oval shape in early days, and nowadays rugby balls are only allowed to be made of leather or other suitable synthetic materials [1]. In addition, the specification of size, weight, and even the air pressure has been strictly defined according to the rules of the International Rugby Board, with the aim of enhancing the stability of hand gripping [2].

In those sports games involving hands contact with equipment, it is essential for players to demand effective

handling skills with the aim of achieving better performance. For example, in rugby, handling skills consist of various activities such as passing balls, catching balls, holding balls, or running with balls. Regardless of position, all players are required to perform those core skills effectively with lower level of errors. This is because ball handling errors are believed intimately associated with the turnover in games. Good ball handling skills could reduce the turnover rate and hence increase the chance of winning [3, 4]. Ball handling is a complex activity that employs hands/fingers interacting with ball surfaces under various conditions and can be easily influenced by different factors. For instance, most players find the balls become very slippery when their hands are sweating or they are playing on raining days, which bring difficulties for them to handle the balls with those wet conditions [5, 6].

In this paper, we study the handling/passing performance of rugby balls under wet conditions using data analytics and data-driven modelling approaches. In recent years, we have witnessed a dramatic increase in our ability to collect, process, and store data, which also happens in sports

engineering. Analysing and utilising these data, such as in data-driven modelling, provides humans with very useful knowledge and some practical tools. In the case that some complicated systems cannot be described by conventional mathematical models, people may apply data-driven modelling techniques to develop practical data-driven models. In the current study, we follow this strategy to study the performance of the rugby sport. First, three different experiments were designed and conducted, including a rugby ball passing test, a skin friction test, and a measurement of skin mechanical properties. Data-driven predictive models were then developed based on the experimental data.

## 2. Related Work

In the recent study of Tomlinson et al. [7], they investigated the passing accuracy of various rugby balls under both dry and moist conditions. The coefficient of the friction of the balls was also measured. The results showed that the passing scores of the balls with a lower friction coefficient are poorer than those balls with a higher friction coefficient. Lewis et al. [8, 9] pointed that the accuracy of a rugby ball pass is closely associated with the frictional behaviours between human fingers and ball texture. High friction forces would be beneficial for players to handle balls in games. In comparison of balls with various textures, they noticed that the balls with patterns of pimples could help ball gripping under dry conditions. This phenomenon might be attributed to the corresponding change in skin viscoelasticity, which leads the skin to deform around pimples and thus increases the contact area, hence the hysteresis mechanism [7].

The human skin is a complex material and is composed of three different layers. This structure gives the skin unique physical-mechanical properties that are very similar to rubbers [10, 11]. In past decades, the frictional behaviour of the skin has been studied widely [12–22]. This is because, as the outer covering of the human body, the skin plays an important role in preventing the risk of body damages from skin friction due to the skin contacting with complex environments/objects. It is generally believed that the frictional properties of the skin are owed to various physical mechanisms, for instance, adhesion mechanisms, interfacial shear mechanisms, and capillary action mechanisms. [23]. In addition, it is deemed that skin frictional properties are closely related to the contact conditions: hydration, load, speed, and material properties. For instance, Bowden and Tabor [24] stated that the skin frictional properties comply with a classical two-term model consisting of an adhesion mechanism and a deformation mechanism. In general, it is expected that the coefficient of the skin friction is mainly affiliated with the adhesion mechanism under dry conditions and can be described by a linear relationship [16, 17].

For the moisture conditions, the coefficient of the skin friction is found to vary with different moisture conditions [13, 15, 21, 22]. In previous studies, both a linear correlation and a “bell-curve” correlation between the coefficient of skin friction and the moisture level have been observed [17, 22, 25, 26]. For example, Yoshimune et al. [25] and Veijgen et al., [26] examined the relationship between the

skin moisture level and the coefficient of the skin friction and pointed that the coefficient of skin friction increases with the moisture level following a positive linear function. However, in recent studies, Adams et al. [17] and Tomlinson et al. [22] found a bell-curve relationship for the coefficient of the skin friction when hydrated fingers contacting with various materials, in which an initial increase in the coefficient of friction while adding water to dry skin. It was found that the coefficient of the skin friction decreases after the moisture level of the skin reaches a certain level.

Handling/passing under wet conditions is one of the key research topics in rugby union; however, there is very limited relevant work available. This research was performed with the aim of investigating how the handling/passing performance is influenced by different moisture levels using data analytics and data-driven modelling techniques. In this study, three different experiments were designed and conducted, including a rugby ball passing test, a skin friction test, and a measurement of skin mechanical properties. Predictive models were then developed based on the experimental data.

## 3. Experiments

*3.1. Rugby Ball Passing Test under Different Moisture Conditions.* In this test, eight male rugby players (age  $20 \pm 2$  years old) from Shandong Sport University took part, and they all have developed similar ball passing/handling skills. The details of all participants were recorded, including height, weight, wrist strength, throwing posture, and handling hand preference. Prior to the test, all participants were given a training session to familiarise the test procedure. In order to investigate the effect of the moisture level on the accuracy of ball handling, the participants were asked to perform the test under four different moisture conditions: a natural state, a medium hydration state, a high hydration state, and an addition of water. A moisture meter was used to measure the moisture level of the skin on the participants' thumb and middle fingers of their right hands (see Liu et al. [19] for more details).

As illustrated in Figure 1, the participants were instructed to run with a rugby ball from position E to zone AB (the distance between E and AB is 4 m) and throw the ball to a target net when arrived at zone AB. The target net was located at a place 7 m away from AB. Then, the participants needed to run toward position F to pick another ball and back to zone AB to complete another shot. Prior the tests, all participants were required to clean their hands and dry them with paper towels. Regarding the natural-state test, it was performed on cleaned and dried hands. For the middle hydration test, the participants were requested to carry out 10 min warm-up activities firstly and then perform the passing test immediately in order to minimise the loss of water from the skin. In the third test of a relative high hydration state, the participants were asked to perform the passing test along 20 min warm-up activities for a higher moisture level. Finally, the participants were asked to perform the test using wet balls. In order to ensure all surfaces of the balls are covered with water, the balls were soaked in



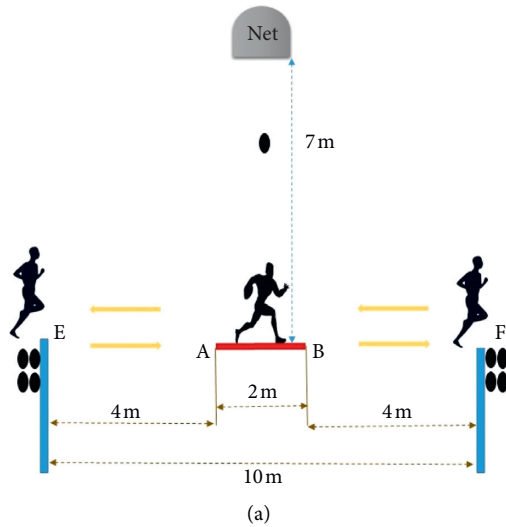


FIGURE 1: (a) Sketch of the passing ball test and (b) images of passing a ball to the net.

water for 1 min before the test. With regard to each moisture condition, the participants had to complete one successful shot. The time taken for completing all shots successfully by each participant was recorded for determining their final passing scores. The scoring scheme used in this study is given in Table 1 [27]. During the test, if any invalid run or invalid data were identified, the test would be repeated until getting the valid data. After the passing test, 32 experimental data were collected.

All obtained data from the rugby ball passing test were entered into Microsoft Excel and analysed by the SPSS software (version 26) using a nonparametric method (Friedman test) and a correlation analysis method. Prior to these inferential statistical analyses, all raw data need to be preprocessed. In this study, the moisture readings were processed for the average data and standard deviations using a descriptive analysis. Due to the big varieties of the moisture readings collected from participants, it is necessary to normalise the data for each player. The normalised moisture readings were set in the range of 0.0 to 1.0. The Friedman test was employed to analyse the difference in the moisture readings among four hydration conditions. In addition, the relationship between the normalised moisture readings and the target scores was also studied by correlation analyses. In all statistical analyses, the  $p$  value that is less than 0.05 was deemed to be statistically significant. The correlation coefficient  $R$  was also calculated to analyse the relationship between the measured data and the predicted results.

**3.2. Measurement of Skin Friction.** A miniature force platform device (Mode: HE6X6, Advanced Mechanical Technology, Inc.) was employed to measure skin friction. It is a low-load, high-accuracy measurement device (22 N in  $X$ - and  $Y$ -axes and 44 N in  $Z$ -axis with an accuracy of 1%) and consists of a force plate, an interface box, and a computer (see [18] for more details). The principle of this measuring

device relies on the strain gauge flexibility technique that allows the forces to be detected in  $X$ -,  $Y$ -, and  $Z$ -axes. During measurement of friction coefficient, the normal force implied to the force acting in the  $Z$ -axis and the corresponding force in  $X$ -/ $Y$ -axis is considered as a friction force; hence, the coefficient of skin friction can be obtained by the proportion of the friction force to the normal force. In the current study, the measurements of the coefficient of the skin friction were performed on the middle fingers of the participants. The participants were instructed to slide the examined fingers along a 5 mm wide acetal strip (roughness of  $0.5 \mu\text{m}$ ) with a normal force of  $1.5 \pm 0.2 \text{ N}$ . Prior the tests, all participants were invited and requested to clean and dry their hands. With regard to a variety of moisture levels in the skin, the participants were asked to soak their right hands in tap water for 20, 40, 80, 120, 160, 200, 240, 280, 320, 360, and 400 s. The moisture level of the skin was recorded by a moisture meter. All friction measurements were repeated three times to obtain average results.

**3.3. Measurement of Skin Mechanical Properties.** This series of tests aimed to explore the mechanical properties of finger-pad skin in response to skin hydration. To achieve this, the middle fingers on the right hands of participants were examined using a noninvasive Cutometer MPA 580 (see [19] for more details). The participants were asked to prepare the examined fingers using the same method of the skin friction measurement. The test firstly was performed with a dry finger and then a soaked finger (400 s of soaking time). With the purpose of avoiding the influence of any chemical on measurements, no treatment was given to the examined finger in twelve hours before the test. A 2 mm diameter probe was held on the surface of the examined finger with a constant pressure of 500 mbar at a room temperature. The measurements were repeated three times to ensure the accuracy of the results.

TABLE 1: The scoring scheme used for the passing test [27].

Time taken (s)	>13.71	13.65-13.71	13.58-13.64	13.51-13.57	13.44-13.50	13.37-13.43	13.30-13.36	13.23-13.29	13.16-13.22	13.09-13.15	<13.08
Target score	0	1	2	3	4	5	6	7	8	9	10

## 4. Data Analysis and Discussion

**4.1. Rugby Ball Passing Test.** As shown in Table 2, the moisture readings rise significantly from the “natural” state to the “hydrated” state when the participants were involved in warm-up activities. This finding is also evaluated by the Friedman test where big differences on the moisture level of the skin were obtained under four moisture conditions ( $p < 0.001$ ). It can be observed that the moisture readings of the examined fingers reach their maximum levels (99 au) in the case of doing 20 m warm-up activities. With regard to the condition of extra water, the tests were performed by saturated fingers with wet balls where additional water was added to rugby ball surfaces, prior to the tests. For the target score, a different phenomenon was observed with respect to various moisture conditions. An initial increase in the target score was found when the moisture level changing from the natural state to the Hydrated 1 state. After that, the target score started to decrease with the increase in moisture readings. The lowest score was found in the case of adding extra water, as given in Table 2. These findings suggest a parabolic relationship of the target score with the moisture level (see Figure 2). A quadratic polynomial model was considered to provide the best fit for these changes in the target score and expressed as  $y = ax^2 + bx + c$  ( $R > 0.9$ ,  $p < 0.03$ ).

Figure 2 shows the data for the passing test under four moisture conditions. For the moisture reading, it was found to be varied with warm-up activities. For 10 min exercise, the moisture reading has a significant increase with time. As shown in Figure 2, the moisture reading for Participant 1 in the natural condition was 69 au and increased to 73 au after taking part in 10 min of warm-up. A similar finding was also noted by Tomlinson et al. [22], in which the moisture levels of the tested fingers were found to increase from 50 au in resting to 90 au when the participants were taking part in various activities. Their findings agree with this study. The changes in the moisture level of the skin could be attributed to the sweating on the skin surface. During various activities, participants’ bodies experience high temperature as a result of body cells converting metabolic energy into thermal energy. In order to maintain the balance of temperature, excessive heat generated is converted into thermoregulatory sweating and evaporated through the skin. Consequently, the sweating accumulated on the skin surface leads to a change in the skin moisture.

The data of the target score with respect to different moisture conditions are also given in Figure 2, where a bell-curve relationship between the target score and the moisture level is observed. The target score is found to have an initial increase with the increase in moisture and then a decrease when the moisture level reaches Hydration 1, except Participant 7. In Participant 7, the target score for Hydrated 2 shows an increasing phenomenon with the increase in the moisture level. This could be attributed by many factors, such as participant’s skill, test conditions, and others. The bell-curve behaviour in the ball pass test was expected because of the curved relationship between skin friction and moisture of the skin. Lewis et al. [8, 9] indicated that the

accuracy of a ball pass is closely associated with the frictional behaviour between the finger pad and ball texture. It is generally believed that a high friction force is desired for a good grip and therefore a higher accuracy of pass rate. In addition, they indicated the coefficient of the skin friction is inversely proportional to the skin’s Young’s modulus. An appropriate increase in the moisture level of the skin could reduce its Young’s modulus and, therefore, result in a rise in the coefficient of the skin friction. In the studies of Tomlinson et al. [7], they found the target score changes with the skin friction coefficient following a linear regression model. In the comparison of the skin friction coefficient between the dry balls and the wet balls, the results show that the wet balls have relative lower coefficients than those of dry balls. Moreover, in their passing accuracy test, the balls with higher friction coefficient give better scores than those with lower friction coefficient.

On the basis of the above findings, it is suggested that the dry balls with a relative low moisture level will present a good friction coefficient and, therefore, a good passing accuracy. Oppositely, wet balls usually give a lower friction coefficient and a lower passing accuracy. This conclusion is in consonance with the findings of this study. When the participants were in activities (up to 10 min), the moisture level was found to increase due to thermoregulatory sweating and this leads to an increase in the target score. While increasing the warm-up time, the mechanical properties of the skin could be affected due to excessive sweating and consequently result in a decrease in skin friction and then a lower target score. With regard to the extra water condition, the lowest target score is obtained. This could be attributed to that the additional water on the ball surface acts as a lubricant at the contact interface. The water-based lubrication film produced a lower friction between the hand and the ball surface and led to a lower target score.

**4.2. Skin Friction.** A curve relationship was observed between the moisture reading and the hydration time, as shown in Figure 3(a). It was noticed that the moisture reading is increased by 55% with hydration. In the natural state, the moisture reading was 53 au and rose to 82 au after being hydrated for 80 s. After that, the moisture reading showed a decreasing tendency with the hydration time and reached a plateau around 200 s. Figure 3(b) shows the data of the skin friction coefficient collected from different moisture conditions. It indicated that the coefficient of the skin friction varies with hydration time with an initial increase in the coefficient of the skin friction when the examined fingers were hydrated for around 80–120 s. Then, it is followed by an approximately 35% of decrease in the coefficient of the skin friction with time. This finding conforms to the work of Tomlinson et al. [22]. In their studies, an initial increase in the coefficient of friction was observed with a small amount of water added to participants’ hands, and then this was followed by a decrease. Additionally, in some other studies, Adams et al. [17] carried out a work on the frictional behaviour of the human skin where a polypropylene probe was sliding on participant’s forearms under different moisture

TABLE 2: The results collected from the target passing test.

Participants	Moisture levels				
	Natural	Hydrated 1	Hydrated 2	Extra water	
1	MMR $\pm$ SD (au)	71.3 $\pm$ 1.8	73.0 $\pm$ 1.8	96.0 $\pm$ 2.8	99.0 $\pm$ 0.0
	Target score	6	9	6	5
2	MMR $\pm$ SD (au)	56.3 $\pm$ 1.3	72.3 $\pm$ 1.0	99.0 $\pm$ 0.0	99.0 $\pm$ 0.0
	Target score	5	7	5	4
3	MMR $\pm$ SD (au)	51.3 $\pm$ 2.8	69.7 $\pm$ 1.8	86.7 $\pm$ 6.9	99.0 $\pm$ 0.0
	Target score	7	8	8	5
4	MMR $\pm$ SD (au)	56.0 $\pm$ 4.4	77.0 $\pm$ 3.6	99.0 $\pm$ 0.0	99.0 $\pm$ 0.0
	Target score	7	9	7	7
5	MMR $\pm$ SD (au)	67.0 $\pm$ 1.6	75.0 $\pm$ 2.8	99.0 $\pm$ 0.0	99.0 $\pm$ 0.0
	Target score	6	8	6	6
6	MMR $\pm$ SD (au)	68.0 $\pm$ 1.7	72.0 $\pm$ 0.0	99.0 $\pm$ 0.0	99.0 $\pm$ 0.0
	Target score	7	8	5	4
7	MMR $\pm$ SD (au)	62.7 $\pm$ 4.3	71.3 $\pm$ 0.8	93.3 $\pm$ 3.9	99.0 $\pm$ 0.0
	Target score	5	7	8	5
8	MMR $\pm$ SD (au)	66.7 $\pm$ 0.8	78.0 $\pm$ 0.4	99.0 $\pm$ 0.0	99.0 $\pm$ 0.0
	Target score	7	9	6	6

Note. MMR represents a mean moisture reading collected from both examined fingers, Hydrated 1 represents a medium hydration state, and Hydrated 2 represents a high hydration state.

conditions. The coefficient of the skin friction obtained was found to be  $0.2\mu$  for the dry skin and rose to  $4.2\mu$  after adding demineralised water to the examined skin. The friction coefficient then dropped to  $0.2\mu$  when water was removed. Based on those observations, they suggested that the coefficient of the skin friction and the hydration time are in consonance with a curve relationship (bell-shaped).

The changes in the coefficient of the skin friction under moisture conditions might be ascribed to the growth of the contact area between the skin and the force probe/plate. When adding water to the interface, some water would be absorbed by skin tissues, which will help soften the skin and lead to an increment in the contact area. According to Amontons laws, the coefficient of the skin friction is proportional to the contact area; thus, a rise in the coefficient of the skin friction can be expected due to the increased contact area. For the water remaining on the skin surface, it develops into "liquid bridges" at the interface. Those liquid bridges formed might introduce viscous shear stress into the contact and therefore lead to increases in the contact area and the coefficient of the skin friction. The capillary adhesion might also be found to increase, which is attributed to the increased contact area.

**4.3. Mechanical Properties of the Skin.** The results of the mechanical properties under hydration are shown in

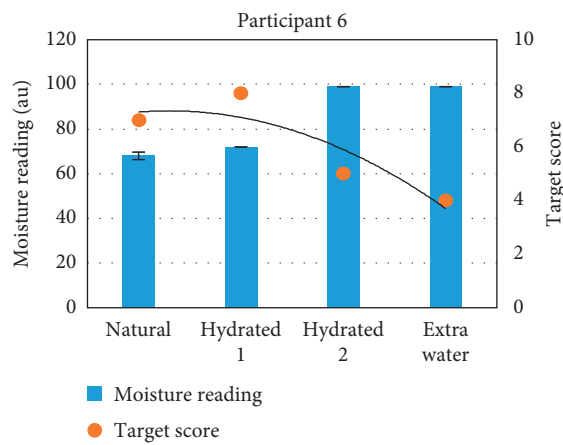
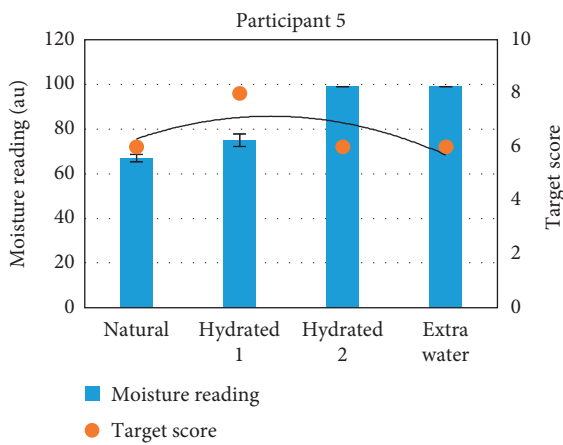
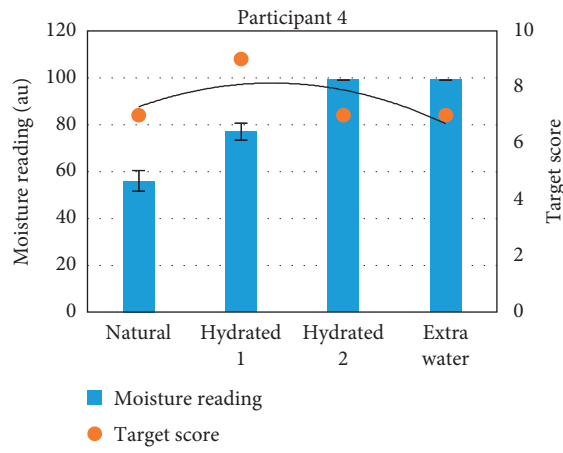
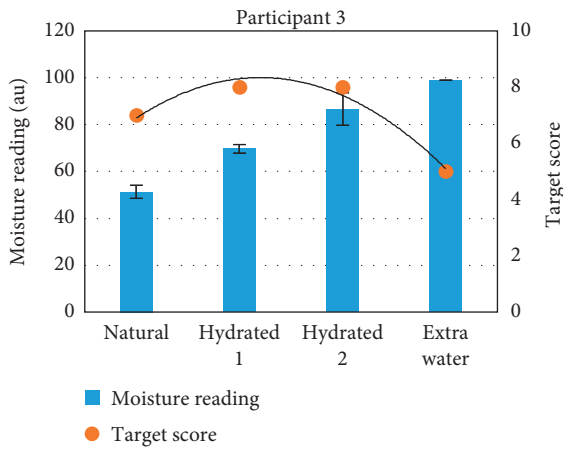
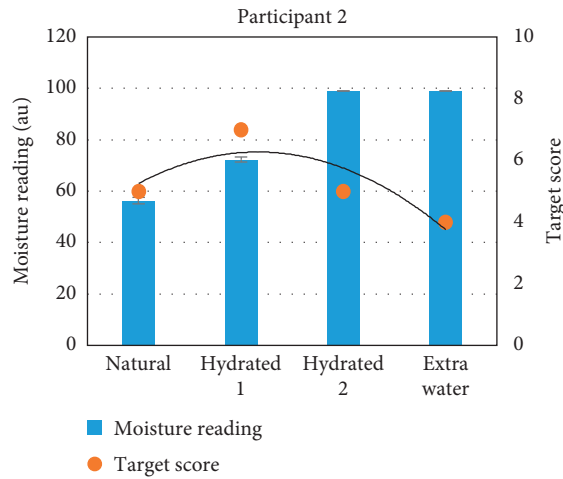
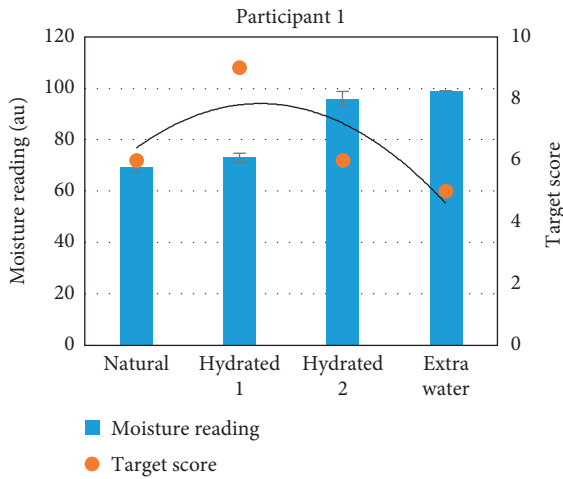
Figure 4. It is noted that the skin deformation is reduced from 0.14 mm to 0.11 mm with the hydration time of 120 s. After that, it is found to gradually recede to 0.11 mm. An opposite phenomenon was found in the global elasticity of the skin ( $U_r/U_e$ , where  $U_r$  is the immediate retraction and  $U_e$  is the immediate distension). Figure 4(b) shows that the global elasticity reaches to 0.7 with the hydration time of 120 s and then decreases to a steady value of 0.5 at 400 s. The viscoelasticity ( $U_v/U_e$ , where  $U_v$  is the delayed distension) was found to be strongly associated with the hydration time as there is a significant reduction observed in Figure 4(c). A curved behaviour was also noted in the normal stiffness with the hydration time, as shown in Figure 4(d). Similar to that of the global elasticity, the normal stiffness has an initial increase of 35% at 120 s, followed by a decline to 0.17 at 400 s.

In previous studies of the skin's mechanical properties, Yuan and Verma [28] indicated that the elastic modulus of the stratum corneum (SC) can be obtained using a noninvasive technique by using a Hysitron TriboScope. The elastic modulus can be defined as a function of unloading stiffness:

$$E = \frac{\sqrt{\pi}}{2\sqrt{A}} * \left( \frac{dN}{dL} \right), \quad (1)$$

where  $E$  is Young's modulus,  $A$  is the contact area,  $N$  is the normal load applied on the skin surface, and  $L$  is the skin deformation subject to the normal load [28]. In their comparative study of elastic modulus in the dry and wet skin, the results show that the elastic modulus of the dry skin is one order of magnitude higher than that of the wet skin. Hendricks and Franklin [15] found that the skin stiffness for finger pads is decreased by about an order of magnitude while increasing its moisture level. The results of Liu et al.'s studies indicated that the elastic modulus and the normal stiffness start to decrease while increasing the thicknesses of the artificial fingers [19]. Their further studies of rubbing tests indicated that one of the possibilities for the reduction in friction force is likely to be the change in SC thickness. According to Liu et al. [19], the adhesion force is in inverse proportion to skin stiffness. The removal of SC does not affect the normal stiffness of the skin, but it does affect the lateral stiffness of the skin. This finding is in good accordance with our observation in Figure 4(d) as no obvious change was noted in the normal stiffness of the skin. In a similar study carried out by Pailler-Mattei et al. [21], they pointed out that the lateral stiffness of the skin is decreased with the removal of the thickness of the SC. They assumed that the change in lateral stiffness might influence the global mechanical properties of the skin and hence influence its friction coefficient.

From the above findings, it is concluded that the coefficient of the skin friction closely correlates with the SC thickness. Therefore, it is expected to observe a decrease in the elastic modulus of the skin while increasing the thickness of the skin related to hydration. Figure 4(a) shows that there is an initial decrease and then an increase in the deformation when the finger pads were hydrated in water up to 400 s. However, the corresponding change in the global elasticity of the skin is not significant. One of the possible causes could



(a)

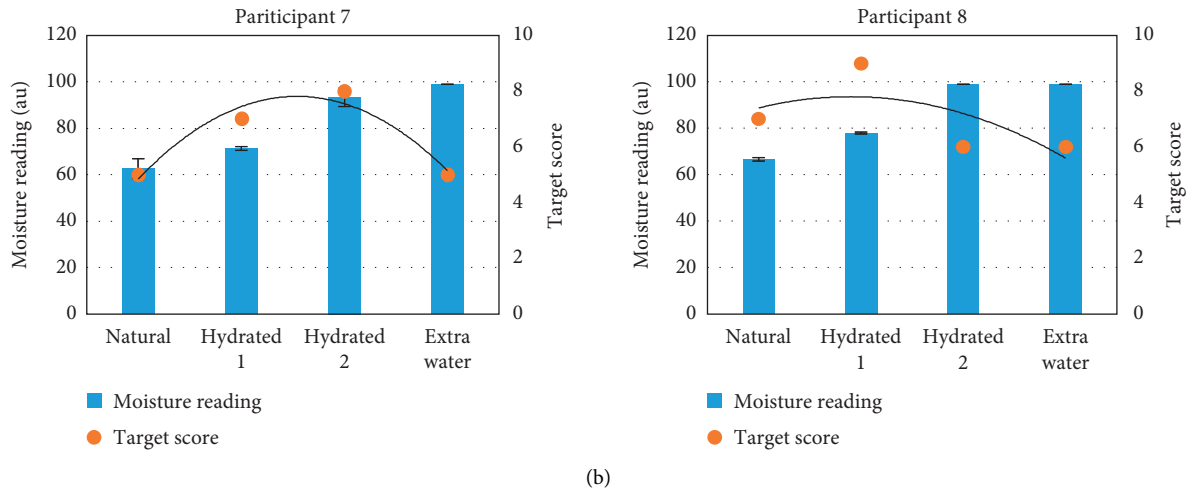


FIGURE 2: The moisture readings and the target scores under different hydration conditions.

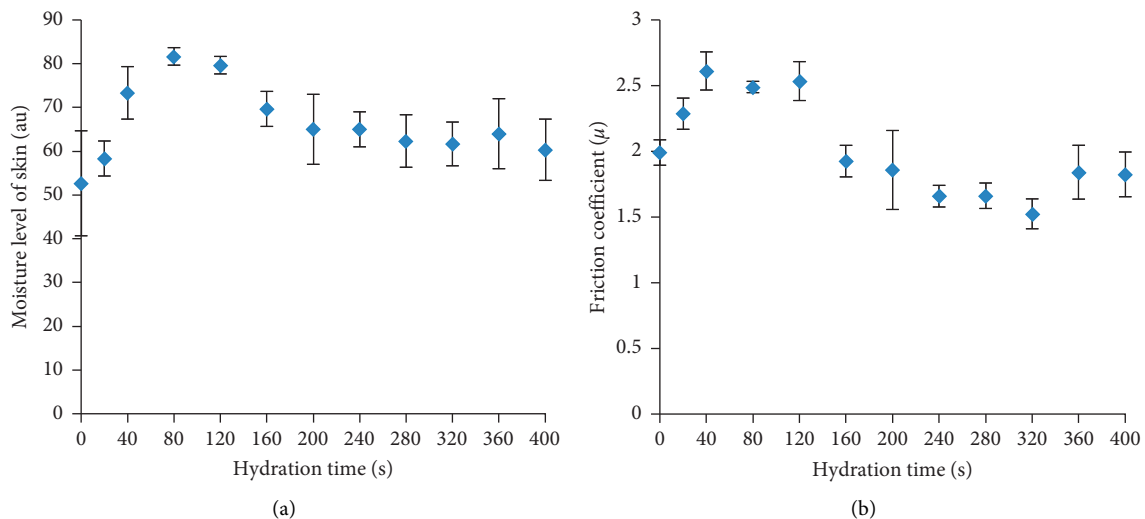


FIGURE 3: (a) The moisture reading versus the hydration time and (b) the skin friction coefficient versus the hydration time.

be that data of the elasticity obtained are of the global elasticity for the whole skin. The global elasticity is believed to be dominated by the dermis layer of the skin, so it might be not very accurate to describe the elastic property of the SC. In addition, in the studies of Liu et al. [27], they indicated that no water is transmitted into the dermis layer of the skin during the hydration treatment. Therefore, it is reasonable to observe that the elasticity of the skin does not have significant changes under wet conditions.

## 5. Data-Driven Models

To generate predictive models for anticipating the passing scores of athletes, two widely used data-driven modelling techniques were employed. The first one is a feedforward artificial neural network (ANN) with a Levenberg–Marquardt training method [29]. In this paper, the ANN model was set to

have a hidden layer with eight sigmoid neurons. The second method used is the adaptive neuro-fuzzy inference system (ANFIS) [30]. It is a Sugeno fuzzy rule-based system that uses the back-propagation gradient descent and least-squares methods in training. In this paper, the ANFIS model constructed has eight fuzzy rules.

In the experiments, six variables were selected as the inputs of models, which are athletes' weight, height, hand length, hand width, wrist strength, and figure moisture level. The output of models is the score of the passing test. The figure moisture level has been observed to be correlated with the ball handling performance in Section 4. The weight and height represent the athletes' physical conditions and often reflect the strength and speed of athletes. The hand length, hand width, and wrist strength directly relate to the ball handling performance of humans. These variables are believed to be associated with the capability of passing rugby

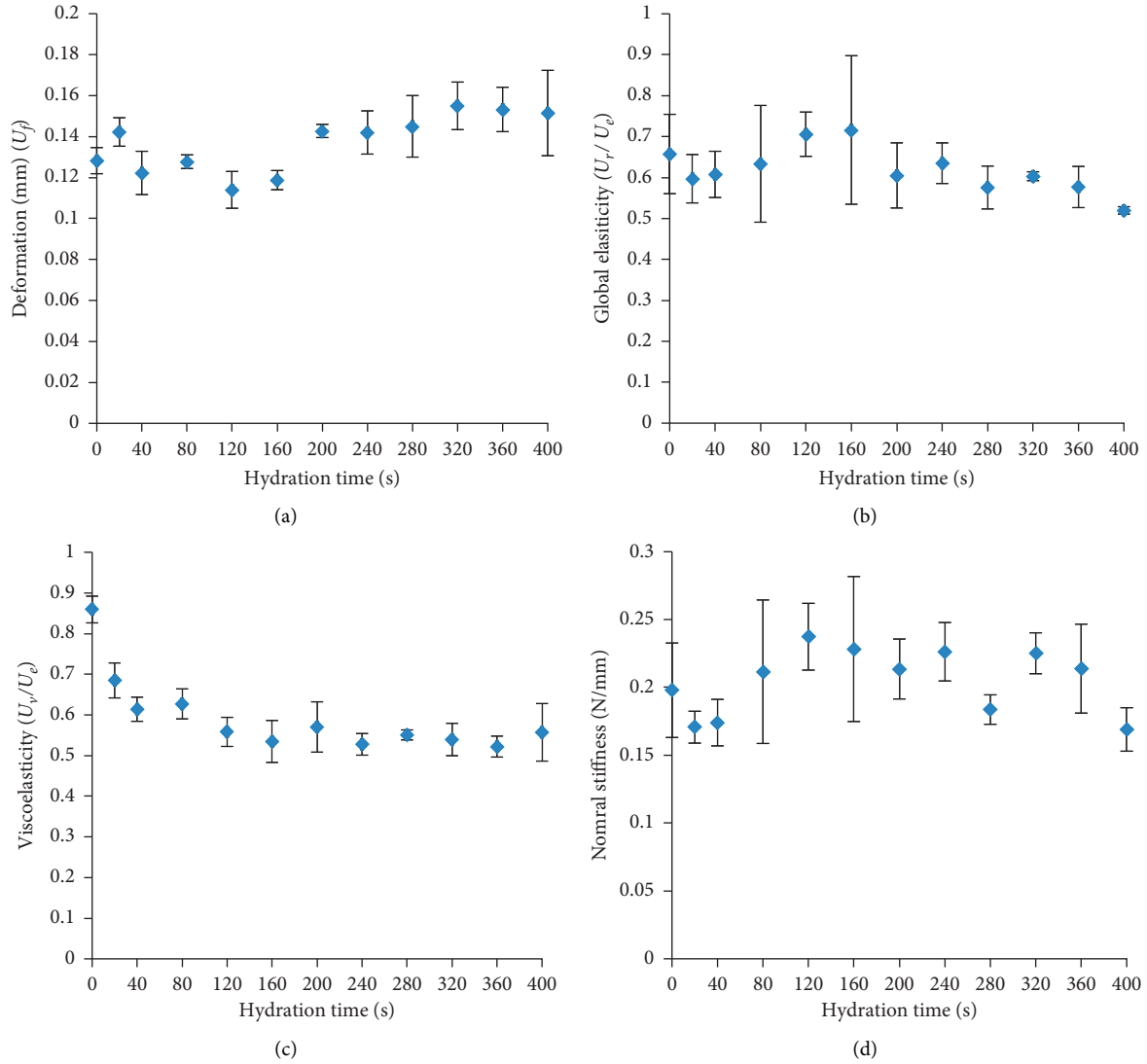


FIGURE 4: (a) Skin deformation ( $U_f$ ) against hydration time, (b) global elasticity ( $U_r/U_e$ ) against hydration time, (c) viscoelasticity ( $U_v/U_e$ ) against hydration time, and (d) normal stiffness against hydration time.

TABLE 3: Basic information about athletes.

Participants	Weight (kg)	Height (m)	Hand length (cm)	Hand width (cm)	Wrist strength (N)
1	73	1.77	19	20	80
2	85	1.8	19	20	85
3	61	1.78	18	21	60
4	80	1.8	19	19	80
5	75	1.8	22	19	50
6	89	1.85	18	19	60
7	64	1.7	18	20	50
8	62	1.76	18	20	60

balls and thus are used as input variables of models. Table 3 shows the basic information about athletes, which was used in modelling. In the experiments, 80% of the collected data were randomly selected and used in training and the remaining 20% of the data were utilised in testing. Table 4 and Figure 5 show the performance of the developed models in prediction. In Table 4, RMSE represents root mean square

error and  $R$  represents Pearson product-moment correlation coefficient.

It can be observed that both the ANN and ANFIS models perform well in the training data, but they are less accurate in testing. Compared with the ANFIS model, the ANN model is less accurate in training; however, it performs much better in testing. The reason lies in that ANN has a validation



TABLE 4: Modelling performance using ANN and ANFIS.

Model	Training		Testing	
	RMSE	$R$	RMSE	$R$
ANN	0.2663	0.9841	0.6158	0.9916
ANFIS	0.1387	0.9942	7.6267	0.8853

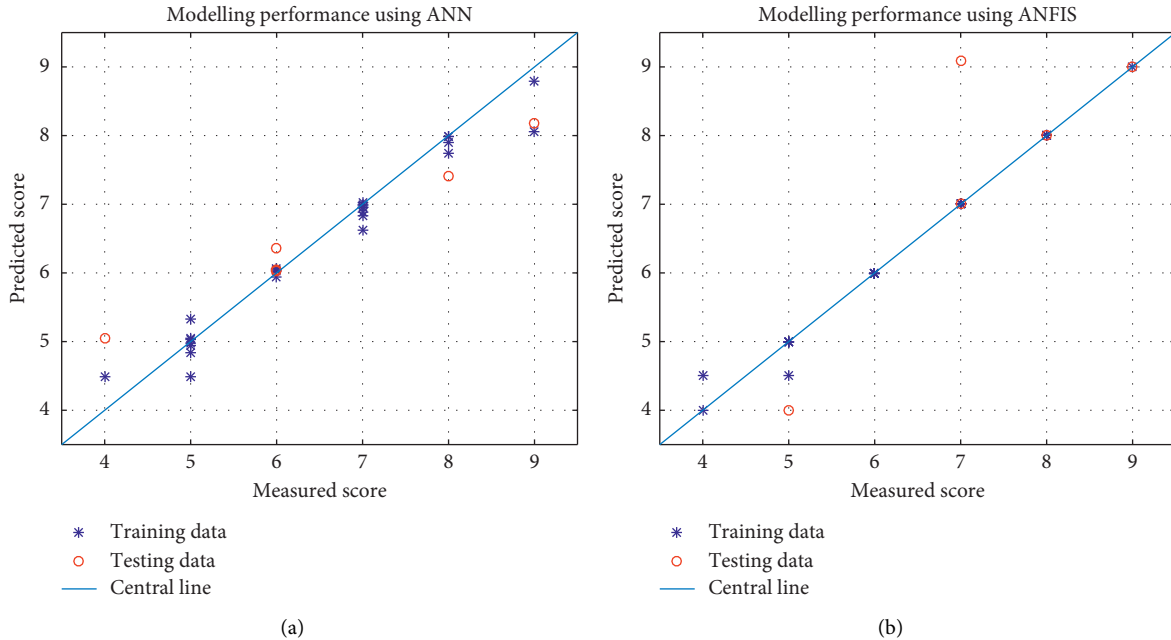


FIGURE 5: Measured passing score versus predicted passing score using (a) ANN and (b) ANFIS.

mechanism that can prevent overtraining, which happened in the training of the ANFIS model.

## 6. Conclusion

The aim of this study was to investigate rugby players' performance under different moisture conditions using data analytics and data-driven modelling methods. In the rugby ball passing test, it has been found that higher target scores can be obtained by increasing the skin moisture to a certain level. When the finger pads were saturated, the passing accuracy started to drop. A quadratic polynomial function has been proposed to describe the relationship between the skin moisture and the target score. The measurement of the skin friction coefficient showed that there is bell-shaped behaviour between the coefficient of the skin friction and the hydration time. Additionally, in the measurement of mechanical properties, the skin elasticity of the finger pads was shown to reduce with hydration. Based on the collected experimental data, intelligent data-driven models have then been developed to anticipate the passing performance of rugby players under different moisture conditions, by employing a neural-network model and a neuro-fuzzy model. The constructed models showed decent accuracy, although with limited training data.

## Data Availability

The data used in this research are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] "Pioneers of the rugby ball," 2019, <http://www.rugbyfootballhistory.com/ball.htm>.
- [2] Board International Rugby, *Law 2 the Ball, Playing Charter*, International Rugby Board, Dublin, Ireland, 2006.
- [3] A. Ross, "Physical characteristics and match performance in rugby sevens," Auckland University of Technology Theses and Dissertations Archive, Auckland, New Zealand, 2015, <https://openrepository.aut.ac.nz/bitstream/handle/10292/9215/RossA.pdf?sequence=1>, Doctoral dissertation.
- [4] L. Vaz, S. Hendricks, and W. Kraak, "Statistical review and match analysis of Rugby World Cups finals," *Journal of Human Kinetics*, vol. 66, no. 1, pp. 247–256, 2019.
- [5] A. Woolford, "Why player jerseys are being blamed for so many people dropping the ball at the Rugby World Cup," 2019, <https://www.walesonline.co.uk/sport/rugby/rugby-news/player-jerseys-being-blamed-many-17029726>.

- [6] P. Williams, "Five things we learnt about rugby in September," *Rugby World*, 2019, <https://www.rugbyworld.com/tournaments/rugby-world-cup-2019/five-things-learnt-rugbyseptember-2-103598>.
- [7] S. E. Tomlinson, R. Lewis, S. Ball, A. Yoxall, and M. J. Carré, "Understanding the effect of finger-ball friction on the handling performance of rugby balls," *Sports Engineering*, vol. 11, no. 3, pp. 109–118, 2009.
- [8] R. Lewis, M. J. Carré, A. Abu Bakar, and S. E. Tomlinson, "Effect of surface texture, moisture and wear on handling of rugby balls," *Tribology International*, vol. 63, pp. 196–203, 2013.
- [9] R. Lewis, M. J. Carré, and S. E. Tomlinson, "Skin friction at the interface between hands and sports equipment," *Procedia Engineering*, vol. 72, pp. 611–617, 2014.
- [10] A. Delalleau, G. Josse, J.-M. Lagarde, H. Zahouani, and J.-M. Bergheau, "A nonlinear elastic behavior to identify the mechanical parameters of human skin *in vivo*," *Skin Research and Technology*, vol. 14, no. 2, pp. 152–164, 2008.
- [11] S. Derler, M. Preiswerk, G.-M. Rotaru, J.-P. Kaiser, and R. M. Rossi, "Friction mechanisms and abrasion of the human finger pad in contact with rough surfaces," *Tribology International*, vol. 89, pp. 119–127, 2015.
- [12] O. S. Dinc, C. M. Ettles, S. J. Calabrese, and H. A. Scarton, "Some parameters affecting tactile friction," *Journal of Tribology*, vol. 113, no. 3, pp. 512–517, 1991.
- [13] L.-C. Gerhardt, V. Strässle, A. Lenz, N. D. Spencer, and S. Derler, "Influence of epidermal hydration on the friction of human skin against textiles," *Journal of the Royal Society Interface*, vol. 5, no. 28, pp. 1317–1328, 2008.
- [14] F. M. Hendriks, "Mechanical behaviour of human epidermal and dermal layers *in vivo*: characterization of non-linear mechanical behaviour of skin using ultrasound," Technische Universiteit Theses and Dissertations Archive, Eindhoven, Netherlands, Doctoral dissertation, 2005.
- [15] C. P. Hendriks and S. E. Franklin, "Influence of surface roughness, material and climate conditions on the friction of human skin," *Tribology Letters*, vol. 37, no. 2, pp. 361–373, 2010.
- [16] S. A. Johnson, D. M. Gorman, M. J. Adams, and B. J. Briscoe, "The friction and lubrication of human stratum corneum," "The friction and lubrication of human stratum corneum," in *Thin Films in Tribology*, D. Dowson, C. M. Taylor, T. H. C. Childs, M. Godet, and G. Dalmaz, Eds., Elsevier Science Publishers, Amsterdam, Netherlands, pp. 663–672, 1993.
- [17] M. J. Adams, B. J. Briscoe, and S. A. Johnson, "Friction and lubrication of human skin," *Tribology Letters*, vol. 26, no. 3, pp. 239–253, 2007.
- [18] X. Liu, Z. Lu, R. Lewis, M. J. Carré, and S. J. Matcher, "Feasibility of using optical coherence tomography to study the influence of skin structure on finger friction," *Tribology International*, vol. 63, pp. 34–44, 2013.
- [19] X. Liu, D. Gad, Z. Lu, R. Lewis, M. Carré, and S. Matcher, "The contributions of skin structural properties to the friction of human finger-pads," *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, vol. 229, no. 3, pp. 294–311, 2015.
- [20] X. Liu, M. J. Carré, Q. Zhang, Z. Lu, S. J. Matcher, and R. Lewis, "Measuring contact area in a sliding human finger-pad contact," *Skin Research and Technology*, vol. 24, no. 1, pp. 31–44, 2018.
- [21] C. Pailler-Mattei, S. Pavan, R. Vargiolu, F. Pirot, F. Falson, and H. Zahouani, "Contribution of stratum corneum in determining bio-tribological properties of the human skin," *Wear*, vol. 263, no. 7–12, pp. 1038–1043, 2007.
- [22] S. E. Tomlinson, R. Lewis, X. Liu, C. Texier, and M. J. Carré, "Understanding the friction mechanisms between the human finger and flat contacting surfaces in moist conditions," *Tribology Letters*, vol. 41, no. 1, pp. 283–294, 2011.
- [23] S. E. Tomlinson, R. Lewis, and M. J. Carré, "Review of the frictional properties of finger-object contact when gripping," *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology*, vol. 221, no. 8, pp. 841–850, 2007.
- [24] F. P. Bowden and D. Tabor, *The Friction and Lubrication of Solids*, Clarendon Press, Oxford, UK, 1954, [https://scholar.google.com/scholar\\_lookup?title=The%20Friction%20and%20Lubrication%20of%20Solids&publication\\_year=1954&author=Bowden%2CFP&author=Tabor%2CD](https://scholar.google.com/scholar_lookup?title=The%20Friction%20and%20Lubrication%20of%20Solids&publication_year=1954&author=Bowden%2CFP&author=Tabor%2CD).
- [25] N. Yoshimune, F. Takaharu, A. Yuichiro et al., "Tactile impression and friction of water on human skin," *Colloids and Surfaces B: Biointerfaces*, vol. 69, no. 2, pp. 264–267, 2009.
- [26] N. K. Veijgen, M. A. Masen, and E. van der Heide, "A novel approach to measuring the frictional behaviour of human skin *in vivo*," *Tribology International*, vol. 54, pp. 38–41, 2012.
- [27] J. Liu, Q. Zhang, and X. Liu, "Understanding handling performance of rugby balls under wet conditions: analysis of finger-ball friction," *International Journal of Performance Analysis in Sport*, vol. 20, no. 5, pp. 782–799, 2020.
- [28] Y. Yuan and R. Verma, "Measuring microelastic properties of stratum corneum," *Colloids and Surfaces B: Biointerfaces*, vol. 48, no. 1, pp. 6–12, 2006.
- [29] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proceedings of International Conference on Neural Networks (ICNN'97)*, pp. 1930–1935, Houston, TX, USA, June 1997.
- [30] J.-S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

## Research Article

# Data-Driven Model for Rockburst Prediction

Hongbo Zhao <sup>1</sup> and Bingrui Chen<sup>2</sup>

<sup>1</sup>*School of Civil and Architectural Engineering, Shandong University of Technology, Zibo 255000, Shandong, China*

<sup>2</sup>*State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan, Hubei 430071, China*

Correspondence should be addressed to Hongbo Zhao; [bxhbzhao@hotmail.com](mailto:bxhbzhao@hotmail.com)

Received 24 May 2020; Revised 12 July 2020; Accepted 25 July 2020; Published 17 August 2020

Guest Editor: Qian Zhang

Copyright © 2020 Hongbo Zhao and Bingrui Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rockburst is an extremely complex dynamic instability phenomenon for rock engineering. Due to the complex and unclear mechanism of rockburst, it is difficult to predict precisely and evaluate reasonably the potential of rockburst. With the development of data science and increasing of case history from rock engineering, the data-driven method provides a good way to mine the complex phenomenon of rockburst and then was used to predict the potential of rockburst. In this study, deep learning was adopted to build the data-driven model of rockburst prediction based on the rockburst datasets collected from the literature. The data-driven model was built based on a convolutional neural network (CNN) and compared with the traditional neural network. The results show that the data-driven model can effectively mine the complex phenomenon and mechanism of rockburst. And the proposed method not only can predict the rank of rockburst but also can compute the probability of rockburst for each corresponding rank. It provides a promising and reasonable approach to predict or evaluate the rockburst.

## 1. Introduction

Rockburst is an extremely complex dynamic instability phenomenon in rock underground excavation. It usually causes injury to workers, damage to equipment, and economic losses. To prevent the rockburst disaster, various methods, such as field monitoring, laboratory test, theory model, empirical model, numerical model, and intelligent method, etc., were adopted to explore the mechanism of rockburst in the last decades [1–21]. With the depth increasing of mining and underground rock excavation, rockburst is becoming more and more serious and is a challenging rock engineering problem in China [14]. Due to the complexity and uncertainty of rockburst, its mechanism is not clearly understood till now. To decrease risk and losses of rockburst, predicting precisely or estimating the reasonable potential of rockburst is critical to the safety and efficient construction of rock underground excavation and mining engineering.

Various methods have been developed to predict or evaluate the rockburst since Cook et al. first proposed a method for predicting the rockburst in mining [22]. Zhou

et al. reviewed the state of the art and the prediction method of rockburst in brief and classified the method into the empirical method, experimental method, analytical method, intelligent method, and numerical method [23]. Sajjad et al. reviewed the prediction method, data preprocessing of rockburst, and developed an intelligent classification model for rockburst prediction [24]. But rockburst was influenced by multiple factors such as rock mass property, in situ stress, geology structure, and engineering position, etc.; it is difficult to predict rockburst precisely. Though a large number of methods were developed to predict or evaluate the rockburst, there is not much progress in the past few decades and not a universally accepted method which is better than other methods to predict rockburst. Rockburst prediction may now be a universal issue for deep buried underground excavation.

Though rockburst is an unsolved engineering issue for rock underground excavation, a deluge of rockburst data have been available; lots of case histories, monitoring in site, and various tests were implemented, analyzed, and published. The mechanism of rockburst was hidden in the above data about rockburst. Extracting the rockburst information

and knowledge from data is a good way to predict or evaluate rockburst. With the development of deep learning, it provides a good way to reveal the mechanism behind data [25]. Deep learning has achieved notable success in the fields of physics [26], chemistry [27], biology [28], and geoscience [29]. The combination of available data, deep learning, and the theory of rockburst offers an exciting new opportunity for expanding our knowledge about rockburst. In this study, deep learning was adopted to build the prediction model of rockburst. Various rockburst data were collected from the literature. The data-driven model of rockburst prediction was built based on deep learning. The prediction rank of rockburst was obtained and was in excellent agreement with the real rank in the engineering practice. The paper is structured as follows: firstly, the idea and algorithm of deep learning are reviewed in Section 2. Secondly, Section 3 introduces the influence factors and datasets of rockburst, and then the data-driven model was built based on deep learning in Section 4. Finally, some conclusions are given in Section 4.

## 2. Deep Learning

Machine learning has proven to be powerful in capturing subtle, complex, nonlinear relationships between predictor and response variables in various research fields. Artificial neural network (ANN), which mimics biological neural networks, is a commonly used supervised machine learning algorithm and has been widely used in rock mechanics and engineering [30–32]. ANN is the basis of most deep learning methods and comprises more layers containing a series of neurons that accept inputs from neurons on the previous layer based on activate functions. Deep learning is a kind of representation machine learning method that demands that a computer must have the ability to automatically reveal data patterns needed for classification or detection [25]. The “deep” here represents the number of layers. With increasing depth of network, the sufficient transformation allows incorporating simpler features into complex features so that the most appropriate hierarchical representations can be extracted from data [25, 33]. Recently, neural networks (NNs) have been widely applied in a large amount of research field by deep neural networks (DNNs). A deep neural network can compute high-level features over data. Some DNNs are introduced briefly in the following.

*2.1. Conventional ANN.* An ANN typically consists of three-layered components: an input layer (inputs), an output layer (outputs), and several hidden layers between them. The general functionality of the hidden layers is to convert the inputs to final outputs. Each hidden layer contains a certain number of parallel processing units, which are referred to as neurons. A neuron is the basic unit of ANN and is used to receive, process, and deliver signals. Figure 1 shows the ANN architecture. ANN algorithm is used to adjust the connected weights between two neurons in the neighbor layer. The weight can be considered as a measure of the strength of the connection between the two neurons. The larger the weight, the stronger the connection. In the

output layer, the outputs are compared to the targets. The difference is called error. The weights in the ANN are iteratively tuned to yield the minimum error. The detailed algorithm is out of the scope of this study and can be found in the relevant literature [34].

*2.2. Convolutional Neural Network.* Convolutional neural network (CNN) is a well-known deep learning architecture inspired by the natural visual perception mechanism of the living creatures [25]. It is a type of feed-forward neural network. CNN neurons are only connected by a limited subarea of the previous layer according to the design of local receptive field (Figure 2). Aside from input and output layers, CNN involves different types of hidden layers, i.e., convolutional layer, pooling layer, and fully connected layer. Convolutional layers are used to abstract local features at different locations among the whole raw input or the intermediate feature maps with learnable filters (kernels). The advantage of convolution operation is reflected mainly in the implementation of weights sharing and spatial correlation among neighbors. Pooling layers, also called subsampling layers, aim to reduce the size of the input layer. Fully connected layers are added to the last pooling layer for classification or as feature representation for further processing and are similar to regular neural networks and contain most of the parameters of CNN. A deep CNN is built by stacking multiple CNNs aiming to integrate the low-level features into a higher level of representations. This kind of design is powerful for seizing local geometric features and spatial patterns and detecting larger-scale features in deeper layers. The advantages of CNN come from the differences in structures and operations of the convolutional layer and pooling layer. In addition, many neurons in the same layer of CNN share the same weight, thereby reducing the degree of freedom in the model.

In this study, deep learning was used to build the data-driven model for rockburst prediction. Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow [35]. It was adopted in the data-driven model based on deep learning.

## 3. Data-Driven Model of Rockburst Prediction

*3.1. Influence Factors and Rank of Rockburst.* It is difficult to properly evaluate rockburst because of its complicated and indistinct mechanism. The occurrence of rockburst was affected by many different factors such as geologic structure, mining and excavation methods, mechanical property of rock mass and in situ stress, etc. To predict precisely rockburst, it is very important to determine the influence factor reasonably. A vast number of single indicators and multi-indicators have been developed for evaluating the occurrence and intensity of rockburst [23]. These indicators are mainly based on properties of rock, energy, depth of excavation and support structure, etc. But the two necessary conditions are the rock mass and its environment which has the capability of accumulated strain energy and stress concentration. The mechanical

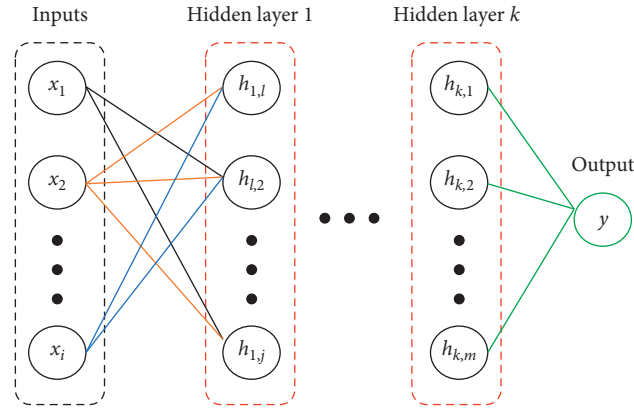


FIGURE 1: Artificial neural network architecture.

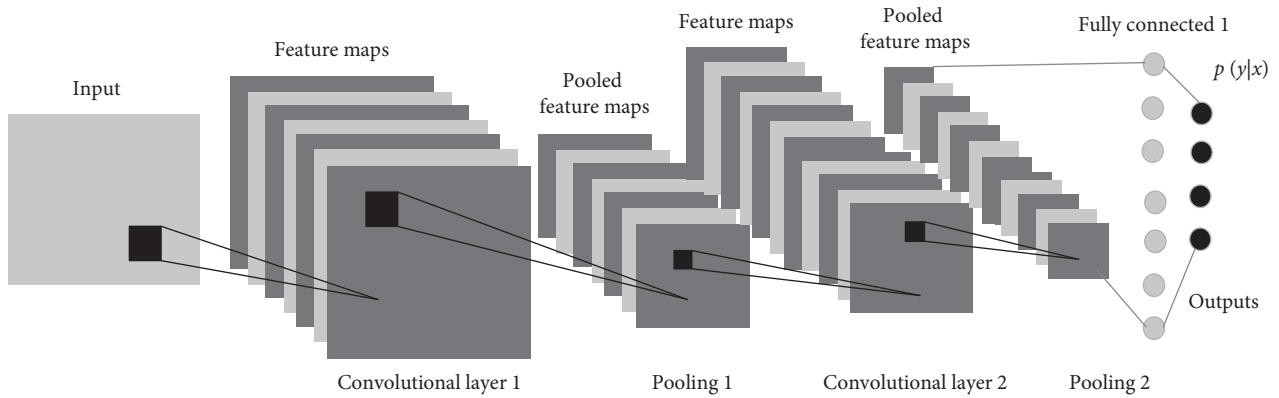


FIGURE 2: Convolutional neural network.

property of rock mass can be characterized by uniaxial compression and tensile strength, and the maximum shear stress of the tunnel wall can reflect the environmental conditions. In this study, uniaxial compression and tensile strength of rock mass, maximum shear stress of tunnel wall, and linear elastic energy were selected to predict the rank of rockburst based on the previous studies in the literature.

Various studies of the potential of rockburst have been conducted in the last decades. Russnes proposed and classified the rockburst intensity into four ranks, i.e., none, weak, moderate, and severe [36]. Brauner classified the rockburst into three ranks based on the intensity of destruction to the surrounding rock mass [37]. Tan classified the rockburst into four ranks based on lots of laboratory tests and field investigations [38]. Cai et al. developed the four ranks of rockburst to evaluate the rockburst liability [39]. The four ranks' method of rockburst classification has been widely used in mining, tunnel, and other rock engineering. So, the four ranks of rockburst were adopted in this study. The input of deep learning was uniaxial compression and tensile strength of rock mass, maximum shear stress of tunnel wall, and linear elastic energy, respectively. The output is the ranks of rockburst, i.e., no rockburst, moderate rockburst, strong rockburst, and violent rockburst. Numbers

1, 2, 3, and 4 were adopted to represent the different ranks of rockburst in deep learning model (1—no rockburst, 2—moderate rockburst, 3—strong rockburst, and 4—violent rockburst).

**3.2. Datasets.** The training data are critical to the learning effect and are necessary to build a deep learning model. In the literature, lots of researchers collected lots of case history and laboratory tests and evaluated the potential of rockburst using different predicted models such as empirical models, numerical models, and intelligent models. In this study, the datasets were collected from the literature based on the previous works [19, 23]. The datasets consist of 165 samples which have four influence factors and a corresponding rank (label) of rockburst. The datasets were divided into 137 training samples and 28 test samples. Appendix listed the training samples (Table 1).

The relationship between each rockburst influence factor and its rank is shown in Figure 3. We can see from Figure 3 that the maximum shear stress of tunnel wall ( $\sigma_\theta$ ) and linear stress energy have higher impacts on the rockburst rank than uniaxial compression strength ( $\sigma_c$ ) and uniaxial tensile strength ( $\sigma_t$ ). It is quite clear that it is impossible to build the prediction model for rockburst based on individual influence factors.

TABLE 1: Training samples.

No.	Maximum shear stress of tunnel wall $\sigma_{\theta}$ (MPa)	Uniaxial compression strength $\sigma_c$ (MPa)	Uniaxial tensile strength $\sigma_t$ (MPa)	Linear stress energy	Rockburst rank
1	90.00	170.00	11.30	9	3
2	90.00	220.00	7.40	7.3	2
3	62.60	165.00	9.40	9	2
4	55.40	176.00	7.30	9.3	3
5	30.00	88.70	3.70	6.6	3
6	48.75	180.00	8.30	5	3
7	80.00	180.00	6.70	5.5	2
8	89.00	236.00	8.30	5	3
9	98.60	120.00	6.50	3.8	3
10	108.40	140.00	8.00	5	4
11	57.00	180.00	8.30	5.00	3
12	50.00	130.00	6.00	5.00	3
13	62.50	175.00	7.25	5	3
14	75.00	180.00	8.30	5	3
15	11.00	115.00	5.00	5.7	1
16	43.40	123.00	6.00	5	3
17	18.80	178.00	5.70	7.40	1
18	34.00	150.00	5.40	7.8	1
19	56.10	131.99	9.44	7.44	3
20	54.20	134.00	9.10	7.1	3
21	70.30	128.30	8.70	6.4	3
22	60.70	111.50	7.86	6.16	4
23	54.20	134.00	9.09	7.08	3
24	70.30	129.00	8.73	6.43	3
25	35.00	133.40	9.30	2.9	2
26	157.30	91.23	6.92	6.27	4
27	148.40	66.77	3.81	5.08	2
28	132.10	51.50	2.47	4.63	3
29	127.90	35.82	1.24	3.67	2
30	107.50	21.50	0.60	2.29	1
31	96.41	18.32	0.38	1.87	1
32	167.20	110.30	8.36	6.83	4
33	118.50	26.06	0.77	2.89	2
34	34.15	54.20	12.10	3.17	2
35	60.00	135.00	15.04	4.86	2
36	60.00	66.49	9.72	2.15	2
37	60.00	106.38	11.20	6.11	2
38	60.00	86.03	7.14	2.85	2
39	60.00	149.19	9.30	3.5	2
40	60.00	136.79	10.42	2.12	2
41	63.80	110.00	4.50	6.31	3
42	2.60	20.00	3.00	1.39	1
43	44.40	120.00	5.00	5.1	2
44	13.50	30.00	2.67	2.03	2
45	70.40	110.00	4.50	6.31	3
46	3.80	20.00	3.00	1.39	1
47	57.60	120.00	5.00	5.1	3
48	19.50	30.00	2.67	2.03	3
49	81.40	110.00	4.50	6.31	4
50	4.60	20.00	3.00	1.39	1
51	73.20	120.00	5.00	5.1	3
52	30.00	30.00	2.67	2.03	4
53	15.20	53.80	5.56	1.92	1
54	88.90	142.00	13.20	3.62	4
55	59.82	85.80	7.31	2.78	3
56	32.30	67.40	6.70	1.1	1
57	30.10	88.70	3.70	6.6	4
58	18.80	171.50	6.30	7	1
59	34.00	149.00	5.90	7.6	2



TABLE 1: Continued.

No.	Maximum shear stress of tunnel wall $\sigma_{\theta}$ (MPa)	Uniaxial compression strength $\sigma_c$ (MPa)	Uniaxial tensile strength $\sigma_t$ (MPa)	Linear stress energy	Rockburst rank
60	38.20	53.00	3.90	1.6	1
61	11.30	90.00	4.80	3.6	1
62	92.00	263.00	10.70	8	2
63	62.40	235.00	9.50	9	4
64	43.40	136.50	7.20	5.6	4
65	11.00	105.00	4.90	4.7	1
66	46.40	100.00	4.90	2.00	2
67	23.00	80.00	3.00	0.85	2
68	46.20	105.00	5.30	2.30	2
69	13.90	124.00	4.22	2.04	1
70	17.40	161.00	3.98	2.19	2
71	19.00	153.00	4.48	2.11	2
72	19.70	142.00	4.55	2.26	2
73	18.70	82.00	10.90	1.5	1
74	28.60	122.00	12.00	2.5	3
75	29.80	132.00	11.50	4.6	3
76	33.60	156.00	10.80	5.2	3
77	26.90	92.80	9.47	3.7	3
78	55.90	128.00	6.29	8.1	4
79	59.90	96.60	11.70	1.8	2
80	68.00	107.00	6.10	7.20	4
81	105.50	187.00	19.20	7.27	3
82	105.50	170.00	12.10	5.76	3
83	105.50	190.00	17.10	3.97	3
84	47.56	58.50	3.50	5	2
85	43.62	78.10	3.20	6	2
86	25.70	59.70	1.30	1.7	1
87	26.90	62.80	2.10	2.4	2
88	40.40	72.10	2.10	1.9	2
89	39.40	65.20	2.30	3.4	3
90	38.20	71.40	3.40	3.6	3
91	45.70	69.10	3.20	4.1	3
92	35.80	67.80	3.80	4.3	3
93	39.40	69.20	2.70	3.8	3
94	40.60	66.60	2.60	3.7	3
95	39.00	70.10	2.40	4.8	3
96	57.20	80.60	2.50	5.5	4
97	55.60	114.00	2.30	4.7	3
98	56.90	123.00	2.70	5.2	3
99	62.10	132.00	2.40	5	3
100	29.70	116.00	2.70	3.7	2
101	29.10	94.00	2.60	3.2	2
102	27.80	90.00	2.10	1.8	1
103	30.30	88.00	3.10	3	2
104	55.60	114.00	2.30	4.7	3
105	41.60	67.60	2.70	3.7	3
106	40.10	72.10	2.30	4.6	3
107	58.20	83.60	2.60	5.9	4
108	56.80	112.00	2.20	5.2	3
109	89.56	190.3	17.13	3.97	3
110	89.56	170.28	12.07	5.76	3
111	89.56	187.17	19.17	7.27	3
112	48	120	1.5	5.8	3
113	63	115	1.5	5.7	3
114	49.5	110	1.5	5.7	3
115	30.9	82.56	6.5	3.2	2
116	89	128.6	13.2	4.9	4
117	12.3	237.1	17.66	6.9	1
118	55.6	256.5	18.9	9.1	3

TABLE 1: Continued.

No.	Maximum shear stress of tunnel wall $\sigma_\theta$ (MPa)	Uniaxial compression strength $\sigma_c$ (MPa)	Uniaxial tensile strength $\sigma_t$ (MPa)	Linear stress energy	Rockburst rank
119	91.3	225.6	17.2	7.3	4
120	61	171.5	22.6	7.5	2
121	108.4	138.4	7.7	1.9	4
122	69.8	198	22.4	4.68	2
123	105	171.3	22.6	7.27	4
124	105	237.16	17.66	6.38	4
125	105	304.21	20.9	10.57	4
126	25.49	54.2	2.49	3.17	2
127	72.07	147.09	10.98	6.53	3
128	21.8	160	5.2	2.22	1
129	20.9	160	5.2	2.22	1
130	12.1	160	5.2	2.22	1
131	75	170	11.3	9	3
132	105	128.61	13	5.76	4
133	105	304	9.12	5.76	3
134	105	306.58	13.9	6.38	4
135	7.5	52	3.7	1.3	1
136	24.93	99.7	4.8	3.8	1
137	14.96	99.7	4.8	3.8	1

Note. 1: no rockburst; 2: moderate rockburst; 3: strong rockburst; 4: violent rockburst.

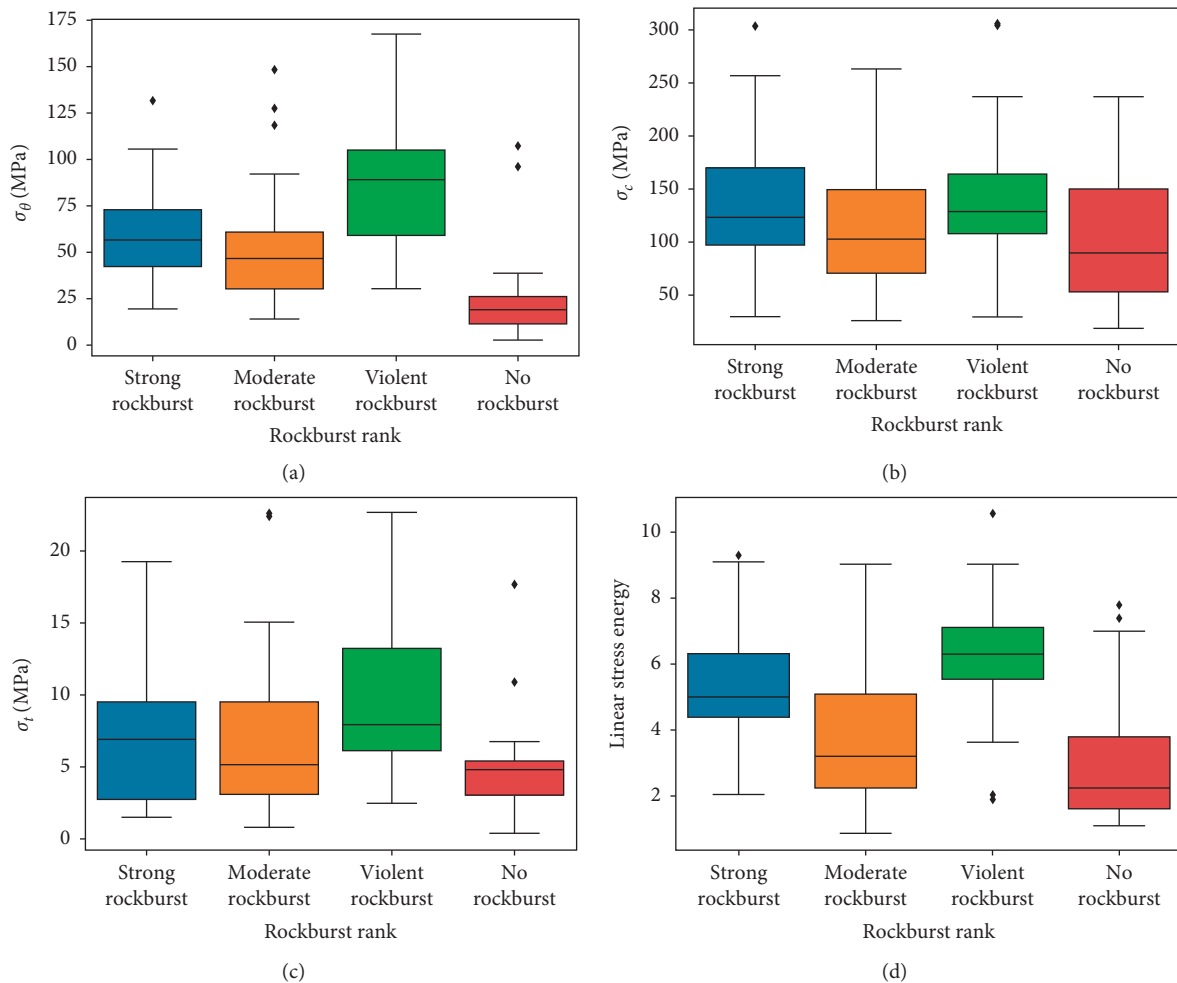


FIGURE 3: The relation between each rockburst influence factor and its rank, (a)  $\sigma_\theta$ , (b)  $\sigma_c$ , (c)  $\sigma_t$ , (d) linear stress energy.

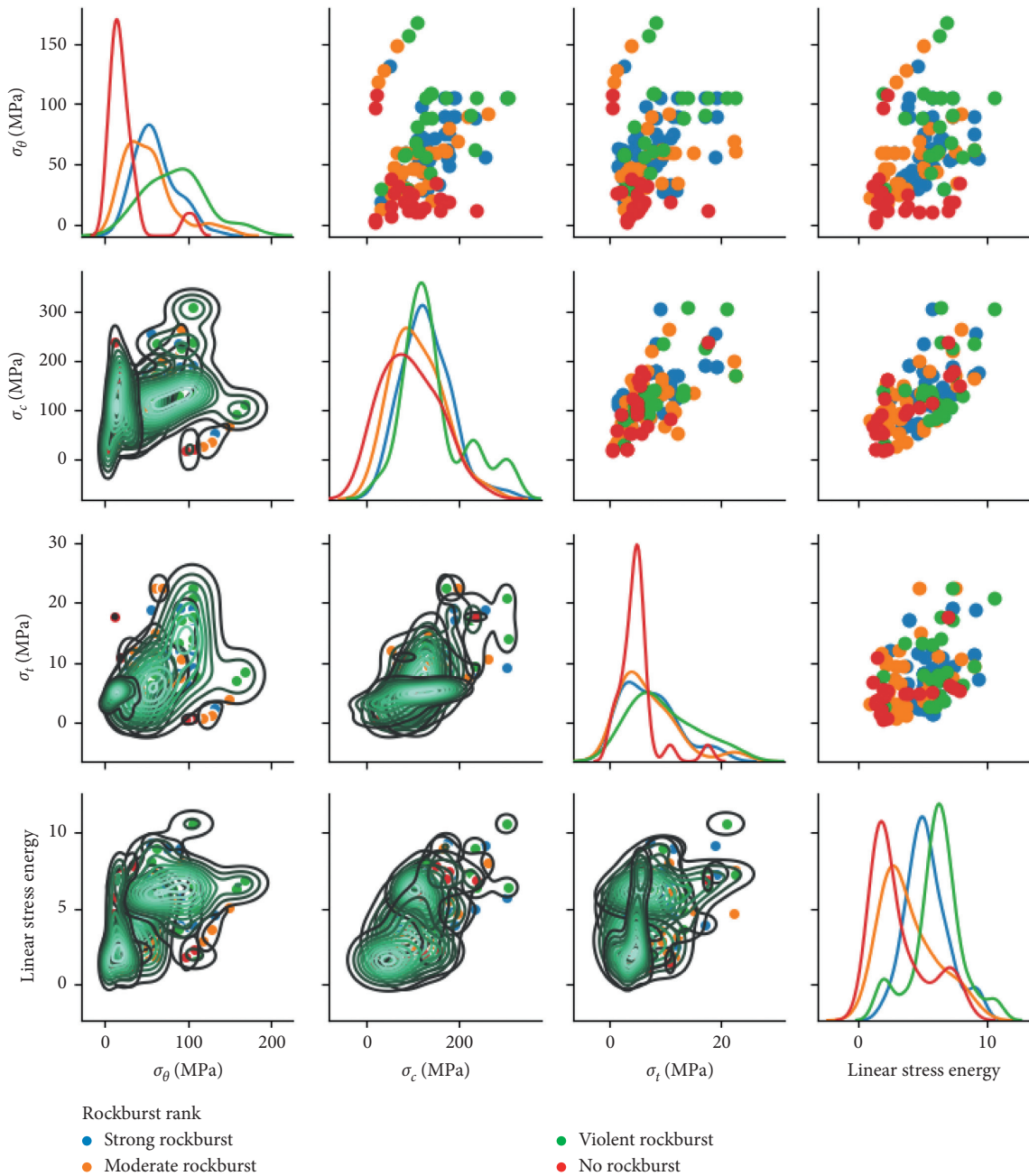


FIGURE 4: The bivariate relation between each pair of rockburst influence factor.

Figure 4 shows the bivariate relation between each pair of rockburst influence factor. It shows that the relationship between rockburst and its influence factor is very complex, uncertain, and nonlinear. There is no apparent dependence between them. It is impossible to evaluate and predict the rockburst using the bivariate relation. It is difficult to characterize the mechanism of rockburst using the traditional statistical method. Figure 5 also shows the relationship between rockburst rank and its influence factors. It is obvious that the rockburst rank depends on the influence factor and their relationship is very complex. For the complex and nonlinear relationship, it is difficult to build the

mathematical model and recognize the rank of rockburst using the traditional mathematical model.

**3.3. Rockburst Prediction.** In the past few decades, machine learning such as neural network and support vector machine has been used to predict or evaluate rockburst. To verify and illustrate the data-driven model of rockburst prediction, NN and CNN were used to build a prediction model and evaluate the potential of rockburst based on the training samples in Table 2. And the comparisons have been implemented and some results were obtained, which

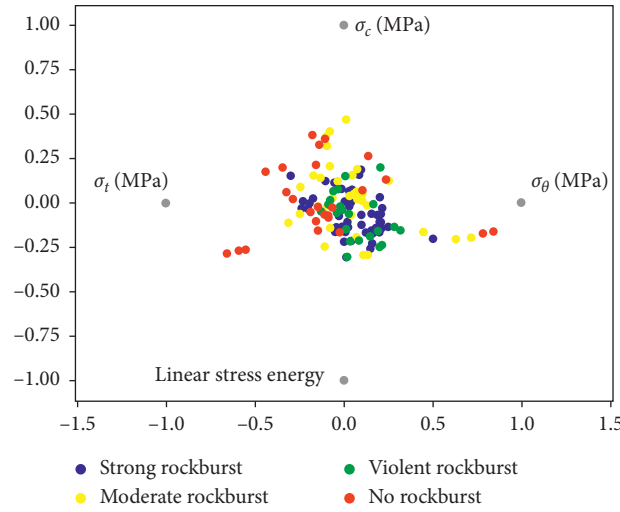


FIGURE 5: The relationship between rockburst rank and its influence factor.

TABLE 2: The test samples and the predicted results using neural network.

No.	Maximum shear stress of tunnel wall $\sigma_\theta$ (MPa)	Uniaxial compression strength $\sigma_c$ (MPa)	Uniaxial tensile strength $\sigma_t$ (MPa)	Linear stress energy	Rockburst rank	
					Real	Predicted
1	34	150	5.4	7.8	1	1
2	60.7	111.5	7.86	6.16	4	4
3	54.2	134	9.09	7.08	3	3
4	70.3	129	8.73	6.43	3	3
5	35	133.4	9.3	2.9	2	2
6	157.3	91.23	6.92	6.27	4	4
7	148.4	66.77	3.81	5.08	2	2
8	132.1	51.5	2.47	4.63	3	2
9	127.9	35.82	1.24	3.67	2	2
10	107.5	21.5	0.6	2.29	1	1
11	96.41	18.32	0.38	1.87	1	1
12	167.2	110.3	8.36	6.83	4	4
13	38.2	53	3.9	1.6	1	1
14	11.3	90	4.8	3.6	1	1
15	92	263	10.7	8	2	2
16	62.4	235	9.5	9	4	4
17	43.4	136.5	7.2	5.6	4	3
18	11	105	4.9	4.7	1	1
19	90	170	11.3	9	3	3
20	90	220	7.4	7.3	2	2
21	62.6	165	9.4	9	2	2
22	55.4	176	7.3	9.3	3	4
23	30	88.7	3.7	6.6	3	3
24	48.75	180	8.3	5	3	3
25	80	180	6.7	5.5	2	2
26	89	236	8.3	5	3	3
27	98.6	120	6.5	3.8	3	3
28	108.4	140	8	5	4	4

proved that deep learning is a promising tool for predicting rockburst precisely.

3.3.1. *NN Model for Rockburst Prediction.* NN was built based on the training samples in Appendix. Figure 6 shows the neural network structure trained based on

the samples in Appendix. 28 testing samples were used to verify the NN model. Table 2 lists the predicted results and their comparison with the real rank of rockburst. The predicted rank of 25 samples was in good agreement with the real rank and Nos. 8, 17, and 22 were not classified correctly. The error ratio was about 11%. The results were in good agreement with the previous research using NN

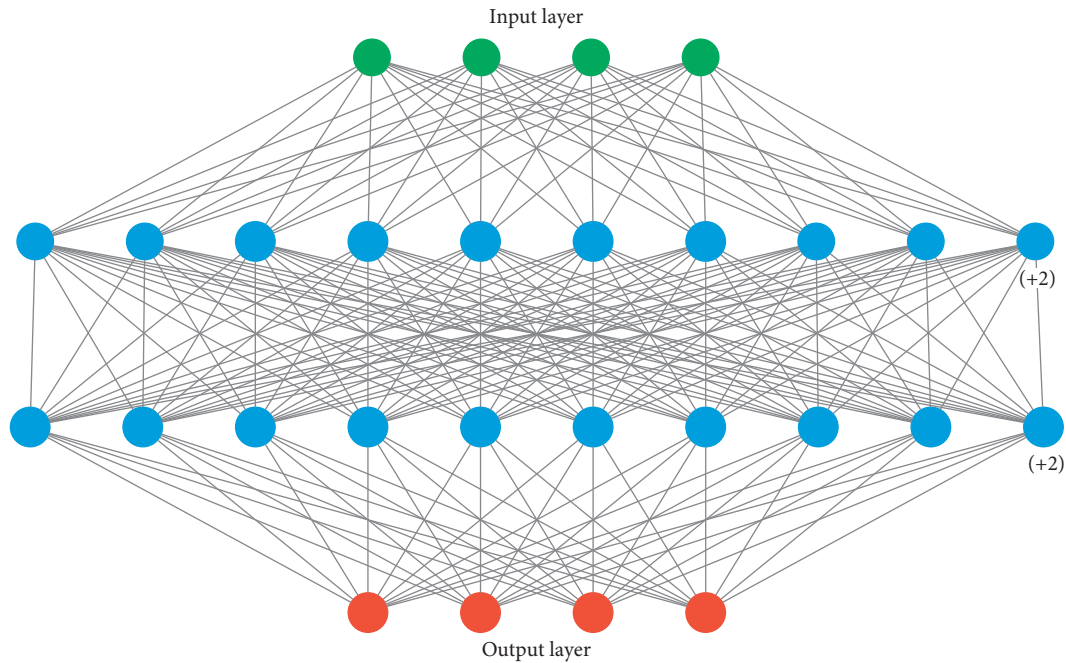


FIGURE 6: Neural network structure.

[17]. This showed that NN can effectively mine the relationship between rockburst and its influence factors. Meanwhile, NN can estimate the probability of rockburst rank. Figure 7 shows the comparison between real rank and predicted rank for testing samples. The real prediction occurred at the neighboring rank for Nos. 8, 17, and 22 samples. As we know, the mechanism of rockburst is complex and not clear. The relationship between rockburst and its influence factors is not clear and uncertain. It is difficult to classify the rank of rockburst which falls in between two neighboring ranks. The rank probability of each sample can be obtained using the NN model. Figure 8 shows the rank probability of rockburst for each testing sample. The probabilistic results were more reasonable than the deterministic value. Figure 9 shows the probability of each rank for Nos. 1, 8, 17, and 22 testing samples. The real rank of No. 1 is no rockburst and is predicted correctly by the NN model. The probabilities of no rockburst and weak rockburst were very close to each other and the probability of no rockburst is a little bigger than the probability of weak rockburst. But the ranks of Nos. 8, 17, and 22 do not classify correctly the rank of rockburst.

**3.3.2. CNN Model for Rockburst Prediction.** CNN was also adopted to build a prediction model of rockburst based on the same training samples. Figure 10 shows the predicted results and their comparison with the real rank of rockburst. It was obvious that the predicted rank of 28 samples was in excellent agreement with the real rank and all samples are classified correctly. The results are better than using the NN model. This showed that deep learning can effectively mine the relationship between rockburst and its influence factors.

Figure 11 shows the rank probability of each sample using the CNN model. Compared with the NN model (Figure 8), the CNN model can reduce the uncertainty of rank and improve the predicted results. In other words, the CNN model can distinguish effectively the neighboring rank of rockburst. The predicted result and rank probability of Nos. 1, 8, 17, and 22 are shown in Figure 12. It further proved the above statement. For the NN model, the probability of no rockburst and weak rockburst is almost the same for the No. 1 sample. The NN model cannot separate the proper rank from the neighboring rank for the No. 8, 17, and 22 samples. But the CNN model can determine correctly the rank of rockburst from the neighboring rank and the probability of the corresponding rank was far from the other ranks (compared with Figure 9).

**3.4. Comparison.** To illustrate and verify the data-driven model of rockburst, the developed method was compared with the traditional empirical criteria of the rockburst including the Russenes criterion [36], the rock brittleness coefficient criterion [40], and the elastic energy index [41]. Table 3 lists the results of 28 testing samples. Additionally, Zhou et al. and Sajjad et al. reviewed the various prediction methods of rockburst [23, 24]. The results of this study are of higher accuracy than other methods (including empirical method and intelligent method). So, the comparison shows that the data-driven model (NN and CNN) has superiority. The data-driven model can reveal the complex and uncertain phenomenon behind data and present well the relationship between rockburst and its influence factor. With the increasing of case history, the data-driven model provides a promising tool for rockburst prediction.

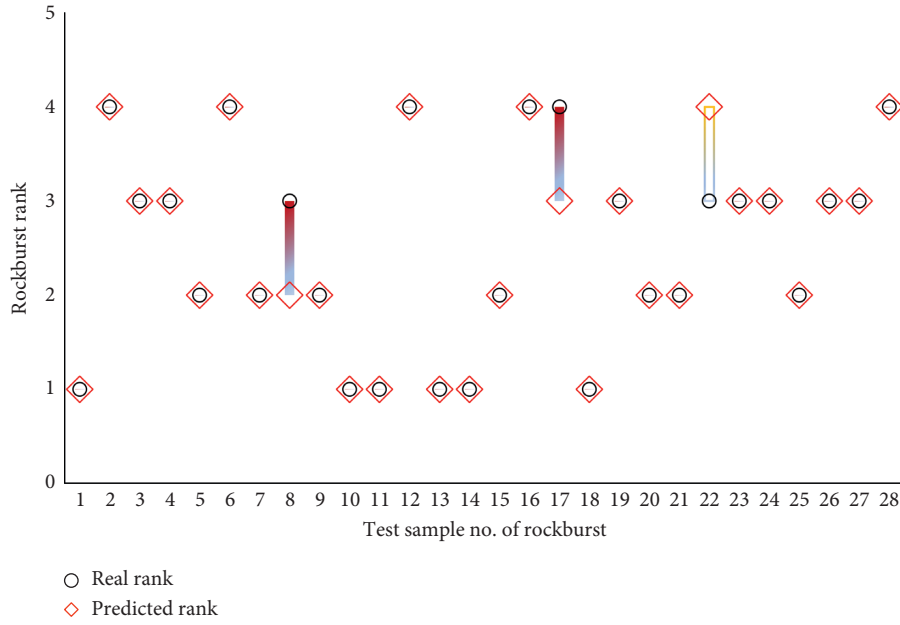


FIGURE 7: The predicted results of deep learning and their comparison for the testing samples.

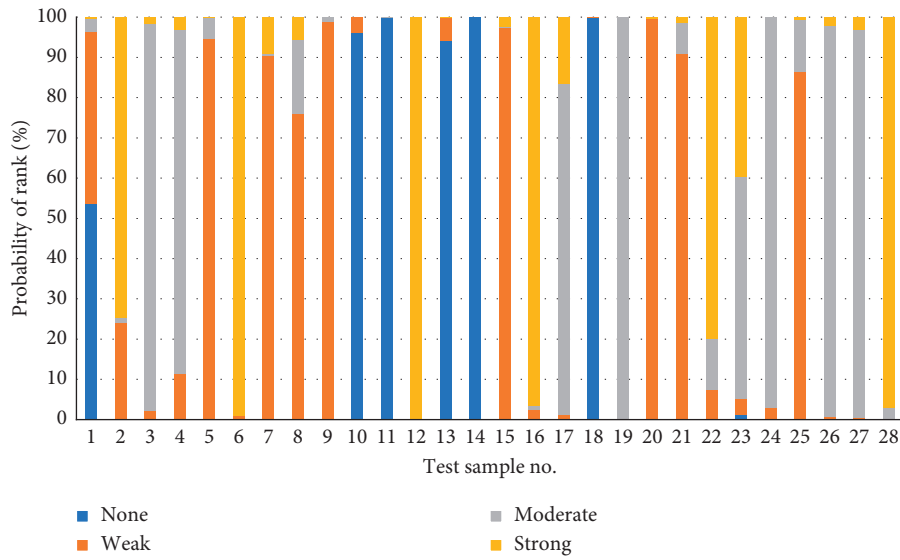


FIGURE 8: The rank probability of rockburst for the testing samples using NN model.

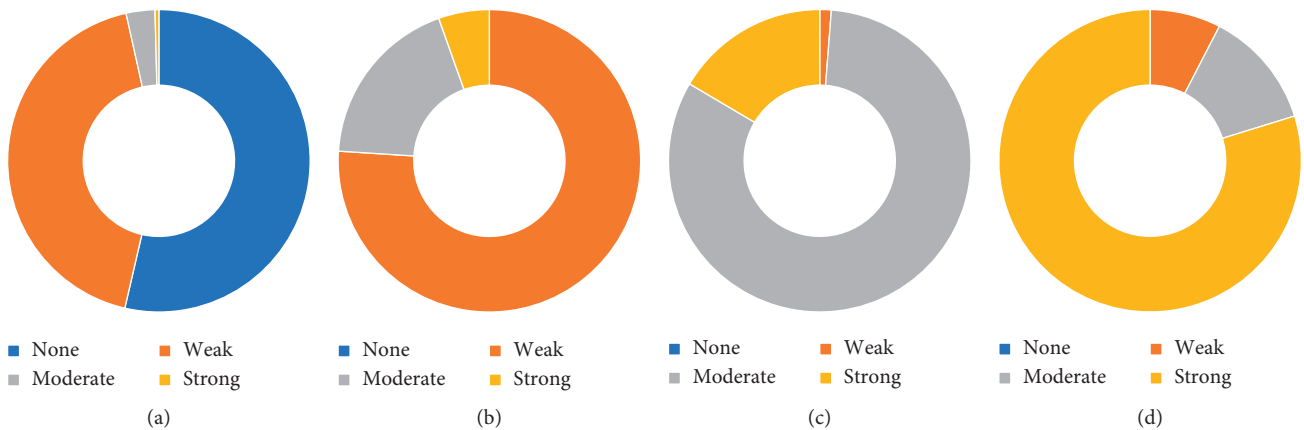


FIGURE 9: The probability of each rank for (a) No. 1, (b) No. 8, (c) No. 17, and (d) No. 22 testing sample.

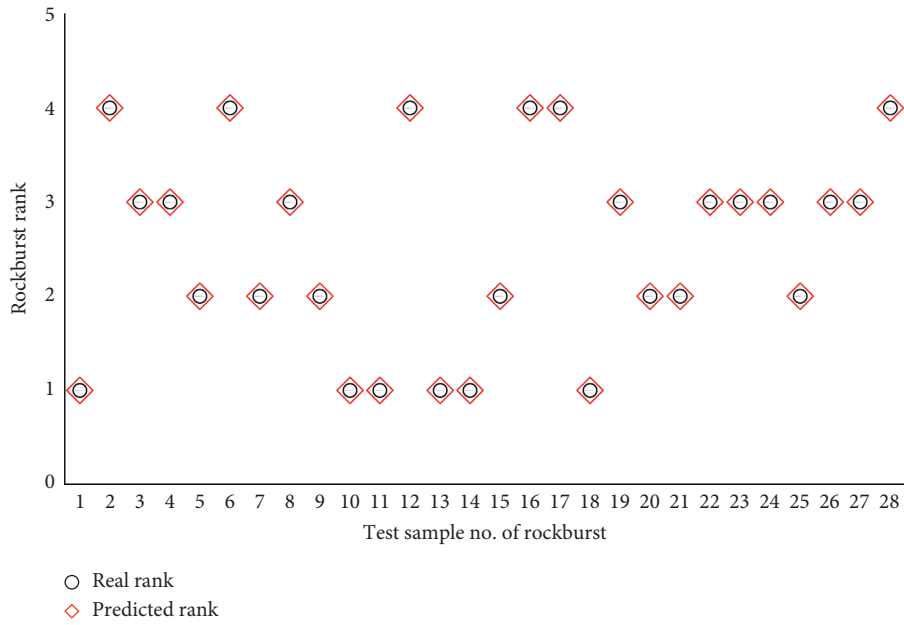


FIGURE 10: The predicted results and their comparison using CNN for the testing samples.

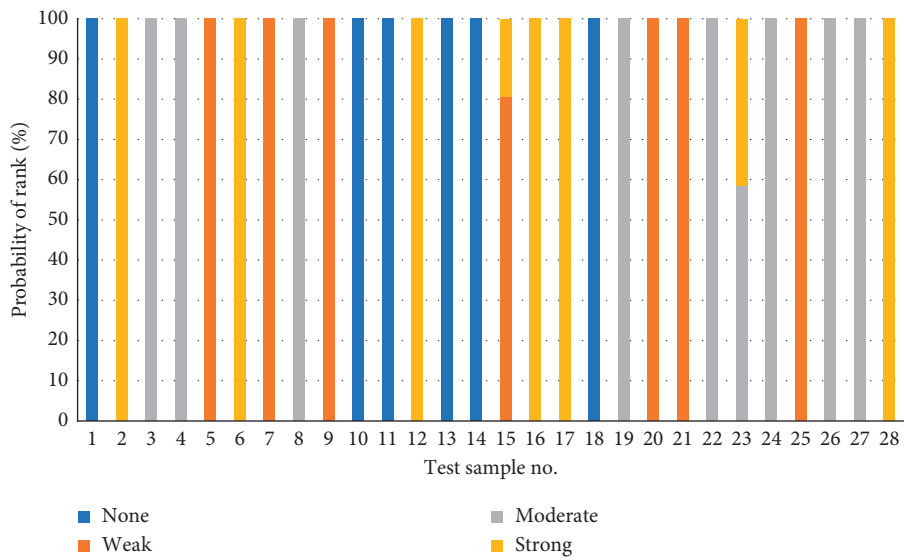


FIGURE 11: The rank probability of rockburst for the testing samples using CNN model.

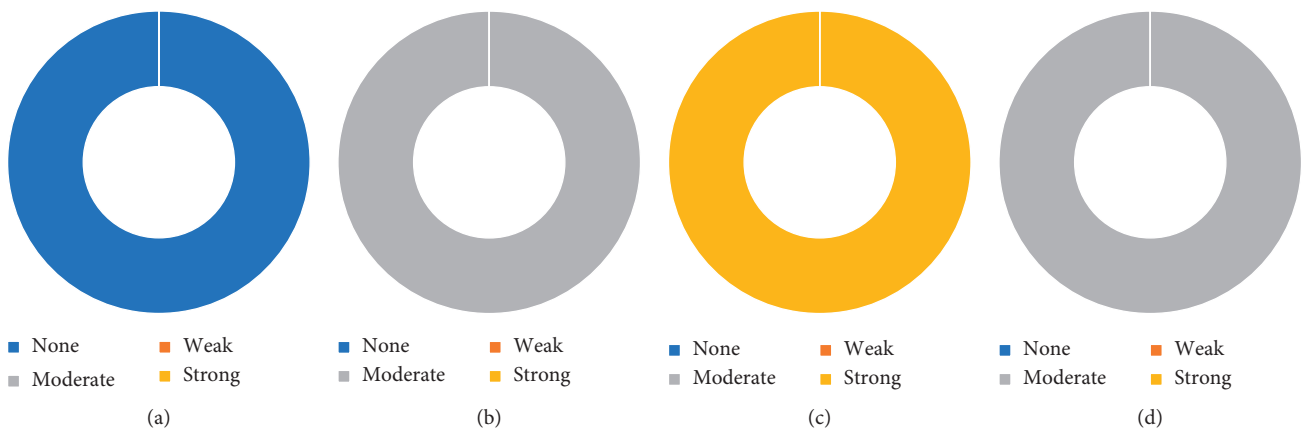


FIGURE 12: The probability of each rank for the (a) No. 1, (b) No. 8, (c) No. 17, and (d) No. 22 testing sample using CNN.



TABLE 3: Comparison with the empirical model.

Sample no.	Russenes criterion	Rock brittleness coefficient criterion	Elastic energy index	NN	CNN	Real
1	2	2	3	1	1	1
2	3	4	3	4	4	4
3	3	3	3	3	3	3
4	3	3	3	3	3	3
5	2	4	2	2	2	2
6	4	4	3	4	4	4
7	4	3	3	2	2	2
8	4	3	2	2	3	3
9	4	2	2	2	2	2
10	4	2	2	1	1	1
11	4	1	1	1	1	1
12	4	4	3	4	4	4
13	4	4	1	1	1	1
14	1	3	2	1	1	1
15	3	3	3	2	2	2
16	2	3	3	4	4	4
17	3	3	3	3	4	4
18	1	3	2	1	1	1
19	3	3	3	3	3	3
20	3	2	3	2	2	2
21	3	3	3	2	2	2
22	3	3	3	4	3	3
23	3	3	3	3	3	3
24	2	3	3	3	3	3
25	3	2	3	2	2	2
26	3	2	3	3	3	3
27	4	3	2	3	3	3
28	4	3	3	4	4	4
Correct rate (%)	42.86	53.57	39.29	89.29	100	—

#### 4. Conclusions

It is critical to evaluate rationally and efficiently the potential rank of rockburst for avoiding and preventing the disaster of rockburst. In this study, a data-driven model was developed to evaluate the rank of rockburst and its probability of corresponding rank using deep learning. Deep learning was adopted to build the relationship between the rank of rockburst and its influence factors based on the collected datasets. The developed method was used to predict the testing samples and compared with the other method. The results showed that the data-driven model is reasonable and feasible to rock engineering with the increasing of rock case history and data. It provides a scientific, promising, and rational way to evaluate the potential of rockburst for rock underground excavation. The following conclusions were obtained.

- (1) Rockburst is a complex dynamic phenomenon and engineering disaster. The relationships between rockburst rank and its influence factor are complex, uncertain, and nonlinear. It is difficult to predict the potential of rockburst and understand the mechanism of rockburst using the traditional method such as empirical method, laboratory or in-site test, numerical method and intelligent method, etc.
- (2) With the development of data science and deep learning technology, a data-driven model provides

a good way to utilize lots of data in the case history and laboratory test for mining the complex mechanism and phenomenon of rockburst. Data-driven models can improve our understanding and evaluate comprehensive rockburst models and data.

- (3) A deep learning model (CNN) can represent well the relationship between rockburst rank and its influence factors. It has been proven to be more powerful and flexible than previous models such as the empirical model, numerical model, and physical model. The combination of data and data-driven model based on deep learning offers an exciting new opportunity for expanding our knowledge about rockburst from data.
- (4) The data-driven model provides a promising and challenging approach for understanding rockburst through combining rock mechanics, engineering geology, rockburst model, and deep learning. It also provides a good way to solve and understand the complex rock mechanics issue in the field of rock engineering.
- (5) The data-driven model is a black box and of poor interpretability. More data and expertise can improve the interpretation of the model.

## Data Availability

All data and models generated or used during the study can be obtained if the reader is required.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] N. Barton, R. Lien, and J. Lunde, "Engineering classification of rock masses for the design of tunnel support," *Rock Mechanics Felsmechanik Mecanique des Roches*, vol. 6, no. 4, pp. 189–236, 1974.
- [2] C. H. Dowding and C.-A. Andersson, "Potential for rock bursting and slabbing in deep caverns," *Engineering Geology*, vol. 22, no. 3, pp. 265–279, 1986.
- [3] J. P. Bardet, "Finite element analysis of rockburst as surface instability," *Computers and Geotechnics*, vol. 8, no. 3, pp. 177–193, 1989.
- [4] W. Müller, "Numerical simulation of rock bursts," *Mining Science and Technology*, vol. 12, no. 1, pp. 27–42, 1991.
- [5] W. D. Ortlepp and T. R. Stacey, "Rockburst mechanisms in tunnels and shafts," *Tunnelling and Underground Space Technology*, vol. 9, no. 1, pp. 59–65, 1994.
- [6] P. K. Kaiser, D. D. Tannant, D. R. McCreath, and R. Jesenak, "Rockburst damage assessment procedure," in *Rock Support in Mining and Underground Construction*, P. K. Kaiser and D. R. McCreath, Eds., pp. 639–647, Balkema, Rotterdam, Netherlands, 1992.
- [7] H. B. Zhao, "Classification of rockburst using support vector machine," *Rock Soil Mech*, vol. 26, no. 4, pp. 642–644, 2005.
- [8] S. Y. Wang, K. C. Lam, S. K. Au, C. A. Tang, W. C. Zhu, and T. H. Yang, "Analytical and numerical study on the pillar rockbursts mechanism," *Rock Mechanics and Rock Engineering*, vol. 39, no. 5, pp. 445–467, 2006.
- [9] S. K. Sharan, "A finite element perturbation method for the prediction of rockburst," *Comput. Struct*, vol. 85, no. 17-18, pp. 1304–1309, 2007.
- [10] G. S. Su, X. F. Zhang, and L. B. Yan, "Rockburst prediction method based on case reasoning pattern recognition," *Journal of Mining and Safety Engineering*, vol. 25, no. 1, pp. 63–67, 2008.
- [11] M. C. He, J. L. Miao, and J. L. Feng, "Rock burst process of limestone and its acoustic emission characteristics under true-triaxial unloading conditions," *International Journal of Rock Mechanics and Mining Sciences*, vol. 47, no. 2, pp. 286–298, 2010.
- [12] W. C. Zhu, Z. H. Li, L. Zhu, and C. A. Tang, "Numerical simulation on rockburst of underground opening triggered by dynamic disturbance," *Tunnelling and Underground Space Technology*, vol. 25, no. 5, pp. 587–599, 2010.
- [13] Q. Jiang, X.-T. Feng, T.-B. Xiang, and G.-S. Su, "Rockburst characteristics and numerical simulation based on a new energy index: a case study of a tunnel at 2,500 m depth," *Bulletin of Engineering Geology and the Environment*, vol. 69, no. 3, pp. 381–388, 2010.
- [14] X. T. Feng, B. R. Chen, C. Q. Zhang, S. J. Li, and S. Y. Wu, *Mechanism, Warning and Dynamical Control of Rockburst Evolution Process*, Science Press, Beijing, China, 2013.
- [15] B.-R. Chen, X.-T. Feng, Q.-P. Li, R.-Z. Luo, and S. Li, "Rock burst intensity classification based on the radiated energy with damage intensity at Jinping II hydropower station, China," *Rock Mechanics and Rock Engineering*, vol. 48, no. 1, pp. 289–303, 2013.
- [16] J. Zhou, X. Li, and X. Shi, "Long-term prediction model of rockburst in underground openings using heuristic algorithms and support vector machines," *Safety Science*, vol. 50, no. 4, pp. 629–644, 2012.
- [17] J. Zhou, X. B. Li, and H. S. Mitri, "Classification of rockburst in underground projects: comparison of ten supervised learning methods," *Journal of Computing in Civil Engineering*, vol. 30, no. 5, Article ID 04016003, 2016.
- [18] N. Li, X. Feng, and R. Jimenez, "Predicting rock burst hazard with incomplete data using Bayesian networks," *Tunnelling and Underground Space Technology*, vol. 61, pp. 61–70, 2017.
- [19] Y. Pu, D. B. Apel, and B. Lingga, "Rockburst prediction in kimberlite using decision tree with incomplete data," *Journal of Sustainable Mining*, vol. 17, no. 3, pp. 158–165, 2018.
- [20] X. Guo, Z. Zhao, X. Gao, Z. Ma, and N. Ma, "Directional sharp-point failure mechanism of rocks surrounding underground circular cavities subjected to large-scale failure," *Mathematical Problem in Engineering*, vol. 2019, Article ID 1415387, 19 pages, 2019.
- [21] G. Wang, X. Yuan, S. Gong, L. Dou, W. Cai, and C. Fan, "Understanding rockburst-generating behaviors and associated seismicity by using a spatial calculation methodology with an energy density index," *Mathematical Problem in Engineering*, vol. 2018, Article ID 6128368, 10 pages, 2018.
- [22] N. G. W. Cook, E. Hoek, J. P. G. Pretorius, W. D. Ortlepp, and M. D. G. Salamon, "Rock mechanics applied to the study of rockbursts," *South African Institute of Mining and Metallurgy*, vol. 66, no. 10, pp. 435–528, 1966.
- [23] J. Zhou, X. Li, and H. S. Mitri, "Evaluation method of rockburst: state-of-the-art literature review," *Tunnelling and Underground Space Technology*, vol. 81, pp. 632–659, 2018.
- [24] A. Sajjad, S. Kourosh, and H. M. Sayyed, "Developing intelligent classification models for rockburst prediction after recognizing significant predictor variables, Section 2: designing classifiers," *Tunnelling and Underground Space Technology*, vol. 84, pp. 522–537, 2019.
- [25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [26] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature Communications*, vol. 5, no. 1, p. 4308, 2014.
- [27] K. T. Schutt, F. Arbabazdah, S. Chmiels, K. R. Muller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, no. 1, Article ID 13890, 2017.
- [28] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning," *Nature Biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.
- [29] M. Reichstein, G. Camps-Valls, B. Stevens et al., "Deep learning and process understanding for data-driven earth system science," *Nature*, vol. 566, no. 7743, pp. 195–203, 2019.
- [30] M. Satar and R. T. Seyed, "Prediction of tunnel convergence using artificial neural networks," *Tunnelling and Underground Space Technology*, vol. 28, pp. 218–228, 2012.
- [31] S. Freitag, B. T. Cao, J. Ninić, and G. Meschke, "Recurrent neural networks and proper orthogonal decomposition with interval data for real-time predictions of mechanised tunnelling processes," *Computers & Structures*, vol. 207, pp. 258–273, 2018.

- [32] X. Gao, M. Shi, X. Song, C. Zhang, and H. Zhang, "Recurrent neural networks for real-time prediction of TBM operating parameters," *Automation in Construction*, vol. 98, pp. 225–235, 2019.
- [33] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [34] S. Haykin, *Neural Networks and Learning Machines*, Pearson Education, Upper Saddle River, NJ, USA, 2009.
- [35] C. Francois, *Deep Learning with Python*, Manning publication, Shelter Island, NY, USA, 2017.
- [36] B. F. Russenes, *Analysis of Rock Spalling for Tunnels in Steep Valley sides*, M.Sc. thesis, Norwegian Institute of Technology, Trondheim, Department of Geology, Trondheim, Norway, 1974.
- [37] G. Brauner, *Rockburst in Coal Mines and Their Prevention*, A. A. Balkema, Rotterdam, Netherlands, 1994.
- [38] Y. A. Tan, "Rockbursting characteristics and structural effects of rock mass," *Science in China Series B Chemistry*, vol. 35, no. 8, pp. 981–990, 1992.
- [39] W. Cai, L. Dou, G. Si, A. Cao, J. He, and S. Liu, "A principal component analysis/fuzzy comprehensive evaluation model for coal burst liability assessment," *International Journal of Rock Mechanics and Mining Sciences*, vol. 81, pp. 62–69, 2016.
- [40] Y. H. Wang, W. D. Li, P. K. K. Lee, Y. Tsui, and L. G. Tham, "Method of fuzzy comprehensive evaluations for rockburst prediction," *Chinese Journal of Rock Mechanics and Engineering*, vol. 17, pp. 493–501, 1998.
- [41] A. Kidybinski, "Bursting liability indices of coal," *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, vol. 18, no. 4, pp. 295–304, 1981.

## Research Article

# A Novel Hybrid CNN-LSTM Scheme for Nitrogen Oxide Emission Prediction in FCC Unit

Wei He,<sup>1,2</sup> Jufeng Li,<sup>2</sup> Zhihe Tang,<sup>2</sup> Beng Wu,<sup>1</sup> Hui Luan,<sup>2</sup> Chong Chen <sup>1</sup>  
and Huaqing Liang <sup>1</sup>

<sup>1</sup>College of Information Science and Engineering, China University of Petroleum-Beijing, Beijing 102249, China

<sup>2</sup>HSE Testing Center, CNPC Research Institute of Safety and Environmental Technology, Beijing 102206, China

Correspondence should be addressed to Chong Chen; [chenchong@cup.edu.cn](mailto:chenchong@cup.edu.cn)

Received 23 April 2020; Revised 22 June 2020; Accepted 28 July 2020; Published 17 August 2020

Guest Editor: Jun Chen

Copyright © 2020 Wei 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.

Fluid Catalytic Cracking (FCC), a key unit for secondary processing of heavy oil, is one of the main pollutant emissions of NO<sub>x</sub> in refineries which can be harmful for the human health. Owing to its complex behaviour in reaction, product separation, and regeneration, it is difficult to accurately predict NO<sub>x</sub> emission during FCC process. In this paper, a novel deep learning architecture formed by integrating Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) for nitrogen oxide emission prediction is proposed and validated. CNN is used to extract features among multidimensional data. LSTM is employed to identify the relationships between different time steps. The data from the Distributed Control System (DCS) in one refinery was used to evaluate the performance of the proposed architecture. The results indicate the effectiveness of CNN-LSTM in handling multidimensional time series datasets with the RMSE of 23.7098, and the R<sup>2</sup> of 0.8237. Compared with previous methods (CNN and LSTM), CNN-LSTM overcomes the limitation of high-quality feature dependence and handles large amounts of high-dimensional data with better efficiency and accuracy. The proposed CNN-LSTM scheme would be a beneficial contribution to the accurate and stable prediction of irregular trends for NO<sub>x</sub> emission from refining industry, providing more reliable information for NO<sub>x</sub> risk assessment and management.

## 1. Introduction

Fluid Catalytic Cracking (FCC) is one of the most important technologies for secondary processing of heavy oil in refining and chemical enterprises [1]. Catalytic cracking reaction and catalyst regeneration are the main chemical processes of FCC. In the catalytic cracking reaction, crude oil is transformed into gasoline and diesel under catalysis during which 40%–50% of nitrogen in feedstock is transferred to coke and deposited on the catalyst [2–4]. Then, coke-covered spent catalysts are burned in the reaction regenerator for catalyst active regeneration, heat balance, energy recovery, and stable operation. During catalyst regeneration process, about 90% of the nitrogen in coke is converted into N<sub>2</sub> and the rest into NO<sub>x</sub> and other reduced nitrogen compounds (NH<sub>3</sub>, HCN, etc.). NO and NO<sub>2</sub> are the most detected NO<sub>x</sub> which have potential risks to human

health. As blood poison, NO would cause hemichypoxia and depress the central nervous system by strongly binding with hemoglobin (HB); NO<sub>2</sub> would cause bronchiectasis (even toxic pneumonia and pulmonary edema) by irritating and corroding the lung tissue [5, 6]. Furthermore, with the development of refining chemical technology, especially catalytic technique, more heavy oil with high percentage of nitrogen (such as residual oil and wax oil) were utilized. Therefore, it is urgent to accurately predict the NO<sub>x</sub> produced during FCC process so as to effectively optimize the noxious gas discharged into the environment subject to the technical and economic conditions.

The FCC process is complex both from the modelling and from the control points of view [7–11]. Fortunately, many researchers have explored and developed semiempirical models, lumped kinetic models, and molecular-based kinetic models [12]. A comprehensive review on FCC

process modelling, simulation, and control was reported by [13]. Many research studies have been conducted using different models for modelling, controlling, and optimizing the FCC process with promising results [14–16]. With the development statistical learning theory, machine learning algorithms have proved effective methods for simulating natural systems in capturing nonlinearity with limited computation costs. The application of machine learning algorithms in the field of FCC is still at an early stage. Michalopoulos et al. [17] and Bollas et al. [18] proved the applicability of Artificial Neural Networks (ANN) in predicting the FCC products and optimized the operation conditions by developing ANN models for determining the steady-state behaviour of industrial FCC units. Zhang [19] established a  $\text{NO}_x$  emission model by Support Vector Machine (SVM) and further optimized the parameters with an improved adaptive genetic algorithm. Gu et al. [20] constructed a boiler combustion model on the basis of Least Support Vector Machines (LSSVM) and successfully forecasted  $\text{NO}_x$  emissions and other parameters which were verified by field data. Recent advantages in artificial intelligence (AI) (lead by deep learning) offered powerful predictive tool for effectively solving the highly complex chemical processes (such as FCC). Shao et al. proposed a new fault diagnosis method of chemical process by combining LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) [21]. Yang et al. integrated deep neural network (“black box model”) with lumped kinetic model (white box model) to create a novel “gray box model” for improving the efficiency and accuracy of simulating FCC process [22]. However, to the best of the authors’ knowledge, there are few research studies using deep learning algorithms for predicting the  $\text{NO}_x$  emission in FCC unit. Some research studies of pollution emission problems have been conducted in power plants [23]. Compared to power plants, the FCC process is relatively complex with more factors involved. Therefore, it is of great difficulty to predict  $\text{NO}_x$  emissions in FCC units.

In this paper, a novel deep learning architecture for predicting  $\text{NO}_x$  emissions in the FCC Unit is proposed. The deep learning architecture is formed by integrating Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) (refer as CNN-LSTM hereafter) with CNN layers extracting features among several variables and LSTM layers learning time series dependencies. The data from the Distributed Control System (DCS) in one refinery was used to demonstrate the performance of CNN-LSTM in the FCC unit. The main contributions of this paper are (1) the proposal of a novel hybrid CNN-LSTM scheme which is able to extract feature among different data sequences and the features between different time steps; (2) the application of the proposed scheme to predict  $\text{NO}_x$  emission during the FCC process with significant results.

## 2. Deep Learning Algorithms

**2.1. Convolutional Neural Network Model (CNN).** CNN is a special kind of neural network which is widely used in the field of image processing [24, 25]. In CNN, a feature map is

used to extract the features from the input of the previous layer with a convolution operation. The pooling layer is used to reduce the computational complexity by reducing the size of the output from one stack layer to the next and at the mean time preserving important information. There are many pooling techniques available, among which maximum pooling is mainly used for pooling windows that contain maximum elements. The convolution layer provides the outputs of the pooling layer and maps it to the next layer. The last layer of CNN is usually fully connected for data classification. Figure 1 shows the basic architecture of CNN.

In neural network training, the accuracy and training speed could be affected by many factors [26]. For example, number of input layer nodes, number of hidden layers, number of hidden layer nodes, and the Internal Covariate Shift (ICS). That is to say, the inputs of the current layer would change according to the variation of parameters in the previous layers which would lead to more training time. In addition, if the inputs are distributed in ranges where the gradient of activation function is low, the ICS would cause the disappearance of gradient. In order to solve these problems, a Dropout method was included as follows.

Dropout (Figure 2) was first proposed by Hinton et al. in order to reduce the overfitting problem in neural networks [27–32]. In dropout procedure, the local feature dependency of the model will be reduced with a probability of  $P$ , and consequently, the generalization ability of the model will be improved effectively.

**2.2. Long Short-Term Memory Network (LSTM).** RNN is a kind of deep neural network which is specially used to process sequential data [33]. Compared with the traditional ANN, the characteristic of RNN is the inclusion of dependencies through time. The basic structure of a RNN is shown in Figure 3.

$$\alpha_t = b + Wh_{t-1} + Ux_t, \quad (1)$$

$$h_t = \tan(\alpha_t), \quad (2)$$

$$o_t = c + Vh_t, \quad (3)$$

$$\hat{y}_t = \text{soft max}(o_t). \quad (4)$$

The left side and the right side in the architecture are the folded form and the expanded form, respectively. In equations (1)~(4),  $t$  is time,  $x$  is the sequence of input data,  $h$  is the hidden layer state of the network,  $o$  is the output vector of the neuron,  $U$  is the parameter matrix from the input layer to the hidden layer,  $V$  is the parameter matrix from the hidden layer to the output layer,  $W$  is the parameter matrix between the hidden layers at different times, and  $\hat{y}_t$  represents the probability output of the predicted value after normalization. All the parameter matrices are shared matrix of the hidden states at different times.

In order to solve the disappearance or explosion of gradient during training RNN, researchers proposed LSTM



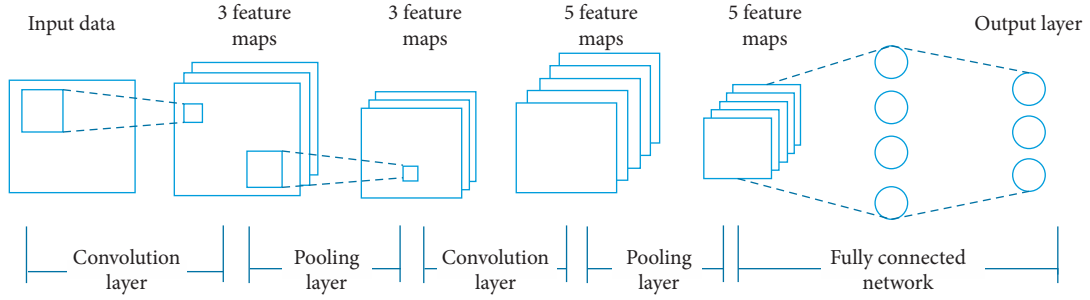


FIGURE 1: The architecture of the CNN.

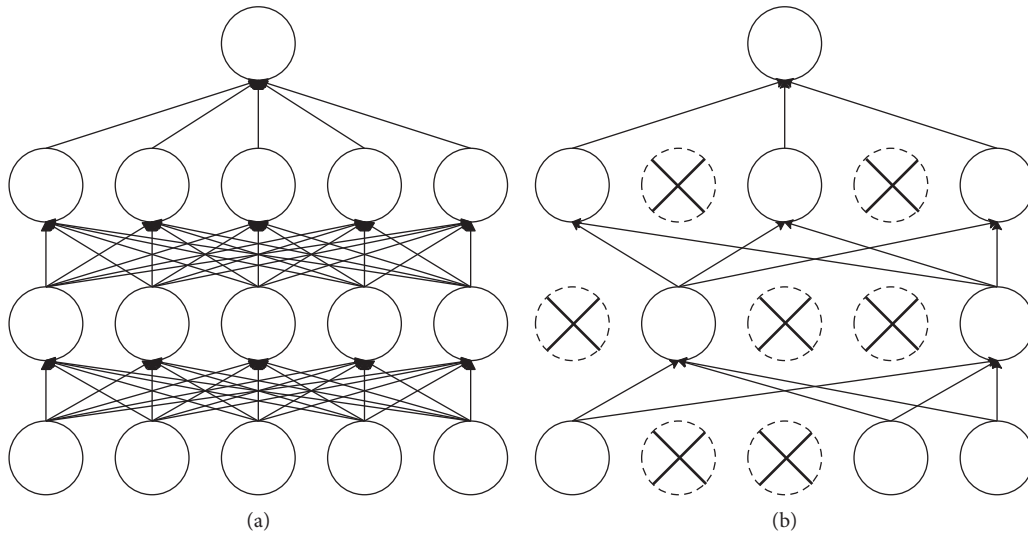


FIGURE 2: Dropout schematic: (a) Standard neural network; (b) after applying dropout.

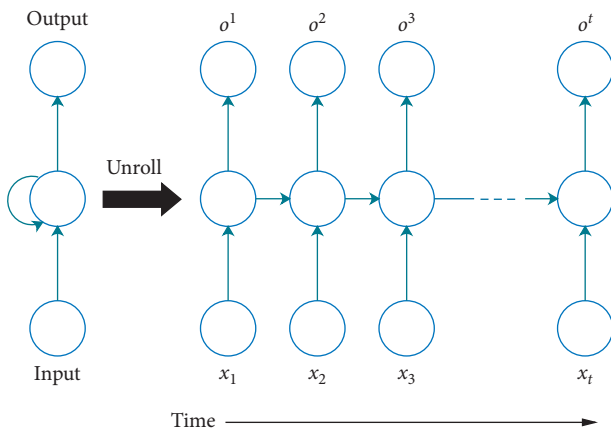


FIGURE 3: The architecture of the RNN.

by introducing gate mechanism in RNN [34, 35]. The gate mechanism is composed of input gate, output gate, and forgetting gate. As a special type of RNN, the neurons in the LSTM model are connected to each other in a directed cycle. The basic structure of LSTM is shown in Figure 4.

The LSTM model saves long-term dependencies using three different gates in an effective way. The structure of LSTM (shown in Figure 4) is similar to RNN. LSTM uses

three gates to regulate and preserve information into every node state. The explanation of LSTM gates and cells is provided in equations (5)~(8):

$$\text{Input Gate } \text{In}_t = \sigma(W_{\text{in}} \cdot [h_{st} - 1], x_t + b_{\text{in}}), \quad (5)$$

$$\text{Memory Cell } C_t = \tanh(W_c \cdot [h_{st} - 1], x_t + b_e), \quad (6)$$

$$\text{Forget Gate } f_t = \sigma(W_f \cdot [h_{st} - 1], x_t + b_f), \quad (7)$$

$$\text{Output Gate } O_t = \sigma(W^o \cdot [h_{st} - 1], x_t + b^o), \quad (8)$$

where  $b$  represents the bias vector;  $W$  is weight matrix;  $x_t$  is the input vector at time  $t$ ; and  $\text{In}$ ,  $f$ ,  $C$ , and  $O$  represent input, forget, cell memory, and output gates, respectively.

### 3. CNN-LSTM

Due to the characteristics of CNN and LSTM, a common thought to combine the advantages is to integrate CNN and LSTM. In this study, a new deep learning scheme was proposed by integrating CNN and LSTM. Two layers of CNN were used to ensure the correlation and effective extraction of multidimensional data. The feature sequence from the CNN layer was considered as the input for LSTM.

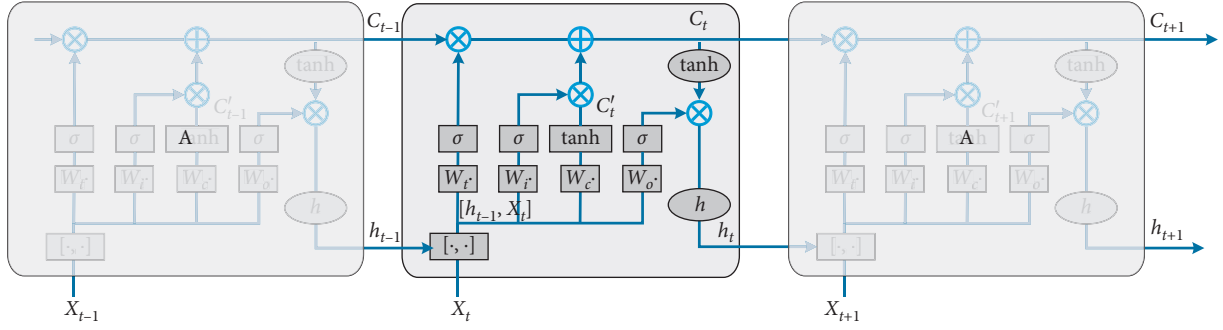


FIGURE 4: The architecture of the LSTM.

The time dependencies were further extracted in the LSTM layer. Three fully connected layers existed in the architecture which refer to FC1, FC2, and FC3. FC1 and FC2 are used to obtain the features extracted by the CNN layer, and FC3 is used to conduct the final data prediction. Figure 5 shows the architecture of the proposed CNN-LSTM.

3.1. CNN Layer. The input data ( $\text{train\_x}$ ) and output data ( $\text{train\_y}$ ) are defined as follows:

$$\text{train\_x}_i = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1q} \\ x_{21} & x_{22} & \cdots & x_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ x_{p1} & x_{p2} & \cdots & x_{pq} \end{bmatrix},$$

$$\text{train\_y}_i = [\text{var}_{t-p+3}, \text{var}_{t-p+4}, \dots, \text{var}_{t+1}, \text{var}_{t+2}, \dots, \text{var}_{t+q}]^T, \quad (9)$$

where  $p$  represents time step and  $q$  represents data features.

The  $i$ th sample from the training set is fed into the network. In the first convolution layer ( $1^{\text{st}}\text{ConV}$ ), the convolution kernel size, number, and step length are denoted as  $\text{filter\_size}=(m, n)$ ,  $\text{filter\_num}$ , and  $\text{strides}$ , respectively.

The  $j$ th convolution kernel  $W_j$  is defined as follows:

$$W_j = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}. \quad (10)$$

The algorithms between  $j$ th convolution kernel  $W_j$  and input  $\text{train\_x}_i$  could be described as follows:

$$W_j \odot \text{train\_x}_i = \text{featureMap\_x}_i = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1b} \\ x_{21} & x_{22} & \cdots & x_{2b} \\ \cdots & \cdots & \cdots & \cdots \\ x_{a1} & x_{a2} & \cdots & x_{ab} \end{bmatrix}. \quad (11)$$

The operation for convolution layer is denoted as  $\odot$ , where

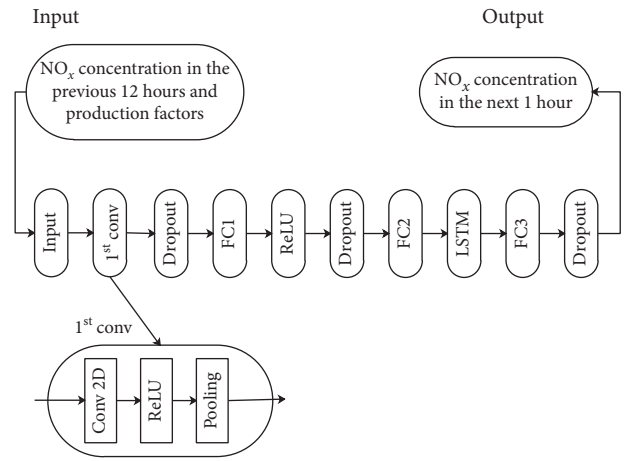


FIGURE 5: Architecture of the proposed CNN-LSTM.

$$a = \frac{p - m + 2 \times \text{padding}}{\text{strides}} + 1, \quad (12)$$

$$b = \frac{q - n + 2 \times \text{padding}}{\text{strides}} + 1.$$

The element  $x$  in the feature map is obtained through multiplying  $W_j$  by Receptive Field, which is recorded as follows:

$$\text{train\_x}_i\text{-field} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}. \quad (13)$$

$x \in \text{featureMap\_x}_i = W_j \odot \text{train\_x}_i\text{-field} = \sum_{k=1}^m \sum_{l=1}^n w_{kl} x_{kl}$ , where  $\odot$  means multiply the elements.

$1^{\text{st}}\text{ConV}$  is calculated as

$$1^{\text{st}}\text{ConV}_{x_i} = W \odot \text{train\_x}_i = [\text{featureMap\_x}_1, \text{featureMap\_x}_2, \dots, \text{featureMap\_x}_k], \quad (14)$$

where  $W = [W_1, W_2, \dots, W_k]$

ReLU is used as the activation function:



$$f_{\text{ReLU}} = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \quad (15)$$

The output of the convolutional layer is nonlinear mapping by the activation function. In pooling layer, the data are compressed and recorded as  $\text{pooling\_size} = (m', n')$ .

For every feature map,

$$x_{i\text{-pool}} = f_{\text{pool}}(\text{featureMap}_x) = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1b'} \\ x_{21} & x_{22} & \cdots & x_{2b'} \\ \cdots & \cdots & \cdots & \cdots \\ x_{a'1} & x_{a'2} & \cdots & x_{a'b'} \end{bmatrix}, \quad (16)$$

where  $a' = a/m', b' = b/n'$

Thus, the  $i$ th sample after convolutional, activation, and pooling layer is

$$X_{\text{pool}_1} = (x_{1\text{-pool}}, x_{2\text{-pool}}, \dots, x_{k\text{-pool}}). \quad (17)$$

The convolutional, activation, and pooling in 2ndConV are similar to those in 1stConV.

Dropout is denoted as  $\lambda$ ;  $\lambda$  takes the value between 0 and 1, which means the percentage of the data that should be discarded. For instance, dropout (0.5) means that 50% of neuron data are discarded randomly.

FC layer dense ( $\alpha$ ) is the output data in the last dimension. For the above input type [none,  $a', b', k$ ], only the last dimension [none,  $a', b', \alpha$ ] is changed after full connection.

Transform [samples, height, width, channels] to [samples, timesteps, features], and then feed them in the LSTM layer. The modular construction of LSTM is shown as follows, in which forget, input, and output gates are included.

**3.2. LSTM Layer.** The forget gate is expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (18)$$

where  $W_f$  represents the weight matrix for the forget gate;  $[h_{t-1}, x_t]$  means concatenation of  $h_{t-1}$  and  $x_t$ ;  $b_f$  represents the offset of the forget gate; and  $\sigma$  represents the sigmoid function. The dimensionality of input layer, hidden layer, and cell state is  $d_x, d_h$ , and  $d_c$ , respectively. In general,  $d_c = d_h$ , the dimensionality of weight matrix for the forget gate, and  $W_f$  is  $d_c \times (d_h + d_x)$ . Actually, the weight matrix ( $W_f$ ) is combined by two matrices  $W_{fh}$  (initem:  $h_{t-1}$ ; dimensionality:  $d_c \times d_h$ ) and  $W_{fx}$  (initem:  $x_t$ ; dimensionality:  $d_c \times d_x$ ),  $W_f$  could be written as follows:

$$\begin{aligned} [W_f] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} &= [W_{fh} \quad W_{fx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \\ &= W_{fh}h_{t-1} + W_{fx}x_t + W_{fx}x_t. \end{aligned} \quad (19)$$

Input gate could be expressed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (20)$$

where  $W_i$  represents the weight matrix for the forget gate and  $b_i$  represents the offset of the input gate.

The cell state for input description is calculated by the last output data and the current input data:

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \quad (21)$$

The current cell state ( $C_t$ ) is as follows:

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{c}_t, \quad (22)$$

where the last cell state ( $C_{t-1}$ ) is multiplied by forget gate ( $f_t$ ) according to different element and the current input cell state ( $\tilde{c}_t$ ) is multiplied by input gate ( $i_t$ ) according to different element.

The new cell state ( $C_t$ ) is established by current memory ( $\tilde{c}_t$ ) and long-term memory  $C_{t-1}$ . On one hand, due to the mechanism of forget and input gate, the new cell state store information from a long time ago or forget the irrelevant content. On the other hand, the output gate controls the effect of long-term memory on current output:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \quad (23)$$

The final output of LSTM is decided by the output gate and cell state (equation (29)):

$$h_t = o_t \circ \tanh c_t. \quad (24)$$

**3.3. Realization of CNN-LSTM.** The CNN-LSTM was realized in Keras using TensorFlow backend based on Figure 5 and the theory described in the previous sections (shown in Algorithm 1). After normalization, the training data (train\_x, train\_y) was fed into the constructed CNN model (1<sup>st</sup> Conv\_model) to train the parameters with loss function (loss\_function which is "mae" in our case) and optimizer (optimizer, which is "adam" in our case). The feature map of CNN was then extracted and reshaped to train the LSTM layer.

## 4. Experiments

**4.1. Datasets.** Several key production factors that affect the nitrogen oxide concentration in the plant were selected from 276 kinds of production factors of catalytic cracking unit. By inquiring experts, the key factors of production include nitrogen content in raw materials, process control parameters of reactor (FCC reaction temperature, catalyst/oil ratio, and residence time), the regeneration process control parameters (regeneration way, dense bed temperature, oxygen content in furnace, and carbon monoxide concentration), and catalyst species (platinum CO combustion catalyst and nonplatinum CO combustion catalyst).

A total of  $2.592 \times 10^5$  of samples collected in half a year were divided into training and validation sets with the proportion of 70% and 30%, respectively. As shown in Table 1, the key production factors were used as input data and the  $\text{NO}_x$  emission were used as labels.

In order to eliminate the dimensional effects among different variables, the original data was standardized using the MinMaxScaler function in Python (equations (25) and (26)):

```

Input: train_x, train_y
Hyper-parameters: filters, kernel_size, pool_size, batch_size, rate
Initialize ()
Normalization (train_x, train_y)
//The first convolution layer
1st ConV_model= Sequential ([Convolution2D (filters, kernel_size, name = "Conv2D_1"), MaxPooling2D (pool_size), Flatten (),
Dense (units, activation), Dropout (rate), Dense (units, activation)])
1st ConV_model.compile (loss_function, optimizer)
1st ConV_model.fit (train_x, train_y, epochs, batch_size)
//Extract the feature map
1st ConV_feature_model= Model (inputs, 1st ConV_model.get_layer ("Conv2D_1").output)
1st ConV_feature_output = 1st ConV_feature_model.predict (train_x)
//LSTM layer
reshape (1st ConV_feature_output)
LSTM_model= Sequential (LSTM (units, activation, recurrent_activation), Dense (units, activation))
LSTM_model.compile (loss_function, optimizer)
LSTM_model.fit (1st ConV_feature_output, train_y, epochs, batch_size)

```

ALGORITHM 1: Pseudocode of CNN-LSTM.

$$X_{\text{std}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \quad (25)$$

$$X_{\text{scaled}} = \frac{X_{\text{std}}}{(\max - \min) + \min}, \quad (26)$$

where  $X_{\max}$  and  $X_{\min}$  are the maximum and minimum values of the data and max and min are maximum and minimum values of the zoom range. In addition, the problem of time prediction was reconstructed into supervised learning.

#### 4.2. Hyperparameters

**4.2.1. CNN.** The hyperparameters in RNN mainly contain weight initialization, learning rate, activation function, epoch numbers, iteration times, etc. Several important hyperparameters include number of convolution layers, number of convolution kernels, and size of convolution kernel are discussed in this study.

**4.2.2. LSTM.** Long-short term memory (LSTM) is a kind of RNN, in which tanh could be replaced by sigmoid activation function, resulting in faster training speed. In LSTM, Adam was used as an optimizer, MSE was used as a loss function, and identity activation function was used to complete the weight initialization. The hyperparameters in LSTM mainly contain number of hidden layer nodes and the number of batch sizes. The number of hidden layer nodes in LSTM have direct influences on the learning results by affecting the ability of nonlinear mapping which is the same as in Feedforward Neural Networks. The batch size have an influence on the computation costs and the learning accuracy by affecting amount of data used for updating the gradient.

**4.2.3. CNN-LSTM.** As a neural network model combined CNN with LSTM, the hyperparameters of CNN-LSTM is

basically the same with CNN and LSTM which mainly include learning rate  $\eta$ , regularization parameter  $\lambda$ , the number of neurons in each hidden layer (such as the full-connected layer and the number of neurons in LSTM), batch size, convolution kernel size, neuron activation function, pool layer size, and dropout rate. All the related hyperparameters were investigated and analysed in Section 5.

**4.3. Performance Criteria.** The performances of different algorithms were evaluated by the Root Mean Square Error (RMSE) (equation (27)) and the coefficient of determination ( $R^2$ ) (equation (28)) [36]. The RMSE value reflects the discrete relationship between predicted and observed values:

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^N (o_n - p_n)^2}{N}}, \quad (27)$$

where  $N$  is the data length,  $o_n$  is the  $n$ th observed value, and  $p_n$  is the  $n$ th predicted value.

The  $R^2$  value reflects the accuracy of the model which ranges from 0 to 1 with 1 denotes perfect match:

$$R^2 = \frac{\text{SSR}}{\text{SST}} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (28)$$

where  $\hat{y}_i$  represents the predicted value,  $\bar{y}$  is the average value, and  $y_i$  is the observed value.

## 5. Results and Discussion

**5.1. CNN-LSTM.** The hyperparameters mentioned above were determined by the trial-and-error method. RMSE and  $R^2$  were considered as objective function to optimize the size and number of convolution kernel, the number of batch size, the number of convolution layers, and the probability of dropout. The results shown in Figure 6 indicate the process of optimizing hyperparameters for the proposed method.

TABLE 1: The setting of input and labels.

Data types	Key production factor types	Parameters	Unit
Input data	Raw materials	Nitrogen content	%
	Process control parameters of reactor	FCC reaction temperature	°C
		Catalyst/oil ratio	%
		Residence time	Second (s)
		Regeneration way	
	Regeneration process control parameters	Dense bed temperature	°C
		Oxygen content in furnace	%
		Carbon monoxide concentration	% mg/m <sup>3</sup>
		Platinum CO combustion catalyst	
	Catalyst species	Nonplatinum CO combustion catalyst	—
Labels	NO <sub>x</sub> emission data		mg/m <sup>3</sup>

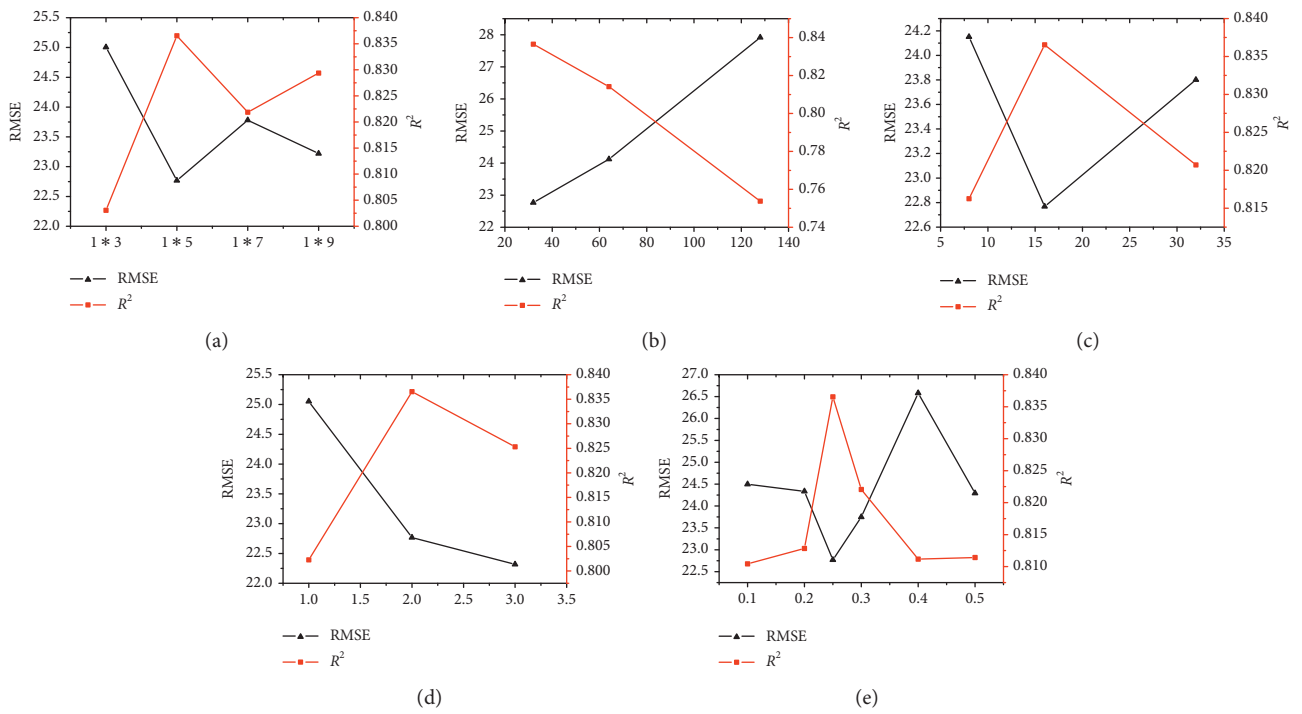


FIGURE 6: Optimization of hyperparameters for CNN-LSTM. (a) Size of convolution kernels. (b) The number of batch size. (c) The number of convolution kernels. (d) The number of convolution layers. (e) The probability of dropout.

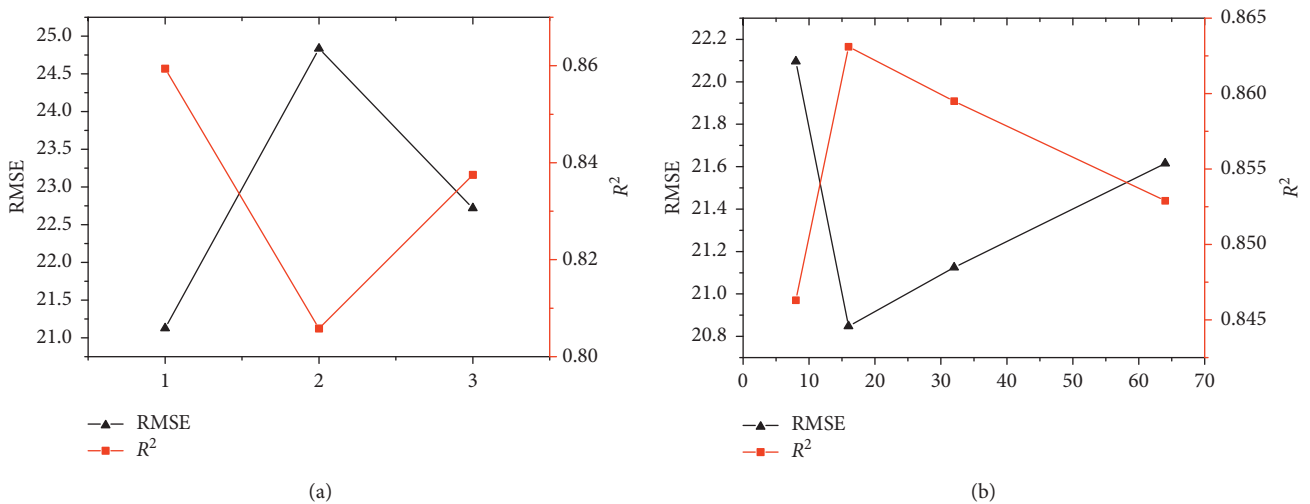


FIGURE 7: Continued.

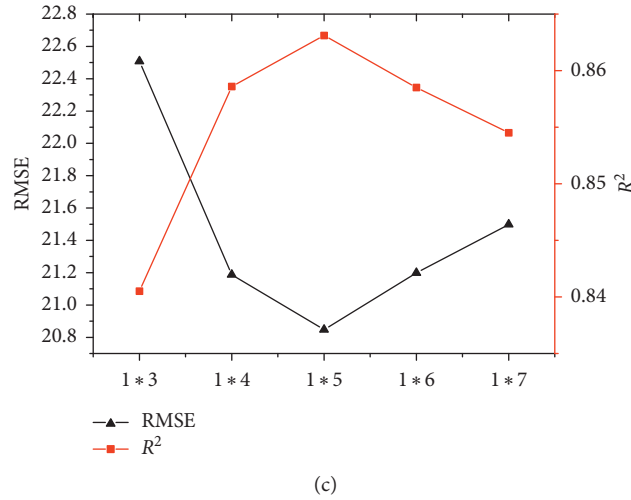


FIGURE 7: Optimization process of hyperparameters for CNN. (a) The number of convolution layers. (b) The number of convolution kernels. (c) Size of convolution kernel.

The network structure adopts two convolutional parts as the CNN layer; the kernel size is  $1 \times 5$  for the first and second CNN layer. Each convolution layer is followed by a Rectified Linear Unit (ReLU) layer (equation (29)) and a maximum pooling layer. The output of CNN part is a 32-dimensional vector after operations. All the vectors form a sequence and feed into the LSTM layer:

$$a_{i,j,k} = \max(z_{i,j,k}, 0), \quad (29)$$

where  $z_{i,j,k}$  is the input of the activation function at location  $(i, j)$  on the  $k$ th channel. ReLU allows neural networks to compute faster than sigmoid or tanh activation functions and train deep network more effectively. In order to train a neural network with strict backpropagation algorithm, the contribution of all samples to the gradient must be considered simultaneously.

With the incorporation of the LSTM network, the proposed CNN-LSTM network can be trained with time series data of FCC unit. A LSTM layer followed by the FC layer is used to assign the predicted value to each frame in the sequence.

The output of the CNN layer passes through two dropout layers and two FC layers to combine the features extracted by the CNN layer. During the training stage, the dropout layer will randomly remove the connection between the CNN layer and the FC layer in each iteration. In our experiments, we set the dropout rate to an empirical value of 0.25, which has shown effectiveness in performance improvement (the experiment on the dropout rate is shown in Figure 6(e)). The convolution layer, convergence layer, and activation function layer are conducted to map the raw data to the feature space in hidden layer. And the full-connected layer plays the role of “classifier” which

maps the learned feature representation to the memory space of the sample.

**5.2. CNN.** The hyperparameters of CNN were also determined by the trial-and-error method. RMSE and  $R^2$  were considered as an objective function to optimize the size and number of convolution kernel and the number of convolution layers. The results shown in Figure 7 indicate the process of optimizing hyperparameters for CNN from which one can conclude that the optimal values for the number of convolution layers, the number of convolution kernels and the size of convolution kernel are 1, 16, and  $1 \times 5$ .

**5.3. LSTM.** The optimization process of hyperparameters for LSTM was shown in Figure 8. RMSE and  $R^2$  were used as quantitative performance criteria to evaluate the hyperparameters (i.e., the number of hidden layer nodes and the number of batch size). The process and results shown in Figure 8 indicated the optimal values for the number of hidden layer nodes and the number of batch size are 40 and 500, respectively.

**5.4. Experiments on Different Methods.** The accuracy of CNN and LSTM and the proposed CNN-LSTM method for the training stage and validation stage were evaluated by  $R^2$  and RMSE. All methods were well tuned and ten test runs were conducted to eliminate the random errors of each method. The average criteria for each method were calculated to evaluate the performance. The results were presented in Tables 2 and 3 and Figure 9, respectively. Compared with traditional CNN, the proposed CNN-

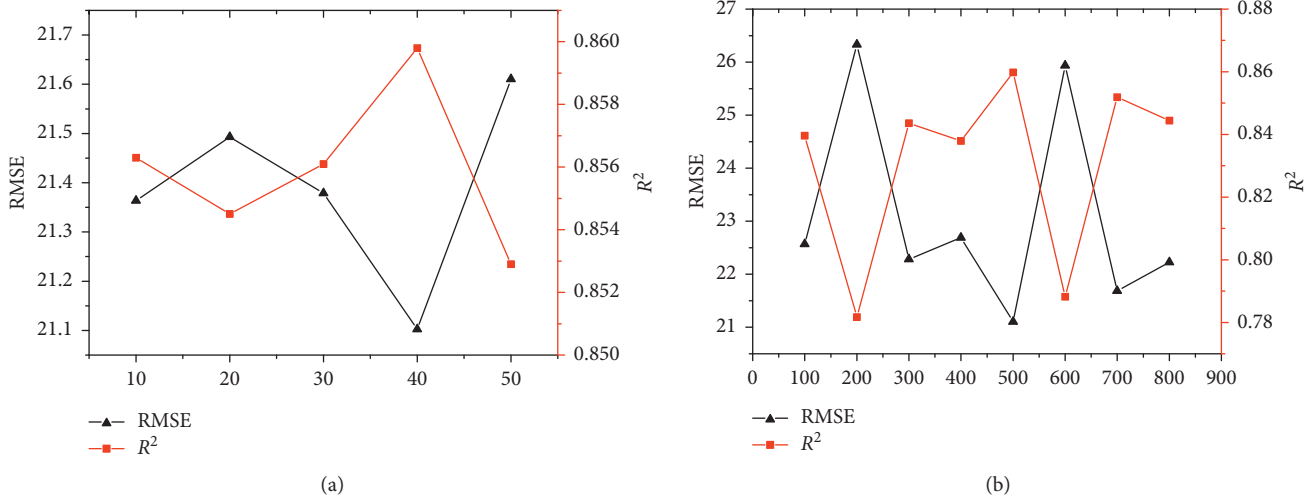


FIGURE 8: Optimization process of hyperparameters for LSTM: (a) number of hidden layer nodes; (b) number of batch sizes.

TABLE 2: The RMSE of different methods.

Test	CNN	LSTM	CNN-LSTM
#1	76.4694	26.3256	22.8220
#2	68.1422	26.0771	22.2389
#3	105.1664	25.6918	22.8374
#4	66.4166	25.4898	26.5900
#5	80.0437	26.0154	23.0910
#6	66.0526	25.8325	23.9463
#7	135.0926	25.7496	27.4446
#8	95.9767	25.7942	21.9647
#9	91.5383	25.8024	23.1579
#10	53.0630	27.0787	22.9959
Average	83.7962	26.2065	23.7089

TABLE 3: The  $R^2$  of different methods.

Test	CNN	LSTM	CNN-LSTM
#1	0.4935	0.7931	0.8222
#2	0.4718	0.7912	0.7923
#3	0.3329	0.7923	0.8060
#4	0.4891	0.7946	0.8367
#5	0.1761	0.7980	0.8358
#6	0.3739	0.7900	0.8014
#7	0.4748	0.7999	0.8130
#8	0.1901	0.7964	0.8448
#9	0.1094	0.7734	0.8452
#10	0.1133	0.5345	0.8328
Average	0.3224	0.7663	0.8237

LSTM was more accurate in  $\text{NO}_x$  emission prediction with average RMSE of 23.7089 and  $R^2$  of 0.8237. The combination of CNN and LSTM integrates the advantages of CNN and LSTM which are capable of extracting the features among different data sequences and the features between different time

steps. The ability of CNN-LSTM is suitable for the characteristics of datasets from refining and chemical enterprises. By describing the local feature relationship under multidimensional and long-term conditions, CNN-LSTM matches the observations better than the other methods.

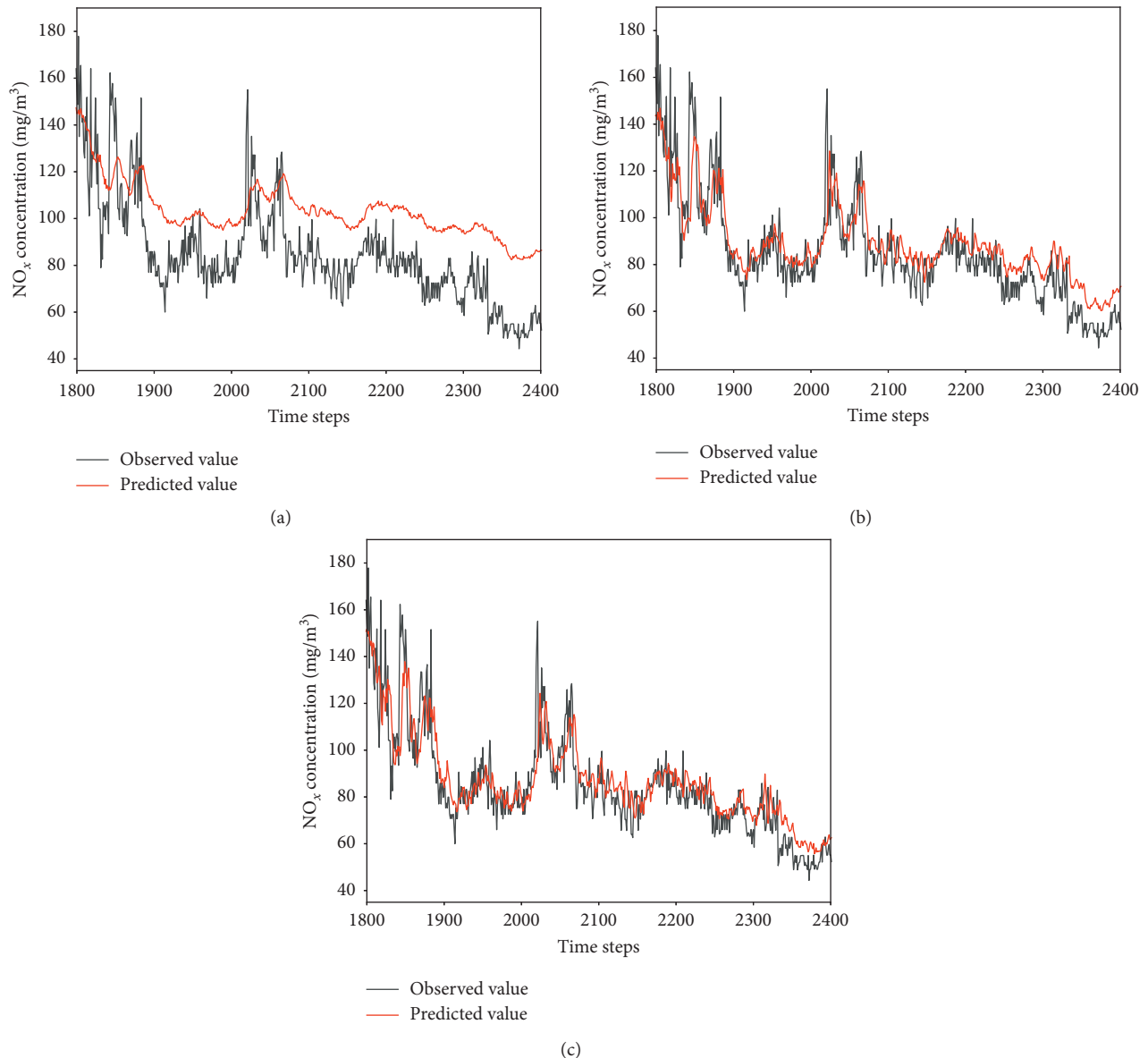


FIGURE 9: Comparisons of the observed and predicted  $\text{NO}_x$  concentrations of (a) CNN; (b) LSTM; (c) CNN-LSTM.

## 6. Conclusions

In this paper, a novel CNN-LSTM scheme combining CNN and LSTM was proposed for the prediction of  $\text{NO}_x$  concentration observed during FCC process. Dropout were introduced to accelerate network training and address the overfitting issue. In our study, a series of hyperparameters (learning rate, regularization parameter, the number of neurons in each hidden layer, small batch data size, convolution kernel size, neuron activation function, pool layer size, and dropout rate) and conditions (raw materials, process control parameters of reactor, regeneration process control parameters, and catalyst species) were selected and optimized. Experiments were conducted to evaluate the proposed scheme with traditional methods (CNN and LSTM) being baseline models. The hyperparameters of all the methods were optimized to obtain the best results. RMSE and  $R^2$  were used to evaluate the performance of different methods. Due to

the capability of extracting features among different data sequences and different time steps, better efficiency and accuracy were obtained by CNN-LSTM than baseline models. This study provides a potential direction of deep learning methods by integrating different architectures for individual advantages. The CNN-LSTM scheme proposed in this paper would be a beneficial contribution to the accurate and stable prediction of irregular trends for  $\text{NO}_x$  emission from refining industry and provided more reliable information for  $\text{NO}_x$  risk assessment and management. Future work will focus on attention and transformer mechanism to obtain better results and explore the application of the proposed scheme on other datasets.

## Data Availability

All data and program files included in this study are available from the corresponding author upon request.



## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by CNPC Basic Research Fund Projects “Research on Prediction and Early Warning System of Air Pollution Emission in FCC and New Control Model” (no. 2017D-5008) and Science Foundation of China University of Petroleum, Beijing (nos. 2462018YJRC007 and 2462020YXZZ025).

## References

- [1] L. A. Lacijan, M. W. Schnaith, P. J. Van Opdorp et al., “FCC refinery solutions for the European market,” in *Petroleum Technology Quarterly*, Palo Alto, CA, USA, 2002.
- [2] I. V. Babich, K. Seshan, and L. Lefferts, “Nature of nitrogen specie in coke and their role in NO<sub>x</sub> formation during FCC catalyst regeneration,” *Applied Catalysis B: Environmental*, vol. 59, no. 3-4, pp. 205–211, 2005.
- [3] P. Li, L. K. Zeng, H. Wang, and X. S. Cheng, “Technological feasibility for controlling NO<sub>x</sub> emission of ceramic furnace,” *Chinese Academy of Sciences*, vol. 22, no. 3, pp. 35–42, 2015.
- [4] J. L. Lin, X. B. Hua, Y. Q. Wu, and C. Yang, “Generation and control of nitrogen oxides in flue gas of catalytic plant regeneration,” *Environmental Protection in Petrochemical Industry*, vol. 28, no. 1, pp. 34–38, 2005.
- [5] R. Ehrlich and M. C. Henry, “Chronic toxicity of nitrogen dioxide,” *Archives of Environmental Health: An International Journal*, vol. 17, no. 6, pp. 860–865, 1968.
- [6] J. Buick, R. Lowry, and T. R. Magee, “Pulmonary aspects of nitrogen dioxide toxicity,” *Resources, Conservation and Recycling*, vol. 27, no. 1-2, pp. 89–97, 1999.
- [7] K. Sertić-Bionda, Z. Gomzi, and M. Mužić, “Modeling of gas oil catalytic cracking in a fixed bed reactor using a five-lump kinetic model,” *Chemical Engineering Communications*, vol. 197, no. 3, pp. 275–288, 2009.
- [8] C. Jia, S. Rohani, and A. Jutan, “FCC unit modeling, identification and model predictive control, a simulation study,” *Chemical Engineering and Processing: Process Intensification*, vol. 42, no. 4, pp. 311–325, 2003.
- [9] P. Kiss and F. Szeifert, “Coordinated control of a fluid catalytic cracking unit,” *Chemical Engineering & Technology*, vol. 21, no. 6, pp. 515–521, 1998.
- [10] M. V. Cristea, Ș. P. Agachi, and V. Marinoiu, “Simulation and model predictive control of a UOP fluid catalytic cracking unit,” *Chemical Engineering and Processing: Process Intensification*, vol. 42, no. 2, pp. 67–91, 2003.
- [11] X. Dupain, E. D. Gamas, R. Madon, C. P. Kelkar, M. Makkee, and J. A. Moulijn, “Aromatic gas oil cracking under realistic FCC conditions in a microriser reactor,” *Fuel*, vol. 82, no. 13, pp. 1559–1569, 2003.
- [12] I. S. Nam and J. R. Kittrell, “Use of catalyst coke content in deactivation modeling,” *Industrial & Engineering Chemistry Process Design and Development*, vol. 23, no. 2, pp. 237–242, 1984.
- [13] C. I. C. Pinheiro, J. L. Fernandes, L. Domingues et al., “Fluid catalytic cracking (FCC) process modeling, simulation, and control,” *Industrial & Engineering Chemistry Research*, vol. 51, no. 1, pp. 1–29, 2012.
- [14] D. V. Naik, V. Karthik, V. Kumar, B. Prasad, and M. O. Garg, “Kinetic modeling for catalytic cracking of pyrolysis oils with VGO in a FCC unit,” *Chemical Engineering Science*, vol. 170, no. 12, pp. 790–798, 2017.
- [15] J. Long, M. S. Mao, and G. Y. Zhao, “Model optimization for an industrial fluid catalytic cracking riser-regenerator unit by differential evolution algorithm,” *Petroleum Science and Technology*, vol. 33, no. 13-14, pp. 1380–1387, 2015.
- [16] S. Sun, H. Yan, and F. Meng, “Optimization of a fluid catalytic cracking kinetic model by improved particle swarm optimization,” *Chemical Engineering & Technology*, vol. 43, no. 2, pp. 289–297, 2020.
- [17] J. Michalopoulos, S. Papadokonstadakis, G. Arampatzis, and A. Lygeros, “Modelling of an industrial fluid catalytic cracking unit using neural networks,” *Chemical Engineering Research and Design*, vol. 79, no. 2, pp. 137–142, 2001.
- [18] G. M. Bollas, S. Papadokonstadakis, J. Michalopoulos et al., “Using hybrid neural networks in scaling up an FCC model from a pilot plant to an industrial unit,” *Chemical Engineering and Processing: Process Intensification*, vol. 42, no. 8-9, pp. 697–713, 2003.
- [19] Z. X. Zhang, *Utility Boiler Combustion Optimization Based on Intelligent Optimization Algorithm*, North China Electric Power University, Beijing, China, 2015.
- [20] Y. P. Gu, W. J. Zhao, and Z. S. Wu, “Combustion optimization for utility boiler based on least squares support vector machine,” *Proceedings of the CSEE*, vol. 30, no. 17, pp. 91–97, 2010.
- [21] B. Shao, M. Li, Y. Zhao, and G. Bian, “Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm,” *Mathematical Problems in Engineering*, vol. 2019, Article ID 1032480, 15 pages, 2019.
- [22] F. Yang, C. Dai, J. Tang, J. Xuan, and J. Cao, “A hybrid deep learning and mechanistic kinetics model for the prediction of fluid catalytic cracking performance,” *Chemical Engineering Research and Design*, vol. 155, no. 1, pp. 202–210, 2020.
- [23] G. Miskell, W. Pattinson, L. Weissert, and D. Williams, “Forecasting short-term peak concentrations from a network of air quality instruments measuring PM<sub>2.5</sub> using boosted gradient machine models,” *Journal of Environmental Management*, vol. 242, pp. 56–64, 2019.
- [24] K. Zhao, T. He, S. Wu et al., “Application research of image recognition technology based on CNN in image location of environmental monitoring UAV,” *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–11, 2018.
- [25] W.-Y. Chen, Y.-F. Liao, and S.-H. Chen, “Speech recognition with hierarchical recurrent neural networks,” *Pattern Recognition*, vol. 28, no. 6, pp. 795–805, 1995.
- [26] L. Gao, P.-Y. Chen, and S. Yu, “Demonstration of convolution kernel operation on resistive cross-point array,” *IEEE Electron Device Letters*, vol. 37, no. 7, pp. 870–873, 2016.
- [27] F. Y. Zhou, L. P. Jin, and J. Dong, “Review of convolutional neural network,” *Chinese Journal of Computers*, vol. 40, no. 2, 2017.
- [28] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate Shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, pp. 448–456, Lille, France, July 2015.
- [29] G. E. Hinton, N. Srivastava, A. Krizhvsy, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012, <https://arxiv.org/abs/1207.0580>.



- [30] F. Dan Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 3, pp. 1930–1935, IEEE, Houston, TX, USA, June 1997.
- [31] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 807–814, Haifa, Israel, June 2010.
- [32] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.
- [34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [35] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [36] P. Zhang, "Mean-mean absolute deviation portfolio model and optimization," *Statistics and Decision-Making*, vol. 1, pp. 14–15, 2009.

## Research Article

# Financial Trading Strategy System Based on Machine Learning

Yanjun Chen,<sup>1</sup> Kun Liu,<sup>1</sup> Yuantao Xie ,<sup>2</sup> and Mingyu Hu<sup>2</sup>

<sup>1</sup>School of Information and Management, University of International Business and Economics, Beijing, China

<sup>2</sup>School of Insurance and Economics, University of International Business and Economics, Beijing, China

Correspondence should be addressed to Yuantao Xie; xieyuantao@uibe.edu.cn

Received 6 May 2020; Accepted 7 July 2020; Published 28 July 2020

Guest Editor: Qian Zhang

Copyright © 2020 Yanjun Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The long-term and short-term volatilities of financial market, combined with the complex influence of linear and nonlinear information, make the prediction of stock price extremely difficult. This paper breaks away from the traditional research framework of increasing the number of explanatory variables to improve the explanatory ability of multifactor model and provides a new financial trading strategy system by introducing Light Gradient Boosting Machine (LightGBM) algorithm into stock price prediction and by constructing the minimum variance portfolio of mean-variance model with Conditional Value at Risk (CVaR) constraint. The new system can capture the nonlinear relationship between pricing factors without specific distributions. The system uses Exclusive Feature Bundling to solve the problem of sparse high-dimensional feature matrix in financial data, so as to improve the ability of predicting stock price, and it can also intuitively screen variables with high impact through the factor importance score. Furthermore, the risk assessment based on CVaR in the system is more sufficient and consistent than the traditional portfolio theory. The experiments on China's stock market from 2008 to 2018 show that the trading strategy system provides a strong logical basis and practical effect for China's financial market decision.

## 1. Introduction

With the development of stock market, the efficiency of artificial subjective investment mode is gradually reduced due to the complex and diverse investment targets. Benefiting from the advancement of data science and statistical method, the former subjective investment mode has been gradually replaced by quantitative investment strategy, which uses data and models to construct investment strategies. New investment model, selecting stocks with investment value by combining the open information in the market with statistical methods, avoids the subjective impact of human to some extent.

As the most widely used quantitative stock selection model at present, multifactor model is based on finding out factors with the highest correlation with the stock return rate, which can predict the stock return to some extent. However, in the empirical test, scholars have found that it could not bring sustained returns to investors due to the low prediction accuracy and the lack of stability of the prediction results.

At the same time, through the empirical study with financial market data, scholars found that the financial market is a dynamic system with high complexity, including long-term and short-term fluctuations and linear and nonlinear information. The formation and change of stock price involve various uncertain factors, and there are complex relationships among them. To further study and analyze financial data and for more accurate prediction, the application of machine learning algorithms in the research of financial time series has been widely concerned by scholars.

Compared with the linear model, machine learning algorithm takes the nonlinear relationship between variables into account. It does not need to be based on the assumption of independence and specific distribution and has higher flexibility and efficiency, making it excel at dealing with big data, especially the huge amount of financial data.

The innovations of this paper are as follows: Firstly, it breaks away from the traditional research framework of improving the explanatory power of multifactor model by increasing the number of explanatory variables and provides a new trading strategy system by introducing one of the

latest machine learning algorithms, LightGBM, into the field of portfolio. LightGBM does not need to consider the specific distribution form of financial data, and it can capture the nonlinear relationship between pricing factors, and the Exclusive Feature Bundling method can solve the problem of sparsity of high-dimensional characteristic matrix and improve the prediction accuracy of stock returns. Secondly, the new system can be used to generate importance score of factors. It directly shows the impact variables have on stock return, which has important practical significance for stock selection. Thirdly, for stock position allocation module, we construct minimum-variance weight method of mean-variance model with CVaR constraint and test the trading strategy system on the real data of China's A-share market. The experiment result shows that the system can bring stable excess return to investors and provides a logical basis and practical effect for China's stock market.

## 2. Literature Review

Since Markowitz proposed mean-variance model to quantify the risk in 1952, scholars have been trying to find ideal models for stock pricing. In 1960s, the CAPM model proposed by Sharpe, John Lintner, and Jan Mossin expresses the relationship between expected return and expected risk as a simple linear relationship [1, 2]; Ross [3] put forward the Arbitrage Pricing Theory on this basis, revealing that stock return is affected by multiple factors rather than a single factor; Fama and French [4, 5] proposed a three-factor model, which includes market value, book value ratio, and P/E ratio of listed companies as compensation for the risk factors that  $\beta$  cannot reflect. However, the classic models above were lacking explanation in the empirical test. At the beginning, scholars attributed that to the omission of explanatory variables, so they successively put forward factors that may lead to excess return: For example, Datar et al. [6] used the data of NYSE for empirical analysis and found that there was a significant negative correlation between stock return and stock turnover. Piotroski [7] selected nine indicators from the perspective of the company's financial indicators, including profitability, robustness, and growth. By comprehensively scoring each indicator, he established a stock pool and selected the stocks according to the scores. Novy-Marx [8] proved that there is a strong correlation between the company's profitability and stock return. Aharoni et al. [9] also found that the company's investment level and stock return are remarkably correlated. In 2014, Fama and French added Robust Minus Weak (RMW) as profit factor and Conservative Minus Aggressive (CMA) as investment factor based on the three-factor model and put forward five-factor model to further explain the stock return [10].

However, none of the models above deviated from the research framework of linear asset pricing under the background of small sample data, that is, extract excess return factors from historical data and then use these factors as independent variables to construct a linear model to evaluate the investment value of stocks.

Since the 1970s, with the increasing availability of empirical data in the financial market and the improvement of computer technology, scholars have found several abnormal phenomena in the financial market through empirical research. These abnormal phenomena are contrary to the basic assumption of CAPM: financial market data obey normal distribution, have no long memory, and satisfy the linear model, which challenges the traditional model [11]. For the stock market, Greene and Fielitz [12] performed a test and confirmed that the American stock market has the feature of long memory, which showed that even if the time interval is very long, it still had significant autocorrelation; that is, the historical events would affect the future for a long time. On the one hand, it proved the importance of historical information and the predictability of return; on the other hand, it also reflected the nature of nonlinear structure of stock market. As for the distribution of financial data, scholars pointed out that, in reality, the distribution of financial data is usually characterized by thick tail and asymmetry [13]. Therefore, the traditional use of normal distribution to fit the actual financial data has limitations. For example, in VaR calculation, due to the thick tail of financial data distribution, the calculation under the assumption of normal distribution will lead to huge errors [14]. In order to find the most reasonable distribution hypothesis, Mandelbrot [15] proposed replacing the normal distribution of financial data with the stable distribution. However, because the tail of the stable distribution is usually thicker than the actual distribution, some scholars proposed using the truncated stable distribution as the distribution of securities returns [16], but where to cut off had become another question.

In recent years, in order to analyze and predict financial data more accurately, machine learning has received wide attention from scholars. Compared with the traditional model, machine learning has a unique advantage in dealing with financial data. First of all, it can automatically identify the hidden features behind the financial data, reducing human intervention. Secondly, the traditional linear asset pricing model is based on the assumption that the financial system is linear. However, scholars' research on the nonlinear characteristics of financial time series, such as long memory and nonpairing distribution, indicates that the stock market system is actually a dynamic system with linear and nonlinear information. Machine learning models can deal with high-dimensional and collinear factors and are not limited to the probability distribution of investment income. Machine learning models do not need to calculate high-dimensional covariance matrix [17].

Mukhejee (1997) firstly proposed the application of support vector machine in nonlinear chaotic time series, which provided the basis for the application of stock series. In the empirical study, Fan and Palaniswami [18] first applied the support vector machine model to stock selection. Based on the data of Australian stock exchange, the model they constructed could identify the stocks that outperform the market and the five-year yield of the equal weight portfolio constructed was 208%, which was far higher than the benchmark return of the large market. Kim [19] used

support vector machine (SVM) and artificial neural network (ANN) to predict the market index and the results showed that SVM had more advantages than ANN in stability. There are also some literatures that focus on the differences between different models in variable selection and modeling characteristics; for example, Xie et al. [20] and Huang et al. [21] set the rise and fall of stock market as dichotomous variables and used linear model, BP neural network, and support vector machine model to predict them. It was found that SVM had better classification performance than other methods, and the combined model performed best in all prediction methods when SVM is combined with other models. Nair et al. [22] used C4.5 decision tree algorithm to extract the characteristics of stock data and then applied it to the prediction of stock trend. They found that the prediction effect of C4.5 decision tree was better than neural network and naive Bayesian model. Zhu et al. [23] applied Classification and Regression Tree (CART) algorithm and traditional linear multifactor model in North American market during the outbreak of financial crisis and found that the stock selection model based on CART algorithm had a significant effect on risk dispersion. Kumar and Thenmozhi [24] used random forest model to predict the up and down direction of Standard and Poor's and found that the result was better than that of SVM. Bogle and Potter [25] used decision tree, artificial neural network, support vector machine, and other machine learning models to predict the stock price of Jamaica stock exchange market and found that, in this market, the prediction accuracy of stock price could reach 90%.

In recent years, the first mock exam has also been made in some areas, such as the sequence dependence of financial time series data and the local association characteristics of different financial market time series data. For example, Xie and Li [26] discussed the joint pricing models and Yan [27] constructed a CNN-GRU neural network, which combines the advantages of convolutional neural network (CNN) and gating loop unit (GRU) neural network. There are also some papers that study the computing power, time consumption, and even hardware layout of various algorithms. For the discussion of machine learning related hardware, refer to Tang et al. [28, 29].

### 3. System Introduction

Based on machine learning algorithm, this paper constructs an optimal trading strategy system, which aims to bring stable excess return to investors. According to Figure 1, this system is divided into four modules: data preprocessing, stock pool selection, position allocation, and risk measurement. The details are as follows.

**3.1. Data Preprocessing.** Because of the noise and format asymmetry in financial data, preprocessing plays an important role in getting accurate prediction results. According to Figure 2, we preprocess the financial data according to the following steps:

Step 1 (financial data processing): due to the differences in the format of financial statements of different companies, the data sets have sparse data spaces. The financial accounts with over 20% missing values are discarded directly. The remaining default values are filled with the average values of the previous and next three quarters. Since the income statement and cash flow statement are process quantities, representing the accumulation of quarterly values, the data of these two financial statements are differentially processed to obtain quarterly data.

Step 2 (market data processing): since the stock market data set is monthly, in order to match the data in the financial statements, it is processed on a quarterly average basis.

Step 3 (data screening): due to the differences in the format of financial statements in different industries, financial data are divided into banking, securities industry, insurance industry, and general business. As the banking, insurance, and security industries are all subindustries under the financial industry, their businesses are complicated and are greatly affected by the macro impact, resulting in the uncertainty and volatility of their stock prices far greater compared to the general business. Therefore, prediction from their financial data alone is difficult. Moreover, after preprocessing, it is found that the data of these industries are too insufficient to make a prediction. In view of the lack of reference in the forecast results, these three industries are deleted from the data, and only the data of general business are retained.

Step 4 (data splicing): we take the company's stock code as the primary key and combine the financial data with market data of the same quarter into a wide table to prepare for feature engineering.

Step 5 (feature engineering): this paper applied machine learning models into stock return forecasting. The reason why machine learning can achieve high prediction accuracy is that it can deal with the non-linear relationship between variables. Unlike simple linear model, the complexity of the model leads to machine learning being regarded as a "black box." Due to the lack of interpretation, it is contradicted by the traditional financial industry. In order to improve the credibility of our model, not only do we use the importance of variables to analyze the important influencing factors of stock price return, but also we accord to the previous literature and construct the characteristics that have been proved to have strong significance in the previous multifactor model. Table 1 shows the calculation method and index implication.

Step 6 (missing value processing and standardization): due to the high requirements of data integrity in stock forecast, we delete the data with the missing rate more than 10%. For the remaining missing value, based on the idea of moving average, we take the same fields of the three records before and after the missing record to

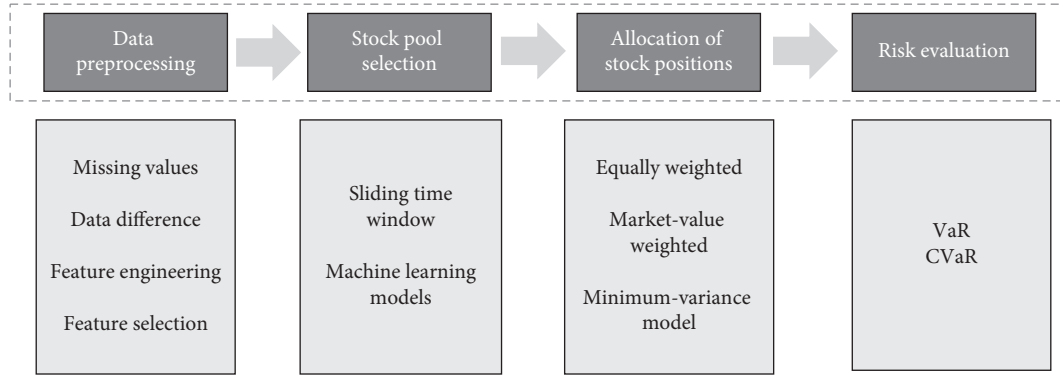


FIGURE 1: Framework of the trading strategy system.

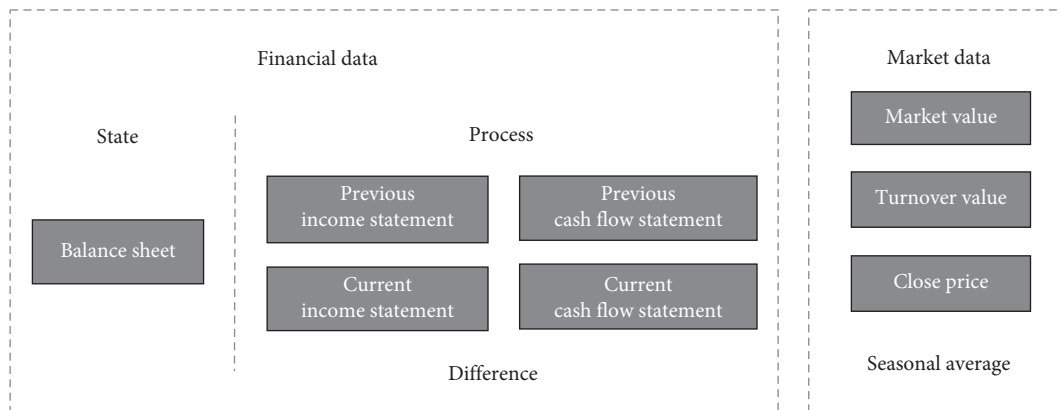


FIGURE 2: Data and data processing.

calculate the moving average value, thus retaining the trend information of the stock as much as possible. Finally, in order to improve the convergence speed and prediction accuracy of the model, the data are standardized.

**3.2. Stock Pool Generation.** This paper uses LightGBM algorithm under the sliding time window training method to predict the quarterly earnings of stocks and compares it with traditional linear model, support vector machine, artificial neural network, random forest, and other machine learning models. The evaluation criteria of models are R2 and RMSE. Based on the prediction results, a stock pool composed of a limited number of stocks is generated, and then the portfolio is constructed according to different position allocation methods.

**3.2.1. Algorithm Principle: LightGBM.** Compared with the generalized linear model, machine learning, as a new model, does not need to be based on the assumption that the variables are independent and obey the distribution of specific functions and has greater flexibility and efficiency. Machine learning algorithms have unique advantages in

dealing with a large amount of data, such as financial market data.

Among many machine learning algorithms, LightGBM algorithm has the characteristics of high speed, high accuracy, high stability, and low memory space. It has wide application space in the financial field with large amount of data and high requirements for prediction accuracy and stability. LightGBM is essentially an enhanced gradient lifting tree that can be used for regression and classification. Compared with the previous gradient lifting tree (GBT), it has the following advantages: LightGBM uses the leaf-wise (best-first) strategy to grow a tree: find the leaf with the largest gain each time and split it while other nodes without the maximum gain do not continue to split. LightGBM does not need artificial trim, so the result is relatively objective and stable. At the same time, LightGBM adopts histogram algorithm and accumulates statistics in the histogram according to the value after discretization as the index to calculate the information gain. LightGBM also adopts two new calculation methods: Gradient-Based One-Side Sampling and Exclusive Feature Bundling, greatly improving the accuracy and efficiency of calculation.

- (1) Given the supervised training set  $X = \{(x_i, y_i)\}_{i=1}^n$ , the goal of LightGBM is to minimize the objective function, which is



TABLE 1: Features and implication.

	Index	Calculation method	Implication
Profitability	ROE	Net profit/owner's equity	Return on equity
	ROA	Net profit/total assets	Return on assets
	ROS	Total profit/operating income	Return on sales
Fluidity	D/E	Liabilities/owner's equity	Debt to equity ratio
	Cash ratio	(Current assets – inventory)/current liabilities	Ratio of quick assets to current liabilities
	Current ratio	Current assets/current liabilities	Ratio of current assets to current liabilities
Operating efficiency	Equity turnover	Sales income/average shareholders' equity	The efficiency of the company in using the owner's assets
	Asset turnover	Sales income/average total assets Average balance of total assets = opening balance + closing balance/2	An important financial ratio to measure the efficiency of enterprise asset management
Valuation index	B/M	Outstanding stock * closing price/shareholder equity	Book-to-market ratio High B/M, considered to be undervalued by the market, resulting in high yield
	P/E	Market price per common share/earnings per common share per year	Price-to-earnings ratio The lower the price earnings ratio is, the lower the profitability of the market price relative to the stock is
	P/B	Share price/net asset per share	Price-to-book ratio The higher the investment value of stocks with low market to net ratio is, the lower the investment value is
	Market value of listed company	Market price per share * total number of shares issued	The total value of shares issued by a listed company at market price
	Turnover	Trading volume/total issued shares	The higher the turnover of a stock is, the more active the stock is in trading
$\beta$ Coefficient	Systematic risk coefficient	Regression of the historical rate of return of a single stock asset to the index rate of return of the same period	$\beta$ describes the systemic risk of a fully diversified portfolio

$$J(\phi) = \sum_i l(\hat{y}_i, y_i), \quad (1)$$

where  $i$  denotes the  $i$ -th sample and  $l(\hat{y}_i, y_i)$  denotes the prediction error of the  $i$ -th sample.

- (2) For optimizing the objective function, LightGBM uses gradient boosting method to train rather than using bagging method to directly optimize the whole objective function. The gradient boosting training method optimizes the objective function step by step. Firstly optimize the first tree and then optimize the second one and so on until the  $K$  tree is completed.
- (3) When generating a new optimal tree, LightGBM uses the leaf-wise algorithm with depth limitation to grow vertically. Therefore, the leaf-wise algorithm is more accurate compared with the level-wise algorithm when they have the same number of splitting times.

When the  $T$ -th tree is generated, every time a newly generated split node is added, and then the objective function can be obtained as follows:

$$J = \frac{1}{2} \left[ \frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right], \quad (2)$$

where  $g_i = \partial_{\hat{y}} l(y_i, \hat{y}^{(t-1)})$ ,  $h_i = \partial_{\hat{y}}^2 l(y_i, \hat{y}^{(t-1)})$ ;  $I_L$  and  $I_R$  are the sample sets of the left and right branches, respectively.

**3.2.2. A Method for High-Dimensional Feature Matrix Sparsity: Exclusive Feature Bundling.** In order to get the data of each quarter, we carried out differential processing on the financial statements of process volume during data pre-processing. However, due to the lack of mandatory regulations on the quarterly reports of listed companies in China, some companies have the problem of time lag in the quarterly reports, and there are a large number of zero values in the data of nonupdated adjacent quarterly reports after differential processing. At the same time, due to business differences between companies, nonshared business accounts in financial statements sometimes are also zero.

In conclusion, the financial data of listed companies with large subjects and different businesses eventually form a high-dimensional feature matrix, and a large number of zero values in the matrix lead to the problem of data sparsity. Traditional statistical methods are often unable to extract enough effective information when dealing with high-dimensional and sparse data, which makes the prediction results inaccurate or even wrong. In order to solve this problem and improve the prediction accuracy, we introduce Exclusive Feature Bundling (EFF) method of LightGBM into the stock pool selection module.

Exclusive Feature Bundling aims to solve the problem of data sparsity by merging mutually exclusive features to reduce the dimension of feature matrix [30]. Since LightGBM stores the features divided into discrete values by constructing histogram instead of storing continuous values directly, we can combine mutually exclusive features by assigning them to different intervals of the same histogram.

For example,  $X_a$  and  $X_b$  are two mutually exclusive features, where  $X_a \in [0, a)$  and  $X_b \in [0, b)$ .

The new bundling feature  $X_c$  can be obtained by adding the value range of  $X_a$  as offset and value range of  $X_b$ , where  $X_c \in [0, a + b)$ :

$$\begin{cases} X_a = X_c, X_b = 0 & \text{when } 0 \leq X_c < a, \\ X_a = 0, X_b = X_c - a & \text{when } a \leq X_c < a + b. \end{cases} \quad (3)$$

**3.2.3. Training Method: Sliding Time Window.** Since the data of the stock market is a time series, the historical information of the company has a great influence on the future stock price. Considering that the sequence has a great influence on the values of the sequence nodes, the prediction model in stock pool selection should consider the key element, "time," in the prediction process, instead of treating the stock prices of all times as the same data and randomly selecting the training set and the test set. Therefore this system does not use cross-validation but uses sliding time window to randomly simulate the prediction process.

In the experiment, suppose that the size of sliding time window was  $N$  quarters. In a unit time window, the first  $N-1$  quarter is the training set, and the last quarter is the test set. The size of the time window should take into account the characteristics of the data set. If it is too short, the natural time period of the test set may be outside the training set, and the time information brought by the time window will be greatly reduced; if it is too long, some unnecessary noise may be introduced.

In essence, the model in each time window is a new model and they are all independent of each other. According to Figure 3, the stock price of each stock in the next quarter in each time window is predicted according to all the information in the latest quarter.

**3.2.4. Stock Selection.** The goal of this paper is to apply a new machine learning model, LightGBM, to the prediction of stock return and to construct a low-risk and high-yield portfolio compared with the stock price prediction models used in previous studies. In order to highlight the risk of different portfolio construction methods, the first  $N$  stocks are selected as the stock pool from the stock list sorted by the yield, only the long purchase rule is allowed in this part to ensure that the yield equals the required value. Then adjust the position of each stock according to different weights to find the optimal portfolio.

**3.3. Stock Position Allocation Method.** This paper uses three methods of stock position allocation: (1) equal-weight method, (2) market-value weighting method, and (3) minimum-variance

method of mean-variance model with CVaR constraint. Combined with the sliding time window training method to predict the quarterly earnings of the stock, we use  $R$ -squared and RMSE of the model as the evaluation criteria, while in the traditional linear model, support vector machine, artificial neural network, random forest, and other machine learning models are used. Based on the prediction results, a stock pool composed of a limited number of stocks is generated, and then the portfolio is constructed according to different position allocation methods.

**3.3.1. Method 1: Equal-Weight Method.** Each stock is assigned the same weight. If there are  $n$  stocks in the stock pool, the weight of each stock is  $w_i = 1/n$ .

**3.3.2. Method 2: Market-Value Weighting Method.** The ratio of market value of stock  $i$  to the market value of all  $n$  stocks in the stock pool is taken as the weight of this stock, and the calculation formula is as follows:

$$w_i = \frac{\text{Market value}_i}{\sum_{i=1}^n \text{Market value}_i}, \quad (4)$$

where  $\text{Market value}_i$  is the market value of the company at the closing of the previous period and it is calculated by multiplying the market price of each share by the total number of shares issued.

**3.3.3. Method 3: Minimum-Variance Weight Method of the Mean-Variance Model with CVaR Constraint.** In 1952, Markowitz published the beginning article of modern portfolio theory "portfolio selection" in the financial magazine, which studies how to allocate risk assets effectively. Markowitz believed that investors only consider two factors of expected return and standard deviation of forecast when making portfolio decision, so portfolio decision was mainly based on the following two points: (1) when the investment return is the same, investors want to minimize the risk; (2) when the risk is the same, investors want to maximize the income. According to the principle of mean-variance efficiency, the optimal portfolio can be expressed by mathematical programming in the process of investing in assets.

Assuming that the return of risk assets obeys normal distribution, consider CVaR constraint in Markowitz mean-variance model, and then the portfolio optimization model based on CVaR constraint is

$$\begin{cases} \min & \sigma_p^2 = \min X^T \Sigma X, \\ \text{s.t.} & \text{CVaR}_\beta = C_2(\beta)\sigma_p - E(r_p) \leq L, \\ & E(r_p) = X^T R, \\ & X^T I = 1, \quad I = (1, 1, \dots, 1)^T, \\ & x_i \in [0, 1], \end{cases} \quad (5)$$

where  $C_2(\beta) = \phi(\Phi^{-1}(\beta))$ ,  $R = (R_1, R_2, \dots, R_n)$ ,  $R_i = E(r_i)$  is the expected return of  $i$ -th stock, and  $X = (x_1, x_2, \dots, x_n)^T$  is the weight of each portfolio. We have the following:



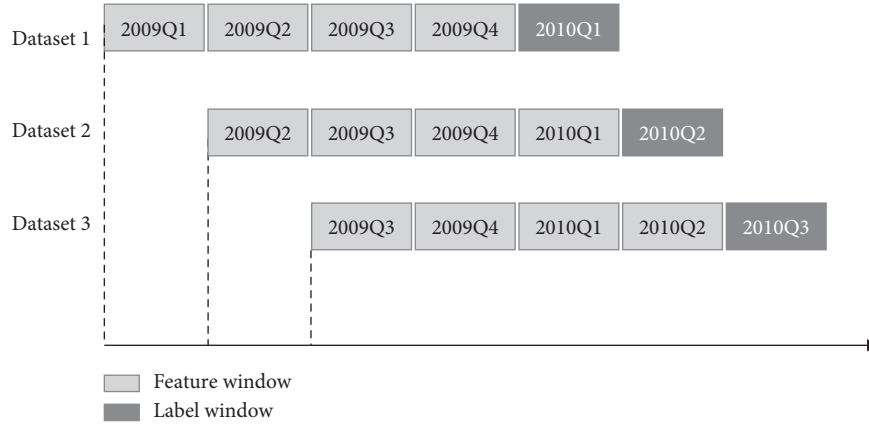


FIGURE 3: Sliding time window model.

- (1) According to the constraint equation,  $CVaR_\beta = C_2(\beta)\sigma_p - E(r_p) \leq L$ , and we obtain

$$CVaR_\beta = C_2(\beta)\sigma_p - E(r_p) = L. \quad (6)$$

- (2) After transformation, we obtain

$$\sigma_p^2 = \frac{1}{(C_2(\beta))^2} (E(r_p) + L)^2 = \frac{1}{(C_2(\beta))^2} \cdot (E^2(r_p) + 2LE(r_p) + L^2). \quad (7)$$

- (3) Define  $R_p = E(r_p) = X^T R$  and  $\sigma_p^2 = X^T \Sigma X$ ; equation (7) can be

$$X^T \Sigma X = \frac{1}{(C_2(\beta))^2} \left( (X^T R)^2 + 2L(X^T R) + L^2 \right). \quad (8)$$

- (4) Since the rank of the system of linear equation (8) is  $n - 2$ , the number of its basic solutions is 1. That is to say,  $x_2, x_3, \dots, x_{n-1}$ , can be linearly expressed by  $x_1$  (obtained by elimination method). Since  $\sum_{i=1}^n x_i = 1$ ,  $x_n$  can also obviously be expressed by  $x_1$ , by substituting  $x_2, x_3, \dots, x_{n-1}$  into equation (8), we can get a quadratic equation of  $x_1$ , and then we can obtain  $x_2, x_3, \dots, x_{n-1}$  by calculating  $x_1$ .
- (5) Therefore, if  $x_1$  has no solution, the constraint line does not intersect the mean-variance effective front, which means CVaR does not play a constraint role; if  $x_1$  has two multiple roots, there is only one intersection  $A$  between the constraint line and the mean-variance effective front, and the weight of  $A$  is  $x_1$ ; if there are two different roots, the constraint line and the mean-variance effective front have two intersections,  $A$  and  $B$ . The expected return and variance of portfolio at  $A$  and  $B$  are  $R_A, R_B, \sigma_A^2$ , and  $\sigma_B^2$ .

**3.4. Risk Evaluation.** This paper mainly studies how to reduce the risk of portfolio. At present, the most common measurement methods of portfolio risk are sensitivity

method, volatility method, VaR method, and CVaR method. They are introduced, respectively, as follows.

**3.4.1. Method 1: Sensitivity Method.** Sensitivity method is a method to measure the risk of financial assets by using the sensitivity of the value of financial assets to market factors. Sensitivity refers to the percentage change in the value of financial assets when market factors change by a percentage unit. The greater the sensitivity of financial assets is, the greater the impact of market factors is and the greater the risk is. The sensitivity of different types of financial assets has different names and forms, such as the duration and convexity of bonds, beta of stocks, Delta, Gamma, and Vega of derivative assets. The problems of sensitivity method lie in the following: (1) The sensitivity is only valid when the range of market factors changes is very small. (2) A certain sensitivity concept is only applicable to a certain class of specific assets or a certain class of specific market factors, which makes it difficult to compare the risks among different kinds of assets. (3) The sensitivity is only a relative proportion concept, which cannot determine the risk loss of a certain portfolio body value.

**3.4.2. Method 2: Volatility Method.** Volatility method is a statistical method, which is usually described by standard deviation or covariance. Variance represents the volatility of the actual rate of return deviating from the average rate of return. The greater the volatility, the greater the uncertainty of the actual rate of return, regardless of whether the actual rate of return is higher than the average rate of return or lower than the actual rate of return. One of the main defects of variance measurement of investment risk is that variance represents positive and negative deviation, generally speaking, investors do not want that the actual return is less than the expected return, but they do not refuse when the actual return is higher than the expected return. Therefore, Markowitz put forward the semi variance method in 1959; that is, the part of the actual income higher than the expected income is not included in the risk, and only the loss is included. There are some problems in both variance method and semi variance method: (1) The method is based on the

assumption that variance exists, but whether the variance of return rate exists is still questionable. (2) Variance implies the assumption of normality, so only the linear correlation structure between risks can be analyzed. In reality, the risk dependence structure may be a nonlinear complex structure. (3) The square deviation does not specifically indicate how much the loss of the portfolio is. (4) This method is not suitable for comparing the risk of assets with different expected return.

**3.4.3. Method 3: VaR Method.** VaR is Value at Risk [31]. VaR is the maximum possible loss expected in the holding period of an investment within a certain confidence level. Its mathematical expression is as follows:

$$\text{Prob}(\Delta p \leq \text{VaR}) = \alpha, \quad (9)$$

where  $\Delta p$  is the loss amount of the portfolio during the holding period  $\Delta t$ , which is the Value at Risk under the given fixed credit level  $\alpha$ , that is, the upper limit of possible loss. The meaning of the expression of the above formula is that the risk loss of the portfolio is not less than the VaR at the level of probability. In the research, the above expression is regarded as a function of VaR on  $\alpha$ , and the probability distribution function of the portfolio return is expressed with  $F(\alpha)$ , which means

$$\text{VaR}_\alpha = F^{-1}(\alpha). \quad (10)$$

The advantages of VaR method are: the following (1) The measurement of risk is simple and clear, the risk measurement standard is unified, and it is easy for managers and investors to understand and grasp. (2) VaR can also be used to compare the risks of different types of financial assets, but its disadvantage lies in the inconsistency.

**3.4.4. Method 4: CVaR Method.** CVaR is a new risk measurement method proposed by Roekafellor and Uryasev (1999), also known as Conditional Value at Risk method, which means that, under a certain confidence level, the loss of portfolio exceeds the mean value of a given VaR, reflecting the average level of excess loss. Its mathematical expression is as follows:

$$\text{CVaR}_\alpha = E(-X | -X \geq \text{VaR}_\alpha), \quad (11)$$

where  $-X$  represents the random loss of the portfolio and  $\text{VaR}_\alpha$  is the Value at Risk under the confidence level.

The advantages of CVaR are as follows: (1) It solves the problem of inconsistency measurement, satisfies the additivity of risk, and improves the defect of VaR. (2) It does not need to realize the form of assumed distribution, and, in any case, its calculation can be realized by simulation. (3) It fully measures tail loss and calculates the average value of tail loss, which considers all tail information larger than VaR rather than based on a single quantile to calculate.

Therefore, this paper mainly measures the risk of stock portfolio based on the VaR and CVaR. Monte Carlo simulation method is used in the specific calculation method, which is the most effective method to calculate VaR and

CVaR as it can solve the nonlinear relationship of various targets well without making assumptions on the distribution of portfolio income.

## 4. Empirical Research

**4.1. Experimental Results and Analysis.** The data used in this paper are the market data of 3676 A-share listed companies in China from 2008 to 2018, as well as the financial data disclosed by the company on a quarterly basis (including the company's balance sheet, profit statement, and cash flow statement).

After data preprocessing and feature engineering, the processed data from the fourth quarter of 2008 to the first quarter of 2018 are selected as the final data set.

Based on the total split times of features, the top 20 variables in the variable importance score obtained by our trading strategy system are as follows.

According to Figure 4, the factor affecting the next stock price most significantly is the closing price of the current stock (CLOSE\_PRICE). According to Charles Dow's technical analysis theory, the historical price of the stock contains a lot of information. The price will evolve in the way of trend, and the history will always be repeated because of human psychology and market behavior. By studying the historical price of the stock, investors can find out the current market and the trend, so as to better detect the stock selection target and the opportunity to build a position. Therefore, the closing price of the current stock is highly related to the next stock price, which is the most important variable to predict the stock price. The second in the list is accounts payment (AP). In order to expand sales and increase market share, enterprises often buy materials and accept services first and then pay service fees and commissions. The time difference between sales and payment also reflects the risk that enterprises may be short of funds and cannot pay in time. Therefore, AP is an important reference index for investment. The third index, the growth rate of the previous period's stock price (Price\_rate), has a great contribution to the prediction of the stock price. It has a positive correlation with the growth of the current period's stock price. Generally, the higher the growth rate of the previous period's stock price is, the higher the growth of the stock is and the greater the possibility of continuous growth in the current period is.

Other financial indicators, such as the balance of cash and cash equivalents at the beginning of balance (N\_CE\_BEG\_BAL), turnover (TURNOVER\_VALUE), cash paid for goods purchased and services received (C\_PAID\_G\_S), surplus reserve (SURPLUS\_RESER), and return on assets (ROA), also have impact on the stock price. The balance of cash and cash equivalents at the beginning of the balance refers to the amount of cash and cash equivalents carried over from the previous year to the current year for current turnover. It reflects the cash stock of the enterprise. Turnover rate measures the frequency of stock turnover in the market within a certain period of time, reflecting the activity of market trading investment. Its calculation formula is as follows: turnover rate = trading volume / total number of shares issued. The higher the turnover of a stock

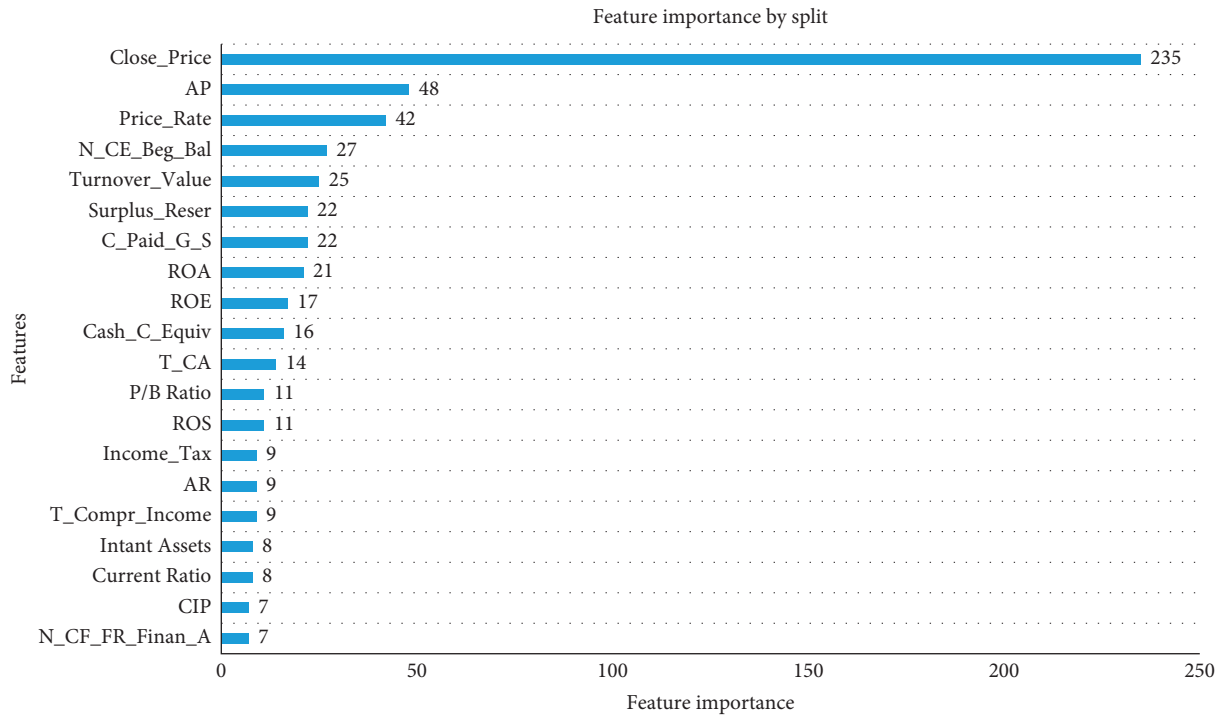


FIGURE 4: Variable importance score.

is, the more active the stock is. The cash paid for purchasing goods and receiving services is the main cash flow generated in the business activities of industrial and commercial enterprises, which reflects the status of the main business. The surplus reserve is the accumulation of earnings that the enterprise keeps in the enterprise from the after tax profits. It can be used to expand production and operation, increase capital (or share capital), or distribute dividends, which has a direct impact on the stock price. The return on assets is an important index to measure the profitability of a company relative to its total assets. The calculation method is as follows: return on assets = net profit/total assets. The higher the index is, the better the asset utilization effect of the enterprise is, indicating that the enterprise has achieved good results in increasing revenue and saving capital use. It can be seen that LightGBM considers a variety of financial indicators comprehensively, and the importance score also provides a reference for the research and analysis of long-term investment in enterprise value.

It is noteworthy that among the top 20 factors obtained by the model, 40% of the factors are constructed by us according to the fundamental theory. On the one hand, it reflects the scientific nature of combining the fundamental aspect theory with the pure machine learning method. On the other hand, it can also be seen that, in the previous literature, the prediction of stocks completely depending on individual fundamental factors may cause inaccuracy.

**4.1.1. Comparisons of Models.** In order to compare the accuracy of LightGBM and other algorithms, this paper uses GLM (generalized linear model), DNN (deep neural network), RF (random forest), SVM (support vector machine),

and LightGBM to predict the next stock price of each quarter. The results are shown in Figure 5. *R*-squared measures the goodness of fit, which is equal to the ratio of the sum of squares of regression to the total sum of squares. The closer *R*-squared is to 1, the better the fitting degree of regression model is. RMSE represents the root mean square error. The smaller RMSE is, the more accurate the prediction is. It can be seen from the left that the *R*-squared under LightGBM model is 0.798; that is, this model can solve the variation of 79.8% of the stock price, which is higher than the other four methods and indicates that LightGBM is the best to fit the stock price. It is noteworthy that *R*-squared of the linear model is almost 0.443, and the correlation is very weak. It is speculated that the reason is that the stock price and a large number of factors do not satisfy the linear relationship at the same time. The linear model is only applicable to the model composed of a few fundamental factors. Even if it contains the most influencing factors as much as possible, such a model still lacks explanation for excess return. It can be seen from (b) that the RMSE of LightGBM model is 6.1829, which is lower than the other four methods, also showing the high accuracy of LightGBM.

**4.1.2. Risk Assessment of Models.** In order to compare the risk of the portfolio of GLM, DNN, RF, SVM, and LightGBM under equal-weight allocation method, market-value weighting method, and minimum-variance weight method of mean-variance model with CVaR constraint, the top *N* stocks with the highest investment income in stock pool are calculated under these 15 conditions, respectively, and the VaR and CVaR of portfolio investment are calculated at the confidence level of  $\alpha = 5\%$ , as shown in Figure 6.

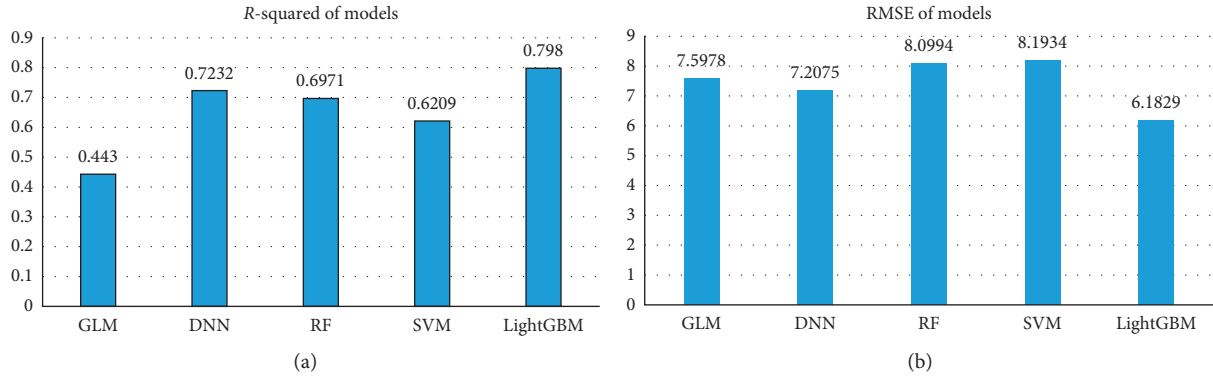


FIGURE 5: R-squared and RMSE comparisons of five models.

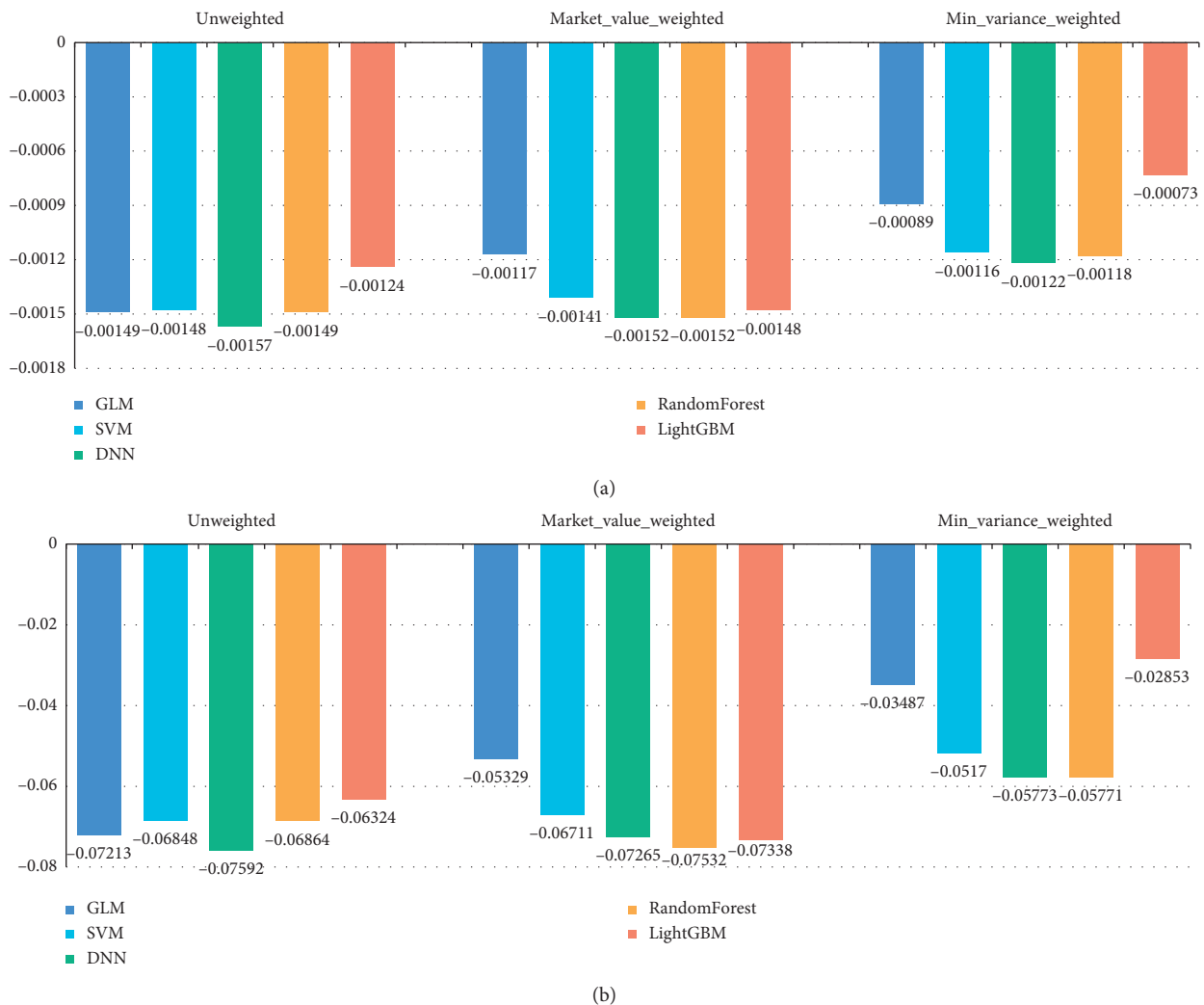


FIGURE 6: Risk assessment of 5 algorithms under 3 position allocation methods. (a) Var. (b) CVaR.

In the selection of position allocation methods, it can be seen from Figure 6 that, whether the risk is measured by VaR or CVaR, minimum-variance weight method is more able to minimize the risk of the portfolio and reduce the loss compared to the other two allocation methods. At the same

time, the overall ranking of algorithms under VaR and CVaR is basically the same. It is because CVaR is based on VaR and CVaR is the optimization of VaR in risk measurement. The two are highly correlated and meet the expectations; thus the model results are reasonable.

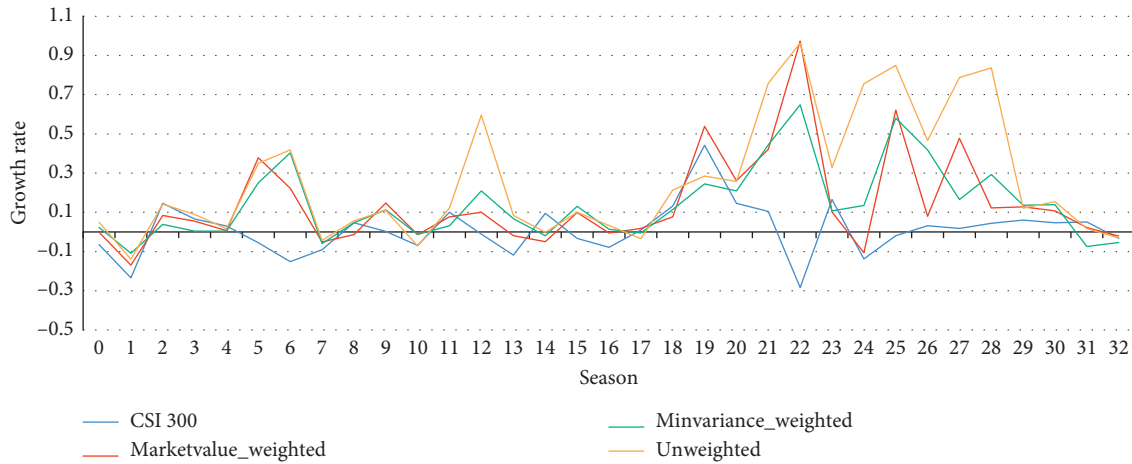


FIGURE 7: Market index tracking results.

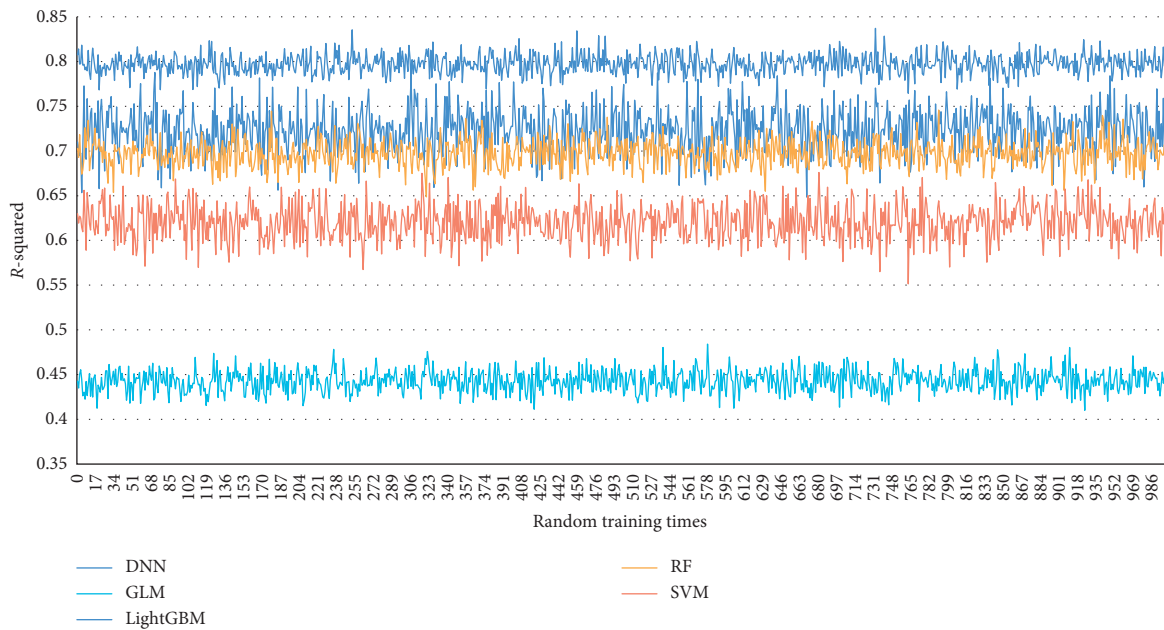


FIGURE 8: 1000 repetitive tests of 5 models.

In algorithm selection, LightGBM has lower risk compared to the other four models under minimum-variance weight method of mean-variance model with CVaR constraint. Its VaR is  $-0.0073$  and CVaR is  $-0.02853$ , which means that, under the normal fluctuation of the stock market, the probability that the return of the optimal portfolio declines by more than 0.73% due to market price change is 5%, and the expected loss of the whole stock portfolio is 2.835%. Under equal-weight allocation method, LightGBM also has lower risk than the other four models, VaR is  $-0.00124$ , and CVaR is  $-0.06324$ , which shows that it can significantly reduce the risk of portfolio and make the expected return of investors more stable than linear models and traditional machine learning algorithms. It is noteworthy that LightGBM is not the best under market-value weighting method. The reason may be that market-value weighting method pays too much attention to the stocks

with higher market value and does not consider the investment value of small- and medium-sized stocks in the stock market well. However, investors in real market often tend to invest in potential stocks with small market value at the early stage of growth, so the reference value of the result of market-value weighting method is limited. On balance, LightGBM has a higher risk reduction effect in a more practical situation.

**4.1.3. Yield Analysis.** In this paper, we calculate the quarterly yield of stocks in the stock pool based on LightGBM under three position allocation methods and use it to track CSI 300 index in China's stock market from 2009 Q4 to 2018 Q1. From Figure 7, it is obvious that the yield obtained by three position allocation methods can outperform the market, and it is more likely to ensure a considerable yield in a bear



market. Moreover, compared with market-value weighted method and equal-weight method, the optimal portfolio constructed by minimum-variance weight method has a stable performance in the actual market. The combination tracking error constructed by market-value weighting method is the largest, and its performance is not as good as the other two methods in the bear market. Therefore, the portfolio constructed by minimum-variance weight method of mean-variance model with CVaR constraints can continuously obtain relatively stable excess income.

**4.1.4. Model Robustness Comparison.** The data structure is different for different stock markets. In order to reflect the influence of different data structures on the model proposed in this paper, we use stochastic simulation technology to verify the robustness. In order to keep these dependency structures [17], this paper uses repeated sampling technology to do random simulation. In order to compare the robustness and generalization ability of models, we randomly select 1 year's original data each time and conduct repeated regression predictions 1000 times, and the results are shown in Figure 8. According to the comparison of curve volatility, LightGBM can still maintain a lower volatility while maintaining a higher goodness of fit; regardless of the accuracy of prediction or the robustness of the algorithm, LightGBM's regression effect on this type of data set is significantly better compared to the other models.

## 5. Conclusion

This paper takes the financial risks and returns of the stock market as the research object and uses the method of machine learning and data mining to build a financial trading strategy system based on LightGBM. During data preprocessing and feature engineering, we construct multiple variables that have been proved to have strong significant features in previous multifactor models. Experimented on the market data of China's A-share listed companies, the prediction model in this system is trained to predict the stock price of next quarter. The method will perform in other stock markets. The variable importance score affecting the next stock price is also generated, and the accuracy of our systems is compared with GLM, DNN, RF, and SVM model. At the same time, VaR, CVaR, and quarterly return of the portfolio based on LightGBM are calculated and compared with the market index.

The results show the following:

- (1) Compared with the traditional linear model, machine learning models do not need to be based on the assumption that the variables are independent and obey the distribution of specific functions, and they have greater advantages in dealing with big data in financial market. The result of LightGBM is 0.798 for  $R$ -squared and 6.1829 for RMSE, which is much better than GLM. The prediction error of LightGBM is also significantly smaller than that of the other machine learning models, which shows that LightGBM has high accuracy.

- (2) Compared with equal-weight method and market-value weight method, the portfolio under minimum-variance weight method of mean-variance model with CVaR constraint has the best risk aversion effect. At the same time, the three position allocation methods can outperform the market and are more likely to ensure a considerable yield in a bear market. In general, the portfolio, constructed by minimum-variance weight method of mean-variance model with CVaR constraint, has the best stability and yield, followed by market-value weighting method and equal-weight method.
- (3) In this paper, we generate feature importance score to find the most important factor affecting the next stock price. The three most influencing factors are the closing price of the current stock, accounts payable, and the growth rate of stock price. Other financial indicators, like the balance of cash and cash equivalents at the beginning of schedule, turnover rate, and cash paid for goods and services, etc., also have great impact on the stock price.

However, there is still room for improvement in this paper: (1) The experimental data in this paper is the quarterly data of stocks, which is of great value for the long-term strategy. In the future, we can try to use the monthly data of stocks or even the daily data. (2) When we allocate the position of stock, we do not think about shorting, and the weights are limited between 0 and 1, so we can try to add the short strategy in the later research. (3) We choose the mean-variance portfolio model with CVaR constraint for position allocation, but the variance itself is inconsistent. In the future, we can consider using the mean-CVaR model to calculate the weight of the portfolio.

## Data Availability

All stock return data are available.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] W. F. Sharpe and F. William, "Capital asset prices: a theory of market equilibrium under conditions of risk," *The Journal of Finance*, vol. 19, no. 3, pp. 425–442, 1964.
- [2] J. Mossin, "Equilibrium in a capital asset market," *Econometrica*, vol. 34, no. 4, pp. 768–783, 1966.
- [3] S. A. Ross, "The arbitrage theory of capital asset pricing," *Journal of Economic Theory*, vol. 13, no. 3, pp. 341–360, 1976.
- [4] E. F. Fama and K. R. French, "The cross-section of expected stock returns," *The Journal of Finance*, vol. 47, no. 2, pp. 427–465, 1992.
- [5] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *Journal of Financial Economics*, vol. 33, no. 1, pp. 3–56, 1993.
- [6] V. T. Datar, N. Y. Naik, and R. Radcliffe, "Liquidity and stock returns: an alternative test," *Journal of Financial Markets*, vol. 1, no. 2, pp. 203–219, 1998.



- [7] J. D. Piotroski, "Value investing: the use of historical financial statement information to separate winners from losers," *Journal of Accounting Research*, vol. 38, pp. 1–41, 2000.
- [8] R. Novy-Marx, "The other side of value: the gross profitability premium," *Journal of Financial Economics*, vol. 108, no. 1, pp. 1–28, 2013.
- [9] G. Aharoni, B. Grundy, and Q. Zeng, "Stock returns and the miller modigliani valuation formula: revisiting the Fama French analysis," *Journal of Financial Economics*, vol. 110, no. 2, pp. 347–357, 2013.
- [10] E. F. Fama and K. R. French, "A five-factor asset pricing model," *Journal of Financial Economics*, vol. 116, no. 1, 2014.
- [11] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [12] M. T. Greene and B. D. Fielitz, "Long-term dependence in common stock returns," *Journal of Financial Economics*, vol. 4, no. 3, pp. 339–349, 1977.
- [13] S.-R. Yang and B. W. Brorsen, "Nonlinear dynamics of daily futures prices: conditional heteroskedasticity or chaos?," *Journal of Futures Markets*, vol. 13, 1993.
- [14] J. Fajardo, A. R. Farias, and J. R. H. Ornelas, "Goodness-of-fit tests focus on value-at-risk estimation," *Brazilian Review of Econometrics*, vol. 26, no. 2, pp. 309–326, 2006.
- [15] B. Mandelbrot, "The variation of certain speculative prices," *The Journal of Business*, vol. 36, no. 4, pp. 394–419, 1963.
- [16] M. Y. Romanovsky, "Truncated levy distribution of sp500 stock index fluctuations. Distribution of one-share fluctuations in a model space," *Physica A*, vol. 287, no. 3-4, pp. 450–460, 2000.
- [17] J. Zhang, Y. T. Xie, and J. Yang, "Risk dependence, consistency risk measurement and portfolio: based on mean-copula-CVaR models," *Journal of Financial Research*, vol. 10, pp. 159–173, 2016.
- [18] A. Fan and M. Palaniswami, "Stock selection using support vector machines," in *Proceedings of the International Joint Conference on Neural Networks. (Cat. No.01CH37222)*, vol. 3, IEEE, Washington, DC, USA, pp. 1793–1798, 2001.
- [19] K. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, 2003.
- [20] Y. T. Xie, J. Yang, and W. Wang, "Comparison analysis on logistic regression and tree models: based on response ratio of credit mail advertising," *Statistics & Information Forum*, vol. 26, no. 6, pp. 96–101, 2011, in Chinese.
- [21] 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.
- [22] B. B. Nair, V. P. Mohandas, and N. R. Sakthivel, "A decision tree- rough set hybrid system for stock market trend prediction," *International Journal of Computer Applications*, vol. 6, no. 9, pp. 1–6, 2010.
- [23] M. Zhu, D. Philpotts, and M. J. Stevenson, "The benefits of tree-based models for stock selection," *Journal of Asset Management*, vol. 13, no. 6, pp. 437–448, 2012.
- [24] M. Kumar and M. Thenmozhi, "Forecasting stock index returns using arima-SVM, arima-ANN, and arima-random forest hybrid models," *International Journal of Banking, Accounting and Finance*, vol. 5, no. 3, p. 284, 2014.
- [25] S. Bogle and W. Potter, "Using hurst exponent and machine learning to build a predictive model for the Jamaica frontier market," in *Transactions on Engineering Technologies*, pp. 397–411, Springer, Singapore, 2016.
- [26] Y. T. Xie and Z. X. Li, "Extension of bonus-malus factor based on joint pricing models," *Statistics & Information Forum*, vol. 30, no. 6, pp. 33–39, 2015.
- [27] H. Yan, "Integrated prediction of financial time series data based on deep learning," *Statistics & Information Forum*, vol. 35, no. 4, pp. 33–41, 2020.
- [28] Z. Tang, R. Zhu, P. Lin et al., "A hardware friendly unsupervised memristive neural network with weight sharing mechanism," *Neurocomputing*, vol. 332, pp. 193–202, 2019.
- [29] Z. Chang, Y. Chen, S. Ye et al., "Fully memristive spiking-neuron learning framework and its applications on pattern recognition and edge detection," *Neurocomputing*, vol. 403, pp. 80–87, 2020.
- [30] G. L. Ke, Q. Meng, T. Finley et al., "LightGBM: a highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3149–3157, 2017.
- [31] P. Jorion, "Risk 2: measuring the risk in value at risk," *Financial Analysts Journal*, vol. 52, no. 6, pp. 47–56, 1996.

## Research Article

# Health Assessment of High-Speed Train Running Gear System under Complex Working Conditions Based on Data-Driven Model

Chao Cheng <sup>1,2,3</sup>, Ming Liu <sup>1</sup>, Bangcheng Zhang <sup>4</sup>, Xiaojing Yin,<sup>4</sup> Caixin Fu,<sup>2</sup> and Wanxiu Teng<sup>2</sup>

<sup>1</sup>School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China

<sup>2</sup>National Engineering Laboratory, CRRC Changchun Railway Vehicles Co., Ltd., Changchun 130062, China

<sup>3</sup>Department of Automation, Tsinghua University, Beijing 100084, China

<sup>4</sup>School of Mechatronic Engineering, Changchun University of Technology, Changchun 130012, China

Correspondence should be addressed to Bangcheng Zhang; zhangbangcheng@ccut.edu.cn

Received 13 April 2020; Revised 15 June 2020; Accepted 23 June 2020; Published 23 July 2020

Guest Editor: Qian Zhang

Copyright © 2020 Chao Cheng 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.

It is very important for the normal operation of high-speed trains to assess the health status of the running gear system. In actual working conditions, many unknown interferences and random noises occur during the monitoring process, which cause difficulties in providing an accurate health status assessment of the running gear system. In this paper, a new data-driven model based on a slow feature analysis-support tensor machine (SFA-STM) is proposed to solve the problem of unknown interference and random noise by removing the slow feature with the fastest instantaneous change. First, the relationship between various statuses of the running gear system is analyzed carefully. To remove the random noise and unknown interferences in the running gear systems under complex working conditions and to extract more accurate data features, the SFA method is used to extract the slowest feature to reflect the general trend of system changes in data monitoring of running gear systems of high-speed trains. Second, slowness data were constructed in a tensor form to achieve an accurate health status assessment using the STM. Finally, actual monitoring data from a running gear system from a high-speed train was used as an example to verify the effectiveness and accuracy of the model, and it was compared with traditional models. The maximum sum of squared resist (SSR) value was reduced by 16 points, indicating that the SFA-STM method has the higher assessment accuracy.

## 1. Introduction

With the continuous improvement of the safety and stability of high-speed trains, the study of the assessment of the health status of the running gear system has received extensive attention in recent years [1–3]. As a critical component to withstand and transmit various loads from the vehicle body and route, while mitigating its dynamic effect, the running gear system is quite prone to failure after a long period of high-speed operation. This requires a health status assessment of the running part to improve its safety and reliability. However, there are many unknown interferences and many noises in the working environment of the running

gear system; thus, it is difficult to accurately assess the status of the running gear system by only using the original monitoring data. Therefore, to reduce the influences of unknown interference and noise under complex conditions and to enhance the accuracy of the assessment, in this study, we designed a data-driven health assessment model to ensure the safety, stability, and reliability operation of running gear systems.

The running gear system of high-speed trains is a complicated electromechanical system composed of many components. Any component may suffer from cracks, corrosion, leakage, and other faults, which can cause its faster degradation [4–6]. The extrusion wear of the wheel

pair and the stiffness degradation of the steel spring will also lead to the degradation of the running gear system. Additionally, the monitoring data collected by the sensor under interference conditions contain noise, which will have a great impact on the health status assessment of the running gear system. Therefore, a health assessment of the running gear system has the following two characteristics:

- (1) There are a lot of random noises caused by sensor factors and unknown interferences caused by internal or external environmental factors in the monitoring data of the running gear system
- (2) Structural complexity due to the close coupling of components in the running gear system is difficult to be described by precise mathematical models through mechanism analysis, which will limit the use of analytical models

Currently, the common methods for health status assessment of complex electromechanical systems are mainly divided into three categories: the method based on semi-quantitative information [7–9], data-driven method [10–13], and model-based method [14–16]. With an increase in the number of sensors in the running gear system, it has become very easy to obtain a large amount of data that can reflect the actual status of the system. Additionally, with the continuous development of feature extraction methods in recent years, data-driven methods have been able to extract feature information from massive amounts of data. Currently, many experts and scholars use data-driven methods to assess the health status of complex electromechanical systems. In [17], studies have combined the odd-even space method with the recursive least square algorithm to propose a fault detection and assessment method for a quadrotor UAV based on a linear time-varying system. With the analytic hierarchy process (AHP), which is a kind of traditional decision-making assessment method [18], Qian et al. [19] have suggested a kind of electric power dispatching control scheme based on health status assessments, and through the AHP, this was carried out to perform a wind turbine health status assessment. This method can improve the operation efficiency of a wind farm to the greatest extent and reduce the fatigue load of the fan fault. Through continuous improvement of the hidden Markov model (HMM) [20, 21], Liu et al. [22] proposed a discrete hidden Markov model (DHMM) based on K-means clustering, and the K-means clustering algorithm was adopted to filter the sample points that are inconsistent with the actual class labels, which could better detect and isolate faults. Recently, deep learning technology has been widely used in status assessment [23, 24], and Liang et al. [25] proposed a convolutional neural network (CNN) method, which can be applied for status assessment of a gearbox. Using a massive amount of training data, a high assessment accuracy can be achieved. However, none of the above methods take into account the problem of accurate extraction of original data features. Too many data features will generate redundant variables and increase the complexity of the algorithm, thus affecting the accuracy of the assessment model.

For this reason, the Hilbert–Huang transform (HHT) and the support vector machine (SVM) have been used in the engine fault intelligent diagnosis method (EFD), which uses HHT to extract features and effectively and provides engine fault diagnosis [26]. Song et al. [27] combined statistical filtering (SF) and the wavelet packet transform (WPT) to propose a new method of signal feature extraction and fault diagnosis for a low-speed mechanical system. Principal component analysis (PCA) is the most classical method used for feature extraction [28], and one study constructed subspaces in different directions using PCA analysis of the principal components, to divide the original feature space into several subspaces automatically, and developed a monitoring and assessment scheme for the model. Moreover, Jiang et al. [29] presented a distributed fault detection and isolation method based on fault-related variable selection and Bayesian reasoning. They use an optimization algorithm to determine the optimal subset of variables for each fault, build a sub-PCA model in each subset, and combine the monitoring results of each subset through Bayesian reasoning. This method significantly reduces the redundancy and complexity and thus improves the monitoring performance. To sum up, there are many status assessment methods based on data that have improved the assessment accuracy after feature extraction, but, for actual work environments, there are many unknown disturbances combined with the considerable noise from the complex electromechanical systems. The traditional feature extraction method is difficult to carry out accurately for feature extraction, which affects the accuracy of the assessment model.

Therefore, a new process monitoring method based on SFA was applied in one study [30, 31] to extract the feature with the slowest change from the original monitoring data, and the process monitoring based on slow data was used to distinguish the deviation between dynamic anomalies and normal operating conditions. The purpose of the SFA method in data processing is to find mapping functions of instantaneous scalar input and output from multidimensional monitoring data so that the output slowness data changes as slowly as possible. Additionally, it carries information reflecting the general trend of the system change and filters the slowness data according to the slowness. The filtered data often represents the short-term noise of complex system changes, and the remaining data more accurately reflect the general trend of system change. Therefore, this paper proposes a SFA-STM method to assess the health status of the running gear systems of high-speed trains. The STM is a classification algorithm based on tensor data, which is very suitable for dealing with nonlinear and non-Gaussian problems. In [32, 33], bearing monitoring data was constructed into a tensor matrix and the method based on STM was applied to fault diagnosis of the bearings. Because the tensor data form retains the space and time form of the original data, the data are fully utilized, which can prevent information loss caused by the multidimensional feature of vectorization, providing a good diagnosis accuracy. In [34], the authors proposed a method based on a hybrid support tensor for the diagnosis and positioning of open circuit faults

of modular multilevel converters, and the classification accuracy was better than that of the support vector machine. Therefore, in this study, we used SFA to extract the slowness data from the monitoring data of the running gear system and to eliminate the noise data. Then, the slowness data reflecting the general trend of the process change was constructed into a tensor matrix and input into the STM model, to enhance the accuracy of STM for the health status assessment of the running gears of high-speed trains. This solves the problem of the many unknown interferences and random noises in monitoring data under actual working conditions, which affect assessment results.

This paper is arranged as follows. Section 2 introduces the structure of the running gear system of a high-speed train and describes the problem. In Section 3, the health assessment model for the running gear system in a high-speed train based on SFA-STM is proposed. Section 4 describes a practical case to verify the method proposed in this paper. Section 5 provides the conclusions of this paper.

## 2. Preliminaries

This part briefly introduces the running department and describes the problems to be solved.

*2.1. Description of Running Gear System.* As shown in Figure 1(a), the running gear system is a complex coupling system composed of multiple components that is located between the high-speed train body and the track and pulls the vehicle along the track. As the core component of a high-speed train, it is mainly composed of a frame, axle box, suspension device, driving device, brake device, and sensor. During the manufacturing phase, many sensors are preinstalled in the running gear system to monitor the status of the running gear system. Each sensor is integrated with multiple sensor units to monitor different physical quantities and collect different types of data, such as temperature, vibrations, and impact. Therefore, a single sensor is also called a composite sensor component. A total of 11 composite sensors are installed in the monitoring bearing support area, and the installation direction should be consistent with the direction of the impact signal. The specific distribution location is shown in Figure 1(b). A1–A4: measurement point of the axle box bearing. B1–B3: motor bearing measurement points (2 is the measurement point of the motor drive bearing and 3 is the measurement point of the motor rotor). C1–C4: gearbox measurement point (3 and 4 are measuring points of small and large gears at the motor end). The distribution of sensors is very complex, and the many sensors can easily obtain considerable monitoring data reflecting the status of the running gear system during the process of train operation. However, because of the complex monitoring environment, the considerable noise and interference factors cannot be ignored; thus, it is particularly important to remove noise and interference from the monitoring data.

To better determine the health status of a running gear system, it is necessary to monitor the temperature, vibrations, impact, speed, and other physical quantities of the

running gear system. For this reason, in this study, we analyzed three subsystems of the running gear system, including the axle box, gearbox, and motor, and we evaluated the status of the running gear system by using the monitoring data from these three subsystems.

*2.2. Problem Description.* When a high-speed train runs at high speed for a long time, the gearbox bearing is prone to cracks and deformation. Additionally, the wheelset of the running gear system will be squeezed and worn when it touches the ground for a long time, and this will be corroded by different environments while staying at a high-temperature status [35]. The air spring also has the possibility of gas leakage, and the stiffness degradation of the shock absorber steel spring can occur. All these factors will cause degradation of high-speed trains. From the distribution of a large number of complex sensors, the running gear system status can be determined based on monitoring data. However, because the complex engineering monitoring environment and running environment are complex, the monitoring data is mixed with large amounts of random noise and uncertainty owing to disturbances. This seriously affects the monitoring data, affecting the accuracy of direction of actual running status, and there are two key reasons for this situation [34]:

- (1) Environmental factors: this is mainly due to the operation environment of high-speed trains and the sensor environment. In the high-speed running of high-speed trains, it is inevitable that they will be affected by road condition factors, interior factors, climate factors, and other disturbances, which cause the sensor monitoring data to fail to reflect the actual status of the running gear system over a short period. This can affect the accuracy of the status assessment of a running gear system.
- (2) Sensor factor: this is mainly determined by the sensor's principle, material, manufacturing process, and other physical characteristics. After long working hours, the quality and function of a sensor will be reduced to different degrees, and a large amount of random noise will be generated, which will continuously affect the monitoring performance. As a result, some errors in the information will be recorded, and accurate monitoring data cannot be obtained, which will seriously affect the accuracy of the status assessment of the running gear system.

According to the above analysis, monitoring data are mainly affected by these two factors, while the traditional feature extraction method has difficulty in removing the influence of uncertainty disturbances and random noise. Thus, a new approach is therefore required to address this issue.

## 3. Methodology

To accurately assess the health status of high-speed train running gear system, a health status assessment model based



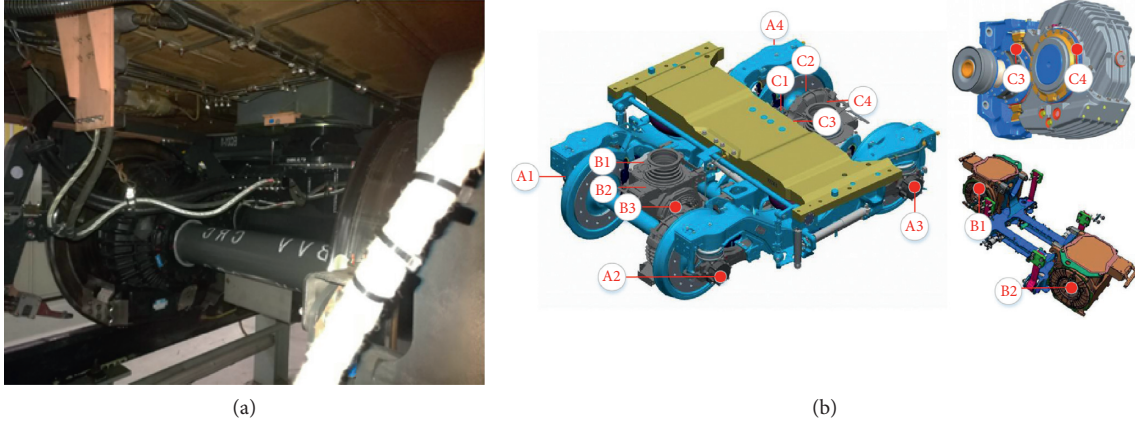


FIGURE 1: Structure diagram of high-speed train running gear system. (a) Real structure diagram of high-speed train running gear and (b) CATIA (computer-aided three-dimensional interactive application) structure diagram of high-speed train running section.

on SFA-STM is proposed. As shown in Figure 2, the model structure is mainly divided into three parts. First, the features reflecting the general trend of system changes are selected as the input for the SFA. Second, the monitoring data processed by SFA is constructed into a tensor matrix. Third, the STM health status assessment model is constructed using the tensor matrix.

**3.1. SFA.** Data reflecting the changing trend of high-speed train running gear systems is covered by random noise and unknown interference, and when these data are used to assess the health status of a running gear system, it will seriously affect the accuracy of the assessment results. Thus, SFA is the most effective method to reduce noise and interference.

The purpose of SFA data processing is to determine the instantaneous scalar input-output mapping function from the multidimensional time input signal so that the change in the output signal is as slow as possible while carrying the information reflecting the general trend of the system change. Mathematically, the purpose of an SFA is for a given  $m$ -dimensional input signal:

$$\mathbf{x}(t) = [x_1(t), \dots, x_m(t)]^T. \quad (1)$$

Find the eigenfunction  $\{g_j(\cdot)\}_{j=1}^m$  of a series of slow feature sets so that

$$\min_{g_j(\cdot)} \langle \dot{s}_j^2 \rangle_t, \quad (2)$$

under constraint

$$\langle s_j \rangle_t = 0 \text{ (zero mean)}, \quad (3)$$

$$\langle s_j^2 \rangle_t = 1 \text{ (unit variance)}, \quad (4)$$

$$\forall i \neq j: \langle s_i s_j \rangle_t = 0 \text{ (decorrelation and order)}, \quad (5)$$

to minimize, where  $\{s_j(t) = g_j(x(t))\}_{j=1}^m$  is a series of slow features, and  $\Delta(s_j) \doteq \langle \dot{s}_j^2 \rangle_t$  is a measure of slowness of  $s_j(t)$ .

$\langle \cdot \rangle_t$  is the time average, and  $\dot{s}$  is the first derivative of  $s$  with respect to time.

For SFA, each slow feature  $s_j(t)$  is a linear combination of input variables:

$$s_j = g_j(\mathbf{x}) = \mathbf{w}_j^T \mathbf{x}, \quad (6)$$

where  $\{\mathbf{w}_j\}_{j=1}^m$  represents the coefficient vector. The mapping from  $\mathbf{x}(t)$  to  $\mathbf{s}(t)$  can be abbreviated as follows:

$$\mathbf{s} = \mathbf{W}\mathbf{x}, \quad (7)$$

where  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_m]^T$  is the coefficient matrix to be optimized by SFA.

Substitute equation (6) into constraint (3), and we obtain

$$\langle s_j \rangle_t = \mathbf{w}_j^T \langle \mathbf{x} \rangle_t = 0. \quad (8)$$

If the input variable  $\mathbf{x}$  is scaled to a zero mean ahead of time, constraint (3) is automatically satisfied.

For the solution of slow feature  $s$ , the following steps are required, after the singular value decomposition (SVD) of the covariance matrix  $\langle \mathbf{x}\mathbf{x}^T \rangle_t$ , and it can be obtained:

$$\langle \mathbf{x}\mathbf{x}^T \rangle_t = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T. \quad (9)$$

Next, the whitening transformation can be expressed as

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \mathbf{x} \doteq \mathbf{Q}\mathbf{x}, \quad (10)$$

where  $\mathbf{Q} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^T$  is the whitening matrix. You can easily get  $\langle \mathbf{z}\mathbf{z}^T \rangle_t = \mathbf{Q} \langle \mathbf{x}\mathbf{x}^T \rangle_t \mathbf{Q}^T = \mathbf{I}$  and  $\langle \mathbf{z} \rangle_t = \mathbf{0}$ , so the purpose of the SFA is to further translate into finding a matrix  $\mathbf{s} = \mathbf{P}\mathbf{z}$  that satisfies  $\mathbf{P} = \mathbf{W}\mathbf{Q}^{-1}$  because

$$\mathbf{s} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{Q}^{-1} \mathbf{z} \doteq \mathbf{P}\mathbf{z}. \quad (11)$$

Then, it can be seen that constraints (4) and (5) can be simply written as follows:

$$\langle \mathbf{s}\mathbf{s}^T \rangle_t = \mathbf{I}. \quad (12)$$

Take equation (11) and bring it to equation (12):

$$\langle \mathbf{s}\mathbf{s}^T \rangle_t = \mathbf{P} \langle \mathbf{z}\mathbf{z}^T \rangle_t \mathbf{P}^T = \mathbf{P}\mathbf{P}^T = \mathbf{I}, \quad (13)$$

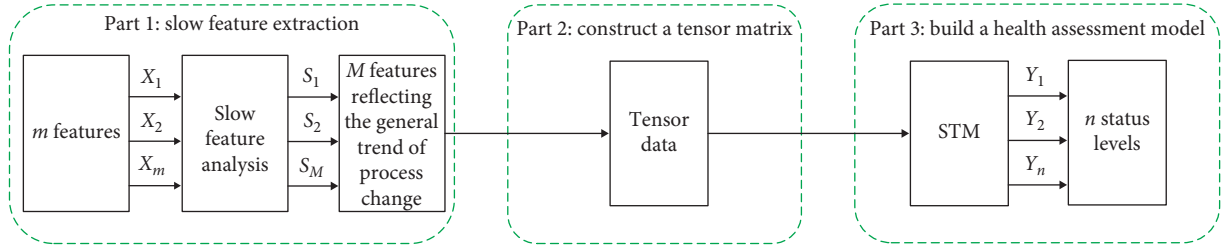


FIGURE 2: Flow chart of the health status assessment model of high-speed train running gear system.

so we know that  $\mathbf{P}$  is an orthogonal matrix. Therefore, the optimization problem of SFA can be simplified to find the orthogonal matrix  $\mathbf{P}$ , thus minimizing  $\mathbf{s} = \mathbf{Pz}$  to  $\langle \dot{s}_j^2 \rangle_t$ . Such problems can be solved by SVD of the covariance matrix  $\langle \dot{z}\dot{z}^T \rangle_t$ :

$$\langle \dot{z}\dot{z}^T \rangle_t = \mathbf{P}^T \mathbf{\Omega} \mathbf{P}. \quad (14)$$

Therefore, the orthogonal eigenvector  $\{P_j\}_{j=1}^m$  and the corresponding eigenvalue  $\{\omega_j\}_{j=1}^m$  can be obtained and verified:

$$\begin{aligned} \langle \mathbf{ss}^T \rangle_t &= \mathbf{P} \langle \mathbf{zz}^T \rangle_t \mathbf{P}^T = \mathbf{I}, \\ \langle \dot{s}_j^2 \rangle_t &= \mathbf{p}_j^T \langle \dot{z}\dot{z}^T \rangle_t \mathbf{p}_j = \omega_j. \end{aligned} \quad (15)$$

Finally, the change matrix  $\mathbf{W}$  can be calculated as

$$\mathbf{W} = \mathbf{P} \mathbf{\Lambda}^{-1/2} \mathbf{U}^T, \quad (16)$$

so the slow feature  $s$  can be calculated as

$$\mathbf{s} = \mathbf{Pz} = \mathbf{P} \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \mathbf{x} = \mathbf{Wx}. \quad (17)$$

The statistical characteristics of the slow feature can be expressed by the following equation:

$$\begin{aligned} E\{\mathbf{s}\} &= \mathbf{0}, \\ E\{\mathbf{ss}^T\} &= \mathbf{I}, \\ E\{\dot{s}\} &= \mathbf{0}, \\ E\{\dot{s}\dot{s}^T\} &= \mathbf{\Omega}. \end{aligned} \quad (18)$$

According to different slowness features, the slowness features are extracted according to the following equation:

$$M = \text{card}\{s_i | \Delta(s_i) < \max_j^q \{\Delta(x_j)\}\}, \quad (19)$$

where  $\text{card}\{\cdot\}$  represents the number of elements in a set and  $\Delta(s_i) = \langle \dot{s}_j^2 \rangle_t$  is the  $s_i(t)$  slowness measure. Set  $q = 0.1$ , and  $\max_j^q \{\Delta(x_j)\}$ , here as the upper quantile of the set  $\{\Delta(x_j)\}$ , is an evaluation criterion for the slowness of  $s_i(t)$ , which can

filter out slowness of the  $M$  largest  $s_i(t)$ , expressed as  $s_d = [s_1, \dots, s_M]^T$ . These slow features reflect the general trend of system changes in the running gear system. For the  $M_e = m - M$  slow features screened out, they represent the short-term noise and unknown disturbance of the system changes. The above proofs have been given in [36].

**3.2. STM Model.** Using the SFA model in the previous section, data reflecting the general trend of the system change of the running gear system was successfully extracted. Then, the nonlinear model between the input and output needs to be built by the STM, to achieve an accurate assessment of the health status of running gear systems of high-speed trains.

Consider a dichotomy problem, where the training set is

$$\begin{aligned} T &= \{(\chi_1, y_1), (\chi_2, y_2), \dots, (\chi_l, y_l)\}, \\ y_i &\in \{+1, -1\}, \end{aligned} \quad (20)$$

where  $l$  represents the number of samples,  $\chi_i \in R^{I_1 \times I_2 \times \dots \times I_N}$  represents the  $i$  input data,  $y_i$  represents the class label of the  $i$  data, and  $y_i \in \{+1, -1\}$ . The optimization problem of the STM model is

$$\begin{aligned} \min_{w^{(n)}|_{n=1}, b, \xi} J(w^{(n)}|_{n=1}, b, \xi) &= \frac{1}{2} \left\| \sum_{n=1}^N w^{(n)} \right\|_F^2 + C \sum_{i=1}^l \xi_i, \\ \text{s.t. } y_i \left( \chi_i \prod_{n=1}^N x_n w^{(n)} + b \right) &\geq 1 - \xi_i, \end{aligned} \quad (21)$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, l.$$

Based on the idea of supervised tensor learning framework, the STM dichotomy model is decomposed into  $N$  suboptimization problems, and the expression form of one of its suboptimization problems after fixing  $\{w^{(m)}\}_{m=1}^N$  and  $m \neq n$  is

$$\begin{aligned} \min_{w^{(n)}, b^{(n)}, \xi^{(n)}} J(w^{(n)}, b^{(n)}, \xi^{(n)}) \\ = \frac{1}{2} \left\| w^{(n)} \right\|_F^2 \prod_{1 \leq m \leq N}^{m \neq n} \left\| w^{(m)} \right\|_F^2 + C \sum_{i=1}^l \xi_i^{(n)}, \text{ s.t. } y_i \left( \left( w^{(n)} \right)^T \left( \chi_i \prod_{1 \leq m \leq N}^{m \neq n} x_m w^{(m)} \right) + b^{(n)} \right) \geq 1 - \xi_i^n, \xi_i^{(n)} \geq 0, \quad i = 1, 2, \dots, l. \end{aligned} \quad (22)$$



In equation (22) above,  $w^{(n)} \in R^{I_n}$  ( $1 \leq n \leq N$ ) is the weight of the  $n$  hyperplane,  $b^{(n)}$  is the offset,  $b^{(n)} \in R$  and  $\xi^{(n)}$  are the relaxation variables, and  $C$  is the penalty factor. What we need to solve are  $N$  optimization problems with a similar form to equation (22); for these  $N$  problems, we can solve them through the optimization method of alternating projection and finally obtain the decision function of the model as

$$D(\chi) = \operatorname{sgn} \left( \chi \prod_{n=1}^N w^{(n)} + b \right). \quad (23)$$

The alternating projection algorithm requires the input tensor to train the sample set  $\{\chi_i \in R^{I_1 \times I_2 \times \dots \times I_N}, y_m \in \{1, -1\}\}_{i=1}^l$ , the iterative control threshold  $\varepsilon$ , and the output  $w^{(n)} \in R^{I_n}$  ( $1 \leq n \leq N$ ),  $b \in R$ .

Step 1:  $w^{(n)}$  is initialized as the unit vector in  $R^{I_n}$ ,  $n = 1, 2, \dots, N$ .

Step 2: iterate through steps 3 and 4 until the algorithm converges.

Step 3: iterate  $w^{(n)}$  for  $n = 1, 2, \dots, N$ , fix  $\{w^{(m)}\}_{m=1}^N$ ,  $m \neq n$ , solve equation (22), and get  $w_n$ .

Step 4: stop the calculation if

$$\sum_{n=1}^N \left( \left| (w_t^{(n)})^T w_{t-1}^{(n)} \right| \left\| w_t^{(n)} \right\|_F^{-2} - 1 \right) < \varepsilon. \quad (24)$$

is met. Otherwise, go to step 2, where  $w_i^{(n)}$  and  $w_{t-1}^{(n)}$  represent the values of the previous iteration step and the current iteration step corresponding to the  $n$  projection weight vector, respectively.

For the STM multiclassification model, the running gear system status label  $L = \{L_1, L_2, \dots, L_s\}$  is defined, and the label is composed of  $s$  information classes. One-against-one (OAO) strategy is adopted to construct  $S(S-1)/2$  binary STM models to model all possible paired classifications. Then, the decision function  $D_{i,j}(\chi)$  between each possible pair of classes  $L_i$  and  $L_j$  ( $i \neq j$ ) is obtained through dichotomy, and the fractional function

$$\text{Score}_j = \sum_{i=1, i \neq j}^s D_{i,j}(\chi). \quad (25)$$

of the sum of the number of labels allocated to the category by the sample is calculated. The classes with the highest scores were considered predictive labels for unclassified samples.

**3.3. The Steps of Health Status Assessment Model for High-Speed Train Running Gear System.** As shown in Figure 3, according to the algorithm described above, the steps to summarize the health status assessment algorithm of the high-speed train running gear system are summarized as follows:

Step 1: use SFA to extract the slow feature from the high-speed train running gear system monitoring data.

Step 1.1: normalize the sample data.

Step 1.2: whiten the data through equations (9) and (10).

Step 1.3: take the derivative of the matrix  $z$ .

Step 1.4: equations (14) and (16) are used to obtain the mapping matrix.

Step 1.5: slowness data  $s$  is obtained by equation (17).

Step 1.6: the slow feature is extracted according to equation (19).

Step 2: the filtered slowness data matrix of the running gear system is constructed into the fourth-order tensor form, as shown in Figure 4.

Step 3: the health status assessment model of the high-speed train running gear system based on STM was constructed.

Step 4: the STM multiclassification model is transformed into multiple binary classification models.

Step 4.1: the optimization problem of STM is divided into  $N$  suboptimization problems.

Step 4.2: the  $N$  suboptimization problems were solved by alternating projection algorithm, and the decision function  $D_{i,j}(\chi)$  was obtained.

Step 5: through equations (23) and (25), the system was assessment.

In the assessment model, the motor temperature, gearbox vibration, and impact of external ring friction of the retaining frame in the axle box were set as the features to assess the health of the running gear system. The specific assessment process will be described through cases in Section 4.

## 4. Case Study

To verify the accuracy of the SFA-STM assessment model proposed in this paper, this section will describe the high-speed train running gear system as an example for experimental verification. For data collection of the running gear system, the axle box, gearbox, and motor monitoring data of No. 2 carriage and position running gear system of the train during a certain month were selected. To ensure that the collected data was the data when the high-speed train was in operation, the monitoring data with speeds above 1000 r/min were screened to verify the training data and test data for the model, and a total of six monitoring indicators, including temperature, vibrations, and impact were selected for positions 1 and 2, respectively. Owing to the influence of the operating environment on the running gear system of high-speed trains during actual operation and considering the complexity of parameters, such as weight, center of gravity, and the suspension of each component, the health status assessment criteria for the running gear system in the high-speed train were set as "normal," "general," and "bad," as shown in Table 1.

*Normal:* under this condition, all parts of the running gear system worked normally, fasteners were stable and not loose, and all indicators were within factory

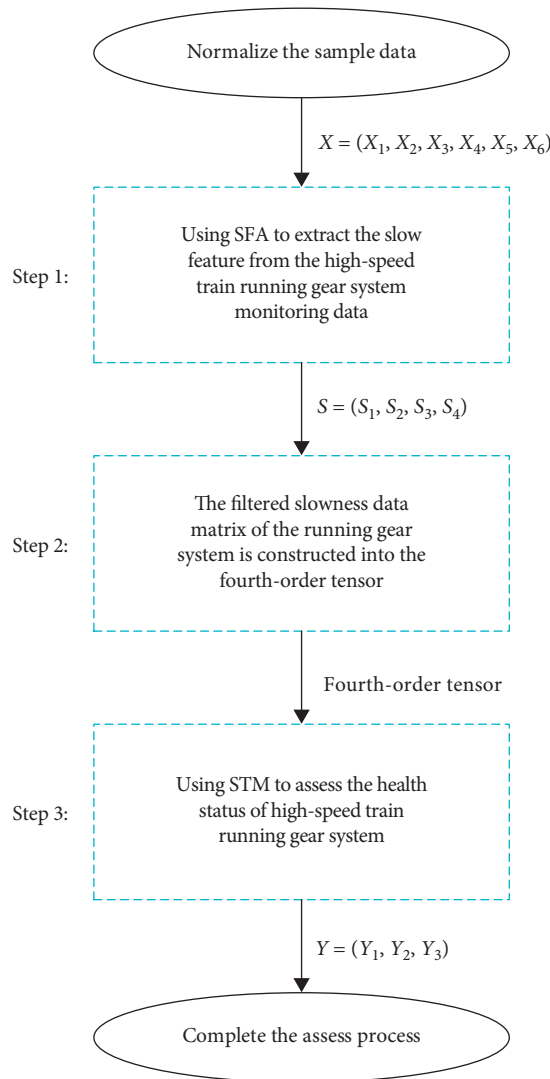


FIGURE 3: The steps of the health status assessment model for the high-speed train running gear system.

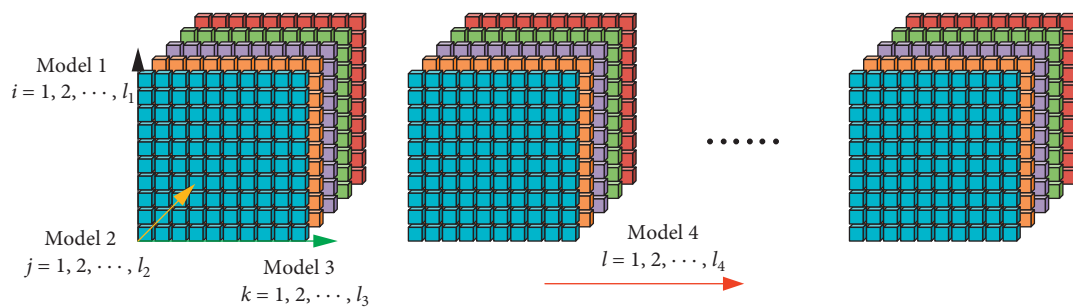


FIGURE 4: Schematic diagram of the fourth-order tensor model.

requirements. The average temperature was less than 30°C, the average vibration peak was less than 22 Hz, the average temperature and amplitude were at a very low levels, and the impact data were also at normal levels, thus ensuring the safe and smooth operation of the high-speed train.

*General:* under this condition, all parts of the running gear system still worked normally, but the parts were slightly worn and deformed, and all indicators were slightly higher than the factory requirements. The average temperature was in the range of 30–35°C, and the average vibration peak was in the range of 22–25 Hz. Under this condition, the

TABLE 1: The referential points.

Status	Normal	General	Bad
Temperature	<30°C	30–35°C	>35°C
Vibration	<22 Hz	22–25 Hz	>25 Hz

operation of the high-speed train is still unaffected, but the appearance, working status, and overall performance of each part should be checked, and preventive and corrective maintenance work should be performed promptly.

*Bad:* under this condition, although all parts of the running gear system could still maintain normal operation, the status of some parts was not good, the operation function decreased, and the critical point of health and failure had been reached. The average temperature was higher than 35°C, and the average vibration peak was above 25 Hz. Under this condition, the system should be completely overhauled to replace or repair the parts with a higher degree of damage to improve the safety of the high-speed train.

**4.1. Data Preprocessing.** Because there are a large number of abnormal environmental noise points in the actual monitoring data of high-speed train running gear systems, it is necessary to preprocess the data. As shown in Figure 5, among them, the short-term fluctuations of temperature, vibration, and shock data are large, and there is much repeated data and abnormal data. Therefore, as shown in Figure 6, the temperature vibration and impact monitoring data of the moving parts at certain positions on the train I and II were preprocessed. The relevant outliers were filtered via mean filtering, and the data were finally compressed to 800 groups. The trend chart is shown in Figure 7.

The temperature sensor was in close contact with the motor element; thus, the external interference was relatively small, whereas the vibration and impact sensors showed more obvious uncertainty in disturbances and random noise owing to the influence of the train speed and acceleration. Although the monitoring data was preprocessed to filter out some of the abnormal value points, the data still included the noise and disturbances as adverse factors, and this could not simply solved by pretreatment. Thus, we needed the following SFA method for further data processing so that the data actually reflected the high-speed train health status.

**4.2. Assessment Model.** The monitoring data of temperature, vibration, shock, and a total of six characteristics at position I and II were normalized. The slowness data is obtained through the SFA, and the coefficient matrix of SFA optimization is

$$W = \begin{pmatrix} -10.37 & -8.42 & -0.64 & 0.79 & 7.28 & -3.35 \\ 1.65 & 1.66 & 0.58 & -3.15 & -6.92 & 4.06 \\ -13.65 & 4.02 & -0.47 & -0.37 & -0.11 & -0.13 \\ 27.59 & 4.14 & 0.36 & -2.91 & -0.03 & -0.22 \\ -4.78 & -0.34 & -8.85 & 2.69 & 0.39 & 0.34 \\ -0.60 & -0.98 & 9.04 & 2.94 & -0.12 & 0.18 \end{pmatrix}, \quad (26)$$

through equation (19), the feature,  $\Delta(s_j)$ , was smaller than the maximum  $\Delta(x_j)$  of 0.1 times, which was selected to screen out four slow features that reflected the overall changing trend of high-speed trains.

After SFA, 800 sets of monitoring data of six features were transformed into 800 sets of slowness data of four features; 800 sets of data were constructed into the fourth-order tensor model of 400 sets of data. Among them, mode 1 of the tensor is the first slow feature and the second slow feature. The mode 3 of the tensor is the third slow feature and the fourth slow feature. Mode 2 is the slowness data of the adjacent two mileage. Mode 4 is 400 sets of the slowness data. The abovementioned information is shown in Figure 8.

The 400 groups of the fourth-order tensor data, which have been constructed, were divided into 150 training data and 250 test data; then, 150 groups of training data were divided into three groups according to the actual status to construct the training set of the STM model. Finally, the decision functions  $D_{1,2}(\chi)$ ,  $D_{1,3}(\chi)$ , and  $D_{2,3}(\chi)$  between the status of each class were obtained through the STM dichotomy model. The corresponding minimization of energy function is  $J_{1,2} = 0.00029988$ ,  $J_{1,3} = 0.00029980$ , and  $J_{2,3} = 0.00029987$ , respectively.

To verify the accuracy of the model, 250 groups of samples to be tested were input into the three decision functions obtained in the STM model, and the class with the highest score for each of them, namely, the prediction label, was calculated through equation (25), to complete the health status assessment of the running gear system.

As shown in Figure 9, the test samples can be accurately classified. Although few samples were misclassified, it can be concluded that the SFA-STM model can accurately assess the health status of the high-speed train running gear system.

**4.3. Simulation.** To more intuitively represent the reliability of the model presented in this paper, four classical data-driven methods (i.e., SVM, naive Bayes (NB), BP neural network, and hidden Markov model (HMM)) are used to compare and analyze this method. The data adopted by the three methods are the same as the data applied by the method proposed in this paper. All of them are 800 groups of monitoring data processed by mean filtering. However, because the training samples and test samples used in the method proposed in this paper are a total of 400 sets of data, to reflect the authenticity of comparative simulation, it is necessary to take the mean value of each adjacent mileage point in the 800 sets of monitoring data and reduce it to approximately 400 sets of training data. This process does not affect the accuracy of the evaluation model because, in engineering practice, two adjacent mileage points almost reflect the health status of the same running gear system. At the same time, the assessment grades of health status of the three models were also “normal,” “general,” and “bad,” with 150 sets of training data and 250 sets of test data.

SVM is used for comparative validation, as shown in Figure 10(a). In the beginning, the green line fits the blue line well, but in the middle, a large number of green points leave the blue line; in the end, the green line fits the blue line well.

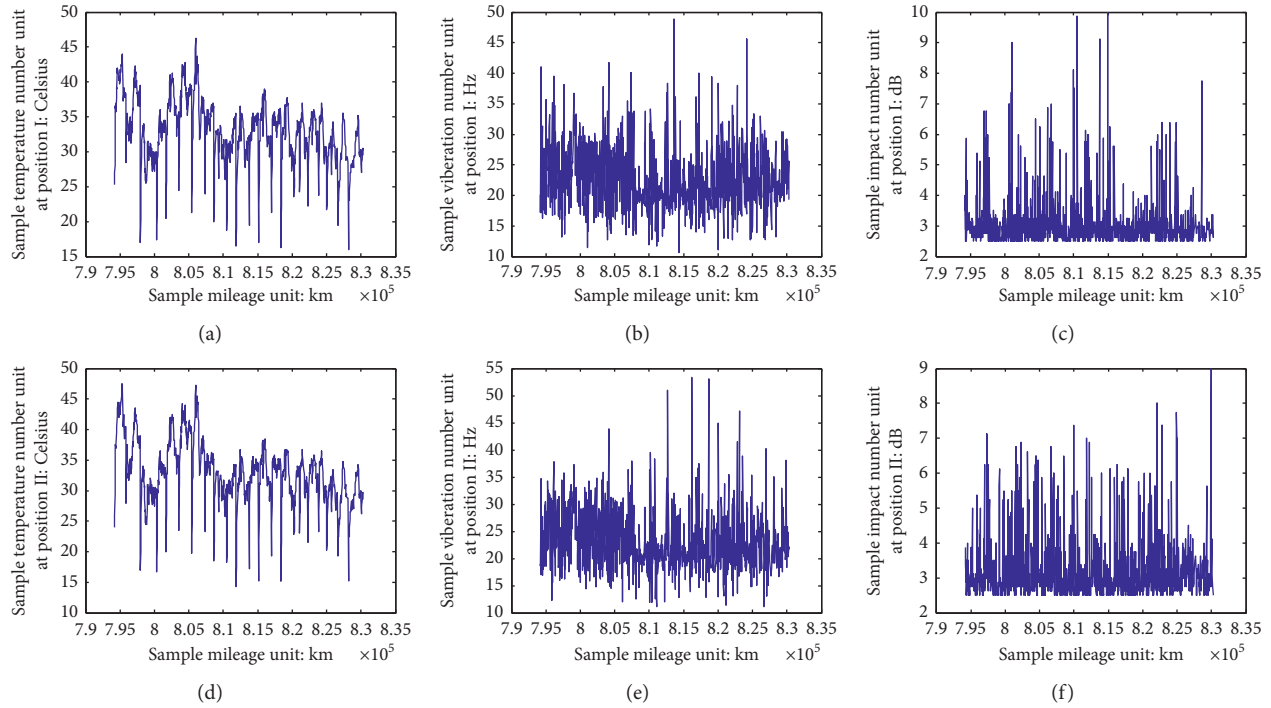


FIGURE 5: Trend chart of original monitoring data of the running gear system. (a) Position I temperature data trend chart. (b) Position I vibration data trend chart. (c) Position I impact data trend chart. (d) Position II temperature data trend chart. (e) Position II vibration data trend chart. (f) Position II impact data trend chart.



FIGURE 6: Actual position of the running gear system.

This indicates that the running gear system can be accurately assessed by the SVM model under normal and bad statuses, while the running gear system cannot be accurately assessed by the model in general status and at the beginning of bad status, which is a considerable limitation.

NB is used for comparative validation, as shown in Figure 10(b). From the fitting degree of the green line to the blue line in the figure, the NB model is still the same as the SVM model, which has many misjudgments in the middle status of the system and also has a certain degree of

misjudgments in good and bad statuses. Compared with the SFA-STM model, the NB model still has difficulty accurately assessing the health status of the system.

The BP neural network is used for comparative validation, as shown in Figure 10(c). The fluctuation of the green line always follows that of the blue line. Although there is a certain degree of abnormal point fluctuation, the overall level of the BP neural network model is more accurate than that of the SVM model and the NB model in assessing the health status of the running gear system. However, the

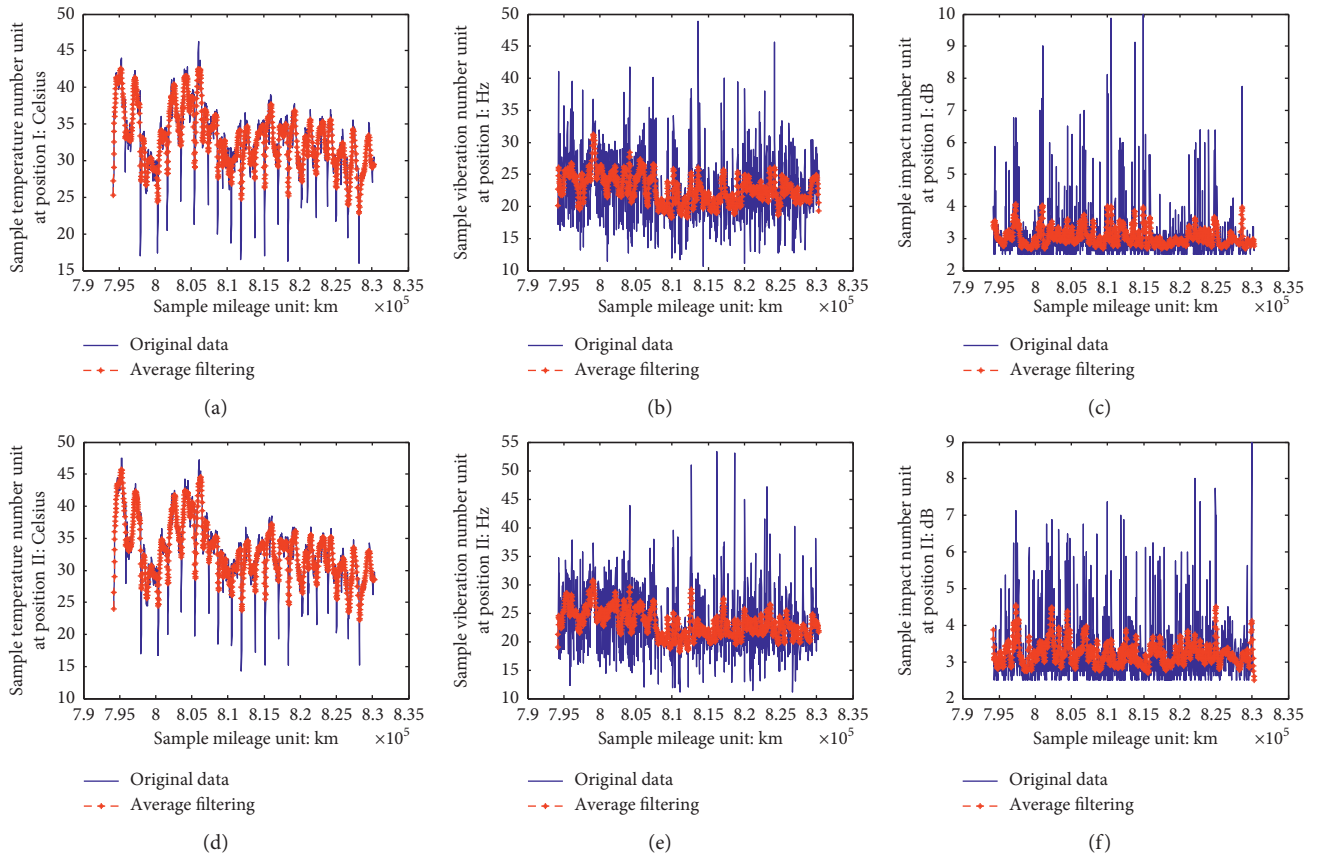


FIGURE 7: Trend chart of mean filter monitoring data. (a) Average filtering position I temperature data trend chart. (b) Average filtering position I vibration data trend chart. (c) Average filtering position I impact data trend chart. (d) Average filtering position II temperature data trend chart. (e) Average filtering position II vibration data trend chart. (f) Average filtering position II impact data trend chart.

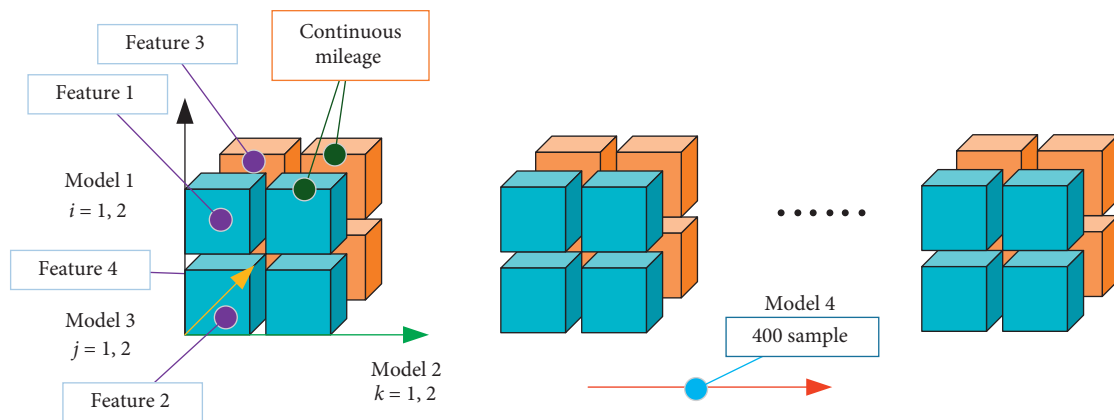


FIGURE 8: Schematic diagram of the tensor model.

abovementioned figure clearly shows the red line of the SFASTM model fits the blue line to a higher degree, which more accurately and intuitively reflects the health status of the system.

HMM is used for comparative validation, as shown in Figure 10(d). From the fitting degree of the green line to the blue line in the figure, the HMM model is still the same as the SVM model, which has a large number of misjudgments in the middle status of the system and also has a certain

degree of misjudgments in good and bad statuses. Because both the NB and HMM models are based on Bayesian classification algorithms, it is not surprising to obtain the same classification results as NB. In general, the HMM model still has difficulty accurately accessing the health status of the running gear system compared with the SFASTM model.

To directly compare the accuracy of the five methods, three indexes are used for comparison. First, the maximum



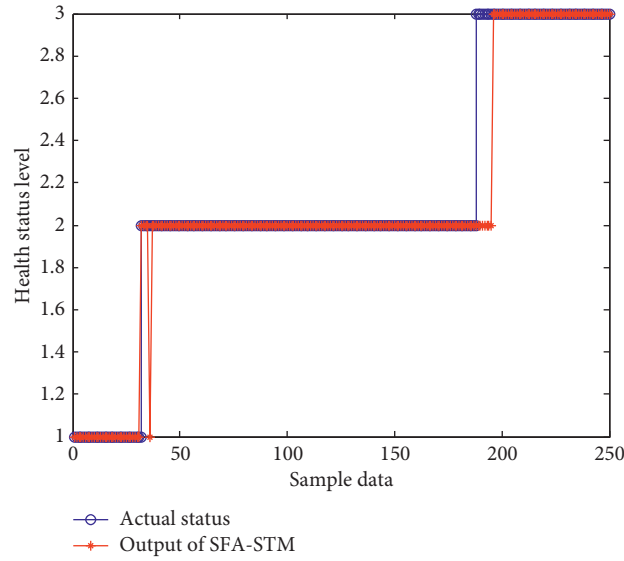


FIGURE 9: The SFA-STM model.

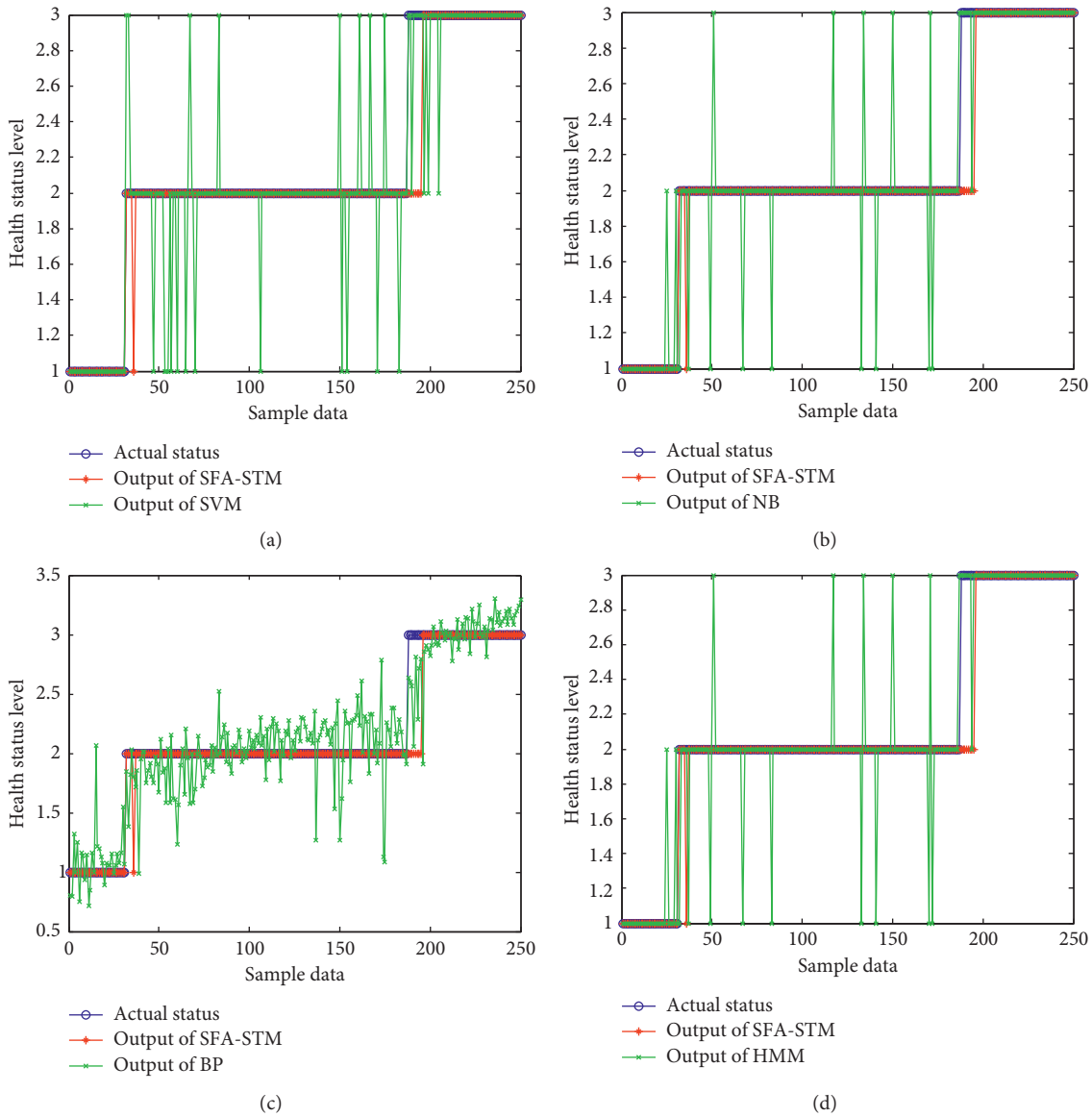


FIGURE 10: Simulation results of the (a) SVM model, (b) NB model, (c) BP model, and (d) HMM model.



TABLE 2: SSR values of the four models.

Model	SSR
SFA-STM	9
SVM	18
NB	25
BP	18.9135
HMM	25

TABLE 3: Table of the training time.

Model	Time (s)
SFA-STM	7.55
SVM	6.43
NB	0.73
BP	0.43
HMM	0.30

TABLE 4: TPR and FPR indicators.

Method	Normal		General		Bad	
	TPR (%)	FPR (%)	TPR (%)	FPR (%)	TPR (%)	FPR (%)
SFA-STM	<b>100</b>	<b>0.47</b>	<b>99.36</b>	8.51	87.3	<b>0</b>
SVM	93.55	4.25	90.38	<b>3.19</b>	<b>98.41</b>	3.41
NB	<b>100</b>	2.02	86.54	4.26	93.65	4.6
HMM	<b>100</b>	2.02	86.54	4.26	93.65	4.6

sum of the squared residuals (SSR) index is adopted, as shown in Table 2. According to equation (27), where  $y_i$  represents the status value obtained from the assessment of  $i$  sample according to the model,  $\hat{y}_i$  represents the actual status value of the  $i$  sample, and  $n$  represents the number of samples. The SSR index of the four methods is given, among which the SSR value of SFA-STM is the lowest, only 9; the SSR value of SVM is 18; the SSR values of NB and HMM are 25, and the SSR value of BP is 18.9135 [37]. Second, the true positive rate (TPR) and false positive rate (FPR) indexes were used to assess the accuracy of classification. Table 3 shows that the SFA-STM model still has the best effect [38]. In the end, as shown in Table 4, compared with the training time of each model, the training time is relatively long because both SFA-STM and SVM adopt the optimization algorithm. In general, it is seen that the SFA-STM model has the highest accuracy in assessing the health status of the high-speed train running gear system.

$$\text{SSR} = \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (27)$$

## 5. Conclusion

In this study, the SFA-STM model is proposed to assess the health status of the high-speed train running gear system, which solves the problem that there are many unknown

interferences and random noise in the monitoring data under complex working conditions, which affect the assessment results. Using an example, it is shown that the SFA-STM system proposed in this study can accurately reflect the actual health status of the running gear system. Compared with the four types of traditional data-driven models, this method has higher applicability to practical engineering problems and provides a new solution to the problem that the general trend of system changes cannot be extracted by the traditional feature extraction method to assess the health status of the complex system under high noise and multiple disturbances.

## Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61903047, 61973046, and 61803044 and Jilin Province Development and Reform Commission under Grant no. 2019C040-3.

## References

- [1] M. Torabi, S. G. M. Mousavi, and D. Younesian, "A high accuracy imaging and measurement system for wheel diameter inspection of railroad vehicles," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8239–8249, 2018.
- [2] H. Chen and B. Jiang, "A review of fault detection and diagnosis for the traction system in high-speed trains," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 450–465, 2020.
- [3] H. Chen, B. Jiang, N. Lu, and W. Chen, *Data-driven Detection and Diagnosis of Faults in Traction Systems of High-Speed Trains*, Springer Nature, New York, NY, USA, 2020.
- [4] C. Yang, C. Yang, T. Peng, X. Yang, and W. Gui, "A fault-injection strategy for traction drive control systems," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5719–5727, 2017.
- [5] Z. Chen, S. X. Ding, T. Peng, C. Yang, and W. Gui, "Fault detection for non-Gaussian processes using generalized canonical correlation analysis and randomized algorithms," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1559–1567, 2018.
- [6] H. Chen, B. Jiang, W. Chen, and Z. Li, "Edge computing-aided framework of fault detection for traction control systems in high-speed trains," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1309–1318, 2020.
- [7] L. Jin, C. Peng, and T. Jiang, "System-level electric field exposure assessment by the fault tree analysis," *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, no. 4, pp. 1095–1102, 2017.

- [8] F. Sihombing and M. Torbol, "Parallel fault tree analysis for accurate reliability of complex systems," *Structural Safety*, vol. 72, pp. 41–53, 2018.
- [9] Y. Dong, J. Zhang, Z. Li, Y. Hu, and Y. Deng, "Combination of evidential sensor reports with distance function and belief entropy in fault diagnosis," *International Journal of Computers Communications & Control*, vol. 14, no. 3, pp. 329–343, 2019.
- [10] H. Ji, X. He, J. Shang, and D. Zhou, "Incipient fault detection with smoothing techniques in statistical process monitoring," *Control Engineering Practice*, vol. 62, pp. 11–21, 2017.
- [11] H. Chen, B. Jiang, and N. Lu, "Data-driven incipient sensor fault estimation with application in inverter of high-speed railway," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8937356, 13 pages, 2017.
- [12] H. Chen, B. Jiang, W. Chen, and H. Yi, "Data-driven detection and diagnosis of incipient faults in electrical drives of high-speed trains," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4716–4725, 2019.
- [13] W. Deng, R. Yao, H. Zhao, X. Yang, and G. Li, "A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm," *Soft Computing*, vol. 23, no. 7, pp. 2445–2462, 2017.
- [14] C. Cheng, W. Wang, H. Luo, B. Zhang, G. Cheng, and W. Teng, "State-degradation-oriented fault diagnosis for high-speed train running gears system," *Sensors*, vol. 20, no. 4, p. 1017, 2020.
- [15] H. Luo, M. Krueger, T. Koenings, S. X. Ding, S. Dominic, and X. Yang, "Real-time optimization of automatic control systems with application to BLDC motor test rig," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4306–4314, 2017.
- [16] Y. Hagh, R. Asl, and V. Cocquempot, "A hybrid robust fault tolerant control based on adaptive joint unscented Kalman filter," *ISA Transactions*, vol. 66, pp. 262–274, 2017.
- [17] W. Han, Z. Wang, and Y. Shen, "Fault estimation for a quadrotor unmanned aerial vehicle by integrating the parity space approach with recursive least squares," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 232, no. 4, pp. 783–796, 2018.
- [18] Y. Liu, W. Ni, and Z. Ge, "Fuzzy decision fusion system for fault classification with analytic hierarchy process approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 166, pp. 61–68, 2017.
- [19] P. Qian, X. Ma, D. Zhang, and J. Wang, "Data-driven condition monitoring approaches to improving power output of wind turbines," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6012–6020, 2019.
- [20] S. Wang, J. Xiang, Y. Zhong, Y. Zhou, and Y. Zhou, "Convolutional neural network-based hidden Markov models for rolling element bearing fault identification," *Knowledge-Based Systems*, vol. 144, pp. 65–76, 2018.
- [21] Y. Gao, F. Villecco, M. Li, and W. Song, "Multi-scale permutation entropy based on improved LMD and HMM for rolling bearing diagnosis," *Entropy*, vol. 19, no. 4, p. 176, 2017.
- [22] J. Liu, Q. Li, W. Chen, and T. Cao, "A discrete hidden Markov model fault diagnosis strategy based on K-means clustering dedicated to PEM fuel cell systems of tramways," *International Journal of Hydrogen Energy*, vol. 43, no. 27, pp. 12428–12441, 2018.
- [23] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, 2017.
- [24] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.
- [25] K. Liang, N. Qin, D. Huang, and Y. Fu, "Convolutional recurrent neural network for fault diagnosis of high-speed train bogie," *Complexity*, vol. 2018, Article ID 4501952, 13 pages, 2018.
- [26] Y. S. Wang, Q. H. Ma, Q. Zhu, X. T. Liu, and L. H. Zhao, "An intelligent approach for engine fault diagnosis based on Hilbert-Huang transform and support vector machine," *Applied Acoustics*, vol. 75, pp. 1–9, 2014.
- [27] L. Song, H. Wang, and P. Chen, "Vibration-based intelligent fault diagnosis for roller bearings in low-speed rotating machinery," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 4, pp. 1887–1899, 2018.
- [28] Z. Ge and Z. Song, "Distributed PCA model for plant-wide process monitoring," *Industrial and Engineering Chemistry Research*, vol. 52, no. 5, pp. 1775–2194, 2013.
- [29] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 377–386, 2016.
- [30] J. Huang, O. K. Ersoy, and X. Yan, "Slow feature analysis based on online feature reordering and feature selection for dynamic chemical process monitoring," *Chemometrics and Intelligent Laboratory Systems*, vol. 169, pp. 1–11, 2017.
- [31] S. Zhang and C. Zhao, "Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3773–3783, 2019.
- [32] Z. He, J. Cheng, J. Li, and Y. Yang, "Linear maximum margin tensor classification based on flexible convex hulls for fault diagnosis of rolling bearings," *Knowledge-Based Systems*, vol. 173, pp. 62–73, 2019.
- [33] L. Bo, G. Xu, X. Liu, and J. Lin, "Bearing fault diagnosis based on subband time-frequency texture tensor," *IEEE Access*, vol. 7, pp. 37611–37619, 2019.
- [34] C. Li, Z. Liu, Y. Zhang, L. Chai, and B. Xu, "Diagnosis and location of the open-circuit fault in modular multilevel converters: an improved machine learning method," *Neurocomputing*, vol. 331, pp. 58–66, 2019.
- [35] J. Yang, Y. Yue, R. Zhu, W. Chen, and M. Li, "Dynamic characteristics of encased differential gear train with journal bearing," *Mathematical Problems in Engineering*, vol. 2020, Article ID 2436191, 15 pages, 2020.
- [36] C. Shang, F. Yang, X. Gao, X. Huang, J. A. K. Suykens, and D. Huang, "Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis," *AIChE Journal*, vol. 61, no. 11, pp. 3666–3682, 2015.
- [37] F. Villecco, "On the evaluation of errors in the virtual design of mechanical systems," *Machines*, vol. 6, no. 3, p. 36, 2018.
- [38] A. Formato, D. Guida, D. Ianniello, F. Villecco, T. Lenza, and A. Pellegrino, "Design of delivery valve for hydraulic pumps," *Machines*, vol. 6, no. 4, p. 44, 2018.

## Research Article

# A Scene Text Detector for Text with Arbitrary Shapes

Weijia Wu,<sup>1</sup> Jici Xing,<sup>2</sup> Cheng Yang,<sup>1</sup> Yuxing Wang <sup>1</sup> and Hong Zhou <sup>1</sup>

<sup>1</sup>Zhejiang University, Key Laboratory for Biomedical Engineering of Ministry, Hangzhou, China

<sup>2</sup>Zhengzhou University, School of Information Engineering Institute, Zhengzhou, China

Correspondence should be addressed to Yuxing Wang; wangyuxing@zju.edu.cn and Hong Zhou; zhouhong\_zju@126.com

Received 11 February 2020; Revised 28 April 2020; Accepted 6 May 2020; Published 11 June 2020

Guest Editor: Qian Zhang

Copyright © 2020 Weijia Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of text detection is crucial for the subsequent recognition task. Currently, the accuracy of the text detector still needs further improvement, particularly those with irregular shapes in a complex environment. We propose a pixel-wise method based on instance segmentation for scene text detection. Specifically, a text instance is split into five components: a Text Skeleton and four Directional Pixel Regions, then restoring itself based on these elements and receiving supplementary information from other areas when one fails. Besides, a Confidence Scoring Mechanism is designed to filter characters similar to text instances. Experiments on several challenging benchmarks demonstrate that our method achieves state-of-the-art results in scene text detection with an F-measure of 84.6% on Total-Text and 86.3% on CTW1500.

## 1. Introduction

Detecting text in the real world is a fundamental computer vision task that directly determines the subsequent recognition results. Many applications in the real world depend on accurate text detection, such as photo translation [1] and autonomous driving [2]. Now, horizontal- [3–5] and oriented-[6–10] based methods no longer meet our requirements, and more flexible pixel-wise detectors [11, 12] have become mainstream. However, precisely locating text instances is still a challenge because of arbitrary angles, shapes, and complex backgrounds.

The first challenge involves text instances with irregular shapes. Unlike other common objects, the shaped instance often cannot be accurately described by a horizontal box or an oriented quadrilateral. Some typical methods (e.g., EAST [8] and TextBox++ [10]) perform well on the common benchmarks (e.g., ICDAR 2013 [13] and ICDAR 2015 [14]) but degrade in curved text challenges, as shown in Figure 1(a).

The second challenge is separating text character boundaries. Although pixel-wise methods do not suffer from a certain shape, they may still fail to separate text areas with adjacent edges, as shown in Figure 1(b).

The third challenge is that text identification may face false positives [15] dilemma because of the lack of context information. Some symbols or characters similar to text may be misclassified.

To overcome the aforementioned challenges, we propose a novel method, called TextCohesion. As shown in Figure 2, our method treats a text instance as a combination of a Text Skeleton and four Directional Pixel Regions, where the previous one roughly represents the shape and profile, and the latter is responsible for refining the original region from four directions. Notably, a pixel belongs to more than one Directional Pixel Regions (e.g., up, left), which means the instance has more chances to be recovered. Furthermore, the confidence score of every Text Skeleton is reviewed, only higher than a threshold is considered as a candidate.

## 2. Related Work

Detecting text in the wild has been widely studied in the past few years. Before deep learning era, most detectors adopt Connected Components Analysis [16–21] or Sliding Window-based classification [22–25].

Now, detectors are mainly based on deep neural networks. There are two main trends in the field of text



FIGURE 1: Text detection challenges: (a) bounding box-based methods suffer from a fixed shape and (b) segmentation-based methods may not separate texts with adjacent boundaries.

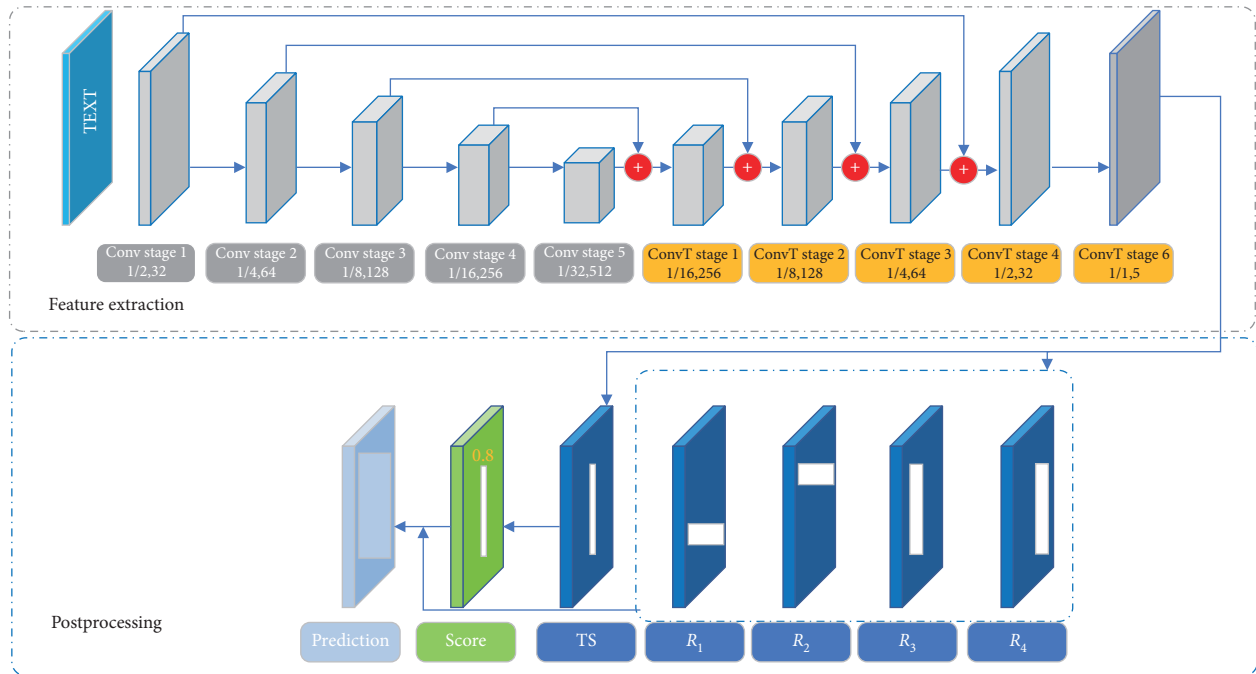


FIGURE 2: The overall procedure of the proposed method consists of Feature Extraction and Postprocessing. Five feature maps are generated from the backbone (e.g., VGG16) and upsampled in the Feature Extraction step. The DPRs and the TS regions are adopted to reconstruct text instances in the postprocessing step.

detection: regression-based and pixel-based. Inspired by the promising of object detection architectures such as Faster R-CNN [26] and SSD [27], a bunch of regression-based detectors are proposed, which simply regress the coordinates of bounding boxes of candidates as the final prediction. TextBoxes [7] adopts SSD and adjusts the default box to relatively long shape to match text instances. PyrBoxes [28] proposes a SSD-based detector equipped with a grouped pyramid to enrich feature. Sheng [29] proposes a novel text detector with learnable anchors to cover all varieties of texts in natural scene. Lyu [30] detects scene text by localizing corner points of text bounding boxes and segmenting text regions in relative positions. By modifying Faster R-CNN, Rotation Region Proposal Networks [31] insert the rotation

branch to fit the oriented shapes of text in natural images. These methods can achieve satisfying performance on horizontal or multioriented text areas. However, they may suffer from the shape of the bounding box, even with rotations. Mainstream pixel-wise methods drew inspirations from the fully convolutional network (FCN) [32], which removes all fully connected layers and is widely used to generate a semantic segmentation map. Convolution transpose operation then helps the shirked feature restore its original size. TextSnake [11] treats a text instance as a sequence of ordered, overlapping disks centered at symmetric order, each of which is associated with potentially variable radius and orientations. It made significant progress on curved text benchmarks. TexeField [33] learns a direction



field pointing away from the nearest text boundary to each text point. An image of two-dimensional vectors represents the direction field. SPCNET [34], based on FPN [35] and Mask R-CNN [36], inserts Text Context Module and Rescore mechanism to leave the lack of context information clues and inaccurate classification score. PSENet [37] projects feature into several maps and gradually expand the detected areas from small kernels to large and complete instances. These pixel-based methods significantly improve the performance of curved benchmarks. However, detection failures are still possible in complex situations. Differs from the previous, the proposed method has more opportunities to recover itself. Specifically, the Text Skeleton represents the profile of the instance, which is smaller and less sticky than the original form. Pixels in text areas are divided into two groups according to four directions: the up-down and left-right. Ideally, a TS can be integrated with any group to restore itself. When some regions fail to reproduce, there is also an opportunity to get additional supplementary from others. We conduct extensive experiments on standard benchmarks, including the horizontal the oriented text, and curved text datasets. Evaluations demonstrate that TextCohesion achieves state-of-the-art or very competitive performance.

### 3. Methodology

The architecture of TextCohesion is depicted in Figure 2, which consists of a feature extraction section and a post-processing section. For image feature extraction, an FCN-based convolutional backbone followed by an up-sampling step is employed. Five feature maps containing a Text Skeleton (TS) and four Directional Pixel Regions (DPRs) are generated after up-sampling. The TS features are evaluated by a Confidence Scoring Mechanism (CSM), and finally obtaining the predicted text regions incorporated with the DPRs regions. To optimize the proposed network, a corresponding loss function of the TS and DPRs is designed. More details are introduced in the following section.

**3.1. Network.** The proposed method inherits the popular VGG16 network by keeping the layers from Conv1 to Conv5, converting the last fully connected layers into convolution layers. The input images are first downsampled to the multilevel features with five convolution blocks, and five feature maps (i.e.,  $P_1, P_2, P_3, P_4, P_5$ ) are generated. Then, these features are gradually upsampled to the original size and mixed with the corresponding output of the previous convolution block. The upsampled process can be described by

$$O = U(P_1 || U_p(P_2 || U_p(P_3 || U_p(P_4 || U_p(P_5))))), \quad (1)$$

where  $O$  is the output of the network, “ $||$ ” refers to feature concatenation, and  $U_p$  is the upsample function (i.e., Conv(1, 1) – Conv(3, 3) – Deconv – ReLu used to resize the feature map matching other layers. Five feature maps with the same resolution are leveraged as the prediction of the network (the blue box shown in Figure 2) after the upsample step. Each prediction is composed of a TS and four DPRs in

the postprocessing. DPRs contain four feature maps according to different directions:  $R_1, R_2, R_3$ , and  $R_4$ . The TS is the skeleton of the text instance that is adopted to separate from each other. The CMS is introduced to reduce false positives in terms of evaluating each TS. For clarity, we take a curved text as an example to demonstrate the process of label generation in the rest of Section 3.

**3.2. Text Skeleton.** Text Skeleton (TS) is an essential component representing the center part of the text instance. As shown in Figure 3(b), the gray area is the TS of the instance. The first step of generating TS is to find the head and tail of the text. Similar to [11], we also use the cosine of adjacent vertices to find the head and tail of text instance, and the remaining two longest sides. The longest two sides along with the text instance (e.g.,  $t_0t_n$  and  $b_0b_n$ ) are called sidelines in the proposed method. Then,  $n$  vertices of even distribution are sampled from the two sidelines (i.e., Top Sideline and Bottom Sideline in Figure 3(a)), respectively. After that the vertices in the centerline (Head – Tail in Figure 3) can be averaged from these sampled vertices:

$$c_i(x, y) = \frac{t_i(x, y) + b_i(x, y)}{2}, \quad (2)$$

where  $\{t_0, t_1, \dots, t_i, \dots, t_n\}$  and  $\{b_0, b_1, \dots, b_i, \dots, b_n\}$  are vertices in two sidelines of the text instance, respectively, and  $\{c_0, c_1, \dots, c_i, \dots, c_n\}$  are a set of vertices belong to the center line. Finally, TS is bold by the center line  $\text{inf}d3$

$$\begin{aligned} e_i &= c_i + (t_i - c_i) \times \beta, \\ f_i &= c_i + (b_i - c_i) \times \beta, \end{aligned} \quad (3)$$

where  $e_i$  and  $f_i$  are pixels that represent the expansion of the center line to both sidelines. The region of  $e_i e_{i+1} f_i f_{i+1}$  form a part of TS, as shown in Figure 3(b).  $\beta$  is a parameter that holds the bold rate, and we set it to 0.2 experimentally. When these vertices are completely processed, TS is generated correspondingly.

**3.3. Directional Pixel Region.** Directional Pixel Regions (DPRs) are used to restore its original form, including  $R_1, R_2, R_3$ , and  $R_4$ . Pixels in text instance but not in TS are considered as falling into DPR. In Figure 3(b),  $t_i t_{i+1} e_{i+1} e_i$  and  $f_i f_{i+1} b_{i+1} b_i$  illustrate a fraction of DPR. The direction of every fraction is determined by the tangent angle between its corresponding center vertices ( $c_i$ ) and the next ( $c_{i+1}$ ). More specifically, the tangent angle of two adjacent center vertices is calculated by the following equation:

$$\tan(\Theta_i) = \frac{y_{c_{i+1}} - y_{c_i}}{x_{c_{i+1}} - x_{c_i}}, \quad (4)$$

where  $x$  and  $y$  refer to the coordinates of the center vertices. By comparing the  $\tan(\Theta_i)$  of center vertices with  $\alpha$ , the regions of  $t_i t_{i+1} e_{i+1} e_i$  and  $f_i f_{i+1} b_{i+1} b_i$  are labeled as DPRs ( $R_1, R_2, R_3, R_4$ ) or background. If  $\Theta_i$  falls into a specific range (e.g.,  $[-30^\circ, 30^\circ]$ ), the pixels within its corresponding

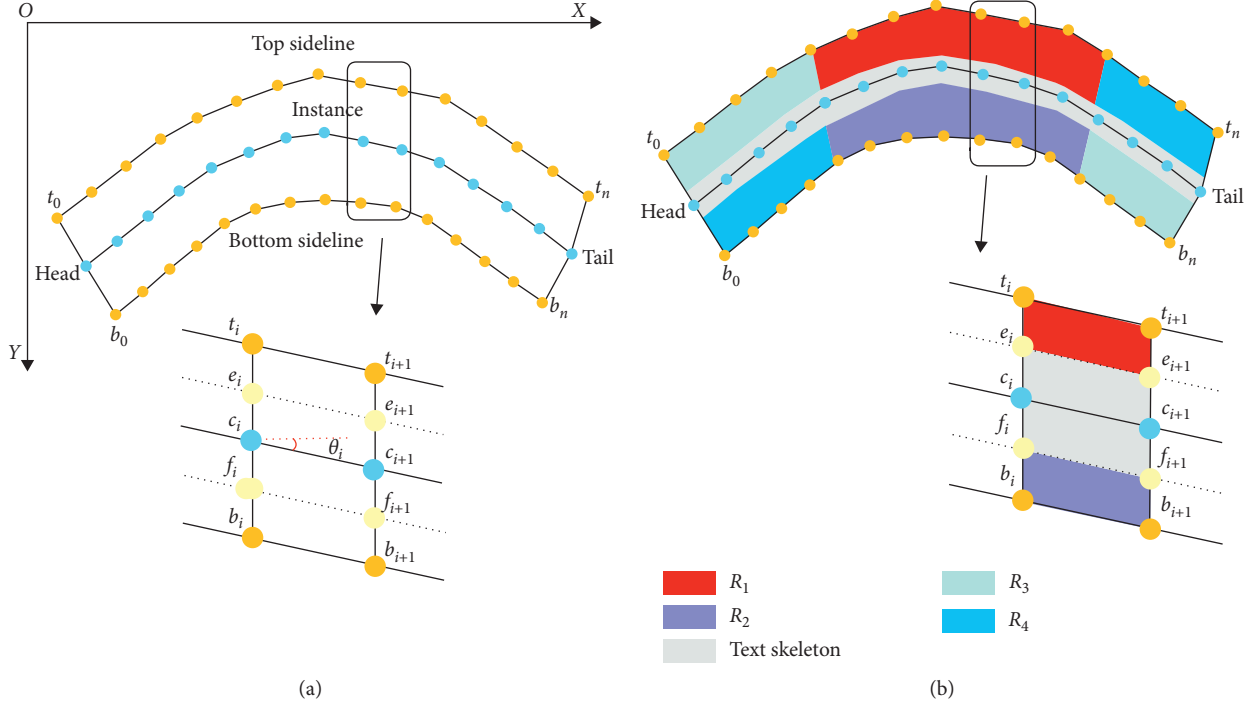


FIGURE 3: Label generation: (a) specific mathematical modeling method and (b) a clear example of the TS or the DPRs.

DPRs  $(t_i t_{i+1} e_{i+1} e_i)$  and  $(f_i f_{i+1} b_{i+1} b_i)$  are considered belonging to the  $R_1$  or  $R_2$ . The  $R_1$  can be calculated as follows:

$$R_{1_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other,} \end{cases} \quad (5)$$

$$\text{condition}_1: \tan(-\alpha) < \tan(\Theta_i) < \tan(\alpha),$$

$$\text{condition}_2: (y_{t_i} + y_{t_{i+1}}) < (y_{c_i} + y_{c_{i+1}}),$$

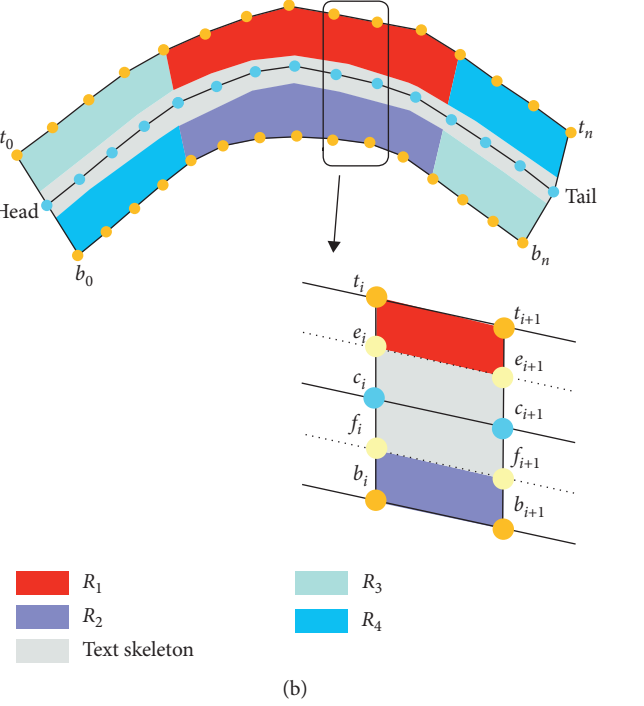
where  $\text{condition}_1$  is used to distinguish the angle of adjacent center vertices and  $\text{condition}_2$  ensures the selected pixels are above the TS.  $\alpha$  is a parameter that controls the boundary of specific directional regions, which is discussed in detail in the experiment section.  $y_{t_i}$  and  $y_{c_i}$  are the vertical coordinates of vertices  $(x, y)$  on the sideline and the center line, respectively. The generating process of the  $R_2$  is similar to the  $R_1$ , but the only difference is that the pixels are located below the TS. Therefore,  $\text{condition}_2$  is reversed naturally:

$$R_{2_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other,} \end{cases} \quad (6)$$

$$\text{condition}_1: \tan(-\alpha) < \tan(\Theta_i) < \tan(\alpha),$$

$$\text{condition}_2: (y_{t_i} + y_{t_{i+1}}) > (y_{c_i} + y_{c_{i+1}}),$$

where  $y_{t_i}$  and  $y_{t_{i+1}}$  are logically equivalent to  $y_{b_i}$  and  $y_{b_{i+1}}$ , which are the vertical coordinates of the sampled vertices on the sidelines. The  $R_3$  and  $R_4$  are generated in the same way, as shown below:



$$R_{3_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other,} \end{cases}$$

$$\text{condition}_1: \left( \tan\left(\alpha - \frac{\pi}{2}\right) > \tan(\Theta_i) \right) \cap \left( \tan(\Theta_i) > \tan\left(\frac{\pi}{2} - \alpha\right) \right),$$

$$\text{condition}_2: (x_{t_i} + x_{t_{i+1}}) < (x_{c_i} + x_{c_{i+1}}),$$

$$R_{4_i} = \begin{cases} 1, & \text{condition}_1 \cap \text{condition}_2, \\ 0, & \text{other,} \end{cases}$$

$$\text{condition}_1: \left( \tan\left(\alpha - \frac{\pi}{2}\right) > \tan(\Theta_i) \right) \cap \left( \tan(\Theta_i) > \tan\left(\frac{\pi}{2} - \alpha\right) \right),$$

$$\text{condition}_2: (x_{t_i} + x_{t_{i+1}}) > (x_{c_i} + x_{c_{i+1}}),$$

(7)

where  $x_{t_i}$  and  $x_{c_i}$  are the horizontal coordinates of vertices on the sideline and the center line, respectively.

**3.4. Confidence Scoring Mechanism.** To filter out false positives, the confidence score is utilized to weight every TS. If the score of TS is lower than a threshold, then all components of this instance are discarded:

$$\begin{cases} \text{TP,} & \frac{\sum_{i=1}^n P_i}{n} > \gamma, \\ \text{FP,} & \text{Other,} \end{cases} \quad (8)$$



where  $n$  is the total number of pixels in the TS.  $p_i$  is the value of the  $i$ th pixel in the TS region. TP and FP refer to the true positives and false positives, respectively.  $\gamma$  is the threshold value to filter out the TS with a lower confidence score, and we set it to 0.6 empirically. TS with high confidence will be retained and processed to form the final prediction with its corresponding DPRs. Instead, TS belonging to FP (FalsePositive) with its components are filtered directly. The TS, as the central area of a text instance, contains the key features of the whole text, which are more valuable to use than the whole features of one text instance.

**3.5. Loss Function.** The proposed method is trained with the following loss function as the three objectives:

$$L = \lambda_1 L_{TS} + L_{DPR} + L_{CSM}, \quad (9)$$

where  $L_{DPR}$  is a smooth  $L_1$  [26] loss and  $L_{TS}$  and  $L_{CSM}$  are crossentropy classification loss functions. The loss of  $L_{TS}$  is computed as follows:

$$L_{TS} = \sum_{n=1}^N w_i \text{CrossEntropy}(TS_i, \widehat{TS}_i), \quad (10)$$

where  $L_{TS}$  is a self-adjust crossentropy loss function and  $w_i$  in equation (10) is a self-adjust weight [9]. For the  $i$ th instance with area =  $S_i$ , every positive pixels within it have a weight of  $w_i = B/S_i$ .  $B$  is the average area of all text instances in one image. In that case, the pixels in text instances with small areas have a bigger weight than the pixels in big text areas. In our experiments, the weight  $\lambda_1$  is set to 3 as the TS is essential than other components. Losses for DPR and CSM are calculated:

$$\begin{aligned} L_{DPR} &= \sum_{n=1}^4 \sum_{i \in \text{DPR}_n} \text{Smooth}L_1(\text{DPR}_i, \widehat{\text{DPR}}_i), \\ L_{CSM} &= \sum_{n=1}^N \text{CrossEntropy}(CS_i, \widehat{CS}_i), \end{aligned} \quad (11)$$

where  $L_{DPR}$  is optimized by a Smooth  $L_1$  loss, and the pixels losses in  $R_1, R_2, R_3$ , and  $R_4$  are calculated, respectively, which means that one pixel can be simultaneously categorized as two regions (e.g.,  $R_1$  and  $R_3$ ).  $L_{CSM}$  is a standard crossentropy function.  $TS_i, \text{DPR}_i$ , and  $CS_i$  are ground truth labels and  $\widehat{TS}_i, \widehat{\text{DPR}}_i$ , and  $\widehat{CS}_i$  are predicted values.

**3.6. Postprocessing.** TextCohesion treats every text instance as TS and four DPRs previously; hence, these components should be grouped, forming the final prediction. The postprocessing algorithm is depicted in Algorithm 1:

Every TS represents a text instance, and after passing through CSM, instances with higher confidence are reserved as candidates. Based on these candidates, the corresponding DPRs can be obtained. The postprocessing mainly includes three steps. (1) The TS is used to differentiate the different text instances. (2) For each TS, the outer pixels as initial points are used to search the corresponding pixels in the DPRs iteratively. (3) The TS is eventually merged with

corresponding searched regions to form the final prediction. The entire postprocessing is shown in Algorithm 1, where Neighbor(.) refers to a function that obtains the directional information of the adjacent pixels.

## 4. Experiment

To evaluate TextCohesion, we conduct extensive experiments on both oriented and curved benchmarks and give a detailed description of these datasets for model training and inference, experimental implementation, results with comparisons, and ablation study, respectively.

**4.1. Datasets.** SynthText [38] is a large scale dataset that contains about 800K synthetic images that are created by blending natural images with text rendered with random fonts, sizes, colors, and orientations. These texts look realistic as the overlaying follows carefully set up configurations and a well-set learning algorithm.

ICDAR2015 [14] contains 1000 training and 500 test images captured by wearable cameras with relatively low resolutions. Each image includes several oriented texts annotated by four vertices of the quadrangles.

ICDAR 2017 MLT (IC17-MLT) [39] is a large scale multilingual text dataset, which includes 7200 training images, 1800 validation images, and 9000 testing images. The dataset is composed of complete scene images that come from 9 languages. Similarly, with ICDAR 2015, the text regions in ICDAR 2017 MLT are also annotated by four vertices of the quadrangle.

CTW1500 [40] is a challenging dataset for curve text detection, which is constructed by Yuliang et al. [18]. It consists of 1000 training images and 500 testing images. Different from traditional text datasets (e.g., ICDAR 2015 and ICDAR 2017 MLT), the text instances in SCUT-CTW1500 are labeled by a polygon with 14 points that can describe the shape of an arbitrarily curve text.

Total-Text [41] is another word-level-based English curve text dataset which is split into training and testing sets with 1255 and 300 images, respectively (Figure 4).

**4.2. Implementation Details.** Training TextCohesion is optimized by SGD with backpropagation [42]. Momentum and weight decay are set to 0.9 and  $5 \times 10^{-4}$ , respectively. Learning rate is initialized to  $10^{-4}$  and decayed by 0.1 every 30 epochs. Following [11], all training images are augmented online with rotated and cropped with areas ranging from 0.24 to 1.69 and aspect ratios ranging from 0.33 to 3. After that noise, blur, and lightness are randomly adjusted and lastly resized to  $512 \times 512$ . We ensure that the text on the augmented images is still legible if they are legible before augmentation. TextCohesion is firstly pretrained on SynthText for 2 epochs and fine-tuned on other datasets. All implementations are deployed on PC with (CPU: Intel(R) Core(TM) i7-7800X CPU @ 3.50 GHz; GPU: GTX 1080).

Inferencing to test the ability of detecting arbitrarily shaped text, we evaluate our method on Total-Text and



FIGURE 4: Visualization of the results on curved text datasets.

**Input:** $t_i \in \text{TS, DPR}$ **Output:**

Result

- (1) **Function Grouping** ( $t_i$ )
- (2)  $T \leftarrow \text{Neighbor}(t_i, \text{up})$
- (3)  $B \leftarrow \text{Neighbor}(t_i, \text{down})$
- (4)  $L \leftarrow \text{Neighbor}(t_i, \text{left})$
- (5)  $R \leftarrow \text{Neighbor}(t_i, \text{right})$
- (6) **if**  $T \neq \text{None}$  **and**  $\text{DPR}[T] == \text{up}$  **then**
- (7)  $T_{\text{cache}} \leftarrow t_i \cup T$
- (8) **Grouping** ( $T_{\text{cache}}$ )
- (9) **else if**  $B \neq \text{None}$  **and**  $\text{DPR}[B] == \text{down}$  **then**
- (10)  $B_{\text{cache}} \leftarrow t_i \cup B$
- (11) **Grouping** ( $B_{\text{cache}}$ )
- (12) **else if**  $L \neq \text{None}$  **and**  $\text{DPR}[L] == \text{left}$  **then**
- (13)  $L_{\text{cache}} \leftarrow t_i \cup L$
- (14) **Grouping** ( $L_{\text{cache}}$ )
- (15) **else if**  $R \neq \text{None}$  **and**  $\text{DPR}[R] == \text{right}$  **then**
- (16)  $R_{\text{cache}} \leftarrow t_i \cup R$
- (17) **Grouping** ( $R_{\text{cache}}$ )
- (18) **else**
- (19) **Return**  $t_i \cup T_{\text{cache}} \cup B_{\text{cache}} \cup L_{\text{cache}} \cup R_{\text{cache}}$
- (20) **end if**

ALGORITHM 1: Postprocessing algorithm.

SCUT-CTW1500, both of them containing the curved instances. Images in the test stage are also resized to  $512 \times 512$ . We report the performance on SCUT-CTW1500 in Table 1, in

which we can find that the Precision (88.0%), Recall (84.6%), and F-measure (86.3%) achieved by TextCohesion significantly outperform the ones of other competitors. Remarkably,

TABLE 1: Experimental results on CTW1500.

Method	Ext.	CTW1500			
		Precision	Recall	F-measure	Time
CTPN [3]	—	60.4	53.8	56.9	0.14
SegLink [44]	—	42.3	40.0	40.8	0.049
EAST [8]	—	78.7	49.1	60.4	0.076
TextSnake [11]	—	65.4	63.4	64.4	0.909
DB-ResNet-18 [43]	—	84.8	77.5	81.0	0.001
PSENet [12]	✓	84.8	78.0	80.9	0.429
DB-ResNet-18 [43]	—	84.8	77.5	81.0	0.018
CRAFT [45]	✓	87.6	79.9	83.6	0.116
DB-ResNet-50 [43]	✓	86.9	80.2	83.4	0.045
TextCohesion (ours)	✓	<b>88.0</b>	<b>84.6</b>	<b>86.3</b>	0.206

Ext. indicates external data.

TABLE 2: Experimental results on Total-Text.

Method	Ext.	Total-Text			
		Precision	Recall	F-measure	Time
SegLink [44]	—	30.0	23.8	26.7	0.049
EAST [8]	—	50.0	36.2	42.0	0.076
MaskSpotter [46]	—	69.0	55.0	61.3	0.208
TextSnake [11]	—	61.5	67.9	64.6	0.909
PSENet [12]	✓	84.0	78.0	80.9	0.429
SPCNet [47]	✓	83.0	<b>82.8</b>	82.9	—
CRAFT [48]	✓	87.6	79.9	83.6	0.116
DB-ResNet-50 [43]	✓	87.1	82.5	<b>84.7</b>	<b>0.031</b>
TextCohesion (ours)	✓	<b>88.1</b>	81.3	84.6	0.206

Ext. indicates external data.

TABLE 3: Experimental results on ICDAR2015.

Method	Ext.	ICDAR2015			
		Precision	Recall	F-measure	Time
CTPN [3]	—	74.2	51.6	60.9	0.14
SegLink [44]	—	73.1	76.8	75.0	0.048
EAST [8]	—	83.6	73.5	78.2	0.076
PixelLink [9]	—	82.9	81.7	82.3	0.333
DB-ResNet-18 [43]	—	86.8	78.4	82.3	<b>0.024</b>
TextSnake [11]	✓	84.9	80.4	82.6	0.909
Mask textspotter [46]	✓	85.8	81.2	83.4	0.208
PSENet [12]	✓	86.9	84.5	85.7	0.429
CRAFT [48]	✓	89.8	84.3	86.9	0.116
SPCNet [15]	✓	88.7	85.8	87.2	—
DB-ResNet-50 [43]	✓	<b>91.8</b>	83.2	87.3	0.083
PMTD [49]	✓	91.3	87.4	89.3	—
TextCohesion (ours)	✓	89.2	<b>90.2</b>	<b>89.7</b>	0.206

Ext. indicates external data.

the recall and F-measure surpass the second-best record by 4.7% and 2.7%, respectively. Besides, the inference time of the proposed method is also compared with other methods, i.e., DB [43]. The testing scale of the input image is resized to  $512 \times 512$  pixels, and the batch size is set to 1 during all the comparison experiments. The main results are reported in Tables 1–4, where an acceptable inference time can be found.

**4.3. Experiments on Curved Text Benchmarks.** To test the ability to detect arbitrarily shaped text, we evaluate our

method on Total-Text and CTW1500, both of them containing the curved instances. Images in the test stage are also resized to  $512 \times 512$ . We report the performance on CTW1500 in Table 1, in which we can find that the Precision (88.0%), Recall (84.6%), and F-measure (86.3%) achieved by TextCohesion significantly outperform the ones of other competitors. Remarkably, the Recall and F-measure surpass the second-best record by 4.7% and 2.7%, respectively.

Our method achieves 88.1%, 81.4%, and 84.6% in Precision, Recall, and F-measure, respectively, outperforming

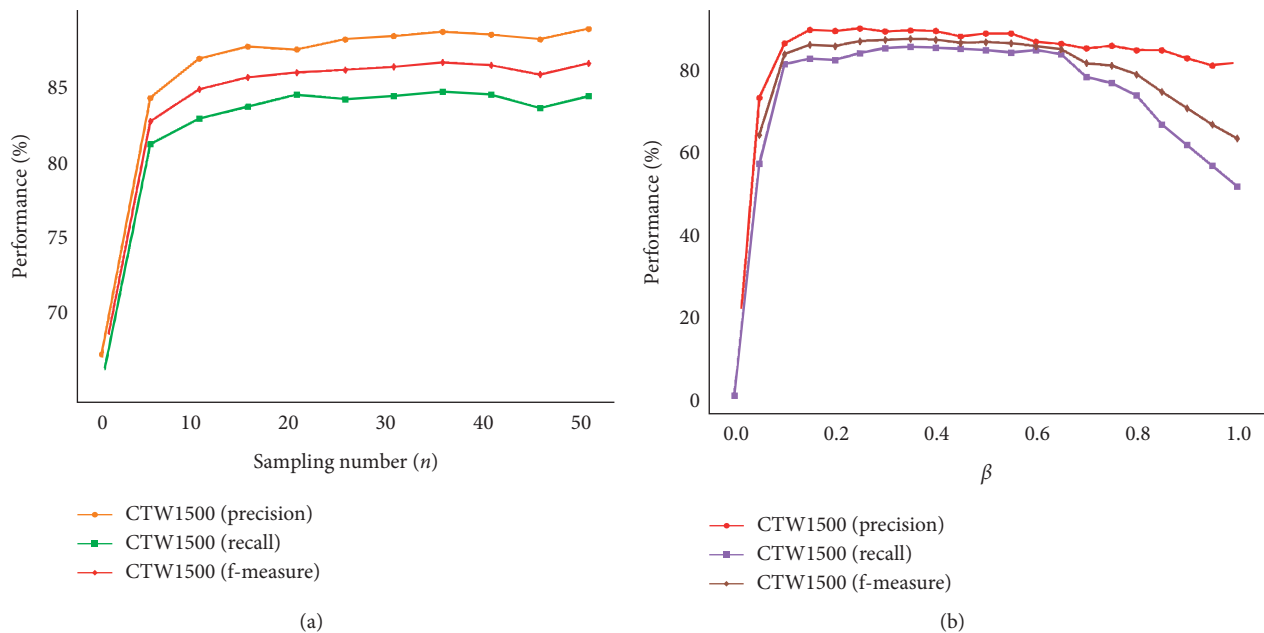
TABLE 4: Experimental results for ICDAR2017.

Method	Ext.	ICDAR2017			
		Precision	Recall	F-measure	Time
Lyu et al. [30]	—	<b>83.8</b>	55.6	66.8	0.175
FOTS [50]	—	81.0	57.6	67.2	0.041
DB-ResNet-18 [43]	—	81.9	63.8	71.7	<b>0.020</b>
PSENet [12]	✓	77.0	<b>68.4</b>	72.5	0.429
CRAFT [51]	✓	80.6	68.2	73.9	0.116
DB-ResNet-50 [43]	✓	83.1	67.9	<b>74.7</b>	0.053
TextCohesion (ours)	✓	81.8	66.0	73.1	0.206

Ext. indicates external data.

TABLE 5: Model results for different values of  $\alpha$  in equation (3) and when using the CSM on CTW1500.

Dataset	$\alpha$	CSM ( $\gamma$ )	Precision	Recall	F-measure
CTW1500	$\pi/6$ (30°)	✓(0.6)	88.0	84.6	86.3
CTW1500	$\pi/4$ (45°)	✓(0.6)	88.5	82.0	85.2
CTW1500	$\pi/3$ (60°)	✓(0.6)	89.5	81.9	85.5
CTW1500	$\pi/6$ (30°)	×	85.2	84.8	85.0
CTW1500	$\pi/6$ (30°)	✓(0.4)	85.3	85.1	85.2
CTW1500	$\pi/6$ (30°)	✓(0.5)	86.3	84.9	85.6
CTW1500	$\pi/6$ (30°)	✓(0.6)	88.0	84.6	86.3
CTW1500	$\pi/6$ (30°)	✓(0.7)	88.2	83.3	85.7

FIGURE 5: Ablation study on sampling number ( $n$ ) and  $\beta$ .

the second competitor with an F-measure of 1.0% on Total-Text. We attribute this excellence to the proposed flexible representation. Instead of taking the text as a whole, the representation treats text as a serial of components and integrates them together to form the final prediction.

**4.4. Experiments on Oriented Text Benchmarks.** In this section, we evaluate TextCohesion on oriented text datasets.

The performance of ICDAR2015 and ICDAR2017 are demonstrated in Tables 3 and 4, which also achieves F-measure of 89.1% and 73.1%, respectively. From these results, it can be observed that our method also achieves very competitive performance in dealing with oriented text. Meanwhile, thanks to the robust feature representation, TextCohesion can as well locate the text instance with small instances and in complex illuminations and variable scales.



#### 4.5. Analyses and Discussion

**4.5.1. Influence of the Number of Samples ( $n$ ).** We sample  $n$  points on the top sideline and bottom sideline for each text instance, and use these points to split text instances better. To further study the Influence of the number of points on the sampling precision, an ablation experiment is performed, as shown in Figure 5(a). Theoretically, the performance of the model will improve with the increase of sampling precision. In the experiment, we found that the performance of the model hardly improve further (around 85%) when the sampling number ( $n$ ) is greater than 10.  $n$  is set to 40 in all experiments.

**4.5.2. Influence of  $\beta$  in Equation (2).**  $\beta$  as an important parameter is used to control the ratio of the TS area to the DPR area. As shown in Figure 5(b), when the value of  $\beta$  is within the range of [0.1, 0.6], the network performs well. In all experiments,  $\beta$  is set to 0.2.

**4.5.3. Influence of  $\alpha$  in Equation (3).**  $\alpha$  is used to delineate the top, bottom, left, and right regions.  $30^\circ$ ,  $45^\circ$ , and  $60^\circ$  are the three specific angles used to investigate the influence of  $\alpha$ . As shown in Table 1, the F-measure is relatively good when  $\alpha$  is  $30^\circ$ , so we set  $\alpha$  to  $30^\circ$  in all experiments.

**4.5.4. Influence of the Confidence Scoring Mechanism.** The CSM is used to filter out the false positives (e.g., those symbols or characters that are similar to text). The influence in the results of the model when using the CSM is shown in Table 5. The precision improves 2.8% after the CSM (0.6) is used. To test the robustness of the proposed model while changing the  $\gamma$  in equation (8), a comparison experiment is set in Table 5, and the F-measure is relatively good when  $\gamma$  is 0.6. In all experiments,  $\gamma$  is set to 0.6.

## 5. Conclusion and Outlook

In this paper, we propose a novel text detector, which achieves upto 86.3% F-measure among common text benchmarks, including text instance with irregular shapes. The text instance modeling method utilized in this detector could precisely detect text with arbitrary boundaries by splitting one text instance into four DPRs and a TS region. Moreover, a Confidence Scoring Mechanism is incorporated into this detector to filter out false positives, which further improves its detection precision. Simulation experiment results show that the proposed text detector performs well in scene text detection. The proposed method might have potential applications in the field of photo translation, autonomous driving, and product identification.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Weijia Wu and Jici Xing contributed equally to this work.

## Acknowledgments

This work was supported by the National Key Research and Development Project (Grant no. 2019YFC0118202), National Natural Science Foundation of China (Grant no. 61803332), and Scientific Research Fund of Zhejiang Provincial Education Department (Grant no. Y201941642).

## References

- [1] C. Yi and Y. Tian, "Scene text recognition in mobile applications by character descriptor and structure configuration," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2972–2982, 2014.
- [2] Y. Zhu, M. Liao, M. Yang, and W. Liu, "Cascaded segmentation-detection networks for text-based traffic sign detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 209–219, 2018.
- [3] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016.
- [4] D. Wu, R. Wang, P. Dai, Y. Zhang, and X. Cao, "Deep strip-based network with cascade learning for scene text localization," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.
- [5] X. Zhu, Y. Jiang, S. Yang, and Wang, "Deep residual text detection network for scene text," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.
- [7] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: a fast text detector with a single deep neural network," *Computer Vision and Pattern Recognition, AAAI*, 2017, <http://arxiv.org/abs/1611.06779>.
- [8] X. Zhou, C. Yao, H. Wen et al., "East: an efficient and accurate scene text detector," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [9] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: detecting scene text via instance segmentation," *Computer Vision and Pattern Recognition, AAAI*, 2018, <http://arxiv.org/abs/1801.01315>.
- [10] M. Liao, B. Shi, and X. Bai, "TextBoxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.
- [11] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: a flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, September 2018.
- [12] W. Wang, E. Xie, X. Li et al., "Shape robust text detection with progressive scale expansion network," in *Proceedings of the*

- Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, June 2018.
- [13] D. Karatzas, F. Shafait, S. Uchida et al., "Icdar 2013 robust reading competition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Washington, DC, USA, August 2013.
- [14] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, and S. Ghosh, "Icdar 2015 competition on robust reading," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, August 2015.
- [15] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *Computer Vision and Pattern Recognition, AAAI*, 2019, <http://arxiv.org/abs/1811.08605>.
- [16] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, CA, USA, pp. 2963–2970, 2010.
- [17] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1241–1248, Sydney, Australia, 2013.
- [18] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognition*, vol. 31, no. 12, pp. 2055–2076, 1998.
- [19] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, IEEE, Providence, RI, USA, June 2012.
- [20] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, 2011.
- [21] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.
- [22] A. Coates, B. Carpenter, C. Case et al., "Text detection and character recognition in scene images with unsupervised feature learning," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, vol. 11, pp. 440–445, Beijing, China, November 2011.
- [23] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proceedings of the 2011 International Conference on Document Analysis and Recognition*, pp. 429–434, IEEE, Beijing, China, September 2011.
- [24] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 1457–1464, IEEE, Barcelona, Spain, November 2011.
- [25] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, Tsukuba, Japan, pp. 3304–3308, 2012.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 2015.
- [27] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, October 2016.
- [28] F. Sheng, Z. Chen, W. Zhang, and B. Xu, "Pyrboxes: an efficient multi-scale scene text detector with feature pyramids," *Pattern Recognition Letters*, vol. 125, pp. 228–234, 2019.
- [29] F. Sheng, Z. Chen, T. Mei, and B. Xu, "A single-shot oriented scene text detector with learnable anchors," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1516–1521, IEEE, Shanghai, China, July 2019.
- [30] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7553–7563, Salt Lake City, UT, USA, June 2018.
- [31] J. Ma, W. Shao, H. Ye et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.
- [32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.
- [33] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.
- [34] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9038–9045, 2019.
- [35] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, Honolulu, HI, USA, July 2017.
- [36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.
- [37] W. Wang, E. Xie, X. Li et al., "Shape robust text detection with progressive scale expansion network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9336–9345, Long Beach, CA, USA, June 2019.
- [38] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2315–2324, Las Vegas, NV, USA, June 2016.
- [39] N. Nayef, F. Yin, I. Bizid et al., "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1454–1459, IEEE, Kyoto, Japan, November 2017.
- [40] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognition*, vol. 90, pp. 337–345, 2019.
- [41] C. K. Ch'ng and C. S. Chan, "Total-text: a comprehensive dataset for scene text detection and recognition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, November 2017.
- [42] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.
- [43] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," 2019, <http://arxiv.org/abs/1911.08947>.



- [44] B. Shi, X. Bai, S. Belongie, and Xiang, "Detecting oriented text in natural images by linking segments," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [45] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2550–2558, Honolulu, HI, USA, July 2017.
- [46] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask Text-Spotter: an end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–88, Munich, Germany, September 2018.
- [47] Q. Wang, J. Gao, M. Zhang, J. Xing, and W. Hut, "SPCNet: scale position correlation network for end-to-end visual tracking," in *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1803–1808, IEEE, Beijing, China, August 2018.
- [48] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.
- [49] J. Liu and X. Liu, "Pyramid mask text detector," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.
- [50] X. Liu, D. Liang, S. Yan et al., "Fast oriented text spotting with a unified network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [51] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9365–9374, Long Beach, CA, USA, June 2019.

## Research Article

# Texts as Lines: Text Detection with Weak Supervision

Weijia Wu,<sup>1</sup> Jici Xing<sup>1,2</sup>, Cheng Yang,<sup>1</sup> Yuxing Wang<sup>1</sup>, and Hong Zhou<sup>1</sup>

<sup>1</sup>Zhejiang University, Key Laboratory for Biomedical Engineering of Ministry, Hangzhou, China

<sup>2</sup>Zhengzhou University, School of Information Engineering Institute, Zhengzhou, China

Correspondence should be addressed to Yuxing Wang; wangyuxing@zju.edu.cn and Hong Zhou; zhouhong\_zju@126.com

Received 11 February 2020; Revised 16 April 2020; Accepted 6 May 2020; Published 10 June 2020

Guest Editor: Qian Zhang

Copyright © 2020 Weijia Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scene text detection methods based on deep learning have recently shown remarkable improvement. Most text detection methods train deep convolutional neural networks with full masks requiring pixel accuracy for good quality training. Normally, a skilled engineer needs to drag tens of points to create a full mask for the curved text. Therefore, data labelling based on full masks is time consuming and laborious, particularly for curved texts. To reduce the labelling cost, a weakly supervised method is first proposed in this paper. Unlike the other detectors (e.g., PSENet or TextSnake) that use full masks, our method only needs coarse masks for training. More specifically, the coarse mask for one text instance is a line across the text region in our method. Compared with full mask labelling, data labelling using the proposed method could save labelling time while losing much annotation information. In this context, a network pretrained on synthetic data with full masks is used to enhance the coarse masks in a real image. Finally, the enhanced masks are fed back to train our network. Analysis of experiments performed using the model shows that the performance of our method is close to that of the fully supervised methods on ICDAR2015, CTW1500, Total-Text, and MSRA-TD5000.

## 1. Introduction

At present, natural scene text detection has attracted more attention due to its practical application requirements, such as scene understanding, visual question answering, autonomous driving, text detection [1], and recognition [2, 3]. Text is one of the most fundamental semantics appearing everywhere in daily life, for example, in traffic signs, commodity packages, and advertising posters. These text instances in the real world have varying sizes, random directions, and arbitrary shapes, making them extremely challenging to label and capture accurately. Unlike other general objectives, scene text usually cannot be described accurately by the axis-aligned rectangle, and most detectors using an axis-aligned rectangle only have an F-measure of below 65%, such as TextSnake [4]. Recently, most scene text detectors based on deep learning have tended to detect texts in different shapes with many coordinates for better performance. However, the above detectors require accurate pixel-level labels with expensive costs. The labelling consumes a large amount of manpower and financial resources, especially for texts with arbitrary shapes in complex environments.

The precision of text detection has a close connection with the labelling methods of datasets. For example, several common datasets, ICDAR2013 [5], ICDAR2015 [6], ICDAR2017 [7], Total-Text [8], CTW1500 [9], and MSRA-TD500 [10], have different labelling methods for various texts. ICDAR2013, as one of the common datasets, was introduced during the ICDAR Robust Reading Competition in 2013 and mainly includes horizontal bounding boxes made by two points at the word level. Because of this labelling peculiarity, text detectors [11, 12] using box regression have a great performance on ICDAR2013. ICDAR2015 was released in the ICDAR2015 Robust Reading Competition for multioriented text detection, using quadrilateral boxes as the annotations, as shown in Figure 1(b). EAST [13] and SPCNet [14], as the representatives of detectors, achieved good results on ICDAR2015. ICDAR2017 was a dataset with texts in nine languages for multilingual scene text detection, using quadrilateral boxes as the annotations as well as ICDAR2015. MSRA-TD500 was released in 2012, and the annotation method is the same as that of ICDAR2015. Unlike the above datasets, Total-Text and



FIGURE 1: Labelling methods. (a) Original image. (b) Bounding box labelling containing a considerable amount of background noise and interference from other text areas. (c) Visualization of segmentation annotation, requiring high labour cost. (d) Proposed annotation methods: coloured regions represent the labelled region, including the grey background and other coloured text lines.

CTW1500 contain many curved texts, which aim to solve the arbitrarily shaped text detection problem. CTW1500 has more than 10k text annotations and at least one curved text per image. Total-Text contains many curved and multi-oriented texts, which require tens of points for accurate labelling. Recently, segmentation-based text detectors [4, 15, 16] have shown promising performance in existing datasets with high-cost labelling. The annotation design becomes more complicated to fit the requirements of text detection in the real world, and the cost also increases.

The bounding box-based labelling method has low labelling costs but cannot fit text instances accurately in the wild, as shown in Figure 1(b). The pixel-based labelling method matches texts with arbitrary shapes in a complex environment but requires high labelling costs, as shown in Figure 1(c). To mitigate this conflict, we explore detecting texts at the pixel level but with a low labelling cost. Precise drawings of the text region are difficult, but using a cross-line to locate text is simple. Therefore, we seek to simplify the complex text labelling as a line named the text line in this work. Compared with the box or full masks, this annotation is extremely simple and contains less pixel information, as shown in Figure 1. Hence, the following two difficulties must be considered:

- (i) A weak text line label loses the text edge information and nearly all of the background information, which is rather problematic for supervised training

- (ii) The loss function focuses only on the labelled area and is not sensitive to the unlabelled ground truth

To solve the above difficulties, a scene text detector based on weakly supervised learning is proposed in our paper. The model is first pretrained on SynthText to make it sensitive to the text region. Subsequently, in the training process of real data, the pretrained model is used to enhance the text line label. In addition, to enhance the weak label better, a soft label  $\in [0, 1]$  containing pixel location (distance) information is used. The contributions of this work are summarized as follows:

- (i) We first propose a scene text detector based on weakly supervised learning that significantly simplifies the annotation process without losing much precision.
- (ii) A modified crossentropy loss function named *degree crossentropy* is proposed. The loss function can optimize the soft label containing distance information.

## 2. Related Work

Scene text detection has received significant attention over the past few years, and numerous deep learning-based methods [17–21] have achieved great progress. Increasing



detectors tend to capture texts at the pixel level to detect texts more precisely.

**2.1. Bounding Box-Level Text Detection Methods.** Bounding box regression-based methods [19, 22] are inspired by general object detection methods such as SSD [23] and Faster R-CNN [24]. TextBoxes++ [25] further regresses to quadrangles instead of horizontal bounding boxes for multioriented text detection. RRD [26] uses rotation-invariant and sensitive features from two separate branches for better long text detection. DSRN [2] maps multiscale convolution features onto a scale invariant space and obtains uniform activation of multisize text instances for detecting texts. Although regression-based methods have achieved state-of-the-art performance, it is still difficult to capture all text information in a bounding box without involving a large proportion of background and even other text instances.

**2.2. Pixel-Level Text Detection Methods.** Pixel-level text detectors draw inspirations from FCN [23] and Mask R-CNN [27]. Using the mask as the annotation, PixelLink [28] performs text/nontext and links prediction at the pixel level. TextSnake [4] learns to predict local attributes, including the text centre line, text region, radius, and orientation, achieving improvements of up to 20% accuracy on curved benchmarks. CRAFT [15] trains a convolutional neural network producing the character region score and affinity score. PSENet [16] projects the feature map into several branches to produce multiple segmentation maps. TextField [29] detects scene text by predicting a direction field pointing away from the nearest text boundary to each text point. Text mountain [30] predicts text centre-border probability and text centre-direction to detect the scene text. Text detectors based on instance segmentation perform better with higher precision annotation.

**2.3. Weak Supervision Semantic Segmentation.** Sun et al. [31] leveraged the power of deep semantic segmentation CNNs while avoiding requiring expensive annotations for training. Rtfnet [32] took advantage of thermal images and fused both the RGB and thermal information in a novel deep neural network. Tang et al. [33] proposed a normalized cut loss for semisupervised learning; the loss combines partial cross-entropy on labelled pixels and normalized cut for unlabelled pixels. Wang et al. [1] proposed a self-supervised approach and developed a pipeline to label drivable areas and road anomalies using RGB-D images automatically.

**2.4. Weak Supervision Text Detection Methods.** WeText [34] trains scene text detection models on a small number of character-level annotated text images, followed by boosting the performance with a much larger number of weakly annotated images at the word/text line level. WordSup [35] trains a character detector by exploiting word annotations in rich large-scale real scene text datasets.

Recently, all detectors have been trained with fully annotated masks, requiring pixel-level accuracy for good

quality prediction. Motivated by weakly supervised semantic segmentation [34, 36–38], we propose a weakly supervised scene text detector to alleviate the labelling consumption without losing high precision.

### 3. Method

In this section, we first introduce the overall pipeline of the proposed network. Second, the label and the procedure for enhancing the text line are described in detail. Furthermore, the designed loss function for weakly supervised learning is introduced. Finally, we list the simple postprocessing mechanism.

**3.1. Overview.** Figure 2 shows the overall pipeline of the proposed method, which is divided into three steps: (1) the model pretrained on a synthetic dataset [17], (2) label enhanced on a real dataset, and (3) training with the enhanced label. In the first step, the model is pretrained on a synthetic dataset with the full mask to make our model sensitive to the text region. In the second step, the pretrained model outputs an activation map of a real image as a supplement to the weakly annotated label (i.e., text line). In the final step, the enhanced label is fed back to optimize the network parameters. The output of the model in the final step forms the final prediction result through a contour search.

#### 3.2. Labelling and Label Enhancement

**3.2.1. Text Line.** In this paper, we define the text line as a line across the text region, as shown in Figure 3. All characters within this text region should be connected with a continuous line (e.g., TL-1 to TL-5). There are no width and curvature requirements for these text lines. However, improper annotations such as TL-6 will result in an obvious decline in text detection accuracy. The BG in Figure 3 represents the background annotation, which has no requirements for the geometric parameters (e.g., shape, width, length, and curvature) of the line. As a result, the TL and the BG constitute the original annotation.

**3.2.2. Soft Label.** The soft label containing the distance (location) information is used in our method. The shortest distance between each text pixel and the background is calculated. Then, we map these distance values to  $[0, 1]$  as the soft label. For pixels concentrated in the centre of the text instance, a strong (high) value that tends to 1 should be given. However, for the estimated edge area, a weak (low) value that tends to 0 should be assigned. As shown in Figure 2 (activation map), the distance-mountain-like activation map is predicted from the model pretrained on SynthText. The shape of the soft label is the same as the distance-mountain shape. The value  $P_i$  of the label is calculated using the following equation:

$$P_i = \frac{D_i}{D_{\max}}, \quad (1)$$

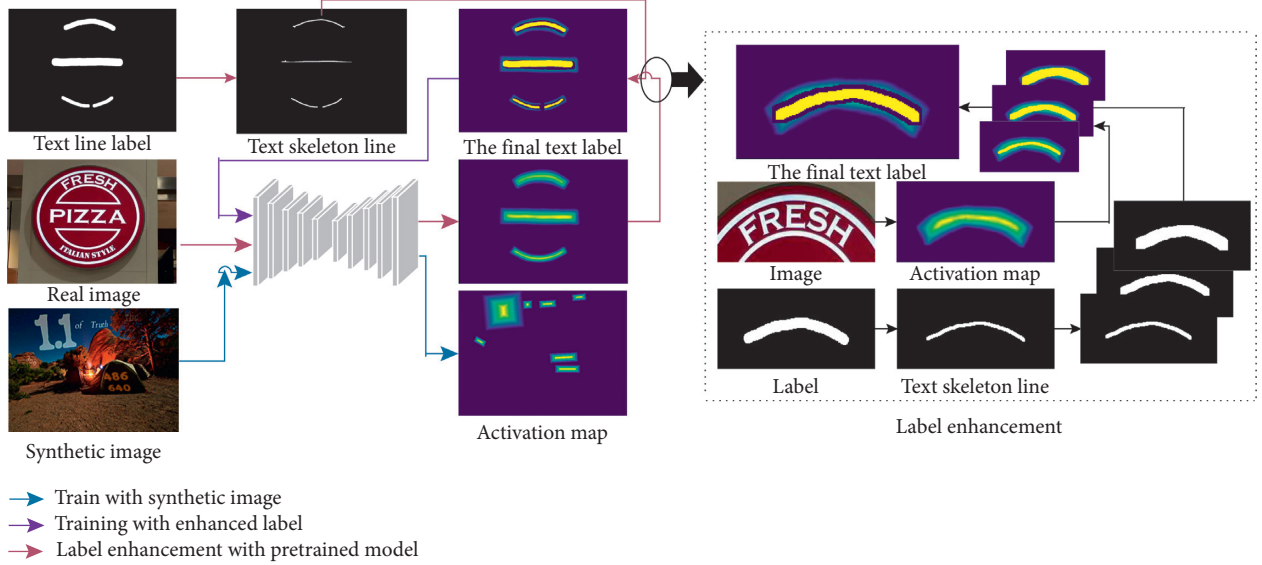


FIGURE 2: The overall procedure of the proposed method. (1) The network pretraining on SynthText for one epoch with full masks. (2) Enhancing the label (i.e., text line) with the pretrained model. (3) The enhanced annotation information is feedback to train the network in a weakly supervised manner.

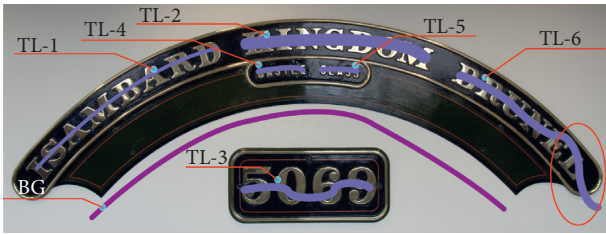


FIGURE 3: Text line labelling: a sample of text line labelling.

where  $D_i$  is the shortest distance between each text pixel ( $i$ ) and the background pixels.  $D_{\max}$  is the maximum value for all  $D_i$ s in the same text instance.

**3.2.3. Label Enhancement.** As shown in Figure 2, label enhancement is an important step in the overall pipeline. The detailed processing of the enhancement is as follows: the network is first pretrained on SynthText for one epoch with full masks, making it sensitive to text areas. The activation maps of real images are generated using the above pretrained model. Then, we extract the text skeleton for the given weakly supervised label. Finally, the intersection of the text activation region and the text skeleton is expanded to obtain more annotation information. The only purpose of label enhancement is to use the text line to locate the correct detection text region in the activation map of the real image and to obtain more supervision information. Enhanced labels only work on the positive part (i.e., text line), while background annotations are excluded.

Figure 2 (right) describes the combination of the text skeleton and activation maps. We first use the text skeleton to locate the corresponding text activation region in the activation map and then attempt to seek the corresponding text edge region through continuous dilation of the

intersection of the text activation region and the text skeleton. Detailed seeking refers to considering a pixel as the edge pixel by estimating whether the pixel value approaches 0. Finally, the values of pixels deemed as edge pixels are used as the supplement to enhance the original annotation (i.e., text line). Note that the values in the activation map are not common binary probabilities (i.e., text/nontext prediction) but represent location (distance) values. Therefore, we can use the value of each pixel in the text region to confirm the relative distance from the background.

**3.3. Network Design.** We chose VGG16 [39] as our feature extractor for a fair comparison with other methods. The images are first downsampled to the multilevel features with five convolution blocks, and five feature maps (i.e.,  $P_1, P_2, P_3, P_4, P_5$ ) are generated in the step. Then, the features are gradually upsampled to the original size and mixed with the corresponding output of the previous convolution block:

$$O = U\left(P_1 \parallel U_p\left(P_2 \parallel U_p\left(P_3 \parallel U_p\left(P_4 \parallel U_p\left(P_5\right)\right)\right)\right)\right),$$

$$U_p(p) = \text{Deconv}(\text{Conv}_{(3,3)}(\text{Conv}_{(1,1)}(p))),$$
(2)

where “ $\parallel$ ” refers to the feature concatenation and  $U_p$  is the upsample function that is used to feed the feature map into the  $\text{Conv}(1, 1)$ - $\text{Conv}(3, 3)$ - $\text{Deconv}$ - $\text{ReLU}$  layers. The difference in  $U$  for  $U_p$  is obtained without the  $\text{ReLU}$  layer and reducing the channel number to 1 as the output. Finally, the output obtained through the sigmoid function is used to calculate the loss of the prediction. In addition to the VGG16, other backbones (i.e., ResNet) were also adopted in a comparative study in Section 4.6 Ablation Study.

**3.4. Loss Function.** The prediction is a two-dimensional feature map, and we map the value to  $[0, 1]$  using the sigmoid function. These values in a text instance are not the confidences of each pixel but represent the degrees of the shortest distance between each pixel and the background. The common binary crossentropy loss function is

$$L_{CE} = -\sum (t \times \ln^y + (1 - t) \times \ln^{1-y}), \quad (3)$$

where  $t$  is the ground truth and  $y$  is the prediction. The common crossentropy is used to evaluate the confidence of a certain category but cannot calculate the loss value with specific meanings (e.g., our distance values).

In that case, we seek to optimize the loss containing distance values by L1 loss:  $|f(x) - Y|$  or L2 loss:  $|f(x) - Y|^2$ . However, we find that L1 and L2 are not sensitive to the distance distribution among  $[0, 1]$ . For instance, the L1 loss between the ground truth of 0.5 and the prediction result of 0.55 is too small and not conducive to backpropagation.

To solve the above difficulty, the *degree crossentropy* is proposed. The *degree crossentropy* can not only evaluate the confidence of category but also deal with the distance information. Losses for the positive and negative pixels are calculated according to

$$\begin{aligned} L_{\text{negative}} &= L_{CE}(x, y) \times (\text{GT}(x, y) == 0), \\ L_{\text{positive}} &= L_{DCE}(x, y) \times U_{cle} \times (\text{GT}(x, y) \neq 0), \end{aligned} \quad (4)$$

where  $L_{CE}(x, y)$  is the traditional crossentropy loss of pixel  $(x, y)$  and  $\text{GT}(x, y)$  is the corresponding ground truth of pixel  $(x, y)$ . Since the enhanced label may not be accurate, we treat the given label and the postenhanced supplements separately.  $U_{cle}$  is a discriminatory mechanism that calculates the losses of the original label and postenhanced part, respectively.  $L_{DCE}(x, y)$  is the *degree crossentropy* loss:

$$L_{DCE} = -\ln\left[1 - \text{abs}(\text{GT} - \text{Pred}_p)\right], \quad (5)$$

where  $\text{Pred}_p$  is the predicted result after the sigmoid function and  $\text{GT}$  is the ground truth. The loss of prediction and any goal  $\in [0, 1]$  is calculated to help us to deal with distance degree information of the text. The specific implementation of  $U_{cle}$  is described by

$$\begin{aligned} U_{cle} &= \begin{cases} 1, & P(x, y) \in \text{TL}(x, y), \\ 1, & P(x, y) \in (\text{DP}(x, y) \cap G), \\ 0, & \text{others,} \end{cases} \\ G &= \{\text{abs}(\text{GT}(x, y) - \text{Pred}(x, y)) > \rho\}, \end{aligned} \quad (6)$$

where  $P(x, y)$  refers to pixel  $(x, y)$  in the entire prediction map.  $\text{TL}(x, y)$  and  $\text{DP}(x, y)$  represent the annotated pixels  $(x, y)$  and postenhanced pixels  $(x, y)$ , respectively.  $G$  is one set of pixels with a difference of more than  $\rho$  between the ground truth and prediction. The postenhanced annotation from the pretrained model may not be quite accurate, and noise interference may exist. Several situations are present in label enhancing. For instance, background pixels are viewed as text pixels as positive annotations. The causes are the annotation differences in the datasets and the

unreliability of the prediction. To make our network learn from noisy or wrong labels, we propose a discriminatory mechanism called  $U_{cle}$ , which calculates the losses of the original label and postenhanced part. In that case, the network performs strong-supervised learning on labelled pixels and distribution supervised learning on post-enhanced pixels. More specifically, the predicted pixel values gradually decrease from the text centre to the edge without fitting the value of the label. The difference between the enhanced annotation and predicted results will be considered reasonable if it is smaller than  $\rho$ . The value of  $\rho$  is set to 0.1 in all the experiments. Therefore, the fault tolerance of  $U_{cle}$  can enhance the robustness of the model and avoid some mistakes from the postenhanced annotation.

**3.5. Postprocessing.** Most segmentation-based methods with segmentation have a common difficulty in which the separation of text instances that are close to each other is challenging. To solve this problem, we propose the apex-edge expansion algorithm that makes full use of the text-mountain shape. Given the prediction result, each text instance appears as a text mountain, as shown in Figure 4(a), where the text centre line region is the peak and the values of the pixels tend to 1. The text edge pixel areas are similar to the feet of the mountain, and their contents are mostly close to zero. Figure 4 presents a vivid example to illustrate the detailed procedure of the apex-edge expansion algorithm.

The detailed procedure of the apex-edge expansion algorithm is shown in Figures 4(b) and 4(c). The post-processing mainly includes three parts. (1) The peak of each text mountain is selected to differentiate the different text instances. The pixel block for which the values of each inner pixel approach 1 is the peak. (2) The *dilate* in OpenCV is used to expand the peak region continuously until reaching the mountain foot or meeting other text areas. The expansion process is divided into many steps  $S_1, S_2, \dots, S_n$ .  $S_i$  ( $i \in [1, n]$ ) represents the entire expansion area in the  $i$ th step.  $S_i - S_{i-1}$  ( $i \in [2, n]$ ) is called the extended area between two adjacent steps. The criterion of expansion ending is that the average score of the extended area approaches 0 or starts to increase. The average score approaching 0 means that the expansion area is close to the background. The increase in the score means that the expansion area begins to cover other text instances. (3) The contour of the whole text instance is represented by many coordinates as the final prediction result after the expansion. The entire postprocessing is shown in Algorithm 1, where  $S_n$  represents the prediction result, and the output  $D_n$  is the set of text instances. Dilation ( $\cdot$ ) is the *dilate* operation in OpenCV. The value and size of the expansion kernel in dilation ( $\cdot$ ) can be changed to realize different direction expansions and different scale expansions. Mean ( $\cdot$ ) is used to calculate the average value of a matrix.  $\cdot$  represents complementing the set.  $\rightarrow$  and  $\Delta$  refer to tending to a number and the value increasing, respectively.



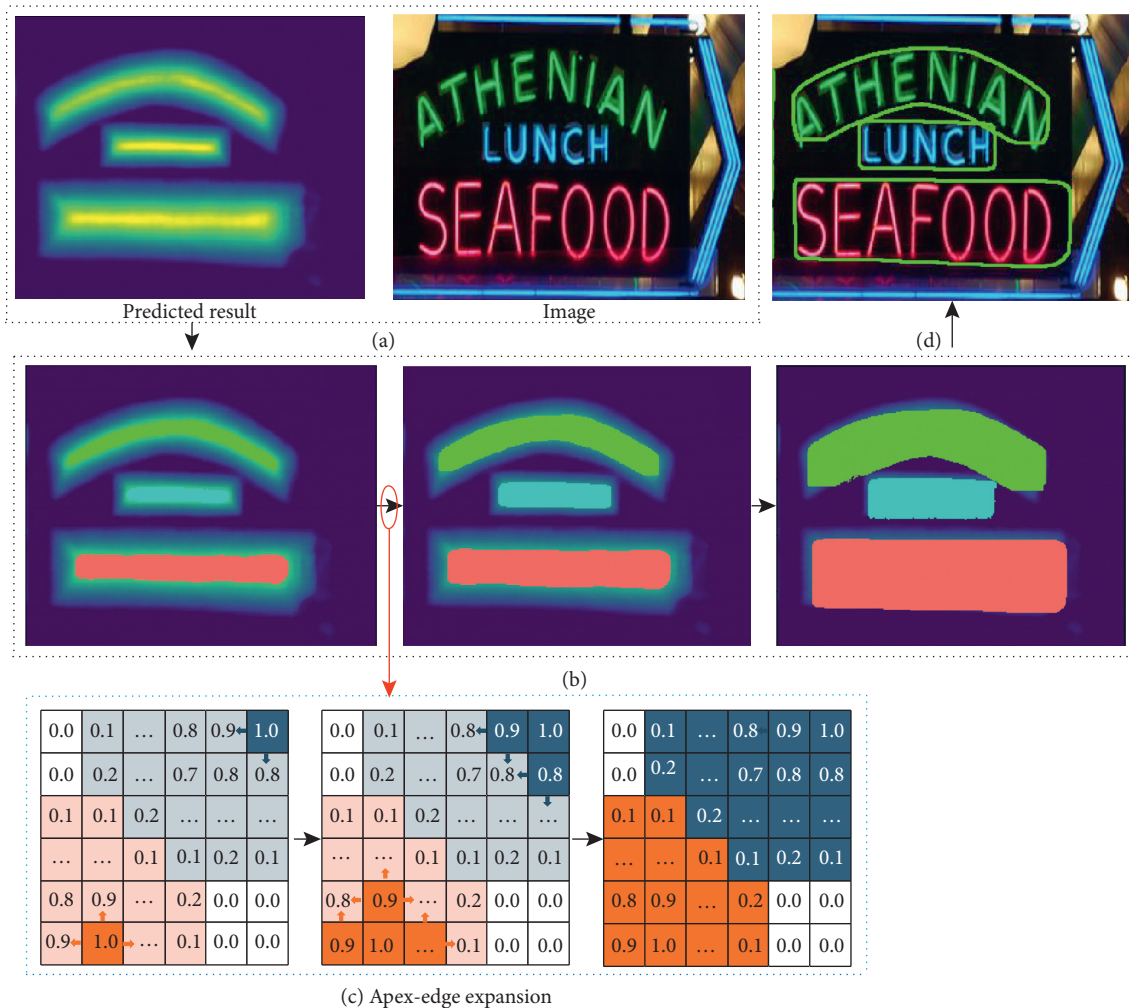


FIGURE 4: Apex-edge expansion: the diagram shows the whole expansion postprocessing process. (a) The original image and the predicted result. (b) Sketch of the expansion showing how to expand from the apex to the edge of the text instance. (c) A more detailed explanation in the two-dimensional decimal matrix. (d) The prediction result.

## 4. Experiments

In this section, we evaluate our approach using ICDAR2015, Total-Text, MSRA-TD500, and CTW1500. The experimental results demonstrate that the performance of the proposed method is comparable to those of the other methods.

**4.1. Datasets.** The datasets used for testing our method are briefly introduced below:

SynthText is a large-scale dataset that contains approximately 800 K synthetic images. These images were created by blending natural images with text rendered with random fonts, sizes, colours, and orientations. We used this dataset to pretrain our model.

ICDAR2015 is a multioriented text detection dataset for English text that includes only 1,000 training images and 500 testing images. The text regions were annotated by four vertices of the quadrilateral.

MSRA-TD500 contains 500 natural images. The indoor images are mainly signs, doorplates, and caution plates,

while the outdoor images are mostly guided boards and billboards in complex backgrounds.

Total-Text is a world-level English curved text dataset that is split into training and testing sets with 1,255 and 300 images, respectively. The text in these images includes more than 3 different text orientations: horizontal, multioriented, and curved.

SCUT-CTW1500 contains 1,000 training images and 500 test images, which contain multioriented text, curved text, and irregularly shaped text. Text regions in this dataset are labelled with 14 scene text boundary points at the sentence level.

Data labelling to test our method: we manually marked Total-Text, CTW1500, and TD500. As shown in Figure 5, the annotation method was brief and inexpensive. For ICDAR2015, the official label was used to fit the text line label for the further verification experiment. The detailed fitting method is simple. The text skeleton as a text line is extracted directly from the full label. All annotations will be released.

```

Input:  $S_n$ : Segmentation result
Output:  $D_n$ : Text instances
 $D_n \leftarrow \phi$ 
 $Apexs \leftarrow [S_n(x, y) \rightarrow 1]$ 
for Apex  $\in$  Apexs do
  if Apex > minimum then
    DK (dilated kernel)  $\leftarrow$  Apex
    while iter < max_iter do
      next_DK  $\leftarrow$  Dilation (DK)//expansion operation
      score_DK  $\leftarrow$  Mean (next_DK  $\cap$   $\overline{DK} \cap S_n$ )
      //get the average score of the extended region
      if score_DK  $\rightarrow$  0 or score_DK  $\Delta$  then
        Enqueue ( $D_n$ , next_DK)//push result into  $D_n$ 
        break
      end if
      DK  $\leftarrow$  next_DK
    end while
  end if
end for

```

ALGORITHM 1: Apex-edge expansion algorithm.



FIGURE 5: The comparison between the text line label and original annotation for three benchmarks. The row on the top is the visualization of the original annotation on (a) CTW1500, (b) Total-Text, and (c) TD500. The middle row is text line annotation and is labelled. The bottom row is the visualization of detection results for the proposed method.

## 4.2. Implementation Details

4.2.1. *Training.* The network was pretrained on SynthText for one epoch and fine tuned on other datasets. We adopted the Adam optimizer as our learning rate scheme. During the pretraining phase, the learning rate was fixed to 0.001. During the fine-tuning stage, the learning rate was initially set to 0.0001 and decayed at a rate of 0.94 every 10,000 insertions. All of the experiments were conducted on a

regular workstation (CPU: Intel (R) Core (TM) i7-7800X CPU @ 3.50 GHz; GPU: GTX 1080). The model was trained with a batch of 4 on one GPU.

VGG16 was adopted as the backbone network for the contrast experiment in our experiments. All of the experiments use the same training strategy: (1) enhancing the text annotation information with the model pretrained on SynthText and (2) training network on the target dataset. To validate the robustness of the proposed method and keep the

same condition in the comparative experiments, all of the models used in label enhancement were the same model pretrained on SynthText for one epoch.

**4.2.2. Data Augmentation.** The images were randomly rotated, cropped, and mirrored at a probability of 0.4. Then, colour and lightness were randomly adjusted. Finally, the images were uniformly resized to  $512 \times 512$ .

**4.2.3. Postprocessing.** We obtained all of the text instances with the apex-edge expansion and then used *findContours* in OpenCV to obtain a set of edge coordinates for each text instance. Finally, the text instances of the regular text datasets (*i.e.*, MSRA-TD500) were described by four coordinate points. Methods such as *minAreaRect* in OpenCV were applied to obtain the bounding boxes of text instances. For curved text datasets, we used a set of coordinate points to describe the text instance (Tables 1 and 2).

**4.3. Detecting Curve Text.** The CTW1500 and Total-Text datasets were used to test the ability of curve text detection. In the experiments, manual text line annotation is used for training. The model pretrained with one epoch on SynthText had two effects: one was to heighten the annotation information, and the other was fine-tuning the pretrained model on other datasets.

The training started with the pretrained model and achieved the best result between 20 and 40 epochs. The F-measure showed a fluctuation of approximately 5%, while the threshold of the peak was in [0.5, 0.8]. For comparative experiments, the threshold of the peak in the apex-edge expansion algorithm was set to 0.6 for CTW1500 and Total-Text for comparative experiments. We continued to expand the peak region until the average score of the extended area approached 0 or met another text instance.

The F-measure of our method with text line was 77.6% on Total-Text, while the F-measure of our method with full masks was 81.1%, as shown in Table 3. The performance with full masks was close to that of the newest method. The difference (3.5%) shows that using the text line can still achieve good results on the challenging poor annotation. The recall (76.7%) was close to the values obtained for the other methods. On CTW1500, our method showed excellent results that were very close to the results obtained by the other strong-supervised methods with an F-measure of 82.3%. The difference (1.9%) between the F-measure of using the text line and that of using the full mask was also acceptable.

**4.4. Detecting Long Text.** TD500 contains many long text scenes and therefore is an excellent dataset for verifying the robustness of the network in long text cases. In the experiment, text line annotation was enhanced by the model pretrained on SynthText. The pretrained model was also used for fine-tuning on TD500. The threshold of the peak in the apex-edge expansion algorithm was set to 0.6, which is the same value as the experiments on CTW1500 and Total-Text. Table 4

TABLE 1: Experimental results for Total-Text. “PT” refers to the model pretrained with one epoch on SynthText. “Ext.” indicates external data. “FM” refers to the model trained with full mask on Total-Text. All listed results were obtained in a strongly supervised manner.

Method	Ext.	Total-Text			
		Precision	Recall	F-measure	FPS
SegLink [40]	—	30.0	23.8	26.7	—
EAST [13]	—	50.0	36.2	42.0	—
Mask TextSpotter [12]	—	69.0	55.0	61.3	—
PSENet [16]	—	81.8	75.1	78.3	3.9
[41]	✓	80.9	76.2	78.5	10.0
TextSnake [4]	✓	82.7	74.5	78.4	—
PSENet [16]	✓	84.0	78.0	80.9	3.9
SPCNet [14]	✓	83.0	<b>82.8</b>	82.9	—
CRAFT [15]	✓	<b>87.6</b>	79.9	<b>83.6</b>	8.6
FM (ours)	PT	83.1	81.6	82.4	10.4
TAS (ours)	PT	78.5	76.7	77.6	<b>11.2</b>

TABLE 2: Experimental results for CTW1500. “PT” refers to the model pretrained with one epoch on SynthText. “Ext.” indicates external data. “FM” refers to the model trained with full mask for strong-supervised learning.

Method	Ext.	CTW1500			
		Precision	Recall	F-measure	FPS
CTPN [42]	—	60.4	53.8	56.9	7.14
SegLink [40]	—	42.3	40.0	40.8	—
EAST [13]	—	78.7	49.1	60.4	—
CTD + TLOC [43]	—	77.4	69.8	73.4	—
PSENet [16]	—	80.6	75.6	78.0	3.9
TextSnake [4]	✓	67.9	<b>85.3</b>	75.6	—
[41]	✓	80.2	80.1	80.1	<b>10</b>
TextField [29]	✓	79.8	83.0	81.4	—
PSENet [16]	✓	84.8	79.7	82.2	3.9
CRAFT [15]	✓	<b>88.2</b>	78.2	<b>82.9</b>	8.6
FM (ours)	PT	86.2	80.5	83.2	9.0
TAS (ours)	PT	83.8	80.8	82.3	9.2

TABLE 3: Effectiveness of label enhancement and the pretrained model on Total-Text.

Method	Pretrain-enhancement	Label	F-measure
Baseline + TL	No	TL	65.0
Baseline + FL	No	Full	81.1
Baseline + TL* + PT	SynthText	TL*	77.6
Baseline + TL + PT $\Delta$	CTW1500	TL	79.1
Baseline + TL + PT	SynthText	TL	77.2

FL and TL refer to using full mask and text line label, respectively. TL\* is the synthetic text line label. PT and PT $\Delta$  are the different models pretrained on SynthText and CTW1500, respectively.

compares the proposed method with state-of-the-art methods on TD500. The proposed method achieved an F-measure of 77.2%, which is competitive with other state-of-the-art detectors trained in a strongly supervised way.

**4.5. Detecting Oriented Text.** All of the parameter settings and training details for ICDAR2015 were the same as those



TABLE 4: Experimental results for SCUT-TD500. “PT” refers to the model pretrained on SynthText. “Ext.” indicates external data. All compared methods were trained in a strongly supervised way. “FM” refers to the model trained with a full mask for strong-supervised learning.

Method	Ext.	MSRA-TD500			
		Precision	Recall	F-measure	FPS
EAST [13]	—	87.3	67.4	76.1	<b>13.2</b>
RRPN [20]	—	82.0	68.0	74.0	—
DeepReg [18]	—	77.0	70.0	74.0	1.1
SegLink [40]	√	86.0	70.0	77.0	8.9
PixelLink [28]	√	83.0	73.2	77.8	3.0
RRD [26]	√	87.0	73.0	79.0	10
TextSnak [4]	√	83.2	73.9	78.3	1.1
[44]	√	87.6	76.2	82.9	—
TextField [29]	√	75.9	87.4	81.3	—
CRAFT [15]	√	<b>88.2</b>	78.2	82.9	8.6
MCN [45]	√	79.0	<b>88.0</b>	<b>83.0</b>	—
FM (ours)	PT	83.2	76.6	79.8	11.6
TAS (ours)	PT	80.6	74.1	77.2	12.0

for the experiments on the curve text datasets. The official label was used to fit the text line label for the further verified experiment on ICDAR2015. Similar to the experiment on TD500, *minAreaRect* in OpenCV was used to obtain the bounding boxes of the text instance, in contrast to several detectors listed in Table 5 that used extra datasets. For instance, the F-measure of PSENet [16] was 80.5% without an extra dataset. The F-measure (79.4%) of our method was already comparatively close to those of the other methods.

**4.6. Ablation Study.** Three groups of comparative experiments were performed to verify the effectiveness of our method.

**4.6.1. Baseline.** The baseline was trained with the text line without label enhancement, and the F-measure of the baseline on Total-Text was 65.0%, as shown in Table 3.

**4.6.2. Label Enhancement.** The results are shown in Table 3, which are further analysed for label enhancement of the model on Total-Text. Training with an unenhanced text line shows an unsatisfactory performance (65.0%), while training with a full mask obtained an F-measure of 81.1%. The large difference (16.1%) indicates that the text line loses important supervision information. After introducing the pretrained model on SynthText to enhance the text line, the performance of the model had an obvious improvement from 65.0% to 77.2%. In addition, using the synthetic text line from the full mask shows better performance (77.6%). The main reason for this is that the manual text line had a larger error in extracting the text skeleton compared to the synthetic text line. In addition, we also compared the performance of the model pretrained on different datasets: synthetic data (i.e., SynthText) and realistic data (i.e., SUCT-CTW1500). The F-measures using SynthText and CTW1500 were 77.2% and 79.1%, respectively. Obviously, the

TABLE 5: Experimental results for ICDAR2015. “PT” refers to the model pretrained on SynthText for one epoch. “Ext.” indicates using external data.

Method	Ext.	ICDAR15			
		Precision	Recall	F-measure	FPS
CTPN [42]	—	74.2	51.6	60.9	7.1
EAST [13]	—	83.6	73.5	78.2	6.5
RRPN [20]	—	82.0	73.0	77.0	—
PSENet [16]	—	81.4	79.6	80.5	1.6
PixelLink [28]	—	82.9	81.7	82.3	7.3
SegLink [40]	√	73.1	76.8	75.0	—
SSTD [46]	√	80.2	73.9	76.9	7.7
WordSup [35]	√	79.3	77.0	78.2	—
RRD [26]	√	85.6	79.0	82.2	6.5
MCN [45]	√	72.0	80.0	76.0	—
TextField [29]	√	80.5	84.3	82.4	—
TextSnake [4]	√	84.9	80.4	82.6	1.1
PAN [47]	√	84.0	81.9	82.9	<b>26.1</b>
PSENet [16]	√	86.9	84.5	85.7	1.6
CRAFT [15]	√	<b>89.8</b>	84.3	86.9	8.6
SPCNet [14]	√	88.7	<b>85.8</b>	<b>87.2</b>	—
TAS (ours)	PT	81.7	77.1	79.4	8.5

performance of our model pretrained with realistic data shows a few advantages. This also indicates an intrinsic limitation of this method and the dependence on the pretrained model.

**4.6.3. Geometric Parameters of the Text Line.** As shown in Table 6, the impact of the width and the offset of the text line was evaluated. For the width of the text line, we used different widths of synthetic or manually marked text lines to test our model. For the manually marked text line, we extracted its skeleton of one-pixel width and dilated the skeleton to different widths while the width was less than that of the original text line. For the synthetic text line, the skeleton of one pixel was extracted from the full mask and used to create different widths. While the width of the text line was the same, using the synthetic text line which usually achieved a better performance than using the manual text line, and the average difference was approximately 0.4%. In addition, with increasing width, the F-measure showed a fluctuation of approximately 1%. The offset of the text line was set to 0 in all experiments to evaluate the influence of the text line width.

Apart from the evaluation of the influence of width, the offset between the synthetic text line and centre line of the text instance was also set to test our detection method. The offset in Table 6 refers to the offset error ratio:  $D_o/D_t$ .  $D_o$  is the distance between the text line and text centre line, and  $D_t$  is the width of the text region. In the experiment, we only performed the experiment on the synthetic text line, while the offset between the manual text line and text centre line was difficult to calculate. The text centre line was calculated from the original coordinate annotation, and then we created the text line by setting the corresponding offset ratio. The curvature and width of the created text line were the same as those of the text centre line. All widths of the text

TABLE 6: Effectiveness of different text line labelling qualities on Total-Text.

Width (pixel)	Method	F-measure	Method	F-measure
1	Baseline + TL	77.2	Baseline + TL*	77.6
3	Baseline + TL	77.8	Baseline + TL*	77.8
5	Baseline + TL	77.5	Baseline + TL*	78.4
7	Baseline + TL	78.1	Baseline + TL*	78.3
Method	Width (pixel)	Offset (%)	F-measure	—
Baseline + TL*	1	0	77.6	—
Baseline + TL*	1	10	77.4	—
Baseline + TL*	1	20	77.8	—
Baseline + TL*	1	30	76.4	—
Baseline + TL*	1	40	75.2	—

TL\* is the synthetic text line with a full mask. TL refers to the manually marked text line.

TABLE 7: Detection results with different backbones on CTW1500.

Backbone	Method	Precision	Recall	F-measure	FPS
VGG16	Baseline + TL	83.8	80.8	82.3	9.2
VGG11	Baseline + TL	82.5	81.2	81.9	11.2
ResNet-18	Baseline + TL	82.4	79.4	80.9	8.7
ResNet-50	Baseline + TL	84.5	80.8	82.7	6.9
ResNet-101	Baseline + TL	84.3	81.6	82.9	6.6

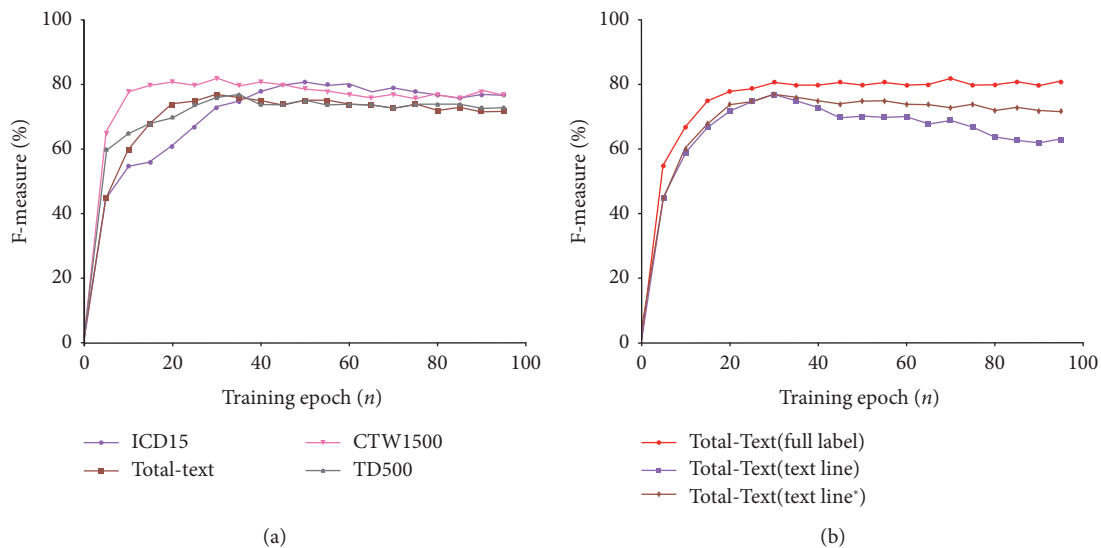


FIGURE 6: Influence of training epochs: (a) results for different datasets with the text line and (b) results for the same dataset with different annotations. \* represents  $U_{cle}$  used in the loss function.

line or text centre line were one pixel in the experiment. While the offset ratio of the text line was below 20%, the F-measure barely fluctuated. While the offset ratio of the text line exceeded 20%, the performance of the model started to be affected slightly, but the fluctuation around 2% was still acceptable.

**4.6.4. Backbone.** As shown in Table 7, a series of experiments comparing different backbones were performed to evaluate its influence on the proposed method. Similar to VGG16, five feature maps generated from VGG11 were gradually upsampled to the original size. For the ResNet

series, four feature maps were used to merge. The F-measure using VGG11 was similar to that of using VGG16, but the latter had a slightly slower inference time. Due to the sophisticated design, the ResNet series had a longer convergence time, but the performance was comparatively accurate and stable.

**4.6.5. Loss Function.** As shown in Figure 6(a), due to the instability of the enhanced annotation, the F-measure decreased after dozens of epochs on four common datasets, particularly for curved text datasets. As shown in Figure 6(b), training with the text line was unstable relative



to the method with full labelling, and the model with full labelling showed better convergence performance with an increasing number of training epochs. After incorporating  $U_{cle}$  into the loss function, the model with the text line showed improved convergence, with convergence fluctuation of approximately 3%.

## 5. Conclusion and Future Work

In this paper, we first introduced a novel text detector based on weakly supervised learning. The most prominent feature of the method was proposing a novel labelling named the text line and the full use of the model pretrained on SynthText. The use of a text line can help the detector decrease the cost of labelling, and the pretrained model can improve the performance of the detector. The experiments showed that the text line with low-cost labelling can be used to train an effective text detector and further verify the feasibility of using a synthetic text dataset to enhance weak labels. Efficient low-cost text detectors have potential applications in the field of photo translation. Synthetic data will play an increasingly important role in the field of deep learning in the future. One reason for this is that the high cost of annotation hinders the application of actual scenes for arithmetic. Another reason is that synthetic data are increasingly similar to real-world images, and the development of auxiliary methods promotes the development of synthetic text. In future work, it will be important to train the methods with synthetic data but apply them to the real world.

## Data Availability

The data are now made public at <https://github.com/xingjici/Texts-as-Lines-Text-Detection-with-Weak-Supervision> and the corresponding code is still cleaning up. Data description can be found in Abstraction sector.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Weijia Wu and Jici Xing contributed equally to this work.

## Acknowledgments

This work was supported by the National Key Research and Development Project (Grant no. 2019YFC0118202), National Natural Science Foundation of China (Grant nos. 61803332 and 11574269), Natural Science Foundation of Zhejiang Province (Grant no. LQ18E050001), Fundamental Research Funds for the Central Universities (Grant no. 2019FZJD005), and Scientific Research Fund of Zhejiang Provincial Education Department (Grant no. Y201941642).

## References

- [1] H. Wang, Y. Sun, and M. Liu, "Self-supervised drivable area and road anomaly segmentation using RGB-D data for

- robotic wheelchairs," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4386–4393, 2019.
- [2] Y. Wang, H. Xie, Z. Fu, and Y. Zhang, "DSRN: a deep scale relationship network for scene text detection," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 947–953, AAAI Press, Macao, China, August 2019.
- [3] H. Xie, S. Fang, Z.-J. Zha, Y. Yang, Y. Li, and Y. Zhang, "Convolutional attention networks for scene text recognition," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 1, pp. 1–17, 2019.
- [4] S. Long, J. Ruan, W. Zhang et al., "TextSnake: a flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–36, Munich, Germany, September 2018.
- [5] D. Karatzas, F. Shafait, S. Uchida et al., "ICDAR 2013 robust reading competition," in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, pp. 1484–1493, IEEE, Washington, DC, USA, August 2013.
- [6] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou et al., "ICDAR 2015 competition on robust reading," in *Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1156–1160, IEEE, Tunis, Tunisia, August 2015.
- [7] N. Nayef, F. Yin, B. Imen et al., "ICDAR 2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT," in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1454–1459, IEEE, Kyoto, Japan, November 2017.
- [8] C. K. Ch'ng and C. S. Chan, "Total-text: a comprehensive dataset for scene text detection and recognition," in *Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 935–942, IEEE, Kyoto, Japan, November 2017.
- [9] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognition*, vol. 90, pp. 337–345, 2019.
- [10] C. Yao, B. Xiang, W. Liu, Yi Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, IEEE, Providence, RI, USA, June 2012.
- [11] He Tong, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5020–5029, Salt Lake City, UT, USA, June 2018.
- [12] P. Lyu, M. Liao, C. Yao, W. Wu, and B. Xiang, "Mask text-spotter: an end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 67–83, Munich, Germany, September 2018.
- [13] X. Zhou, C. Yao, W. He et al., "East: an efficient and accurate scene text detector," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5551–5560, Honolulu, HI, USA, July 2017.
- [14] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9038–9045, 2019.
- [15] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9365–9374, Long Beach, CA, USA, June 2019.

- [16] Li Xiang, W. Wang, W. Hou, R.-Ze Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," 2018, <https://arxiv.org/abs/1903.12473>.
- [17] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2315–2324, Las Vegas, NV, USA, June 2016.
- [18] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 745–753, Venice, Italy, October 2017.
- [19] M. Liao, B. Shi, and X. Bai, "Textboxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.
- [20] J. Ma, W. Shao, H. Ye et al., "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.
- [21] S. Zhang, Y. Liu, L. Jin, and C. Luo, "Feature enhancement network: a refined scene text detector," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.
- [22] Y. Jiang, X. Zhu, X. Wang et al., "R2CNN: rotational region cnn for orientation robust scene text detection," 2017, <https://arxiv.org/abs/1706.09579>.
- [23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, Neural Information Processing Systems Foundation Inc. La Jolla, CA, USA, 2015.
- [25] M. Liao, B. Shi, B. Xiang, X. Wang, and W. Liu, "Textboxes: a fast text detector with a single deep neural network," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.
- [26] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and B. Xiang, "Rotation-sensitive regression for oriented scene text detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5909–5918, Salt Lake City, UT, USA, June 2018.
- [27] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, Venice, Italy, October 2017.
- [28] D. Deng, H. Liu, X. Li, and C. Deng, "Pixellink: detecting scene text via instance segmentation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.
- [29] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.
- [30] Y. Zhu and J. Du, "Textmountain: accurate scene text detection via instance segmentation," 2018, <https://arxiv.org/abs/1811.12786>.
- [31] T. Sun, Y. Sun, M. Liu, and D.-Y. Yeung, "Movable-object-aware visual slam via weakly supervised semantic segmentation," 2019, <https://arxiv.org/abs/1906.03629>.
- [32] Y. Sun, W. Zuo, and M. Liu, "RTFNet: RGB-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2576–2583, 2019.
- [33] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised CNN segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1818–1827, Salt Lake City, UT, USA, June 2018.
- [34] S. Tian, S. Lu, and C. Li, "WeText: scene text detection under weak supervision," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1492–1500, Venice, Italy, October 2017.
- [35] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: exploiting word annotations for character based text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4940–4949, Venice, Italy, October 2017.
- [36] Q. Li, A. Arnab, and P. H. S. Torr, "Weakly-and semi-supervised panoptic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 102–118, Munich, Germany, September 2018.
- [37] Di Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3159–3167, Las Vegas, NV, USA, June 2016.
- [38] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to segment under various forms of weak supervision," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3781–3790, Boston, MA, USA, June 2015.
- [39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [40] B. Shi, B. Xiang, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2550–2558, Honolulu, HI, USA, July 2017.
- [41] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6449–6458, Long Beach, CA, USA, June 2019.
- [42] Z. Tian, W. Huang, He Tong, He Pan, and Yu Qiao, "Detecting text in natural image with connectionist text proposal network," in *European Conference on Computer Vision*, pp. 56–72, Springer, Berlin, Germany, 2016.
- [43] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: new dataset and new solution," 2017, <https://arxiv.org/abs/1712.02170>.
- [44] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7553–7563, Salt Lake City, UT, USA, June 2018.
- [45] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and L. G. Wang, "Learning markov clustering networks for scene text detection," 2018, <https://arxiv.org/abs/1805.08365>.
- [46] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3047–3055, Venice, Italy, October 2017.
- [47] W. Wang, E. Xie, X. Song et al., "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8440–8449, Seoul, South Korea, November 2019.