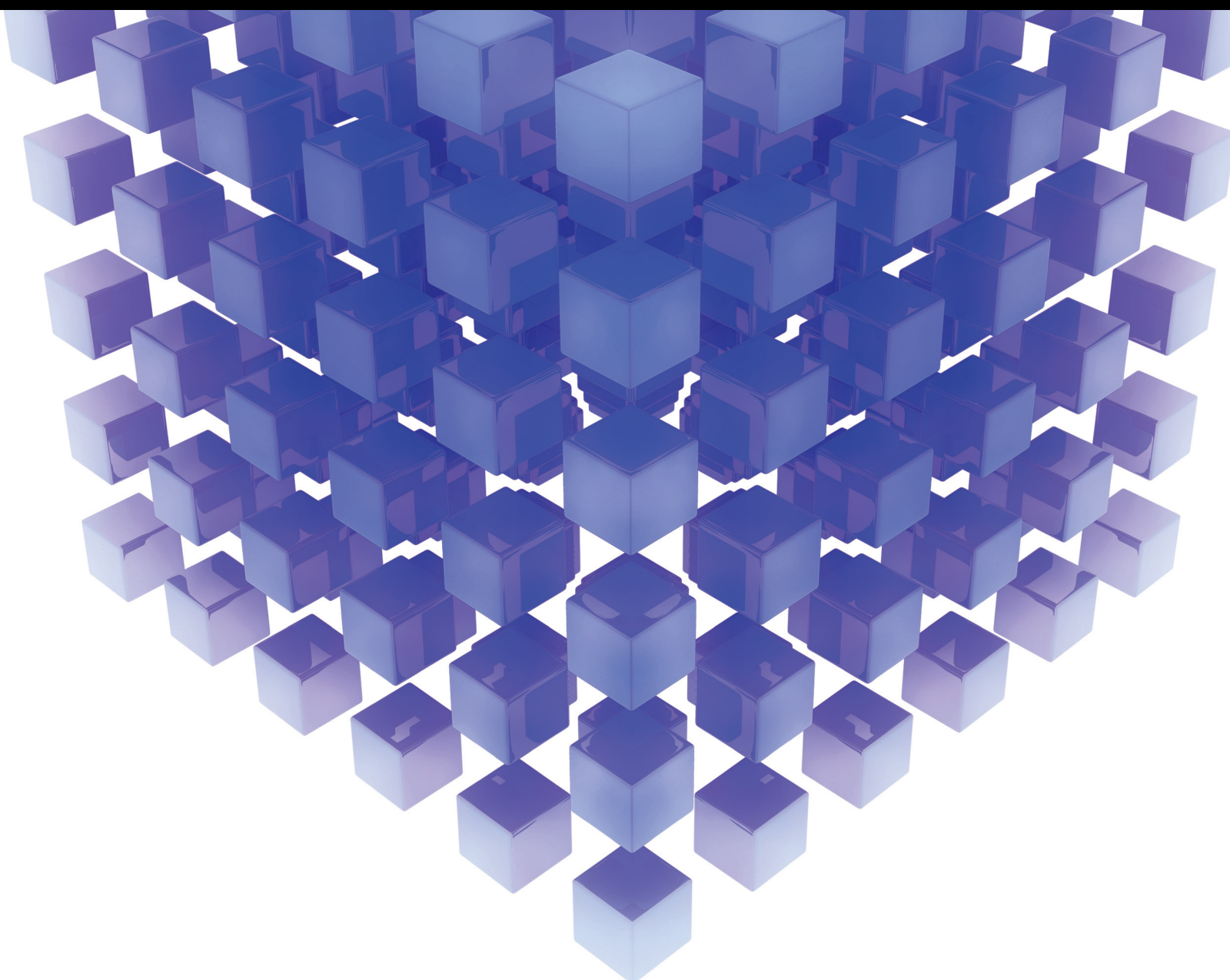


# Advanced Deep Learning Methods for Online Monitoring and Health Assessment

Lead Guest Editor: Shengwei Fei

Guest Editors: Abolfazl Gharaei and Chaoqun Duan





---

# **Advanced Deep Learning Methods for Online Monitoring and Health Assessment**

Mathematical Problems in Engineering

---

**Advanced Deep Learning Methods  
for Online Monitoring and Health  
Assessment**

Lead Guest Editor: Shengwei Fei

Guest Editors: Abolfazl Gharaei and Chaoqun  
Duan




---

Copyright © 2023 Hindawi Limited. All rights reserved.

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

# Chief Editor

Guangming Xie , China

## Academic Editors

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

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

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

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



































José António Fonseca De Oliveira  
Correia , Portugal  
Jose Renato De Sousa , Brazil  
Michael Defoort, France  
Alessandro Della Corte, Italy  
Laurent Dewasme , Belgium  
Sanku Dey , India  
Gianpaolo Di Bona , Italy  
Roberta Di Pace , Italy  
Francesca Di Puccio , Italy  
Ramón I. Diego , Spain  
Yannis Dimakopoulos , Greece  
Hasan Dinçer , Turkey  
José M. Domínguez , Spain  
Georgios Dounias, Greece  
Bo Du , China  
Emil Dumic, Croatia  
Madalina Dumitriu , United Kingdom  
Premraj Durairaj , India  
Saeed Eftekhari Azam, USA  
Said El Kafhali , Morocco  
Antonio Elipse , Spain  
R. Emre Erkmen, Canada  
John Escobar , Colombia  
Leandro F. F. Miguel , Brazil  
FRANCESCO FOTI , Italy  
Andrea L. Facci , Italy  
Shahla Faisal , Pakistan  
Giovanni Falsone , Italy  
Hua Fan, China  
Jianguang Fang, Australia  
Nicholas Fantuzzi , Italy  
Muhammad Shahid Farid , Pakistan  
Hamed Faruqi, Iran  
Yann Favennec, France  
Fiorenzo A. Fazzolari , United Kingdom  
Giuseppe Fedele , Italy  
Roberto Fedele , Italy  
Baowei Feng , China  
Mohammad Ferdows , Bangladesh  
Arturo J. Fernández , Spain  
Jesus M. Fernandez Oro, Spain  
Francesco Ferrise, Italy  
Eric Feulvarch , France  
Thierry Floquet, France

Eric Florentin , France  
Gerardo Flores, Mexico  
Antonio Forcina , Italy  
Alessandro Formisano, Italy  
Francesco Franco , Italy  
Elisa Francomano , Italy  
Juan Frausto-Solis, Mexico  
Shujun Fu , China  
Juan C. G. Prada , Spain  
HECTOR GOMEZ , Chile  
Matteo Gaeta , Italy  
Mauro Gaggero , Italy  
Zoran Gajic , USA  
Jaime Gallardo-Alvarado , Mexico  
Mosè Gallo , Italy  
Akemi Gálvez , Spain  
Maria L. Gandarias , Spain  
Hao Gao , Hong Kong  
Xingbao Gao , China  
Yan Gao , China  
Zhiwei Gao , United Kingdom  
Giovanni Garcea , Italy  
José García , Chile  
Harish Garg , India  
Alessandro Gasparetto , Italy  
Stylianos Georgantzinou, Greece  
Fotios Georgiades , India  
Parviz Ghadimi , Iran  
Ştefan Cristian Gherghina , Romania  
Georgios I. Giannopoulos , Greece  
Agathoklis Giaralis , United Kingdom  
Anna M. Gil-Lafuente , Spain  
Ivan Giorgio , Italy  
Gaetano Giunta , Luxembourg  
Jefferson L.M.A. Gomes , United Kingdom  
Emilio Gómez-Déniz , Spain  
Antonio M. Gonçalves de Lima , Brazil  
Qunxi Gong , China  
Chris Goodrich, USA  
Rama S. R. Gorla, USA  
Veena Goswami , India  
Xunjie Gou , Spain  
Jakub Grabski , Poland

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



Irfan Kaymaz , Turkey  
Vahid Kayvanfar , Qatar  
Krzysztof Kecik , Poland  
Mohamed Khader , Egypt  
Chaudry M. Khalique , South Africa  
Mukhtaj Khan , Pakistan  
Shahid Khan , Pakistan  
Nam-Il Kim, Republic of Korea  
Philipp V. Kiryukhantsev-Korneev ,  
Russia  
P.V.V Kishore , India  
Jan Koci , Czech Republic  
Ioannis Kostavelis , Greece  
Sotiris B. Kotsiantis , Greece  
Frederic Kratz , France  
Vamsi Krishna , India  
Edyta Kucharska, Poland  
Krzysztof S. Kulpa , Poland  
Kamal Kumar, India  
Prof. Ashwani Kumar , India  
Michal Kunicki , Poland  
Cedrick A. K. Kwuimy , USA  
Kyandoghere Kyamakya, Austria  
Ivan Kyrchei , Ukraine  
Márcio J. Lacerda , Brazil  
Eduardo Lalla , The Netherlands  
Giovanni Lancioni , Italy  
Jaroslaw Latalski , Poland  
Hervé Laurent , France  
Agostino Lauria , Italy  
Aimé Lay-Ekuakille , Italy  
Nicolas J. Leconte , France  
Kun-Chou Lee , Taiwan  
Dimitri Lefebvre , France  
Eric Lefevre , France  
Marek Lefik, Poland  
Yaguo Lei , China  
Kauko Leiviskä , Finland  
Ervin Lenzi , Brazil  
ChenFeng Li , China  
Jian Li , USA  
Jun Li , China  
Yueyang Li , China  
Zhao Li , China































Zhen Li , China  
En-Qiang Lin, USA  
Jian Lin , China  
Qibin Lin, China  
Yao-Jin Lin, China  
Zhiyun Lin , China  
Bin Liu , China  
Bo Liu , China  
Heng Liu , China  
Jianxu Liu , Thailand  
Lei Liu , China  
Sixin Liu , China  
Wanquan Liu , China  
Yu Liu , China  
Yuanchang Liu , United Kingdom  
Bonifacio Llamazares , Spain  
Alessandro Lo Schiavo , Italy  
Jean Jacques Loiseau , France  
Francesco Lolli , Italy  
Paolo Lonetti , Italy  
António M. Lopes , Portugal  
Sebastian López, Spain  
Luis M. López-Ochoa , Spain  
Vassilios C. Loukopoulos, Greece  
Gabriele Maria Lozito , Italy  
Zhiguo Luo , China  
Gabriel Luque , Spain  
Valentin Lychagin, Norway  
YUE MEI, China  
Junwei Ma , China  
Xuanlong Ma , China  
Antonio Madeo , Italy  
Alessandro Magnani , Belgium  
Toqeer Mahmood , Pakistan  
Fazal M. Mahomed , South Africa  
Arunava Majumder , India  
Sarfranz Nawaz Malik, Pakistan  
Paolo Manfredi , Italy  
Adnan Maqsood , Pakistan  
Muazzam Maqsood, Pakistan  
Giuseppe Carlo Marano , Italy  
Damijan Markovic, France  
Filipe J. Marques , Portugal  
Luca Martinelli , Italy  
Denizar Cruz Martins, Brazil

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

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

George A. Papakostas , Greece  
Xosé M. Pardo , Spain  
You-Jin Park, Taiwan  
Manuel Pastor, Spain  
Pubudu N. Pathirana , Australia  
Surajit Kumar Paul , India  
Luis Payá , Spain  
Igor Pažanin , Croatia  
Libor Pekař , Czech Republic  
Francesco Pellicano , Italy  
Marcello Pellicciari , Italy  
Jian Peng , China  
Mingshu Peng, China  
Xiang Peng , China  
Xindong Peng, China  
Yuexing Peng, China  
Marzio Pennisi , Italy  
Maria Patrizia Pera , Italy  
Matjaz Perc , Slovenia  
A. M. Bastos Pereira , Portugal  
Wesley Peres, Brazil  
F. Javier Pérez-Pinal , Mexico  
Michele Perrella, Italy  
Francesco Pesavento , Italy  
Francesco Petrini , Italy  
Hoang Vu Phan, Republic of Korea  
Lukasz Pieczonka , Poland  
Dario Piga , Switzerland  
Marco Pizzarelli , Italy  
Javier Plaza , Spain  
Goutam Pohit , India  
Dragan Poljak , Croatia  
Jorge Pomares , Spain  
Hiram Ponce , Mexico  
Sébastien Poncet , Canada  
Volodymyr Ponomaryov , Mexico  
Jean-Christophe Ponsart , France  
Mauro Pontani , Italy  
Sivakumar Poruran, India  
Francesc Pozo , Spain  
Aditya Rio Prabowo , Indonesia  
Anchasa Pramuanjaroenkij , Thailand  
Leonardo Primavera , Italy  
B Rajanarayan Prusty, India

Krzysztof Puszynski , Poland  
Chuan Qin , China  
Dongdong Qin, China  
Jianlong Qiu , China  
Giuseppe Quaranta , Italy  
DR. RITU RAJ , India  
Vitomir Racic , Italy  
Carlo Rainieri , Italy  
Kumbakonam Ramamani Rajagopal, USA  
Ali Ramazani , USA  
Angel Manuel Ramos , Spain  
Higinio Ramos , Spain  
Muhammad Afzal Rana , Pakistan  
Muhammad Rashid, Saudi Arabia  
Manoj Rastogi, India  
Alessandro Rasulo , Italy  
S.S. Ravindran , USA  
Abdolrahman Razani , Iran  
Alessandro Reali , Italy  
Jose A. Reinoso , Spain  
Oscar Reinoso , Spain  
Haijun Ren , China  
Carlo Renno , Italy  
Fabrizio Renno , Italy  
Shahram Rezapour , Iran  
Ricardo Rianza , Spain  
Francesco Riganti-Fulginei , Italy  
Gerasimos Rigatos , Greece  
Francesco Ripamonti , Italy  
Jorge Rivera , Mexico  
Eugenio Roanes-Lozano , Spain  
Ana Maria A. C. Rocha , Portugal  
Luigi Rodino , Italy  
Francisco Rodríguez , Spain  
Rosana Rodríguez López, Spain  
Francisco Rossomando , Argentina  
Jose de Jesus Rubio , Mexico  
Weiguo Rui , China  
Rubén Ruiz , Spain  
Ivan D. Rukhlenko , Australia  
Dr. Eswaramoorthi S. , India  
Weichao SHI , United Kingdom  
Chaman Lal Sabharwal , USA  
Andrés Sáez , Spain

Bekir Sahin, Turkey  
Laxminarayan Sahoo , India  
John S. Sakellariou , Greece  
Michael Sakellariou , Greece  
Salvatore Salamone, USA  
Jose Vicente Salcedo , Spain  
Alejandro Salcido , Mexico  
Alejandro Salcido, Mexico  
Nunzio Salerno , Italy  
Rohit Salgotra , India  
Miguel A. Salido , Spain  
Sinan Salih , Iraq  
Alessandro Salvini , Italy  
Abdus Samad , India  
Sovan Samanta, India  
Nikolaos Samaras , Greece  
Ramon Sancibrian , Spain  
Giuseppe Sanfilippo , Italy  
Omar-Jacobo Santos, Mexico  
J Santos-Reyes , Mexico  
José A. Sanz-Herrera , Spain  
Musavarah Sarwar, Pakistan  
Shahzad Sarwar, Saudi Arabia  
Marcelo A. Savi , Brazil  
Andrey V. Savkin, Australia  
Tadeusz Sawik , Poland  
Roberta Sburlati, Italy  
Gustavo Scaglia , Argentina  
Thomas Schuster , Germany  
Hamid M. Sedighi , Iran  
Mijanur Rahaman Seikh, India  
Tapan Senapati , China  
Lotfi Senhadji , France  
Junwon Seo, USA  
Michele Serpilli, Italy  
Silvestar Šesnić , Croatia  
Gerardo Severino, Italy  
Ruben Sevilla , United Kingdom  
Stefano Sfarra , Italy  
Dr. Ismail Shah , Pakistan  
Leonid Shaikhet , Israel  
Vimal Shanmuganathan , India  
Prayas Sharma, India  
Bo Shen , Germany  
Hang Shen, China

Xin Pu Shen, China  
Dimitri O. Shepelsky, Ukraine  
Jian Shi , China  
Amin Shokrollahi, Australia  
Suzanne M. Shontz , USA  
Babak Shotorban , USA  
Zhan Shu , Canada  
Angelo Sifaleras , Greece  
Nuno Simões , Portugal  
Mehakpreet Singh , Ireland  
Piyush Pratap Singh , India  
Rajiv Singh, India  
Seralathan Sivamani , India  
S. Sivasankaran , Malaysia  
Christos H. Skiadas, Greece  
Konstantina Skouri , Greece  
Neale R. Smith , Mexico  
Bogdan Smolka, Poland  
Delfim Soares Jr. , Brazil  
Alba Sofi , Italy  
Francesco Soldovieri , Italy  
Raffaele Solimene , Italy  
Yang Song , Norway  
Jussi Sopanen , Finland  
Marco Spadini , Italy  
Paolo Spagnolo , Italy  
Ruben Specogna , Italy  
Vasilios Spitas , Greece  
Ivanka Stamova , USA  
Rafał Stanisławski , Poland  
Miladin Stefanović , Serbia  
Salvatore Strano , Italy  
Yakov Strelniker, Israel  
Kangkang Sun , China  
Qiuqin Sun , China  
Shuaishuai Sun, Australia  
Yanchao Sun , China  
Zong-Yao Sun , China  
Kumarasamy Suresh , India  
Sergey A. Suslov , Australia  
D.L. Suthar, Ethiopia  
D.L. Suthar , Ethiopia  
Andrzej Swierniak, Poland  
Andras Szekrenyes , Hungary  
Kumar K. Tamma, USA

Yong (Aaron) Tan, United Kingdom  
Marco Antonio Taneco-Hernández , Mexico  
Lu Tang , China  
Tianyou Tao, China  
Hafez Tari , USA  
Alessandro Tasora , Italy  
Sergio Teggi , Italy  
Adriana del Carmen Téllez-Anguiano , Mexico  
Ana C. Teodoro , Portugal  
Efstathios E. Theotokoglou , Greece  
Jing-Feng Tian, China  
Alexander Timokha , Norway  
Stefania Tomasiello , Italy  
Gisella Tomasini , Italy  
Isabella Torricollo , Italy  
Francesco Tornabene , Italy  
Mariano Torrisi , Italy  
Thang nguyen Trung, Vietnam  
George Tsiatas , Greece  
Le Anh Tuan , Vietnam  
Nerio Tullini , Italy  
Emilio Turco , Italy  
Ilhan Tuzcu , USA  
Efstratios Tzirtzilakis , Greece  
FRANCISCO UREÑA , Spain  
Filippo Ubertini , Italy  
Mohammad Uddin , Australia  
Mohammad Safi Ullah , Bangladesh  
Serdar Ulubeyli , Turkey  
Mati Ur Rahman , Pakistan  
Panayiotis Vafeas , Greece  
Giuseppe Vairo , Italy  
Jesus Valdez-Resendiz , Mexico  
Eusebio Valero, Spain  
Stefano Valvano , Italy  
Carlos-Renato Vázquez , Mexico  
Martin Velasco Villa , Mexico  
Franck J. Vernerey, USA  
Georgios Veronis , USA  
Vincenzo Vespri , Italy  
Renato Vidoni , Italy  
Venkatesh Vijayaraghavan, Australia



Anna Vila, Spain  
Francisco R. Villatoro , Spain  
Francesca Vipiana , Italy  
Stanislav Vitek , Czech Republic  
Jan Vorel , Czech Republic  
Michael Vynnycky , Sweden  
Mohammad W. Alomari, Jordan  
Roman Wan-Wendner , Austria  
Bingchang Wang, China  
C. H. Wang , Taiwan  
Dagang Wang, China  
Guoqiang Wang , China  
Huaiyu Wang, China  
Hui Wang , China  
J.G. Wang, China  
Ji Wang , China  
Kang-Jia Wang , China  
Lei Wang , China  
Qiang Wang, China  
Qingling Wang , China  
Weiwei Wang , China  
Xinyu Wang , China  
Yong Wang , China  
Yung-Chung Wang , Taiwan  
Zhenbo Wang , USA  
Zhibo Wang, China  
Waldemar T. Wójcik, Poland  
Chi Wu , Australia  
Qihong Wu, China  
Yuqiang Wu, China  
Zhibin Wu , China  
Zhizheng Wu , China  
Michalis Xenos , Greece  
Hao Xiao , China  
Xiao Ping Xie , China  
Qingzheng Xu , China  
Binghan Xue , China  
Yi Xue , China  
Joseph J. Yame , France  
Chuanliang Yan , China  
Xinggang Yan , United Kingdom  
Hongtai Yang , China  
Jixiang Yang , China  
Mijia Yang, USA  
Ray-Yeng Yang, Taiwan

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

## Contents

---

### **A Deep Learning Approach to Detect Microsleep Using Various Forms of EEG Signal**

S. K. B. Sangeetha, Sandeep Kumar Mathivanan, V. Muthukumaran , N. Pughazendi, Prabhu Jayagopal, and Md Salah Uddin 


Research Article (8 pages), Article ID 7317938, Volume 2023 (2023)

### **Prediction of Flank Wear during Turning of EN8 Steel with Cutting Force Signals Using a Deep Learning Approach**

S. K. Thangarasu , T. Mohanraj , K. Devendran , M. Rajalakshmi , Subrata Chowdhury , and Saravanakumar Gurusamy 


Research Article (12 pages), Article ID 5401372, Volume 2023 (2023)

### **Bayesian Network Structure Learning by Ensemble Learning and Frequent Item Mining**

Guoxin Cao and Haomin Zhang 


Research Article (15 pages), Article ID 3119316, Volume 2023 (2023)

### **A New Collaborative Filtering Algorithm Integrating Time and Multisimilarity**

Qin Liu 

Research Article (7 pages), Article ID 2340671, Volume 2022 (2022)

### **An IGWOCNN Deep Method for Medical Education Quality Estimating**

Lin Shi and Lei Zheng 

Research Article (5 pages), Article ID 9037726, Volume 2022 (2022)

## Research Article

# A Deep Learning Approach to Detect Microsleep Using Various Forms of EEG Signal

**S. K. B. Sangeetha,<sup>1</sup> Sandeep Kumar Mathivanan,<sup>2</sup> V. Muthukumaran ,<sup>3</sup> N. Pughazendi,<sup>4</sup> Prabhujayopal,<sup>2</sup> and Md Salah Uddin **<sup>5</sup>

<sup>1</sup>Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamilnadu, India

<sup>2</sup>School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

<sup>3</sup>Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamilnadu, India

<sup>4</sup>Department of Computer Science and Engineering, Panimalar Engineering College, Poonamallee, Chennai 600123, Tamilnadu, India

<sup>5</sup>Department of Multimedia and Creative Technology, Daffodil International University, Dhaka-1207, Bangladesh

Correspondence should be addressed to Md Salah Uddin; jahid.mct@diu.edu.bd

Received 20 August 2022; Revised 6 November 2022; Accepted 24 November 2022; Published 16 February 2023

Academic Editor: Abolfazl Gharaei

Copyright © 2023 S. K. B. Sangeetha 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.

Electroencephalography (EEG) is a reliable method for identifying the onset of sleepiness behind the wheel. Using EEG technology for driving fatigue detection still presents challenges in extracting informative elements from noisy EEG signals. Due to their extensive computational parallelism, which is similar to how the brain processes information, neural networks have been explored as potential solutions for extracting relevant information from EEG data. The existing machine learning frameworks suffer from high computing costs and slow convergence, both of which contribute to low classification accuracy and efficiency due to the large number of hyper parameters that need to be improved. It is necessary to automate this micronap detection process before it can be used in real-time scenarios. To distinguish between micronap and non-micronap states, a deep neural network (DNN) framework is developed in this research using different EEG representations as input. Additional EEG representations utilized in this investigation include cleaned EEG as a time series, log-power spectrum, 2D-spatial map of log-power spectrum, and raw EEG. Finally, traditional machine learning algorithms are evaluated for their effectiveness in detecting micronaps from these EEG inputs. The findings suggest that micronap detection can be greatly improved by combining cleaned EEG with DNN.

## 1. Introduction

In many professions and everyday activities, the ability to sustain concentration is essential. When a process or system is semiautomated, the importance of this element increases. Automation has permeated all professions in today's world. Even though maximum blood alcohol levels and the enforcement of speed limits have recently decreased, traffic accidents continue to be a major source of fatalities and injuries [1]. Among the many circumstances where it is essential for the individual affected and those closest to them to concentrate, the brief amount of time during which the

drivers involved in these incidents lost awareness or concentration is frequently unknown to them at the time of the accident. The fatigue masked by the accident disguises their drowsiness. A micronap is an unintentional, brief spell of unconsciousness that can last up to 30 seconds. Micronaps can be distinguished from tiredness by their complete lack of visuomotor responsiveness and partial eye closure. Micronaps can occur without warning. A person might not be aware of having experienced these light sleep stages [2].

Micronaps are more likely due to physical exhaustion, mental exhaustion, circadian rhythm problems, and boredom from repetitive work. However, even in subjects who



are not sleep deprived and perform repetitive activity without warning signs such as fatigue, these types of lapses are possible [3]. The drivers involved in these collisions were unaware of the short period of time they lost consciousness or concentration or how the postcrash adrenaline masked their fatigue. Major safety issues are raised by this, especially, for those who engage in high-risk professions including driving, flying, navigating, traveling by boat, and process control, all of which call for consistent, uninterrupted visuomotor function. Therefore, accurate detection of approaching micronaps has the potential to prevent fatal mishaps and save lives. Micronaps are challenging because they occur suddenly and unintentionally [4].

Machine learning has been used in several studies to identify microsurgers in EEG data. These experiments were primarily concerned with developing a model for identifying short naps and testing different feature extraction, selection, and reduction strategies. It is challenging to maintain relationships when using such algorithms with electroencephalogram (EEG), multivariate signal, and dynamic time series signal as input. Additionally, since the features provided to machine learning algorithms are hand-picked by algorithm or system designers, they have difficulty understanding the dynamics of microvoltages due to selectivity invariance. Recent innovations such as deep learning (DL) algorithms that can extract, analyze, and capture information from unstructured data offer the full answer. With DL, greater focus may be put on model creation to enhance the effectiveness of micronap detection [5, 6].

The performance of the model is significantly influenced by the feature selection. Features that are derived from data using a specific handcrafted process based on expert knowledge are referred to as handcrafted features. This might limit how the characteristics and signals the data provide can be portrayed. In contrast, learners are built from a dataset using a training approach to achieve a certain goal (e.g., gender recognition). The objective of this article was to create a dependable technique for detecting brain activity micropulses that could be used as the foundation for a real-time system that would warn the subject of his condition and avert a deadly accident.

The main contribution of this article is as follows:

- (1) To design deep neural networks (DNN) to classify the micronap and nonmicronap classes
- (2) To use various EEG representations as an input form to the designed DNN and analyze its performance

The contribution of real-time data to the identification of micronaps is another important finding in this article. After this thorough introduction, Section 2 provides a survey of the literature. The data, an illustration of a micronap detection system, the validation processes, and the performance metrics, are all provided in Section 3. The various EEG input data formats are discussed in Section 4 along with a comparison of the outcomes, followed by the conclusion in Section 5.

*1.1. Related Study.* We will review key concepts in EEG-based sleep, exhaustion, and sleepiness research. Forms of lapsing and an overview of several micronap detection

studies are also reviewed. Understanding the nature and properties of the EEG information associated with brain activity in various contexts is required in order to accomplish this [7]. Additionally, this is required to back up the choice to employ EEG to detect micronaps. The human brain is revealed by an electroencephalogram (EEG), a measurement of potentials. It is a simple test that shows how the brain changes over time [8]. EEG is frequently used by medical professionals and researchers to study brain activity and identify neurological problems. Modern research in many fields depends on EEG. A patient's brain death, the severity of a stroke or head injury, epileptic activity, sleep issues, and many other things can all be determined with this technique in medicine. It is useful in other studies investigating various cognitive processes such as memory or attention as well as in linguistic and clinical studies such as aphasia [9, 10].

Experts have to spend a lot of time visually checking the stages of sleep in order to assign a score. In order to diagnose and treat sleep-related diseases, automatic classification of sleep stages is preferred. A response lapse is a period of time during which a person is unable to respond to a continuous task. Depending on the underlying cognitive systems, there are several types of blackouts. Some typos cause delays in rapid response, while others lead to incorrect responses. Some errors may result in complete sensory-motor collapse. A brief interruption that causes a delay or lack of reaction in the main work without making the person unconscious is called an attention lapse [11]. In certain situations, a person could unconsciously engage in a secondary task such as walking, looking, or driving. The unwelcome loss of consciousness related to sleep happens during micronaps. The individual enters the stage of light sleep during this brief (up to 30 s) time. The behavioral signs of micronaps include head nodding, lack of facial expression, and partial eye closure. Sleep is defined as a period of inactivity lasting more than 30 seconds [12, 13].

The current benchmark for micronap state identification, which reflects the top outcomes for micronap state detection on unpruned data, was attained using a range of classifiers and an ensemble of features. Despite numerous studies on the subject, there is still no technique that performs well enough for use in practical situations [14]. The accuracy of EEG data was tested using a variety of convolutional neural network (CNN) topologies. To create learning representations that are efficient and resistant to intrinsic EEG noise as well as inter- and intrasubject variation, deep recurrent convolutional neural networks are used [15]. To avoid the need for handcrafted features, deep belief networks, an unsupervised feature learning architecture, were applied to the sleep data [16]. Compared with handcrafted features, the deep belief network (DBN) technique improved the sleep classification accuracy. The task identifies seizures using a range of formats and machine learning algorithms. Numerous applications based on EEG have shown the efficacy of CNNs and recurrent neural networks (RNNs), including epilepsy, seizures, and different stages of sleep. EEG has been successfully compressed using a convolutional auto encoder (CAE) with the best subspace

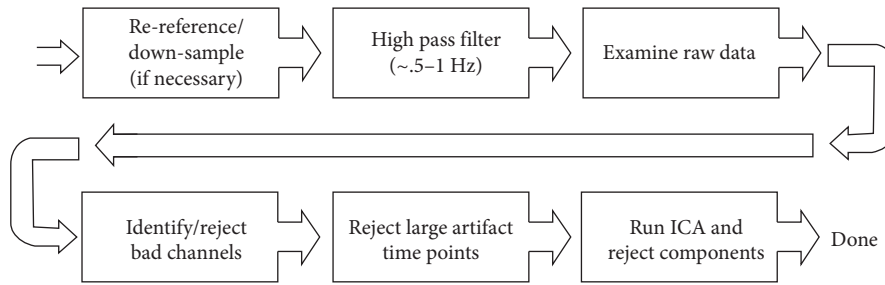


FIGURE 1: Preprocessing EEG signal.

reduction [17]. DNNs' architecture permits the addition of signal processing techniques to conventional statistical data. DNNs are very extendable and flexible. From single-layer shallow patterns to numerous successive convolutional layers, they can have different numbers of convolutional layers [18, 19]. This article used deep learning to analyze scalp EEG signals to identify the phases and onsets of micronaps.

## 2. System Methodology

**2.1. Dataset Description.** This is the first behavioral and EEG dataset ever collected using deep learning. The main motivation for selecting the deep dataset (<https://www.kaggle.com/>) for this work's initial experimentation is that it has been widely used in previous research projects that have shown a number of discoveries and characteristics of micronaps [20]. As a result, a basic standard was established, especially for the detection of micronaps. Ten healthy individuals between the ages of 20 and 45 were examined. None of the participants had any neurological or sleep issues in the past or present, and all of them had enhanced visual acuity. In addition, all subjects reported a sound sleep the previous night (mean = 7.35 h, with a minimum of 6 h) and were, therefore, not considered sleep-deprived. Sixteen electrodes (10–20 international standards) were placed on the subjects' scalp at the following locations: F7, F3, F4, F9, T7, C3, CZ, C4, T8, P7, PZ, P4, P8, and O1 during the EEG capture task at a sampling frequency of 240 Hz [21].

EEG and face images were captured at a frame rate of 15 frames per second (fps). Based on visual indications including head nodding, head jerking, and protracted eyelid closure, the facial video helped identify gaps brought on by micronaps. The gold standard has been developed through the use of visual signals and performance tracking. The validation of training and test data was made easier by this designation [22, 23]. All subjects took part in the experiment, which took place in a 4 m<sup>2</sup> space equipped with a PC screen and all necessary devices for EEG data collection. The distance of the object from the screen was between 75 and 115 cm. All subjects were fitted with an EEG headset and pretests were performed to ensure the integrity of the entire recording pipeline. An experimenter oversaw the session from a nearby room with a one-way glass window. Subjects were instructed to refrain from excessive head movements, eye blinks, and facial muscle contractions before the study

began. Subjects were instructed to silently count the number of target flashes to help maintain focus.

**2.2. EEG Preprocessing.** Each subject's scalp's recorded EEG signals completed the preprocessing illustrated in Figure 1 as follows:

- (i) The reference was reset to a common average reference in order to improve the signal-to-noise ratio. The EEG data were then band-pass filtered from 0.5 to 1 Hz using a high pass filter.
- (ii) The independent component analysis (ICA) components were projected to the calibration data space using its own covariance matrix.
- (iii) To take into account the EEG's nonstationarity, it is divided into 2 min sections with 50% overlap.
- (iv) Each epoch's calibration data were discovered and utilized to clean the same epoch. The original EEG data were then cleaned by concatenating the epochs. To prevent discontinuity, the overlapping portions of succeeding epochs were averaged.

A behavioral expert divided episodes of blackouts into one of four groups as follows: Type 0 is with drooping lids and periods of noticeably increased tracking imprecision but a response rate above zero (complete eye closure). Type 1: episodes persist longer than 500 ms, have drooping eyes, and have a flat or jumbled response (complete or partial slow eye closure). Micronaps are defined as episodes lasting less than 30 seconds, while naps are classified as episodes lasting over 15 seconds. Type 2 is strange incidents and forced eye closure. Type 3 is lack of droopy eyes and flat or senseless shouts. Micronap events were transformed into labels at 250 Hz (gold standard). Only Type 1 was employed to learn, recognize, and detect the onsets of micronaps in this investigation, and only the responsive states were used as the gold standard [24–26].

**2.3. Deep Neural Network Micronap Detection Framework.** Figure 2 depicts the overall micronap detecting mechanism. Figure 3 depicts the proposed DNN framework. As previously noted, the EEG data obtained from the scalp are preprocessed to reduce artifacts and serves as an input to visualize and extract suitable features on its own. The first layer specifies the input dimensions. ReLU and max-pooling layers are inserted between a series of convolutional layers

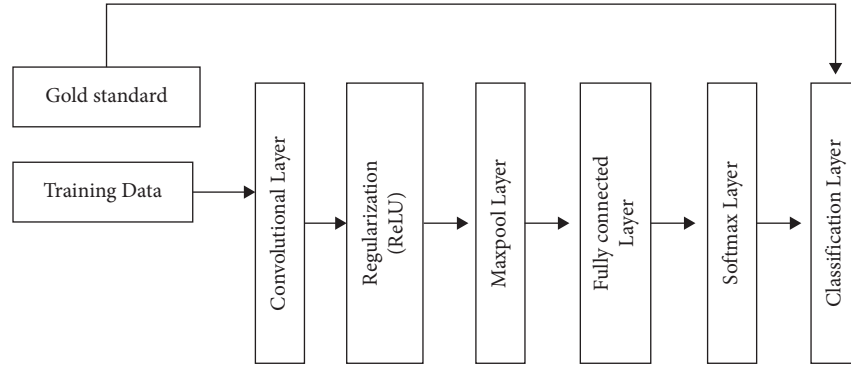


FIGURE 2: Generic DNN framework.

that make up the intermediate layers. To produce a single output, the pooling layer subsamples tiny rectangular blocks from the convolutional layer. The softmax layer and fully linked layers are used in the final layer to categorize patterns.

**2.4. Micronap State Detection.** Both intrasubject (during and between sessions) and intersubject variance has an impact on the number of occurrences and their corresponding lengths. Usually, discrete labels are applied while classifying things. This work's gold standard was discretized or sampled at 2.5 Hz. We used a sliding window with duration of  $W$  to segment the entire 30 minutes of EEG data. Figures 4 and 5 show the EEG during sleep and the typical EEG, respectively. Figure 6 uses the EEG segments as inputs to show the micronap states at  $r$ . This procedure is used every 0.5 s up until all states of the gold standard have been recognised. The classifier can only decide whether the behavioral state that correlates with it is a micronap or a responsive state (i.e.,  $r = 0$ ) at the end of the EEG window. For  $r > 0$ , where  $r$  is the distance between the state under consideration and the window's end, the same conclusions are true.

**2.5. Micronap Onset Detection.** It is impossible to pinpoint and mark the precise onsets of micronaps due to the existence of ambiguous labels. The first instance of a micronap state's definite occurrence after the responsive state is considered to be the official beginning of a micronap in this article. All responsive states and micronap onsets were the industry standard for onset detection. EEG segments of  $W$  duration in the time domain or frequency domain, with or without spatial information, served as the inputs for the identification of the micronap states at  $r$ .

**2.6. Classification.** The classification process used a 0.5-second temporal resolution. Each individual experienced micronaps at a distinct frequency and length, which produced quite diverse imbalance ratios. Table 1 displays the frequency and length of micronaps by subject as well as imbalance ratios of states and onsets. According to Table 1, with subject 9 having the largest imbalance ratio, all subjects, with the exception of subjects 5 and 6, show considerable imbalance ratios between micronap states and responsive

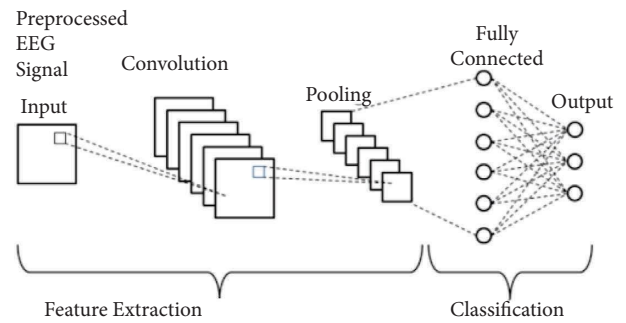


FIGURE 3: Proposed DNN layers.

states. When it comes to the onset of micronaps, the imbalance ratio gets worse. The imbalance ratio between the two classes significantly worsens when it comes to onsets. Table 1: Number of states =  $4 * \text{Time between events}$  (temporal resolution of 0.5 s).

**2.7. Validation and Performance Measures.** To evaluate the model's real performance in classifying the micronap and responsive classes, data from a test subject must be completely hidden from the validation and training processes. In other words, one makes an estimation of the model's behavior with an entirely hypothetical topic. For this purpose, the planning and experimentation phases of this article employed the following strategy:

- (i) keep one subject out of a total of ten for independent testing.
- (ii) train the deep learning model with the remaining 9 subjects.
- (iii) utilize a leave-one-subject-out cross-validation technique (LOSO-CV). The deep learning model's regularization as well as its hyperparameters (number of filters, size of filters, number of layers, types of layers, and number of layers) are altered in order to obtain the optimal performance.
- (iv) Each layer's hyperparameters were successively swept through a set of values. Automatically, the best area under curve (AUC) is used to determine the ideal values.



FIGURE 4: Normal EEG.

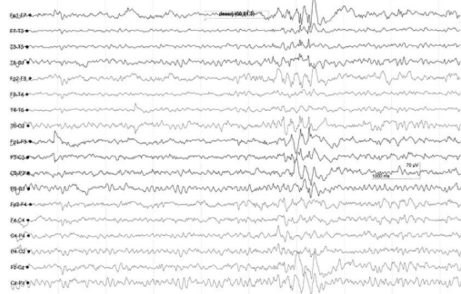


FIGURE 5: Sleep state-EEG.

- (v) Up until all 10 test individuals have been employed, we repeat processes 1 through 4.
- (vi) calculate the performance measurement averages.

The measures of evaluation serve a dual purpose in designing classifiers and evaluating their performance. The concepts present in a confusion matrix for a binary (two-class) classification problem can be used to comprehend the majority of threshold metrics. Three crucial parameters, sensitivity (Sn), specificity (Sp), and precision, can be estimated using the matrix (P). In this article, two more parameters are utilized to assess the model’s test performance in addition to the geometric mean, phi, and others. The area under the receiver operating characteristic (ROC) curve and the area under the precision-recall (PR) curve are two more curve-based measures with measurements that are independent of threshold. The performance of the models is compared using the paired nonparametric Mann–Whitney *U* test. Every input format that the EEG and DL models generated was compared. In order to identify the best input modalities and network architectures, the performances of various combinations of input modalities and network designs on a certain window size and dataset were examined.

### 3. Experimentation Results

A comparison of performance metrics between various input combinations and DNN was also conducted. The outcomes of detecting the micronap state utilizing the following EEG inputs are shown in this section.

**3.1. Cleaned EEG as a Time Series.** Different EEG window lengths are utilized to find the best EEG window length and matching model, including 0.5 s, 1.5 s, 2.5 s, 3.5 s, 4.5 s, and

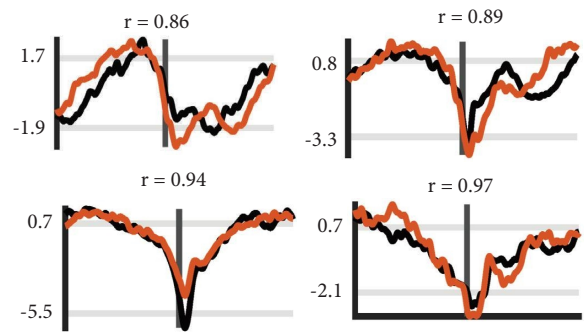


FIGURE 6: Tracking vs. gold standard performance (measures: time vs. amplitude).

5 s. The cross-validation ROC was used to determine which model was best for each subject. Table 2 indicates that the cross-validation ROC was the greatest for window durations of 3.5 s and 4.5 s. The 3.5 s window was found to be the ideal window length because the 4.5 s window requires more processing for the same performance.

**3.2. Log-Power Spectrum.** For several subjects, the Maxpool layer performed best in cross validation when the log-spectrum was used as the input, and its size and stride were both 11. This is what would have happened if the layer had not been there. Similar to this, the dropout layer had an effect on the performance of certain subjects. For state detection, the average values for Sn, Sp, P, Phi, ROC, and PR were 0.62, 0.70, 0.15, 0.22, and 0.76, respectively. Figure 7 displays a comparison of performance metrics when the log-power spectrum is the input.

**3.3. 2D-Spatial Map of Log-Power Spectrum and Raw EEG.** The input dimension was 56×56, where the number four represented a mixture of the frequency bands for delta (δ), theta (θ), alpha (α), and beta (β). There were 4 possible band combinations that could be entered as input. By feeding the individual bands to the DNN, the data contained in the individual bands to help with micronap state detection are also evaluated. The dual combination of bands, which consisted of the bands 6 and 8, 8 and α, and 6 and α, was also investigated in addition to the examination of each individual band. Table 3 shows the average performance metrics.

Because it would include all artifacts, including EOG, raw-EEG was chosen as the time series input to the DNN. Figure 8 illustrates how well state detection works.

### 4. Machine Learning Approaches

The performance metrics are derived when several types of EEG representations are given as input, trained with SVM, KNN, and LSTM classifiers, and then tested. Table 4 lists the average state detection performance metrics phi, ROC, and PR. There were no appreciable improvements in phi and ROC scores between the classifiers. DNN was used as an end-to-end solution along with EEG, although the increase in phi was just

TABLE 1: Micronaps and responsive.

Subjects	No. of events (ms)	No. of states (ms)	Responsive	State imbalance	Event imbalance
Person 1	23	456	34686	1:28	1:580
Person 2	34	5435	12723	1:44	1:620
Person 3	2	455	23728	1:30	1:2778
Person 4	44	344	24264	1:58	1:532
Person 5	16	234	8467	1:102	1:1566
Person 6	42	2444	1276	1:5	1:484
Person 7	37	24	16767	1.62	1:687
Person 8	24	4563	27234	1.248	1:544
Person 9	5	675	14676	1.32	1:3464
Person 10	28	25	33479	1.50	1:764

TABLE 2: Performance metrics comparison.

EEG window length	Sn	Sp	P	Phi	ROC	PR
0.5	0.56	0.65	0.04	0.12	0.6	0.7
1.5	0.59	0.67	0.05	0.17	0.62	0.72
2.5	0.64	0.71	0.13	0.24	0.66	0.75
3.5	0.66	0.73	0.15	0.27	0.69	0.76
4.5	0.65	0.73	0.18	0.28	0.68	0.77
5	0.53	0.73	0.23	0.26	0.6	0.76
Average	0.61	0.70	0.13	0.22	0.64	0.74

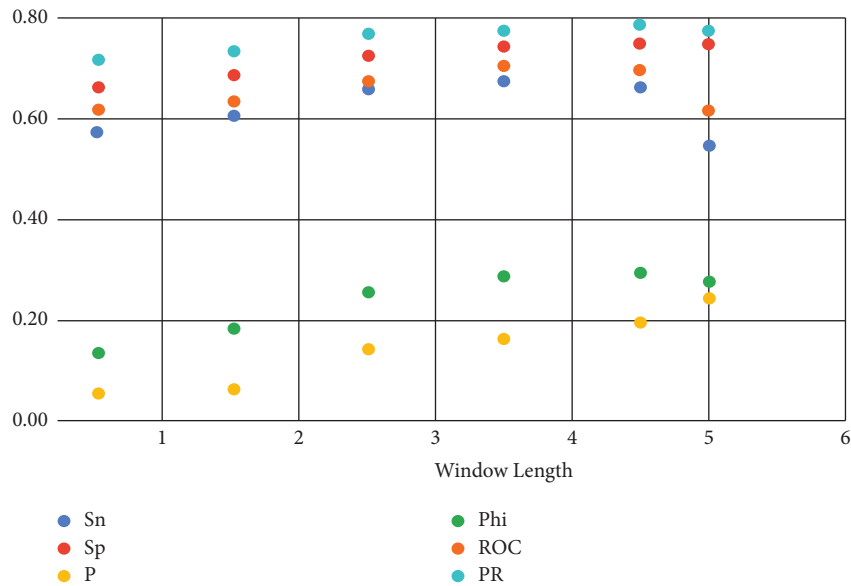


FIGURE 7: Comparison metrics analysis.

marginal (from 0.59 to 0.62). However, DNN has a superior performance as shown in Figure 9 in terms of sensitivity (Sn) (0.84), compared to SVM (0.78) and LSTM (0.72). When it comes to state detection, the DNN model surpassed the LSTM model (0.52) in terms of PR (0.68). The SVM has outperformed the LSTM in terms of PR (0.62). With EEG as the input, the average performance metric for state detection had the highest values in terms of phi, ROC, and PR. EEG waves may, therefore, be a valuable source of data for identifying onset or status.

In comparison to previous EEG input transformations, state detection performance was best when DNN and cleaned EEG were combined. When identifying micronap states, the 6 and 8 band combination held important information. A DNN's filter structure is essential for extracting information from the data. What the DNN will represent in the data throughout its learning phase depends on the filter size. The DNN was employed in this work as a series network, which imposed limitations on the design. Furthermore, regardless of whether the input was a representation

TABLE 3: Band analysis.

Bands	Sn	Sp	P	Phi	ROC	PR
6, 8, $\alpha$	0.5	0.76	0.17	0.2	0.59	0.76
6	0.39	0.68	0.1	0.08	0.46	0.66
8	0.45	0.75	0.15	0.19	0.56	0.73
$\alpha$	0.39	0.62	0.01	0.01	0.48	0.58
$\beta$	0.23	0.68	0.01	0.03	0.37	0.49
6, 8	0.66	0.74	0.26	0.29	0.69	0.8
8, $\alpha$	0.56	0.75	0.15	0.21	0.64	0.76
6, $\alpha$	0.41	0.69	0.06	0.08	0.51	0.65
6, $\beta$	0.56	0.75	0.15	0.21	0.64	0.76
8, $\beta$	0.39	0.62	0.01	0.01	0.48	0.58

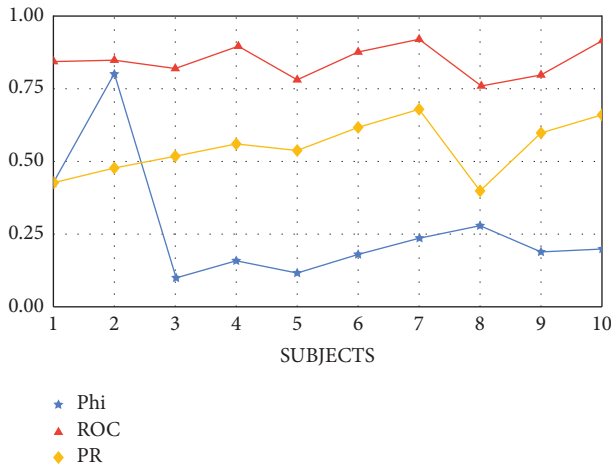


FIGURE 8: Subject-wise state detection.

TABLE 4: DNN vs. traditional approaches.

	Phi	ROC	PR
SVM	0.04	0.51	0.62
KNN	0.11	0.34	0.50
LSTM	0.18	0.48	0.52
DNN	0.24	0.64	0.68

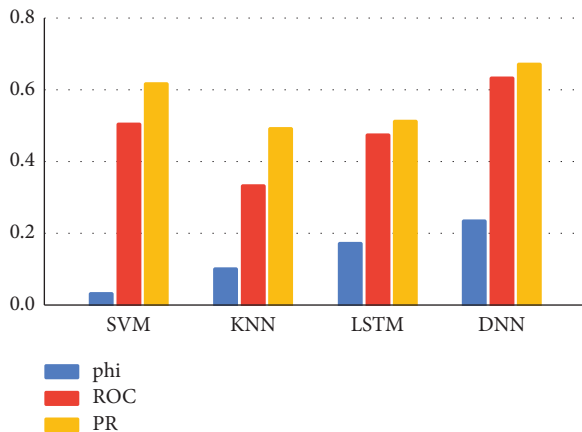


FIGURE 9: DNN vs. traditional approaches.

in the time domain or the frequency domain, the DNN always interpreted it as a 2D input. To comprehend characteristics connected to micronaps and important aspects in the decision-making process, layer-wise feature analysis must be conducted. However, a parallel DNN framework can be created to enable the simultaneous application of several filter sizes to the same input (EEG alone). CNN will be able to learn about traits from a variety of perspectives as a result.

### 5. Conclusion

This article summarized the various modalities of EEG inputs and demonstrated the performance of a DNN-based micronap categorization system. The many representations of EEG employed as an input to the DNN model include cleaned EEG time series, log-power spectrum, 2D-map of log-power spectrum, and raw EEG. The hyperparameters and DNN model architectures were tweaked. The use of a cost-based error function was used to address the imbalance in the courses during training. LOSO-CV was used to test each model in order to estimate its generalized performance. Sensitivity, specificity, accuracy, phi, ROC, and PR were the performance metrics used to assess the model. The best average performance for state detection ( $r=0$ ) was a phi of 0.24, ROC of 0.64, and PR of 0.68, attained using DNN. Despite having a reasonably good sensitivity and specificity for onset detection, the average performance measure phi was extremely poor. This was brought on by a significant number of false detections and a large ratio of imbalance between the onsets of micronaps and responsive states. The DNN learned the changes in activity between the prefrontal, central, and occipital parts of the brain, as well as the delta, theta, and alpha bands chronologically. To determine which raw-EEG signal contained more specific information that the DNN could extract for identifying micronaps, experiments were also conducted with raw-EEG signals without any pre-processing. The purpose of the micronap state detection system is to continuously monitor the level of attentiveness and forecast impending micronaps before they happen. If a micronap state prediction is incorrect, an attempt will be made to identify the subsequent micronap state. The micronap onset detection system, on the contrary, works nonstop to forecast only the beginning of an impending micronap. However, the entire micronap event is lost if an onset detection is missed which can be taken for future work.

### Data Availability

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

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] K. R. Parkes, "Age and work environment characteristics in relation to sleep: additive, interactive and curvilinear effects," *Applied Ergonomics*, vol. 54, pp. 41–50, 2016.
- [2] A. R. Hassan and M. I. H. Bhuiyan, "Automated identification of sleep states from EEG signals by means of ensemble empirical mode decomposition and random under sampling boosting," *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 201–210, 2017.
- [3] R. Sharma, R. B. Pachori, and A. Upadhyay, "Automatic sleep stages classification based on iterative filtering of electroencephalogram signals," *Neural Computing & Applications*, vol. 28, no. 10, pp. 2959–2978, 2017.
- [4] Y. Chu, X. Zhao, J. Han, Y. Zhao, and J. Yao, "SSVEP-based brain-computer interface controlled functional electrical stimulation system for upper extremity rehabilitation," in *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, no. 7, pp. 947–956, Bali, Indonesia, December 2014.
- [5] S. K. B. Sangeetha, R. Dhaya, D. T. Shah, R. Dharanidharan, and K. P. S. Reddy, "An empirical analysis of machine learning frameworks for digital pathology in medical science," *Journal of Physics: Conference Series*, vol. 1767, Article ID 012031, February 2021.
- [6] G. C. Gutiérrez-Tobal, D. Álvarez, J. V. Marcos, F. Del Campo, and R. Hornero, "Pattern recognition in airflow recordings to assist in the sleep apnoea-hypopnoea syndrome diagnosis," *Medical, & Biological Engineering & Computing*, vol. 51, no. 12, pp. 1367–1380, 2013.
- [7] S. K. B. Sangeetha and R. Dhaya, "Deep learning era for future 6G wireless communications—theory, applications, and challenges," *Artificial Intelligent Techniques for Wireless Communication and Networking*, pp. 105–119, 2022.
- [8] M. Y. Chen, H. S. Chiang, A. K. Sangaiah, and T. C. Hsieh, "Recurrent neural network with attention mechanism for language model," *Neural Computing & Applications*, vol. 32, no. 12, pp. 7915–7923, 2020.
- [9] D. Kanthavel, S. K. B. Sangeetha, and K. P. Keerthana, "An empirical study of vehicle to infrastructure communications—an intense learning of smart infrastructure for safety and mobility," *International Journal of Intelligent Networks*, vol. 2, pp. 77–82, 2021.
- [10] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver drowsiness detection model using convolutional neural networks techniques for android application," in *Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 237–242, IEEE, Doha, Qatar, February 2020.
- [11] R. Malekian, A. F. Kavishe, B. T. Maharaj, P. K. Gupta, G. Singh, and H. Waschefort, "Smart vehicle navigation system using hidden Markov model and RFID technology," *Wireless Personal Communications*, vol. 90, no. 4, pp. 1717–1742, 2016.
- [12] D. Mollicone, K. Kan, C. Mott et al., "Predicting performance and safety based on driver fatigue," *Accident Analysis & Prevention*, vol. 126, pp. 142–145, 2019.
- [13] D. Singh, B. Pati, C. R. Panigrahi, and S. Swagatika, "Security issues in IoT and their countermeasures in smart city applications," in *Advanced Computing and Intelligent Engineering*, pp. 301–313, Springer, Singapore, 2020.
- [14] M. Miranda, A. Villanueva, M. J. Buo, R. Merabite, S. P. Perez, and J. M. Rodriguez, "Portable prevention and monitoring of driver's drowsiness focuses to eyelid movement using internet of things," in *Proceedings of the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pp. 1–5, IEEE, Baguio City, Philippines, November 2018.
- [15] S. Abraham, T. Luciya Joji, and D. Yuvaraj, "Enhancing vehicle safety with drowsiness detection and collision avoidance," *International Journal of Pure and Applied Mathematics*, vol. 120, no. 6, pp. 2295–2310, 2018.
- [16] O. I. Khalaf, K. A. Ogudo, and S. K. B. Sangeetha, "Design of graph-based layered learning-driven model for anomaly detection in distributed cloud IoT network," *Mobile Information Systems*, vol. 2022, Article ID 6750757, 9 pages, 2022.
- [17] S. K. B. Sangeetha, M. S. Kumar, H. Rajadurai, H. Rajadurai, V. Maheshwari, and G. T. Dalu, "An empirical analysis of an optimized pretrained deep learning model for COVID-19 diagnosis," *Computational and Mathematical Methods in Medicine*, vol. 2022, Article ID 9771212, 10 pages, 2022.
- [18] H. W. Loh, C. P. Ooi, S. G. Dhok, M. Sharma, A. A. Bhurane, and U. R. Acharya, "Automated detection of cyclic alternating pattern and classification of sleep stages using deep neural network," *Applied Intelligence*, vol. 52, no. 3, pp. 2903–2917, 2022.
- [19] Y. Wang, Z. Xiao, S. Fang, W. Li, J. Wang, and X. Zhao, "BI-Directional long short-term memory for automatic detection of sleep apnea events based on single channel EEG signal," *Computers in Biology and Medicine*, vol. 142, Article ID 105211, 2022.
- [20] M. Malekzadeh, P. Hajibabae, M. Heidari, and B. Berlin, "Review of deep learning methods for automated sleep staging," in *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0080–0086, IEEE, Las Vegas, NV, USA, January 2022.
- [21] J. Cui, Z. Lan, Y. Liu et al., "A compact and interpretable convolutional neural network for cross-subject driver drowsiness detection from single-channel EEG," *Methods*, vol. 202, pp. 173–184, 2022.
- [22] L. Kaulen, J. T. C. Schwabedal, J. Schneider, P. Ritter, and S. Bialonski, "Advanced sleep spindle identification with neural networks," *Scientific Reports*, vol. 12, no. 1, Article ID 7686, 2022.
- [23] R. R. Immanuel and S. K. B. Sangeetha, "Recognition of emotion with deep learning using EEG signals - the next big wave for stress management in this covid-19 outbreak," *Periodico di Mineralogia*, vol. 91, no. 5, 2022.
- [24] R. L. C. Souza, A. Ghasemi, A. Saif, and A. Gharaei, "Robust job-shop scheduling under deterministic and stochastic unavailability constraints due to preventive and corrective maintenance," *Computers & Industrial Engineering*, vol. 168, Article ID 108130, 2022.
- [25] A. Gharaei, A. Amjadian, A. Amjadian et al., "An integrated lot-sizing policy for the inventory management of constrained multi-level supply chains: null-space method," *International Journal of Systems Science: Operations & Logistics*, pp. 1–14, 2022.
- [26] A. Amjadian and A. Gharaei, "An integrated reliable five-level closed-loop supply chain with multi-stage products under quality control and green policies: generalised outer approximation with exact penalty," *International Journal of Systems Science: Operations & Logistics*, vol. 9, no. 3, pp. 429–449, 2022.

## Research Article

# Prediction of Flank Wear during Turning of EN8 Steel with Cutting Force Signals Using a Deep Learning Approach

S. K. Thangarasu <sup>1</sup>, T. Mohanraj <sup>2</sup>, K. Devendran <sup>3</sup>, M. Rajalakshmi <sup>4</sup>,  
Subrata Chowdhury <sup>5</sup> and Saravanakumar Gurusamy <sup>6</sup>

<sup>1</sup>Department of Mechatronics Engineering, Kongu Engineering College, Erode, Tamilnadu, India

<sup>2</sup>Department of Mechanical Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

<sup>3</sup>Department of Computer Science Engineering, Kongu Engineering College, Erode, Tamilnadu, India

<sup>4</sup>Department of Biomedical Engineering, Sethu Institute of Technology, Virudhunagar, Tamilnadu, India

<sup>5</sup>Department of Computer Science and Engineering, Sreenivasa Institute of Technology and Management Studies, Chittoor, Andhra Pradesh, India

<sup>6</sup>Department of Electrical and Electronics Technology, FDRE Technical and Vocational Training Institute, Addis Ababa, Ethiopia

Correspondence should be addressed to S. K. Thangarasu; [skthangarasu@gmail.com](mailto:skthangarasu@gmail.com) and Saravanakumar Gurusamy; [saravanakumar.gurusamy@etu.edu.et](mailto:saravanakumar.gurusamy@etu.edu.et)

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

Academic Editor: Chaoqun Duan

Copyright © 2023 S. K. Thangarasu 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.

Currently, manufacturing industries focus on intelligent manufacturing. Prediction and monitoring of tool wear are essential in any material removal process, and implementation of a tool condition monitoring system (TCMS) is necessary. This work presents the flank wear prediction during the hard turning of EN8 steel using the deep learning (DL) algorithm. The turning operation is conducted with three levels of selected parameters. CNMG 120408 grade, TiN-coated cemented carbide tool is used for turning. Cutting force and flank wear are assessed under dry-cutting conditions. DL algorithms such as adaptive neuro-fuzzy inference system (ANFIS) and convolutional auto encoder (CAE) are used to predict the flank wear of the single-point cutting tool. The DL model is developed with turning parameters and cutting force to predict flank wear. The different ANFIS and CAE models are employed to develop the prediction model. Grid-based ANFIS structure with Gauss membership function performed better than ANFIS models. The ANFIS model's average testing error of 0.0074011 mm and prediction accuracy of 99.81% are achieved.

## 1. Introduction

The required component can be manufactured through conventional and unconventional machining processes [1, 2]. Automated industries focus on conventional machining processes such as turning, milling, and grinding. Tool wear monitoring is essential to enhance product quality during the machining process. To monitor the machining process with high precision, the tool condition monitoring system (TCMS) should be an integral part of the automated manufacturing systems. TCMS can enhance production and enrich the performance of the machining system by improving the tool life, diminishing downtime and scrappage,

and avoiding damage through continuous monitoring and predictive analysis [3]. Tool condition has been monitored by measuring the various sensor signals like cutting force, vibration [4], and sound signals [5]. Cutting force (CF) can be measured with piezoelectric or strain gauge-based sensors [6, 7]. The I-Kaz method analyzed a low-cost strain gauge sensor that measured the CF signals. The investigational outcomes were compared with the adaptive neuro-fuzzy inference system (ANFIS) based prediction model, which produced a maximum of 5.08% error [8]. The ANFIS model with “gbellmf” was used to forecast the surface roughness in milling operation by considering the spindle speed, feed, and depth of cut as input parameters [9].



Tool wear is more sensitive to CF signals, and CF signals are used to indicate tool conditions. Also, CF signals are gradually increased with a rise in tool wear. CF is measured directly through a dynamometer, and the tool wear is measured indirectly by analyzing the CF signals [10]. Prediction of tool wear and optimization of drilling parameters were carried out by ANFIS and genetic algorithm (GA), respectively. A smaller feed was suggested to minimize the thrust force and torque [11]. Response surface methodology (RSM) was employed to investigate process parameters' effects on CF, surface roughness (Ra), and cutting temperature. Out of two forecasting models, such as an artificial neural network (ANN) and ANFIS, the ANFIS model predicts the response more accurately. The optimum process parameters were selected using the RSM [12].

Estimation of tool life was carried out by the ANFIS model using the surface roughness and CF. The accuracy of the proposed model was 92.62% [13]. ANFIS model was used to estimate the material removal rate (MRR) and surface roughness (Ra) while turning stainless steel 202. The depth of cut and spindle speed was found as the most influencing parameter for MRR and Ra, respectively [14]. CF was estimated using ANFIS with spindle speed, feed, and depth of cut during the turning process, and the CF was predicted with an average error of 2.59% [15].

The neural network was used to forecast the CF, Ra, and tool wear while turning the CP-Ti grade II workpiece. The proposed model was suitable for predicting the responses within a 5% error [16]. A discrete and continuous monitoring, hybrid policy-based tool replacement mechanism was developed. The hybrid policy was optimized by particle swarm optimization (PSO) algorithm. All policies' results were analyzed, and PSO performed better than the others [17]. Online monitoring of grinding wheel wear was investigated in the grinding of Ti-6Al-4V titanium alloy. The result indicated that the Gaussian membership function-based ANFIS model was more intelligent for monitoring the grinding process [18].

Prediction and optimization of drilling parameters for Ra were investigated while drilling galvanized steel. ANFIS model with the "gbellmf" membership function produced the minimum error. The output of the ANFIS model was given as input to the GA to find the optimum process parameter. The spindle speed and tool angle was Ra's most influencing parameters [19]. Prediction of CF and Ra was carried out in the Al-20Mg<sub>2</sub>Si-2Cu metal matrix composite with ANFIS. Feed, speed, and particle size were the inputs, and Ra was the output. The ANFIS model was suitable for forecasting the Ra with minimum error [20]. An investigation was conducted to compare the wear of ceramic tools during the machining of AISI D2 steel and glass fibre reinforced plastics (GFRP) material. The flank wear was smooth and higher for machining the GFRP, whereas low flank wear was observed while machining the AISI D2 steel [21].

An experimental study was carried out on ASTM A36 mild steel, and the wavelet denoising technique was used to decompose the noise signals, and then, the thresholding was done. The decomposed signals were trained with the ANFIS

to investigate the relationship between the machining parameters and chatter. The depth of cut was the most influencing parameter on the chatter severity [22]. An investigation was carried out to measure the CF, Ra, cutting power, and MRR while turning EN-GJL-250 cast iron using coated and uncoated silicon nitride ceramic tools. The study observed that Ra was affected by feed followed by speed than the depth of cut, and the CF was influenced mainly by the cutting depth followed by feed and cutting speed. ANN model was used for better prediction, and the RSM was used to identify the optimal process parameters and analyze their interactions [23]. RSM was used to perform the regression analysis and optimize the multiresponse problems [24].

Inconel 690 was machined with TiAlN-coated solid carbide insert under minimum quantity lubrication (MQL). The two models, gene expression programming (GEP) and ANN, were used to predict tool wear. The analysis reported that a speed above 100 m/min was not desirable for machining [25]. The surface morphology and chip were analyzed while turning the functionally graded (FG) specimen under nanofluid-assisted minimum quantity lubrication (NFMQL) conditions. They observed that the NFMQL method provided an ecofriendly, green, clean, and sustainable manufacturing process [26]. The tool vibration, Ra, and chip morphology was analyzed during the hard turning of hot work AISI H13 steel under multiwalled carbon nanotubes (MWCNTs) mixed nanofluid with MQL condition. They observed segmented-type serrated saw-toothed chip morphology and increased flank wear and tool vibration responsible for machined surface finish degradation [27]. The experimental investigation was carried out using Taguchi's  $L_{27}$  orthogonal array. Inconel 718 was turned with a carbide-coated insert. The Mamdani inference system with rule reduction and Sugeno subtractive clustering method was used to model the CF, in which the latter produced the minimum error [28].

Acoustic emission (AE) and CF signals were used to predict the built-up edge formation during AISI 304 stainless steel machining. The discrete wavelet and wavelet packet transform filter the noise signals. Finally, the ANFIS model was used to predict the built-up edge height [29]. A new method of diagnosing the tool wear using the stacked sparse autoencoder is proposed. The different signals obtained from various sensors like accelerometer, dynamometer, and AE sensor were used as inputs. The signals were analyzed with wavelet analysis. The accuracy of this model is 99.63%, and it is more suitable for predicting the values very shortly [30].

During the hard turning of AISI D6 steel, the responses like Ra, CF, crater wear length, crater wear width, and flank wear were measured. Different machine learning algorithms, such as polynomial regression, random forests regression, gradient-boosted trees, and adaptive boosting-based regression, were used to predict the machining characteristics. Polynomial and random forests regression performed better than other algorithms [31]. The spindle structure autoencoder wraps the signal without any parameter mining, and the high ratio compression can be transmitted with a low bandwidth cost [32]. A conditional autoencoder was applied to monitor the wind turbine blade condition. The

performance of the conditional autoencoder was compared with the classical and convolutional autoencoder methods. From the analysis, the conditional autoencoder's accuracy was higher than the other two methods [33]. A deep network method is used to forecast the tool wear with the help of motor power using the deep learning neural network theory, which can increase learning speed and enhance the training process [34].

Furthermore, most of the earlier research works exposed that flank wear was forecasted based on the control factors and CF. Many research articles have reported the prediction of flank wear during the turning operation with vibration signals using machine learning and deep learning algorithms. To the authors' best knowledge, the utilization of CF signals and turning parameters with the CAE model are not found in the literature. This work aims to develop the ANFIS and CAE model to forecast the flank wear during the dry turning of EN8 medium carbon steel. The prediction accuracy of ANFIS is compared with the CAE model.

The rest of the paper has been structured as follows: Section 2 deals with the experimental method, design of experiments, and measurement process. The modeling and prediction using ANFIS and CAE models are presented in Sections 3 and 4. The obtained results are discussed in Section 5.

## 2. Materials and Methods

**2.1. Workpiece and Tool.** The workpiece was EN8 medium carbon steel with a Brinell hardness of 255 BHN. EN8 has good tensile strength, is used for manufacturing various machine parts, and is easy to machine [35]. The TiN-coated cemented carbide tool (CNMG120408) was selected as the cutting tool. The tool insert was attached to a tool holder (PCLNR2525M12), which has a  $-6^\circ$  of back rake angle,  $5^\circ$  of clearance angle,  $-6^\circ$  of negative cutting-edge inclination angle,  $95^\circ$  of significant cutting edge, 0.8 mm nose radius, and 3.9 Nm of insert tightening torque.

**2.2. Experimental Setup.** The experiment was performed in a conventional lathe, as shown in Figure 1. The CF during the turning operation was captured using a lathe tool dynamometer. For acquiring the CF, a data acquisition system with the necessary software and hardware [36] was arranged and is shown in Figure 2.

**2.3. Design of Experiments.** With the help of central composite design (CCD), the effectiveness of the turning process was studied. A suitable variant in the CCD is the face-centered CCD, in which  $\alpha = 1$  [37, 38]. The design matrix is presented in Table 1.

**2.4. Flank Wear Measurement.** Flank wear was measured with profile projector PJ-A3000 (Make: Mitutoyo). The XY measurement range and resolution were 100 mm  $\times$  100 mm and 0.001 mm, respectively. The instrument used for measurement is shown in Figure 3.

## 3. Modeling Using an Adaptive Neuro-Fuzzy Inference System

ANFIS is a well-known hybrid neuro-fuzzy inference system to model intricate problems was established [26]. ANFIS is a beneficial neural network (NN) method for affording the result for nonlinear and approximating functions. ANFIS employs different learning approaches to a fuzzy logic (FL) system [39].

**3.1. ANFIS Prediction Model.** The ANFIS catches the merits of both NN and FL principles, and the ANFIS model can efficiently and optimally forecast the response. The parameters related to the membership functions (MF) are changed through the learning process. The parameter computation is simplified by a gradient vector that measures the fuzzy inference system (FIS), which is used to model the given input/output parameters. ANFIS uses a back propagation neural network (BPNN) or a combination of least squares estimation and BPNN (hybrid) for MF parameter estimation [40]. With the help of MATLAB, the ANFIS model was developed, and the graphical user interface was used for training and testing the model.

**3.1.1. ANFIS Methodology.** The following steps are to be considered for developing the ANFIS model to predict flank wear:

*Step 1.* Define the architecture of the ANFIS.

The architecture of ANFIS is displayed in Figure 4. The input parameters, such as speed, feed, and depth of cut, were controlled, and flank wear was recorded. The corresponding cutting force was measured using the strain gauge dynamometer coupled with an amplifier and was given as one of the input parameters for the ANFIS model to forecast the flank wear. The dynamometer data were acquired using a USB 6221M series, NI DAQ Card.

*Step 2.* Set the input and output parameters and MF.

An MF describes how every data in the input is drawn to a membership value between 0 and 1. Several MFs are used for modeling, including trapezoidal, triangular, piecewise linear, and Gaussian. The selection of MF is based on the nature of the problem and their experience. The triangular MF was used for the distribution of the input variable. The triangular membership function (trimf) was just a curtailed triangle. It is a straight-line MF and has the benefit of easiness. The number of data points should be larger than the number of tuning parameters; hence, the three MFs were considered for every input.

**3.1.2. Input and Output Parameters.** There are four input parameters for the ANFIS model: speed, feed, depth of cut, and CF. Cutting speed has a significant influence on tool life. When the cutting speed increases, the cutting temperature increases, increasing the flank wear [26]. The initial MF for speed is presented in Figure 5.

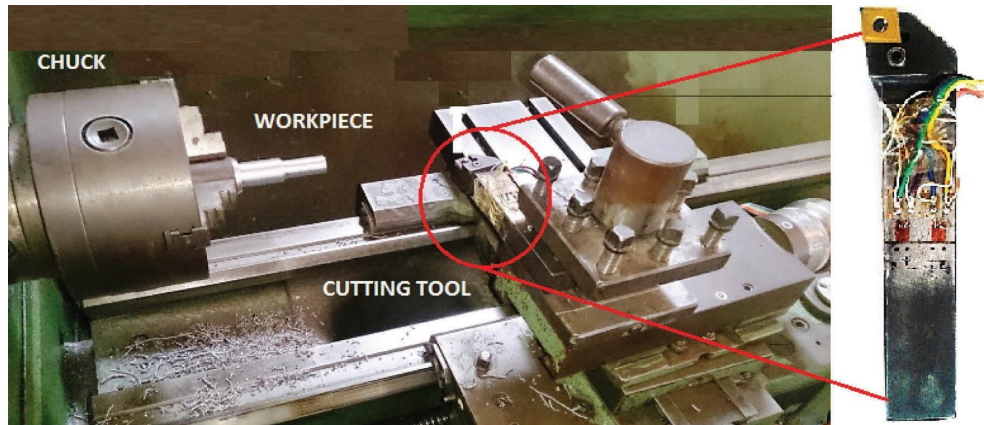


FIGURE 1: Cutting force measurement during the turning process.

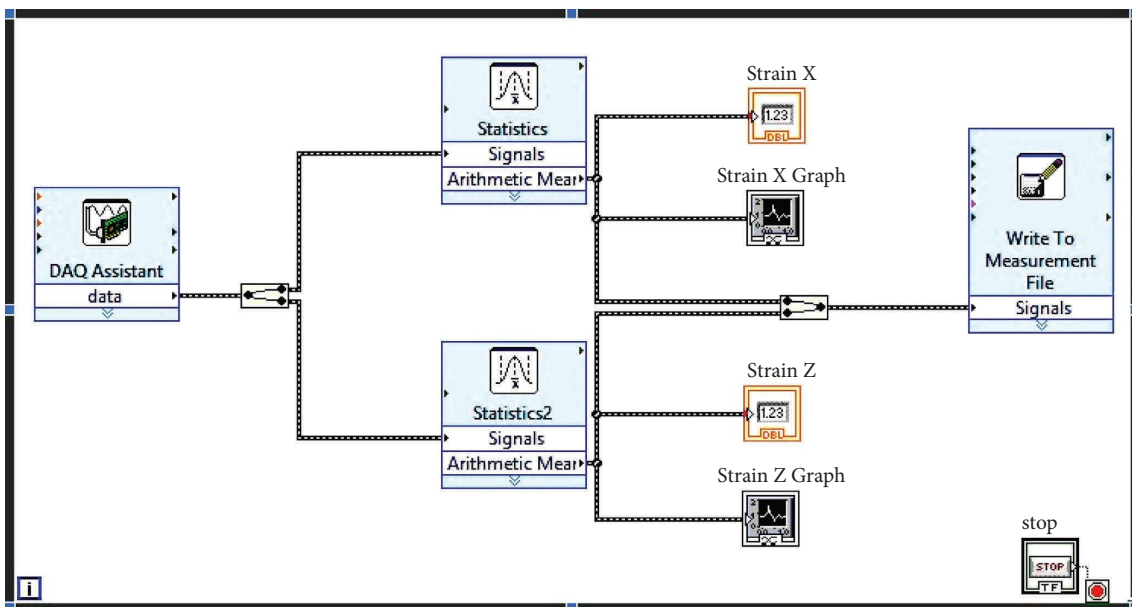


FIGURE 2: Illustration of signal acquisition.

Control factors and CF were considered input parameters for predicting the tool wear during this study. The CF was increased when tool wear also increased. The CF produced during the metal cutting impacted the heat generation in the cutting zone, tool wear, machined surface quality, and workpiece accuracy [27, 41].

**3.1.3. Output Parameter of the ANFIS.** The response of the ANFIS is flank wear which is the gradual failure of the tool due to continuous contact with the workpiece during the machining operation. The flank wear was considered as output.

Figure 6 displays the fuzzy architecture of the ANFIS model where the triangular membership function was implemented. Figure 7 is the rule editor where all the rules were fed into the fuzzy system [42]. It consists of 81 fuzzy rules (4 input parameters with three membership functions). ANFIS architecture is shown in Figure 7.

**Step 3.** ANFIS model training with the given input data.

For the training of the ANFIS model, 14 sets (out of 20 sets—70%) of experimental data were selected, and the training was performed with the chosen ANFIS model. The epoch was regulated until the error reached  $<0.001$  [7]. The ANFIS learning scenario for the prediction can be analyzed from Figure 8 with epoch 20. The average error was achieved as 0.00067123 mm.

**Step 4.** Testing of the ANFIS model with the experimental data.

The turning data sets (6 sets—30%) obtained from the experiments were used to test with the trained ANFIS model. The response values were loaded into the ANFIS GUI and tested across the parametric value. The user can use checking data and demo data to be loaded into the GUI and can be used for prediction. Testing data was applied to the developed ANFIS model and found

TABLE 1: Experimental layout.

Std. order	Cutting speed (mm/min)	Feed rate (mm/rev)	Depth of cut (mm)	Cutting force (N)	Modeling
1	90	0.2	0.3	959	Training
2	270	0.2	0.3	1137	Testing
3	90	0.4	0.3	1047	Testing
4	270	0.4	0.3	1062	Training
5	90	0.2	0.6	996	Training
6	270	0.2	0.6	1084	Testing
7	90	0.4	0.6	1002	Training
8	270	0.4	0.6	1033	Training
9	90	0.3	0.45	1037	Testing
10	270	0.3	0.45	1034	Training
11	180	0.2	0.45	1042	Training
12	180	0.4	0.45	1043	Training
13	180	0.3	0.3	1040	Testing
14	180	0.3	0.6	1081	Training
15	180	0.3	0.45	1015	Testing
16	180	0.3	0.45	971	Training
17	180	0.3	0.45	1190	Training
18	180	0.3	0.45	1032	Training
19	180	0.3	0.45	997	Training
20	180	0.3	0.45	983	Training

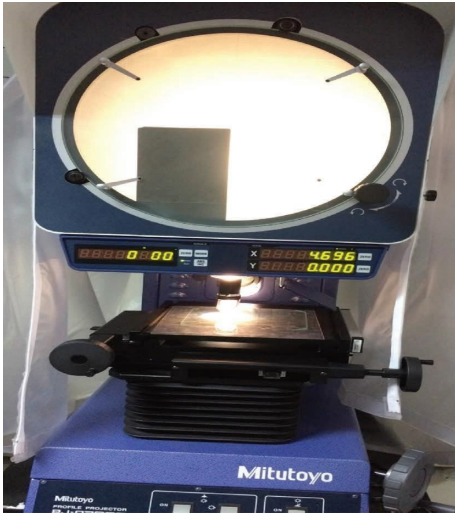


FIGURE 3: Profile projector for flank wear measurement.

an average error of 0.0074011 mm, shown in Figure 9. The blue circles were the actual data, and the red star was the predicted one with the same input parameters criterion.

*Step 5.* ANFIS model testing with experimental data.

The ANFIS model was tested with the experimental data to validate the accuracy of the model. The ANFIS can predict any value between this range. Finally, the output obtained from the ANFIS was compared with the experimental data obtained from the turning operation. The close assent displayed that the developed ANFIS can be

employed to forecast flank wear. To assess the accuracy of the ANFIS model, the error  $E_i$  is defined using the following equation:

$$E_i = \frac{\text{Actual value}_i - \text{Predicted value}_i}{\text{Actual value}_i} \times 100. \quad (1)$$

#### 4. Modeling Using Convolutional Autoencoder

Convolutional auto encoder (CAE) is an unsupervised model with convolutional layers. CAEs are typically used to reduce and compress the size of the input dimension and extract robust features. CAE is distinguished from traditional autoencoders by convolutional layers. These layers stand out for their appealing ability to extract knowledge and learn the internal representation of data. CAE system comprises two convolutional neural network (CNN) models, the encoder and the decoder. The primary function of the encoder is to convert the original input data into a lower-dimensional latent representation. The decoder is in charge of reconstructing the compressed latent representation to generate output data. Training of the CAE is nearly identical to the standard ANN. The backpropagation method minimizes the cost function and the enhanced weight and bias matrix. The structure of the purported model is shown in Figure 10. The mean square error function (MSE) and cross-entropy loss function, which can define reconstruction error, are frequently used as the cost function in autoencoder training.

The model is built in Google Colab, a free Jupyter notebook environment used to implement the CAE. Colab is a free cloud-based service provided by Google, and the best feature of Colab is that no prior installation is required.

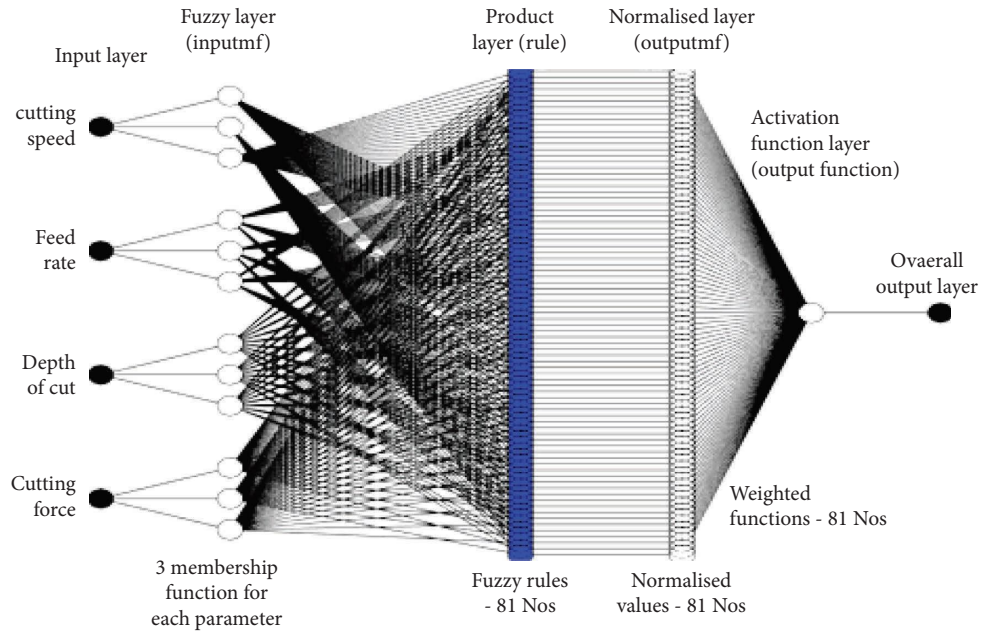


FIGURE 4: ANFIS structure for four input and single output parameters.

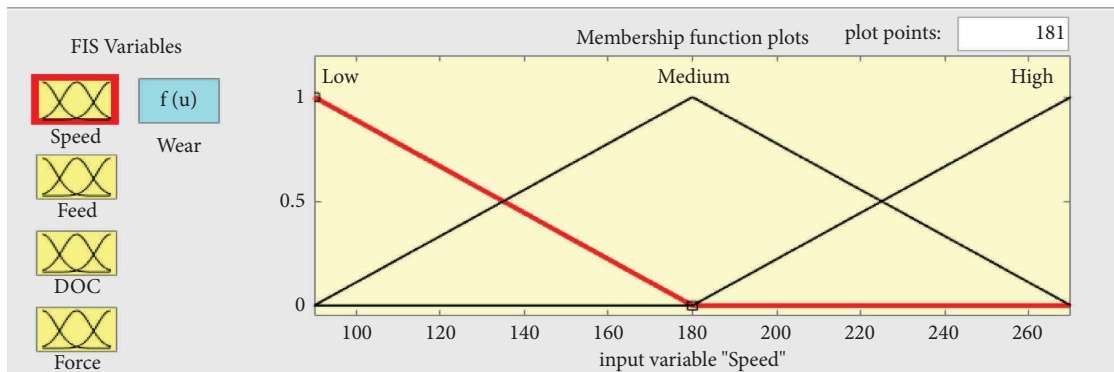


FIGURE 5: The initial MF for "Speed".

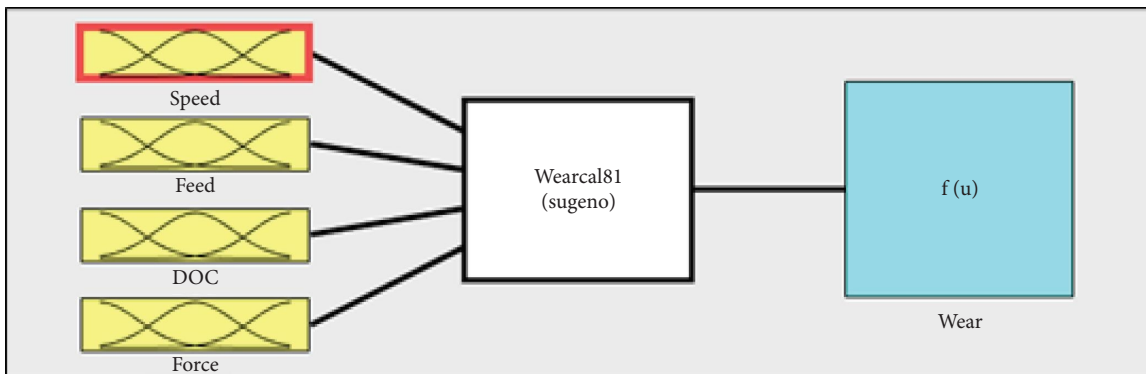


FIGURE 6: Architecture of ANFIS.

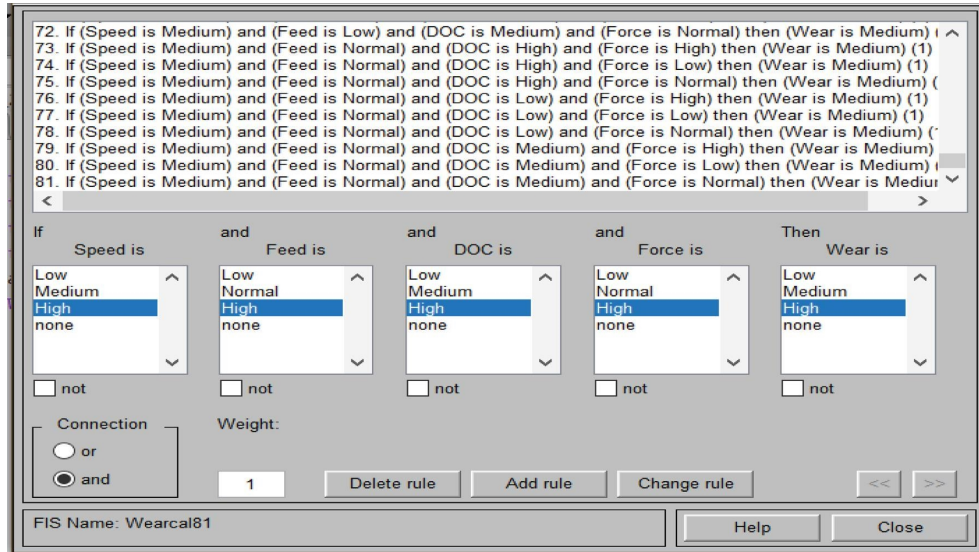


FIGURE 7: Rule editor.

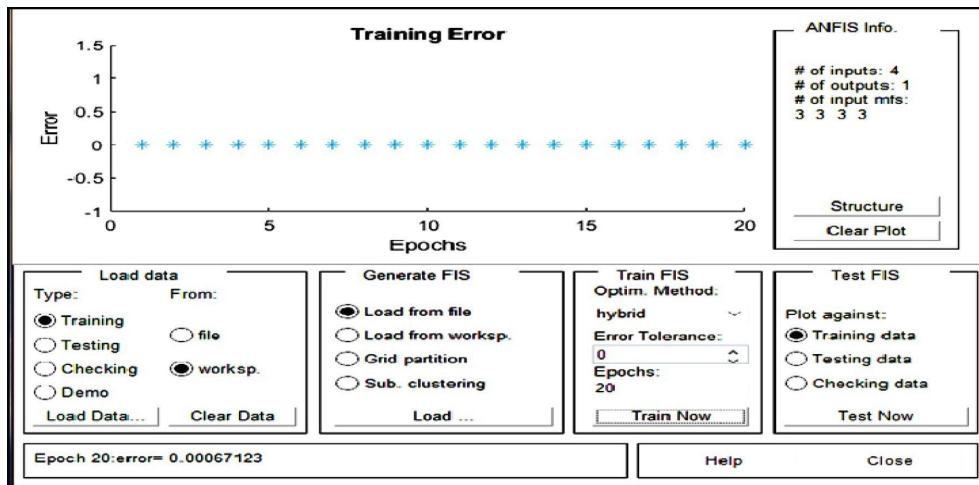


FIGURE 8: Training error.

Many data science and machine learning libraries are pre-installed in Colab, including Pandas, NumPy, TensorFlow, Keras, and OpenCV [43].

4.1. *End-to-End-Framework.* Unlike the conventional compression methods, the CAE is an end-to-end standard that combines compression and encoding methods. CAE model can persistently encode the input into small-size data by mapping the hidden layers without requiring signal transformation. Convolutional encoding (CE) and convolutional decoding (CD) are linked together during the training phase to learn mapping functions for compression and reconstruction. In CE, the input data is encoded into the compressed data as given by the following equation:

$$y = f_{w,b}(x). \quad (2)$$

In the hidden layers,  $x$  is the input signal,  $y$  is the compressed data,  $w$  is the weight, and  $b$  is the bias. In CE,  $f(\cdot)$  signifies the compression mapping function. In CD, the compressed data  $y$  are reconstructed to form the initial signal  $x$ , as defined below.

$$x = g_{w',b'}(y). \quad (3)$$

In CD,  $w'$  is the weight, and  $b'$  is the bias. The reconstruction mapping function is denoted by  $g(\cdot)$ . As a result, this end-to-end standard can condense the input to a small-scale set using the CE mapping function without an independent encoding method. Furthermore, the compressed data can be restored in CD to obtain the actual output, as shown in Table 2. It overcomes the compression quality limitation imposed by programmed expansion biases and mined parameters in conventional

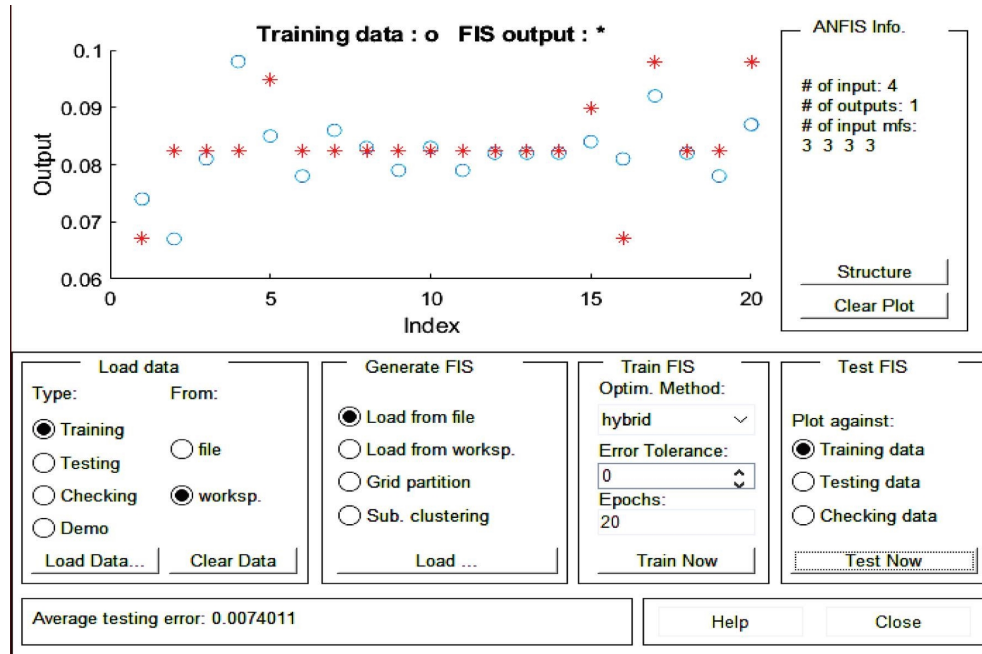


FIGURE 9: Testing error.

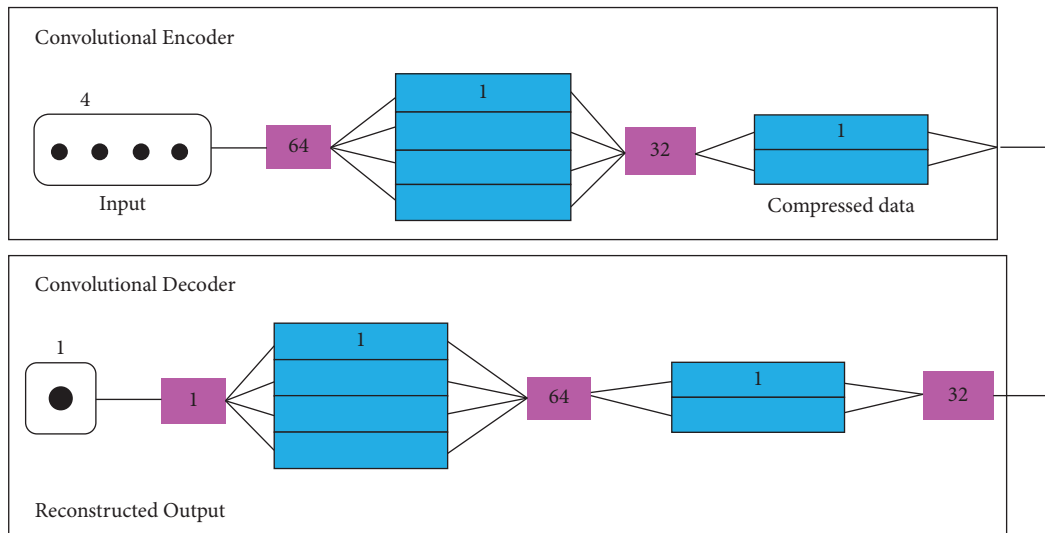


FIGURE 10: Structure of the CAE model.

TABLE 2: The details of layers and parameters using the CAE model.

Section	No	Layer name	Filters × kernels	Activation function	Output size	Parameters
CE	1	Input layer	—	—	—	—
	2	1D convolution	64 × 2	Sigmoid	(None, 1, 64)	448
	3	Dropout	—	—	—	0
	4	1D convolution	32 × 2	Sigmoid	(None, 1, 32)	4128
CD	5	1D transposed convolution	32 × 2	Sigmoid	(None, 1, 32)	2082
	6	Dropout	—	—	—	0
	7	1D transposed convolution	64 × 2	Sigmoid	(None, 1, 64)	4160
	8	1D transposed convolution	1 × 2	Sigmoid	(None, 1, 1)	129
Reconstructed output	9	—	—	—	(None, 1, 1)	—
Total no. of parameters						10,945

TABLE 3: Prediction accuracy of different membership functions.

S. no.	Membership function-ANFIS structure	Optimization method	Training error (mm)	Testing error (mm)
1	Gauss MF	BPNN	0.18915	1.9795
2	Gauss MF	Hybrid	<b>0.000134</b>	<b>0.0704</b>
3	TRAF MF	Hybrid	0.000966	0.0849
4	Gauss 2MF	Hybrid	0.000164	0.0783
5	PRINMF	Hybrid	0.000166	0.0788
6	DSIGMF	Hybrid	0.000165	0.0784
7	PSIGMF	Hybrid	0.000407	0.0783
8	Subtractive clustering	Hybrid	0.002239	0.0929

TABLE 4: Experimental vs. Predicted values.

Std. order	Flank wear (experimental)	Predicted flank wear-ANFIS	% Error	Predicted flank wear-CAE	% error
1	0.071	0.067	5.63	0.078	-9.86
2	0.067	0.073	-8.96	0.072	-7.46
3	0.081	0.083	-2.47	0.09	-11.11
4	0.098	0.093	5.10	0.08	18.37
5	0.085	0.090	-5.88	0.082	3.53
6	0.078	0.083	-6.41	0.085	-8.97
7	0.086	0.083	3.49	0.085	1.16
8	0.083	0.083	0.00	0.085	-2.41
9	0.079	0.083	-5.06	0.083	-5.06
10	0.082	0.083	-1.22	0.083	-1.22
11	0.079	0.083	-5.06	0.082	-3.80
12	0.082	0.083	-1.22	0.083	-1.22
13	0.082	0.083	-1.22	0.085	-3.66
14	0.082	0.083	-1.22	0.081	1.22
15	0.084	0.090	-7.14	0.083	1.19
16	0.081	0.077	4.94	0.082	-1.23
17	0.092	0.098	-6.52	0.083	9.78
18	0.082	0.083	-1.22	0.082	0.00
19	0.078	0.083	-6.41	0.081	-3.85
20	0.087	0.090	-3.45	0.078	10.34

ways. The end-to-end model, operated through mapping functions in hidden layers, attempts to obtain adequate compressed data that recreate the actual data with minor errors.

**4.2. Training.** The training of the CAE takes place using the backpropagation technique and adjusting the parameters to decrease the loss functions based on a minibatch of training models given in equation (4). The batch is randomly chosen from the training dataset. Training can be accomplished by repetitions of the forward and backward pass.

$$L = \sum_{n=1}^N (D_n - F(D_n))^2, \quad (4)$$

where  $N$  is the minibatch size;  $D_n$  is the  $n^{\text{th}}$  input;  $F(\cdot)$  is a function of CAE consisting CE and CD; and  $L$  is the loss function. The loss function (4) is known as the square loss, which is effective in regression analysis. Let  $\varphi$  be trainable matrix parameters and rules can be updated using

$$\varphi_{t+1} = \varphi_t + \frac{\eta}{N} \sum_{n=1}^N \nabla_{\varphi_t} L(D_n), \quad (5)$$

where  $\eta$  is the learning rate;  $\nabla$  is the gradient operator; and  $t$  is the -training step. Parameters, including batch size and the number of iterations, impact the accuracy. After training, CAE can be divided into CE and CD to accomplish compression and restoration. The tuning parameters like  $w$ ,  $w'$ ,  $b$ , and  $b'$  are regulated in training to curtail the modification between the restored output and actual output to attain the optimal parameter settings.

**4.3. Performance Evaluation.** The evaluation criteria for assessing the proposed method's reconstruction quality are mean squared error (MSE) and percentage error.

- (1) Mean squared error: the mean squared error (MSE) is the standard metric for assessing the performance of most regression algorithms. The less data there are, the smaller the aggregated error, MSE, as shown in

$$L = \frac{1}{N} \left[ \sum (\hat{Y} - Y)^2 \right]. \quad (6)$$



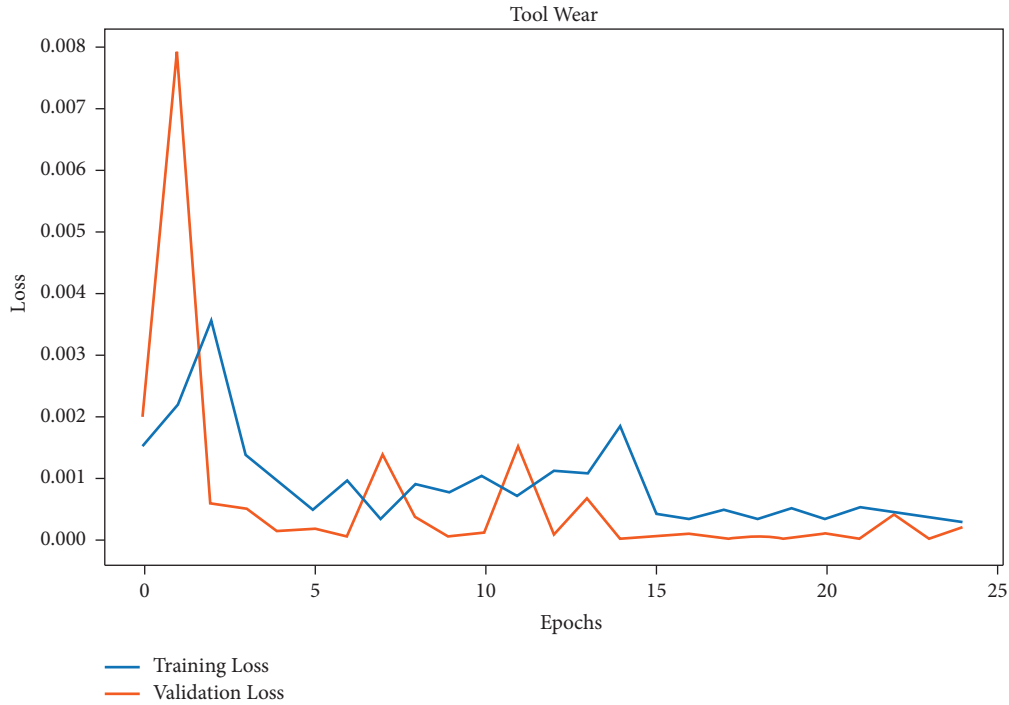


FIGURE 11: Fluctuation of the loss value vs. epochs for the proposed CAE.

- (2) Percentage error: the percentage error is the difference between the actual and predicted values, as described in equation (7). It can be expressed as an absolute or relative error.

$$\delta = \left| \frac{\nu_A - \nu_E}{\nu_E} \right| \times 100\%. \quad (7)$$

## 5. Results and Discussion

The analysis was carried out between the actual data and the prediction model that depends on the various input parameters such as cutting speed, feed, depth of cut, and CF. The various MF used in ANFIS modeling reports different results. Optimizing methods like BPNN and hybrid were employed to fine-tune the membership functions. Initially, modeling was performed with the “Gauss” membership function with BPNN and the hybrid optimization method. The hybrid optimization method performed better than BPNN.

A hybrid method comprises backpropagation for the MF parameters related to the input MF. It estimates the least squares for the MF parameters connected with the output MF, which is used to reduce the error values. The grid-based ANFIS structure performed better than the subtractive clustering-based ANFIS structure; hence, grid-based model was selected [40]. The training and testing error for various MF are presented in Table 3. From Table 3, it was found that “Gauss” MF performed better than other functions. The comparison of the actual and predicted tool wear values is given in Table 4.

The CF is a significant parameter to be examined in the machining processes to determine the machined component’s tool life, tool breakage, flank wear, and Ra. When the prediction value produced from the model is within the prescribed limit, the cutting tool is in satisfactory condition and can do the machining operation. If it exceeds the limit, the cutting tool is not in good condition, so it needs to be replaced. The CAE model was trained for data compression on 20 sets. Each value contains three measurements for training and three measures for validation. In the initial training phase, the model’s training error reduced from 0.0035 mm to 0.001 mm during the 25 iterations. The validation error reduced from 0.008 mm to 0.001 mm during the 25 iterations, and the training error and validation error for the trails are shown in Figure 11.

## 6. Conclusions

The turning of EN8 steel was performed in dry conditions to accelerate the tool wear. The CF and flank wear were measured during the turning operation. The DL model was developed with speed, feed, cutting depth, and CF signals to predict flank wear. Based on the investigational results, the subsequent conclusions have arrived.

- (i) Flank wear was mainly dependent on cutting speed and depth of cut.
- (ii) An increase in CF indicates an increase in tool wear. Hence, CF is used as an indicator.

- (iii) The hybrid optimization method performed better than BPNN in regulating the membership functions in ANFIS.
- (iv) Various MFs were used to train the ANFIS model, in which the grid-based structure with Gauss MF reveals the best minimum training and testing error.
- (v) Grid-based ANFIS structure performed better than subtractive clustering-based ANFIS structure.
- (vi) ANFIS model was used to predict the flank wear with the minimum and maximum average error of 0.602% and 5.77%, respectively.
- (vii) The overall prediction accuracy of the ANFIS model was 99.81% with the Gauss membership function.
- (viii) The average prediction error for ANFIS and CAE models is 4.131% and 5.273%, respectively.
- (ix) The CAE model flourishes the limitations of conventional neural networks, which arbitrarily initialize the weights of the network; hence, the proposed method can be efficiently used to predict the flank wear of the tool under various machining conditions.
- (x) Compared with the conventional backpropagation method, the proposed method is more appropriate for describing the hidden wear feature from the collected data and provides better prediction accuracy.
- (xi) In the future, the proposed method can monitor tool wear states online under different machining conditions. Therefore, this method is expected to be more widely used in tool wear monitoring in metal cutting processes.
- (xii) For other processes like milling and drilling, the model has to be trained with selected input parameters and used to predict the tool condition.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] S. Pradhan, S. R. Das, P. C. Jena, and D. Dhupal, "Machining performance evaluation under recently developed sustainable HAJM process of zirconia ceramic using hot SiC abrasives: an experimental and simulation approach," *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science*, vol. 236, no. 2, pp. 1009–1035, 2022.
- [2] S. Pradhan, S. R. Das, B. K. Nanda, P. C. Jena, and D. Dhupal, "Machining of hardstone quartz with modified AJM process using hot SiC abrasives: analysis, modeling, optimization, and cost analysis," *Surface Review and Letters*, vol. 28, no. 02, Article ID 2050049, 2021.
- [3] L. Dan and J. Mathew, "Tool wear and failure monitoring techniques for turning—a review," *International Journal of Machine Tools and Manufacture*, vol. 30, no. 4, pp. 579–598, 1990.
- [4] P. Krishnakumar, K. Rameshkumar, and K. Ramachandran, "Tool wear condition prediction using vibration signals in high speed machining (HSM) of titanium (Ti-6Al-4 V) alloy," *Procedia Computer Science*, vol. 50, pp. 270–275, 2015.
- [5] T. Mohanraj, S. Shankar, R. Rajasekar, N. Sakthivel, and A. Pramanik, "Tool condition monitoring techniques in milling process—a review," *Journal of Materials Research and Technology*, vol. 9, no. 1, pp. 1032–1042, 2020.
- [6] T. Mohanraj, T. Deepesh, R. Dhinesh, S. Jayaprakash, and S. Sai Krishna, "Design and analysis of a strain gauge based eight-shaped elliptical ring dynamometer for milling force measurement," *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 17, pp. 3125–3134, 2020.
- [7] T. Mohanraj, S. Shankar, R. Rajasekar, and M. Uddin, "Design, development, calibration, and testing of ingeniously developed strain gauge based dynamometer for cutting force measurement in the milling process," *Journal of Mechanical Engineering and Sciences*, vol. 14, no. 2, pp. 6594–6609, 2020.
- [8] M. Rizal, J. A. Ghani, M. Z. Nuawi, and C. H. C. Haron, "Online tool wear prediction system in the turning process using an adaptive neuro-fuzzy inference system," *Applied Soft Computing*, vol. 13, no. 4, pp. 1960–1968, 2013.
- [9] I. Maher, M. E. H. Eltaib, A. A. D. Sarhan, and R. M. El-Zahry, "Investigation of the effect of machining parameters on the surface quality of machined brass (60/40) in CNC end milling—ANFIS modeling," *International Journal of Advanced Manufacturing Technology*, vol. 74, no. 1-4, pp. 531–537, 2014.
- [10] T.-I. Liu, B. Jolley, and C.-H. Yang, "Online detection and measurements of tool wear for precision boring of titanium components," *Proceedings of the Institution of Mechanical Engineers - Part B: Journal of Engineering Manufacture*, vol. 230, no. 7, pp. 1331–1342, 2016.
- [11] L. H. Saw, L. W. Ho, M. C. Yew et al., "Sensitivity analysis of drill wear and optimization using Adaptive Neuro fuzzy-genetic algorithm technique toward sustainable machining," *Journal of Cleaner Production*, vol. 172, pp. 3289–3298, 2018.
- [12] B. Sen, U. K. Mandal, and S. P. Mondal, "Advancement of an intelligent system based on ANFIS for predicting machining performance parameters of Inconel 690—A perspective of metaheuristic approach," *Measurement*, vol. 109, pp. 9–17, 2017.
- [13] V. Jain and T. Raj, "Tool life management of unmanned production system based on surface roughness by ANFIS," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 2, pp. 458–467, 2017.
- [14] I. Shivakoti, G. Kibria, P. M. Pradhan, B. B. Pradhan, and A. Sharma, "ANFIS based prediction and parametric analysis during turning operation of stainless steel 202," *Materials and Manufacturing Processes*, vol. 34, no. 1, pp. 112–121, 2019.
- [15] V. Jain and T. Raj, "Prediction of cutting force by using ANFIS," *International Journal of System Assurance Engineering and Management*, vol. 9, no. 5, pp. 1137–1146, 2018.
- [16] A. Khan and K. Maity, "A comprehensive GRNN model for the prediction of cutting force, surface roughness and tool

- wear during turning of CP-Ti grade 2,” *Silicon*, vol. 10, no. 5, pp. 2181–2191, 2018.
- [17] A. Zaretalab, H. S. Haghghi, S. Mansour, and M. S. Sajadieh, “Optimisation of tool replacement time in the machining process based on tool condition monitoring using the stochastic approach,” *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 2, pp. 159–173, 2019.
  - [18] D. Nguyen, S. Yin, Q. Tang, P. X. Son, and L. A. Duc, “Online monitoring of surface roughness and grinding wheel wear when grinding Ti-6Al-4V titanium alloy using ANFIS-GPR hybrid algorithm and Taguchi analysis,” *Precision Engineering*, vol. 55, pp. 275–292, 2019.
  - [19] R. Kumar and N. R. J. Hynes, “Prediction and optimization of surface roughness in thermal drilling using integrated ANFIS and GA approach,” *Engineering Science and Technology, an International Journal*, vol. 23, no. 1, pp. 30–41, 2020.
  - [20] M. Marani, V. Songmene, M. Zeinali, J. Kouam, and Y. Zedan, “Neuro-fuzzy predictive model for surface roughness and cutting force of machined Al-20 Mg 2 Si-2Cu metal matrix composite using additives,” *Neural Computing & Applications*, vol. 32, no. 12, pp. 8115–8126, 2020.
  - [21] M. A. Khan, A. S. Kumar, S. T. Kumaran, M. Uthayakumar, and T. J. Ko, “Effect of tool wear on machining GFRP and AISI D2 steel using alumina based ceramic cutting tools,” *Silicon*, vol. 11, no. 1, pp. 153–158, 2019.
  - [22] S. Kumar and B. Singh, “Chatter prediction using merged wavelet denoising and ANFIS,” *Soft Computing*, vol. 23, no. 12, pp. 4439–4458, 2019.
  - [23] A. Laouissi, M. A. Yaltese, A. Belbah, S. Belhadi, and A. Haddad, “Investigation, modeling, and optimization of cutting parameters in turning of gray cast iron using coated and uncoated silicon nitride ceramic tools. Based on ANN, RSM, and GA optimization,” *International Journal of Advanced Manufacturing Technology*, vol. 101, no. 1-4, pp. 523–548, 2019.
  - [24] S. Pradhan, D. Dhupal, S. R. Das, and P. C. Jena, “Experimental Investigation and optimization on machined surface of Si<sub>3</sub>N<sub>4</sub> ceramic using hot SiC abrasive in HAJM,” *Materials Today Proceedings*, vol. 44, pp. 1877–1887, 2021.
  - [25] B. Sen, M. Mia, U. K. Mandal, and S. P. Mondal, “GEP-and ANN-based tool wear monitoring: a virtually sensing predictive platform for MQL-assisted milling of Inconel 690,” *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 1-4, pp. 395–410, 2019.
  - [26] S. Pradhan, S. R. Das, P. C. Jena, and D. Dhupal, “Investigations on surface integrity in hard turning of functionally graded specimen under nano fluid assisted minimum quantity lubrication,” *Advances in Materials and Processing Technologies*, vol. 8, pp. 1714–1729, 2021.
  - [27] S. Mahapatra, A. Das, P. C. Jena, and S. R. Das, “Turning of hardened AISI H13 steel with recently developed S3P-ALTiSiN coated carbide tool using MWCNT mixed nanofluid under minimum quantity lubrication,” *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science*, vol. 237, no. 4, pp. 843–864, 2022.
  - [28] D. Rodić, M. Sekulić, M. Gostimirović, V. Pucovsky, and D. Kramar, “Fuzzy logic and sub-clustering approaches to predict main cutting force in high-pressure jet assisted turning,” *Journal of Intelligent Manufacturing*, vol. 32, pp. 21–36, 2020.
  - [29] Y. Seid Ahmed, A. Arif, and S. C. Veldhuis, “Application of the wavelet transform to acoustic emission signals for built-up edge monitoring in stainless steel machining,” *Measurement*, vol. 154, Article ID 107478, 2020.
  - [30] L. E. Escajeda Ochoa, I. B. Ruiz Quinde, J. P. Chuya Sumba, A. V. Guevara Jr, and R. Morales-Menendez, “New approach based on autoencoders to monitor the tool wear condition in HSM,” *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 206–211, 2019.
  - [31] A. Das, S. R. Das, J. P. Panda et al., “Machine Learning-based modeling and optimization in hard turning of AISI D6 steel with advanced ALTiSiN-coated carbide inserts to predict surface roughness and other machining characteristics,” *Surface Review and Letters*, vol. 29, no. 10, Article ID 2250137, 2022.
  - [32] F. Wang, Q. Ma, W. Liu et al., “A novel ECG signal compression method using spindle convolutional auto-encoder,” *Computer Methods and Programs in Biomedicine*, vol. 175, pp. 139–150, 2019.
  - [33] L. Yang and Z. Zhang, “A conditional convolutional autoencoder-based method for monitoring wind turbine blade breakages,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6390–6398, 2021.
  - [34] D. Wang, R. Hong, and X. Lin, “A method for predicting hobbing tool wear based on CNC real-time monitoring data and deep learning,” *Precision Engineering*, vol. 72, pp. 847–857, 2021.
  - [35] T. Sk, S. Shankar, T. Mohanraj, and K. Devendran, “Tool wear prediction in hard turning of EN8 steel using cutting force and surface roughness with artificial neural network,” *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science*, vol. 234, no. 1, pp. 329–342, 2020.
  - [36] S. Shankar, S. Thangarasu, T. Mohanraj, and D. Pravien, “Prediction of cutting force in turning process: an experimental and fuzzy approach,” *Journal of Intelligent and Fuzzy Systems*, vol. 28, no. 4, pp. 1785–1793, 2015.
  - [37] S. Shankar, T. Mohanraj, and A. Pramanik, “Tool condition monitoring while using vegetable based cutting fluids during milling of Inconel 625,” *Journal of Advanced Manufacturing Systems*, vol. 18, no. 04, pp. 563–581, 2019.
  - [38] J. Jena, A. Panda, A. Behera, P. Jena, S. R. Das, and D. Dhupal, “Modeling and Optimization of Surface Roughness in Hard Turning of AISI 4340 Steel with Coated Ceramic Tool,” *Innovation in Materials Science and Engineering*, Springer, Berlin, Germany, 2019.
  - [39] M. Buragohain and C. Mahanta, “A novel approach for ANFIS modelling based on full factorial design,” *Applied Soft Computing*, vol. 8, no. 1, pp. 609–625, 2008.
  - [40] S. Shankar, T. Mohanraj, and R. Rajasekar, “Prediction of cutting tool wear during milling process using artificial intelligence techniques,” *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 2, pp. 174–182, 2019.
  - [41] S. Shankar, T. Mohanraj, and S. K. Thangarasu, “Multi-response milling process optimization using the Taguchi method coupled to grey relational analysis,” *Materials Testing*, vol. 58, no. 5, pp. 462–470, 2016.
  - [42] S.-I. Kwak, U.-S. Ryu, G. Kim, and M.-H. Jo, “A fuzzy reasoning method based on compensating operation and its application to fuzzy systems,” *Iranian Journal of Fuzzy Systems*, vol. 16, pp. 17–34, 2019.
  - [43] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

## Research Article

# Bayesian Network Structure Learning by Ensemble Learning and Frequent Item Mining

Guoxin Cao<sup>1</sup> and Haomin Zhang <sup>1,2</sup>

<sup>1</sup>College of Science, Guilin University of Technology, Guilin 541000, China

<sup>2</sup>Guangxi Colleges and Universities Key Laboratory of Applied Statistics, Guilin University of Technology, Guilin 541000, China

Correspondence should be addressed to Haomin Zhang; zhanghm@glut.edu.cn

Received 10 September 2022; Revised 27 October 2022; Accepted 24 November 2022; Published 11 February 2023

Academic Editor: Abolfazl Gharaei

Copyright © 2023 Guoxin Cao and Haomin Zhang. 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.

Aiming at the common problem of low learning effect in single structure learning of a Bayesian network, a new algorithm EF-BNSL integrating ensemble learning and frequent item mining is proposed. Firstly, the sample set is obtained by sampling the original dataset using Bootstrap, which is mined using the Apriori algorithm to derive the maximum frequent items and association rules so that the black and white list can be determined. Secondly, considering that there may be wrong edges in the black and white list, the black and white list is used as the penalty term of the BDeu score and the initial network is obtained from the hill climbing algorithm. Finally, repeat the above steps 10 times to obtain 10 initial networks. The 10 initial networks were integrated and learned by the integrated strategy function to obtain the final Bayesian network. Experiments were carried out on six standard networks to calculate  $F_1$  score and HD. The results show that the EF-BNSL algorithm can effectively improve  $F_1$  score, reduce HD, and learn the network structure that is closer to the real network.

## 1. Introduction

Bayesian network (BN), a probabilistic graphical model, was proposed by Pearl [1] in 1988 and is capable of effective inference and analysis of uncertain knowledge, which is one of the hot spots of research in machine learning. BN has a solid theory base and has been broadly applied in many industries, including transportation [2], industrial production [3–5], economy [6–8], medicine [9], and agriculture [10, 11]. It is obvious that the innovative research on BN theories and methodologies, as well as the scientific construction of more effective algorithms, will definitely promote the ability of problem solving in practical areas [12, 13].

The fundamental theory of BN mainly contains structure learning, parameter learning, and Bayesian inference. Parameter learning is to learn network parameters on the known structure of the network. Structure learning includes the learning of parameters and the learning of the network structure, which is the focus and difficulty of BN learning. The basis of parametric learning and Bayesian inference is

BN structure learning. However, finding the optimal structure of the BN is a NP-hard problem [14]. The computation complexity grows exponentially as the number of nodes increases.

There are three main structure learning methods, which are constraint-based, fraction-based, and hybrid methods. Constraint-based methods usually use conditional independence tests or mutual information to identify dependency relationships between variables. Spirtes and Meek [15] proposed the SGS algorithm, the first structure learning algorithm, which determines the network structure mainly the conditional independence between nodes, but the learning efficiency grows exponentially. Spirtes et al. [16] improved the SGS algorithm by proposing the PC algorithm, which is enabled to construct BN from sparse networks and allows the use of chi-square tests without the necessity of a specific independence test. Qi et al. [17] proposed the WMIF algorithm to learn the network structure using the weakest-first strategy. The core concept of the score-based approach is to find the best structure based on the scoring function by

traversing all possible structures. Cooper and Herskovits [18] proposed the K2 algorithm, but the algorithm required a prior upper limit on the ordering of the nodes and the number of parents per node. Lee and Beek [19] proposed a local greedy search method coupled with a perturbation factor and used the idea of a knockdown algorithm to improve the performance of a local greedy search by using a metaheuristic. The scoring functions commonly used are MDL [20], AIC [21], BIC [22], and BDeu [23]. The common search algorithms are the K2 algorithm [18], hill climbing algorithm [24], and genetic algorithm [25]. The essential idea of hybrid learning is to decrease the size of the search space by first testing for independence and then using a scoring search method to gain the most optimal network structure. The first hybrid learning algorithm was the CB algorithm proposed by Singh and Valtorta [26], which starts by attempting a constraint-based PC algorithm to identify the order of the nodes and then learns the structure using the K2 algorithm. Alonso-Barba et al. [27] proposed the I-ACO-B algorithm, which first reduces the complexity of the search space using independence tests, and then uses an ant colony algorithm for scoring search to obtain the optimal network structure.

Recently, Eggeling et al. [28] proposed the idea of introducing structural priors and Wang et al. [29] proposed the use of constraints, both of which can greatly decrease the search space of structure learning, which is an idea for improving the existing structure learning algorithms. Xiao et al. [30] proposed an algorithm to merge association rules and knowledge for network structure learning. Sun et al. [31] proposed a new hybrid approach by integrating the PC and PSO algorithm, which takes parts of the output of the PC algorithm as structure priors to improve the initial solutions to BN structure learning. Li et al. [32] proposed BN structure learning based on frequent term mining. Wang and Qin [33] proposed BN structure learning based on ensemble and feedback strategies, but only integrated multiple BNs are

obtained through BDeu score. Based on this paper, a new BN structure learning algorithm named the EF-BNSL algorithm based on ensemble learning and frequent item mining is designed. The EF-BNSL algorithm flowchart is shown in Figure 1.

## 2. Materials and Methods

*2.1. Bayesian Network.* A BN can be shown by  $BN = (G, P)$ , where  $G = (V, E)$  means the directed acyclic graph (DAG),  $V = \{x_1, x_2, \dots, x_n\}$  represents the set of nodes,  $E$  represents the set of directed edges, and  $P$  is the probability distribution among the nodes, which is commonly represented by the conditional probability table (CPT).

The joint probability distribution of the node set  $V = \{x_1, x_2, \dots, x_n\}$  can be represented as

$$P(x_1, x_2, \dots, x_n) = \prod P(x_i | pa(x_i)), \quad (1)$$

where  $pa(x_i)$  is the parent node of  $x_i$  and  $P(x_i | pa(x_i))$  is a conditional probability. For node  $x_i$ , we consider the relationship with  $pa(x_i)$ , and the number of parameters is much smaller than computing the joint probability directly.

As shown in Figure 2, the nodes  $A$  and  $B$  have no parent node, so their probabilities are directly  $P(A)$  and  $P(B)$ . The parents of node  $C$  are nodes  $A$  and  $B$ , so the probability of node  $C$  is  $P(C|A, B)$ . Similarly, the parent node of the node  $D$  is the node  $C$ . The probability of node  $D$  is  $P(D|C)$ . Finally, we can get the CPTs of all nodes.

The goal of Bayesian network structure learning is to obtain the most suitable network structure for the dataset. The scoring function is used to measure the fitness of the BN structure for the dataset. The search algorithm is a search strategy and can find the best BN structure when the scoring function is determined. In this paper, we chose the BDeu score [23]. The BDeu score is given as follows:

$$F(G | D) = \log (P(G)) + \sum_{i=1}^n \sum_{j=1}^{q_i} \left[ \log \left( \frac{\Gamma(\eta_{ij})}{\Gamma(\eta_{ij} + N_{ij})} \right) + \sum_{k=1}^{r_i} \log \left( \frac{\Gamma(\eta_{ijk} + N_{ijk})}{\Gamma(\eta_{ijk})} \right) \right], \quad (2)$$

where  $P(G)$  is the prior probability,  $n$  is the number of nodes,  $q_i$  is the number of parents of the  $i$  node,  $r_i$  is the number of values taken by the  $i$  node,  $N_{ijk}$  is the number of samples, which satisfy the node  $x_i = k$ ,  $pa(x_i) = j$  in the dataset, and  $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ ,  $\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk}$ . Usually, we assume that  $\eta_{ijk} = \eta / r_i q_i$ , where  $\eta$  is the given equivalent sample size.

After choosing the scoring function, the problem of structure learning is transformed into an optimization problem which we must find the structure with the highest

score among the possible structures. In this paper, we chose the hill climbing search algorithm [24].

*2.2. Analysis of Association Rules.* Association rules can be applied to identify association relationships between variables. The rule form of the association rule is  $(V_i \longrightarrow V_j)$ , which indicates the association rule of the  $i$  variable with the  $j$  variable, and  $V_i$  is called the premise of the association rule, and  $V_j$  is called the result of the association rule. The

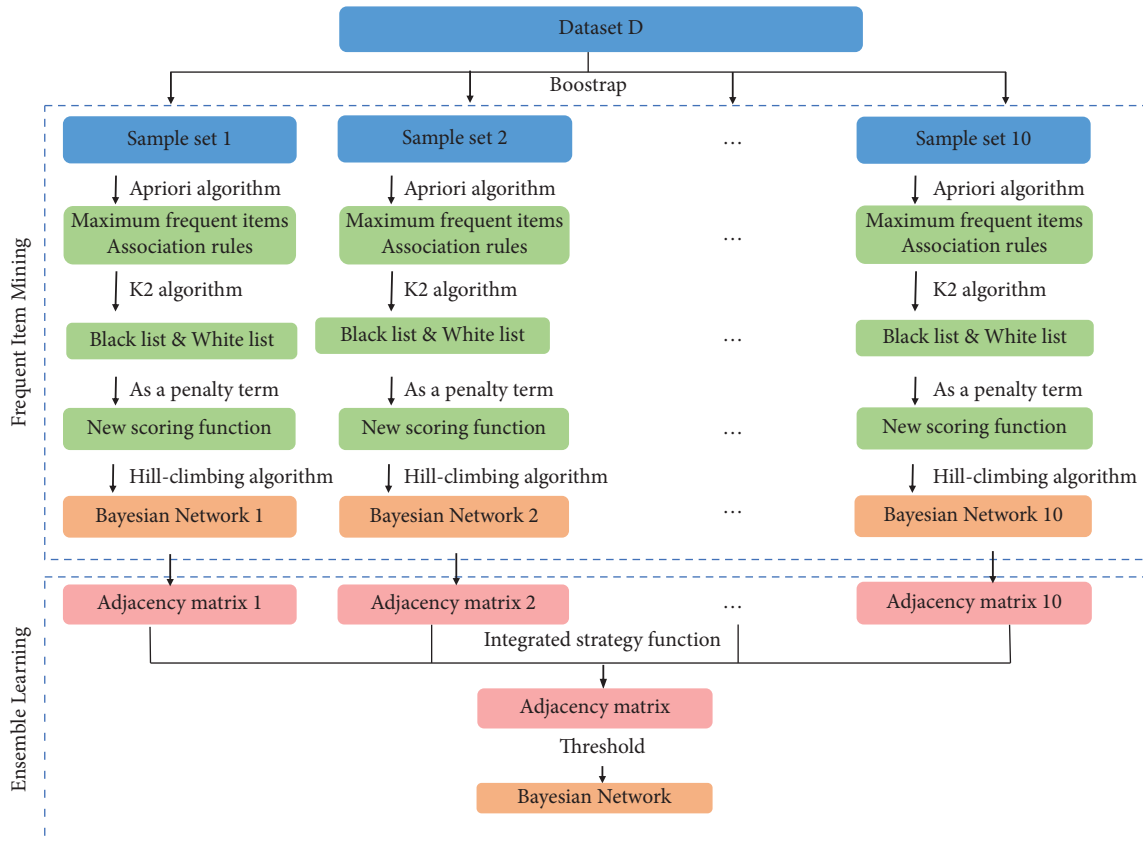


FIGURE 1: The flowchart of the EF-BNSL algorithm.

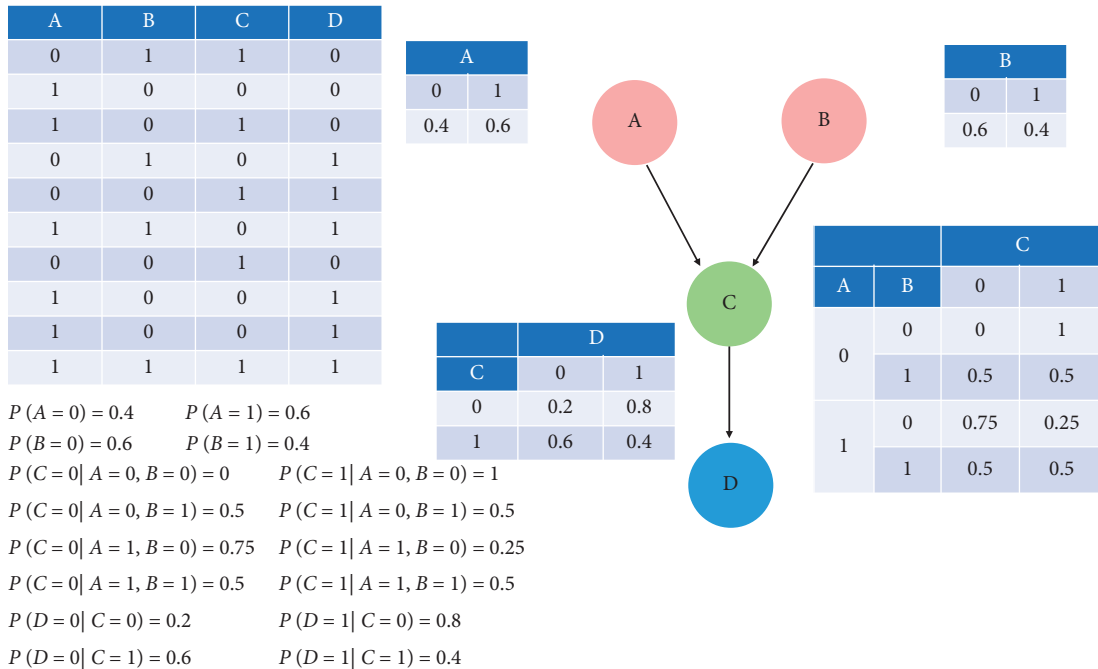


FIGURE 2: An illustrative example of the BN structure and its associated CPT.

definition of association rules requires the introduction of support and confidence, which is defined as follows:

$$\text{Support}(V_i \longrightarrow V_j) = \text{Support}(V_i \cup V_j) = P(V_i V_j), \quad (3)$$

$$\text{Confidence}(V_i \longrightarrow V_j) = \frac{\text{Support}(V_i \longrightarrow V_j)}{\text{Support}(V_i)} = P(V_j | V_i), \quad (4)$$

where  $\text{Support}(V_i \longrightarrow V_j)$  is the support of the association rule  $(V_i \longrightarrow V_j)$ , which is the probability  $P(V_i V_j)$  of  $V_i$  and  $V_j$  occurring simultaneously.  $\text{Confidence}(V_i \longrightarrow V_j)$  is the confidence of the association rule  $(V_i \longrightarrow V_j)$ , which is the conditional probability of  $V_j$  occurring when  $V_i$  occurs.

The basic idea of the Apriori algorithm is to calculate the support of the item set by scanning the dataset several times and finding all frequent item sets to generate association rules. The set1 is said to be a superset of the set2 if every element of the set2 is in the set1. If all supersets of the frequent item set are nonfrequent item sets, the frequent item set is a maximal frequent item (MFI) set. The Apriori algorithm can mine the frequent item sets with support greater than the minimum support and find the MFI by filtering the association rules that satisfy  $\{(V_i \longrightarrow V_j) | \text{Support}(V_i \longrightarrow V_j) \geq \text{min\_support} \cap \text{Confidence}(V_i \longrightarrow V_j) \geq \text{min\_confidence}\}$  to obtain the strongly associated rule set  $\text{Ass} = \cup (V_i \longrightarrow V_j)$ .

**2.3. Ensemble Learning.** Ensemble learning is a new machine learning method which is widely used for classification and regression tasks. Ensemble learning means using multiple identical or different learning algorithms to solve a unified problem by some combination [33].

The most classic ensemble learning is boosting and bagging. The boosting algorithm generates a training set and trains the model by sampling the original dataset with put-back, while the bagging algorithm generates multiple datasets by sampling them with a repeatable sampling technique (bootstrap) and trains the model on them separately and then combines the multiple models to obtain a more stable model. The primary aim of BN structure learning is to determine the directed edges between nodes, and bagging can effectively reduce the possible multilateral and antiedge problems, making the learning results more stable and reliable [34, 35].

### 3. EF-BNSL Algorithm

**3.1. Building the Initial Network.** Since the K2 algorithm is more sensitive, a certain incorrect result may mislead the construction of the whole BN, so this study uses the strong association rule set  $\text{Ass}$  to correct the BN structure of the MFI set  $\text{Max\_freq\_item}$  to increase its robustness.

The process is as follows: firstly, the variables in the MFI set are selected sequentially from the data  $D = \{V, D_v\}$  and  $D_{\text{Max\_freq}} = \{D = \{V, D_v\} | V \in \text{Max\_freq\_item}_i\}$ . Secondly, the BN structure  $\text{BN}_{\text{freq\_}i}$  is obtained by using the K2 algorithm for  $D_{\text{Max\_freq}}$ , respectively,  $\text{BN}_{\text{max\_}f} = \cup \text{BN}_{\text{freq\_}i}$ . Lastly, if the node pair  $(V_i \longrightarrow V_j) \in \text{BN}_{\text{max\_}f}$  and  $(V_i \longrightarrow V_j) \in \text{Ass}$ , then the node pair is added to the white\_list. Otherwise, it is considered impossible to have a dependency and is added to the black\_list.

The pseudocode of the black and white list algorithm is listed in Algorithm 1.

Since the learned edges in the black and white list may have errors, a penalty term can be set in order to give the model certain error tolerance. A new score function is obtained by incorporating the black and white list as a penalty term. Finally, the initial BN is obtained from the hill climbing search algorithm. The new score function  $F_{\text{new}}(G|D)$  is given as follows:

$$F_{\text{new}}(G|D) = F(G|D) + \omega \Phi((G|D)) = F(G|D) + \omega \left( \sum_{k=1}^{r_w} n_k - \sum_{t=1}^{r_b} n_t \right), \quad (5)$$

where  $F(G|D)$  is the BDeu score,  $\omega$  is the given weight,  $\Phi((G|D))$  is the penalty term,  $r_w$  is the number of elements of the node pair  $(V_i \longrightarrow V_j) \in \text{white\_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$ ,  $r_b$  is the number of elements of the node pair  $(V_i \longrightarrow V_j) \in \text{black\_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$ ,  $n_k$  is the number of samples of the node pair  $(V_i \longrightarrow V_j) \in \text{white\_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$ , and  $n_t$  is the number of samples of the node pair  $(V_i \longrightarrow V_j) \in \text{black\_list} \cap (V_i \longrightarrow V_j) \in G_{PC}$ .

**3.2. Ensemble Learning.** In this paper, we adopt the idea of ensemble learning and use an integrated strategy function  $W$  for calculating the final score of each edge. The integrated strategy function  $W$  is defined as follows:

$$W = \sum_{i=1}^n \sqrt{\frac{N_{\text{node}}}{N_{ki}}} M_{ki} \odot \frac{\sum_{i=1}^n M_{ki}}{n} \log(B), \quad (6)$$

where  $n$  is the number of samples,  $N_{ki}$  is the number of edges of the BN  $G_{\text{max\_}i}$  by combining the initial network using the  $i$  training sample,  $N_{\text{node}}$  is the number of nodes in the dataset,  $M_{ki}$  is the adjacency matrix of the BN  $G_{\text{max\_}i}$

```

(i) Input: Dataset  $D = \{V, D_v\}$ , MFI set Max_freq_item
      strong associated rule set Ass
(ii) Output: white list white_list, black list black_list
(1)   For  $i$  in range(len(Max_freq_item)):
(2)     Data_node = data[Max_freq_item[i]]
(3)     BN_freq_i = K2(Data_node)//BN obtained using K2 algorithm
(4)     For edge in BN_freq_i.edges(): //Loop BN of edges
(5)       If edge in Ass: //Determine whether the edges of a BN are in Ass
(6)         white_list ← edge//Get a white list
(7)     BN_max_f =  $\cup$  BN_freq_i//Aggregate the set of BN structures
(8)     If  $(V_i \rightarrow V_j) \notin \text{BN}_{\text{max}_f} \cap (V_i \rightarrow V_j) \notin \text{Ass}$ :
(9)       black_list ←  $(V_i \rightarrow V_j)$ //Get black list

```

ALGORITHM 1: Black and white list algorithm.

```

Input: Dataset  $D$ , the scoring function  $F_{\text{new}}(G|D)$ , threshold  $\theta$ 
Output: Bayesian network BN
(1)   For  $i$  in range (10): //Loop 10
(2)      $D_i \leftarrow \text{Bootstrap}(D)$ //Bootstrap sampling
(3)      $G_{\text{max}_i} \leftarrow \text{HillClimbSearch}(D_i, F_{\text{new}}(G|D_i))$ //Obtain the initial BN
(4)      $M_{ki} \leftarrow G_{\text{max}_i}$ //Represented by adjacency matrices
(5)      $M_s = M_s + M_{ki}$ //Count the number of times each edge has been learned
(6)      $M_x = \text{sqrt}(N_{\text{node}}/N_{ki} * M_{ki})$ //Assign weights to each edge
(7)      $M_k = M_k + M_x$ //Calculate the total weight value of each edge
(8)      $W_k = M_k \odot (M_s)/(n \log(D))$ //Calculate the integrated strategy function  $W_k$ 
(9)      $W_k = \text{normalize}(W_k)$ //normalized by the maximum-minimum
(10)  For  $p$  in range( $nrow(W_k)$ ):
(11)    For  $q$  in range( $ncol(W_k)$ ):
(12)      If  $W_k[p, q] < \theta$ ://Control the number of edges by the threshold  $\theta$ 
(13)         $W_k[p, q] = 0$ //0 means that no edge exists
(14)      Else:
(15)         $W_k[p, q] = 1$ //1 denotes the existence of directed edges ( $p \rightarrow q$ )
(16)  BN ←  $W_k$ //The adjacency matrix is transformed into a BN

```

ALGORITHM 2: EF-BNSL algorithm.

obtained using the  $i$  training sample,  $B$  is the sample size, and  $\odot$  is the Hadamard product.

The EF-BNSL algorithm flow is as follows: firstly, the original dataset is sampled 10 times using bootstrap, and each of the 10 sample sets is learned to obtain 10 initial BNs. The 10 initial BNs are denoted by the adjacency matrices as  $M_{ki}$ .

Secondly, the 10 adjacency matrices  $M_{ki}$  are integrated by the integration strategy function to obtain the score matrices  $W_k$  for each edge. The score matrix  $W_k$  is normalized by the maximum-minimum.

Thirdly, set the threshold  $\theta$ , if the condition  $W_k[p, q] < \theta$  is satisfied then we can set  $W_k[p, q] = 0$ , if the condition  $W_k[p, q] > \theta$  is satisfied then  $W_k[p, q] = 1$ . We can get the adjacency matrix  $W_k$ .

Lastly, iterate over the adjacency matrix  $W_k$ , if  $W_k[p, q] = 1$ , then add the directed edge ( $p \rightarrow q$ ), and finally get the ensemble learning BN. The number of edges of the BN can be effectively controlled by setting the size of the threshold  $\theta$ . If the threshold  $\theta$  is set too small, there will be many of redundant edges, and if it is set too large, the situation of fewer edges will be very dreadful.

The pseudocode of the EF-BNSL algorithm is listed in Algorithm 2.

## 4. Experiments and Result Analysis

**4.1. Experiment Preparation.** In this paper, we use the python integrated environment Anaconda3 with python version 3.8.5 for programming. Six standard networks of different sizes were downloaded from the Bayesian Network Repository [36]. BN was loaded through the pyAgrum package and the corresponding size dataset was generated, frequent term mining and association rule analysis were performed through the mlxtend package, the sklearn package for Bootstrap sampling, and the pgmpy package for the BN structure learning.

Three networks of different types were chosen for the experiment, including the small size networks (up to 20 nodes), Asia and Sachs, the medium size networks (20–50 nodes), Alarm and Insurance, and the large size networks (50–100 nodes), and Hailfinder and Hepar2. The six standard networks are shown in Table 1.



TABLE 1: The six standard networks used in the experiments.

Standard network	Type	Sample size	Nodes	Edges
Asia	Small size	20000	8	8
Sachs	Small size	20000	11	17
Alarm	Medium size	20000	37	46
Insurance	Medium size	20000	27	52
Hailfinder	Large size	20000	56	66
Hepar2	Large size	20000	70	123

4.2. *Evaluation Indicators.* To verify the effectiveness of the EF-BNSL algorithm, the  $F_1$  score and the Hamming distance (HD) are chosen to evaluate the generated BN and defined as follows:

$$\text{recall} = \frac{TP}{(TP + FN)}, \quad (7)$$

$$\text{precision} = \frac{TP}{(TP + FP)}, \quad (8)$$

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (9)$$

$$HD = FP + FN, \quad (10)$$

where TP is the number of edges in both the current network and the standard network, FP is the number of edges in the current network but not in the standard network, and FN is the number of edges that do not exist in the current network but exists in the standard network, recall is the recall rate, and precision is the precision rate.

The  $F_1$  score is a combination of recall rate and precision rate. A larger  $F_1$  score means that the BN is closer to the real network. The HD is the sum of FP and FN. A smaller HD means a smaller number of erroneous edges and indicates that the learned BN is closer to the standard network.

4.3. *Results.* To verify the performance of the EF-BNSL algorithm, we conducted experiments on six networks. The following is an example of the Asia network. The Asia network is a small BN, which is used as a fictional medical example. The network asks whether a patient has tuberculosis, lung cancer, or bronchitis and consists of eight nodes and eight edges. Each random variable is discrete and can take two states. The original Asia standard network is shown in Figure 3.

The experimental process is divided into three stages. The first stage of the experiment is to perform frequent item mining and association rule analysis on the sample set. Firstly, we load the BIF file to generate a dataset with a sample size of 20000, and then we perform sampling by using bootstrap with a sample size of 1000. The sample set variables are transformed into Boolean variables. The minimum support and minimum confidence were set to 0.75 and 0.95 [32], respectively. The minimum support and minimum confidence can control the number of edges in the black and white list, which can be selected on the basis of the dataset distribution characteristics.

By using the Apriori algorithm with the given minimum support, the MFIs and association rules can be obtained. An association rule with a confidence level greater than the given minimum confidence is a strong association rule.

The MFI set Max\_freq\_item is {lung, tub, asia, xray, either}, and the strong association rule results are shown in Table 2.

On the basis of obtaining the MFI set and strong association rule results, the black and white list can be obtained from the proposed Algorithm 1 above. The black and white list of results are shown in Table 3.

In this experiment, the edges (either, xray) and (lung, either) in the white list are in the original Asia standard network, and the accuracy of the white list reaches 100%. On the other hand, all the edges in the black list are judged correctly except for the edge (tub, either) error, and the accuracy rate of the black list reaches 85.7%. To give the model some fault tolerance, so we do not use the list directly as the black and white list of BN. Instead, they are given certain weight scores and added to the scoring function as penalty terms. Also, we find that the number of edges in the white list is smaller than the number of edges in the black list in the experiment. Different weights can be further set for the edges of the black and white list in future studies.

The second stage of the experiment is to update the scoring function by the black and white list and get the initial network. Firstly, according to (3), the black and white list is added to the scoring function as a penalty term. The penalty term weight  $\omega$  was set to 0.5 [32]. If an edge in the black list is contained in BN, the new scoring function becomes smaller. Conversely, if the edges in the white list are contained in BN, the new scoring function becomes larger. The network of the maximum score is obtained from the hill climbing search algorithm, which is the initial BN.

As shown in Figure 4(a), the BN is a network obtained by direct structure learning using BDeu score, which has 14 edges. Figure 4(b) shows the CPT of this BN structure. For example, the lung node has only tub node as its parent node, so there are four possible outcomes for the CPT of the lung node.  $P(\text{lung} = 0 \text{tub} = 0) = 0$ ,  $P(\text{lung} = 1 \text{tub} = 0) = 1$ ,  $P(\text{lung} = 0 \text{tub} = 1) = 0.0633$ , and  $P(\text{lung} = 1 \text{tub} = 1) = 0.9367$ . As shown in Figure 5(a), the BN is a network obtained by structure learning using an updated scoring function, with has 12 edges. Figure 5(b) shows the CPT of this BN structure.

By comparing with the original Asia standard network, we can able to calculate the  $F_1$  score and HD for these two BNs using equations (7)–(10). The  $F_1$  score and HD of the BN in Figure 4(a) is 0.45 and 9, respectively. The  $F_1$  score

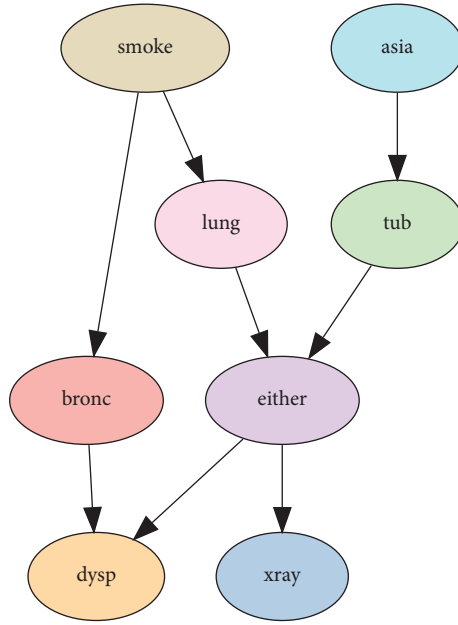


FIGURE 3: The original Asia standard network.

TABLE 2: The strong association rule results of the Asia network.

Association rule	Support	Confidence
(Either, xray)	0.8880	0.9579
(tub, Asia)	0.9760	0.9869
(Xray, Asia)	0.8800	0.9888
(Asia, tub)	0.9760	0.9889
(Either, Asia)	0.9170	0.9892
(Lung, Asia)	0.9260	0.9893
(Lung, either)	0.9270	0.9904
(Lung, tub)	0.9270	0.9904
(Xray, either)	0.8880	0.9978
(Xray, lung)	0.8880	0.9978
(Either, lung)	0.9270	1.0000
(Either, tub)	0.9270	1.0000
(Xray, tub)	0.8900	1.0000

TABLE 3: Asia network’s black list and white list.

Black list	White list
(Lung, xray)	(Either, xray)
(Asia, xray)	(Lung, either)
(Asia, lung)	
(Tub, lung)	
(Tub, either)	
(Asia, either)	
(Tub, xray)	

and HD of the BN in Figure 5(a) is 0.5 and 7, respectively. Improved scoring functions for BN learning outperformed the BN obtained by direct learning using the BDeu score.

The third stage of the experiment is to use Algorithm 2 to obtain the final ensemble learning BN. Firstly, bootstrap was used to sample the dataset for 10 times, and the 10 sample sets were, respectively, learned to obtain 10 initial BNs.

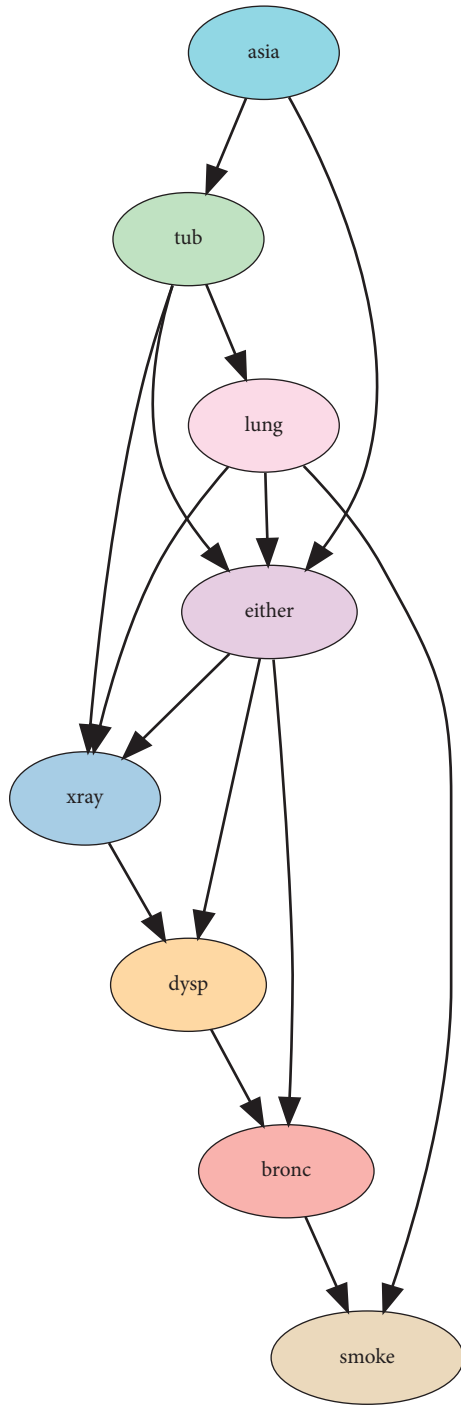
Secondly, the 10 initial BNs were denoted by the adjacency matrices as  $M_{ki}$ , and then the 10 adjacency matrices

$M_{ki}$  were integrated by the integration strategy function to obtain the score matrix  $W_k$ . Meanwhile, the score matrix  $W_k$  was normalized by the maximum-minimum.

Thirdly, set the threshold  $\theta = 0.4$  [33], if the condition  $W_k[p, q] < 0.4$  was satisfied, then we can set  $W_k[p, q] = 0$ ; if the condition  $W_k[p, q] > 0.4$  was satisfied then  $W_k[p, q] = 1$ . We can get the adjacency matrix  $W_k$ .

Lastly, iterate over the adjacency matrix  $W_k$ . If the condition  $W_k[p, q] = 1$ , we can add the directed edge ( $p \rightarrow q$ ) in the network. The final ensemble learning BN was obtained.

The final ensemble learning BN is shown in Figure 6(a), which has 10 edges. Figure 6(b) shows the CPT of this BN structure. For example, the xray node has only either node as its parent node, so there are four possible outcomes for the CPT of the xray node.  $P(xray = 0either = 0) = 0.9706$ ,  $P(xray = 1either = 0) = 0.0294$ ,  $P(xray = 0either = 1) = 0.0483$ , and  $P(xray = 1either = 1) = 0.9517$ . If we know that the value of node either is 0, then we can deduce that the probability that node Xray is equal to 0 is 0.9706. The  $F_1$



(a)

FIGURE 4: Continued.

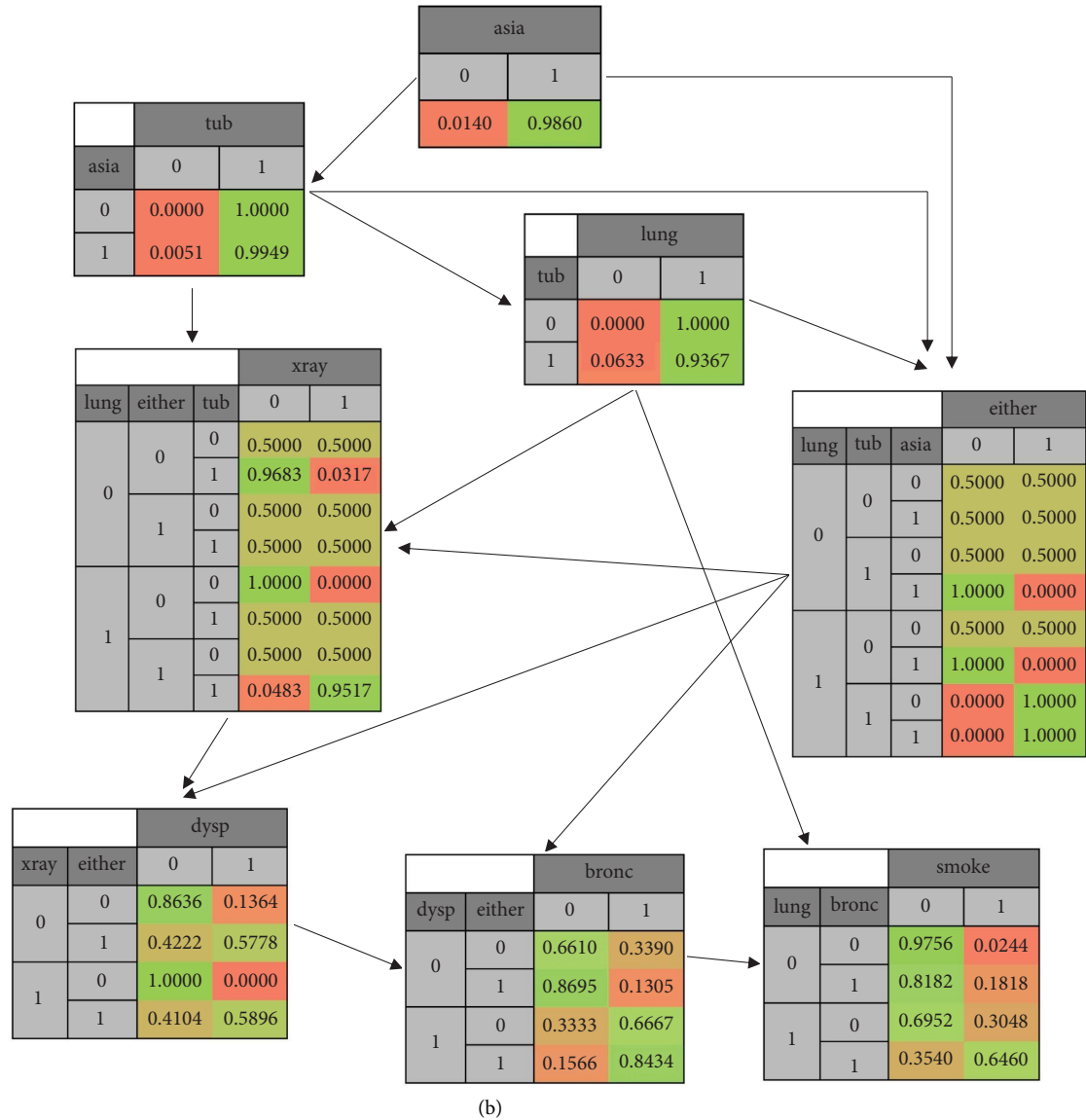


FIGURE 4: (a) The BN of using the BDeu score. (b) The CPT of using the BDeu score.

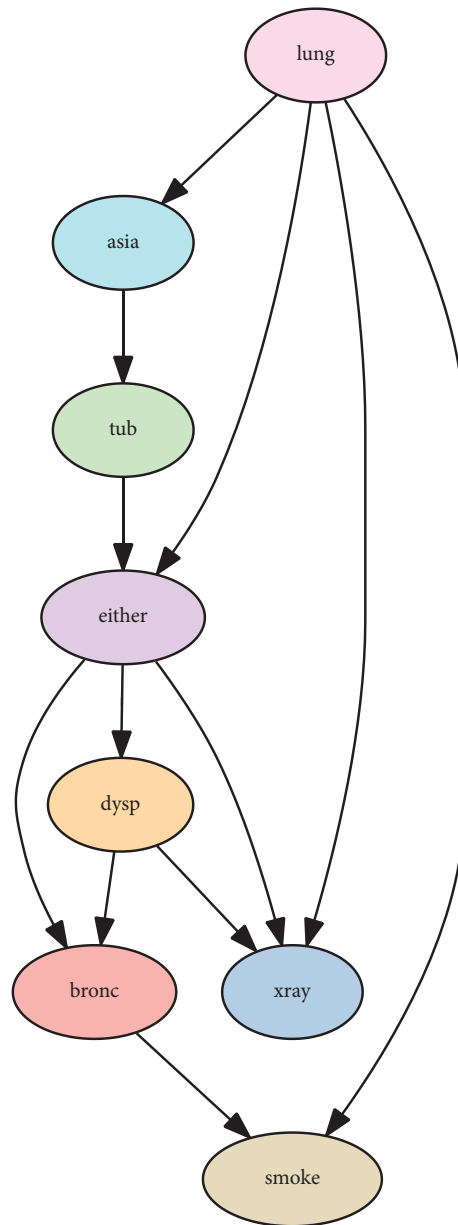
score and HD of the final ensemble learning BN is 0.45 and 9, respectively. These two evaluation indicators show that the EF-BNSL algorithm is better than the BDeu score.

Because of the randomness of the data and the uncertainty of the search process, BN structure learning results can be varied. Therefore, to improve the reliability of the experimental results, the above experimental procedure was repeated 10 times, and the results obtained by calculating the HD and  $F_1$  are shown in Table 4. Finally, the average of HD and  $F_1$  was calculated for 10 times of results.

It is obvious from Table 4 that the BN obtained using the EF-BNSL algorithm is larger in  $F_1$  score than the BN obtained directly using the BDeu score. The BN obtained using the EF-BNSL algorithm is smaller in HD than the BN obtained directly using the BDeu score. The EF-BNSL algorithm on the Asia network outperforms the algorithm that uses the BDeu scoring function directly and is closer to the original standard Asia network.

Through the previous experimental method for three different types of standard networks, including small networks (Asia and Sachs), medium networks (Alarm and Insurance), and large networks (Hailfinder and Hepar2), the comparison results of six standard networks are shown in Table 5.

From the experimental results, it can be seen that the EF-BNSL algorithm proposed in this paper achieves better learning performance than the algorithm that directly using the BDeu score. The EF-BNSL algorithm does not show a greater advantage when the samples are small. Because of the lack of information in small sample data, it is difficult to mine more information of association rules. When the sample size is larger, the EF-BNSL algorithm has a better learning effect. More association rules can be mined and the black and white list can be more accurate. By using the EF-BNSL algorithm, there is a significant performance improvement in Asia and Sachs for the small size networks, and



(a)

FIGURE 5: Continued.

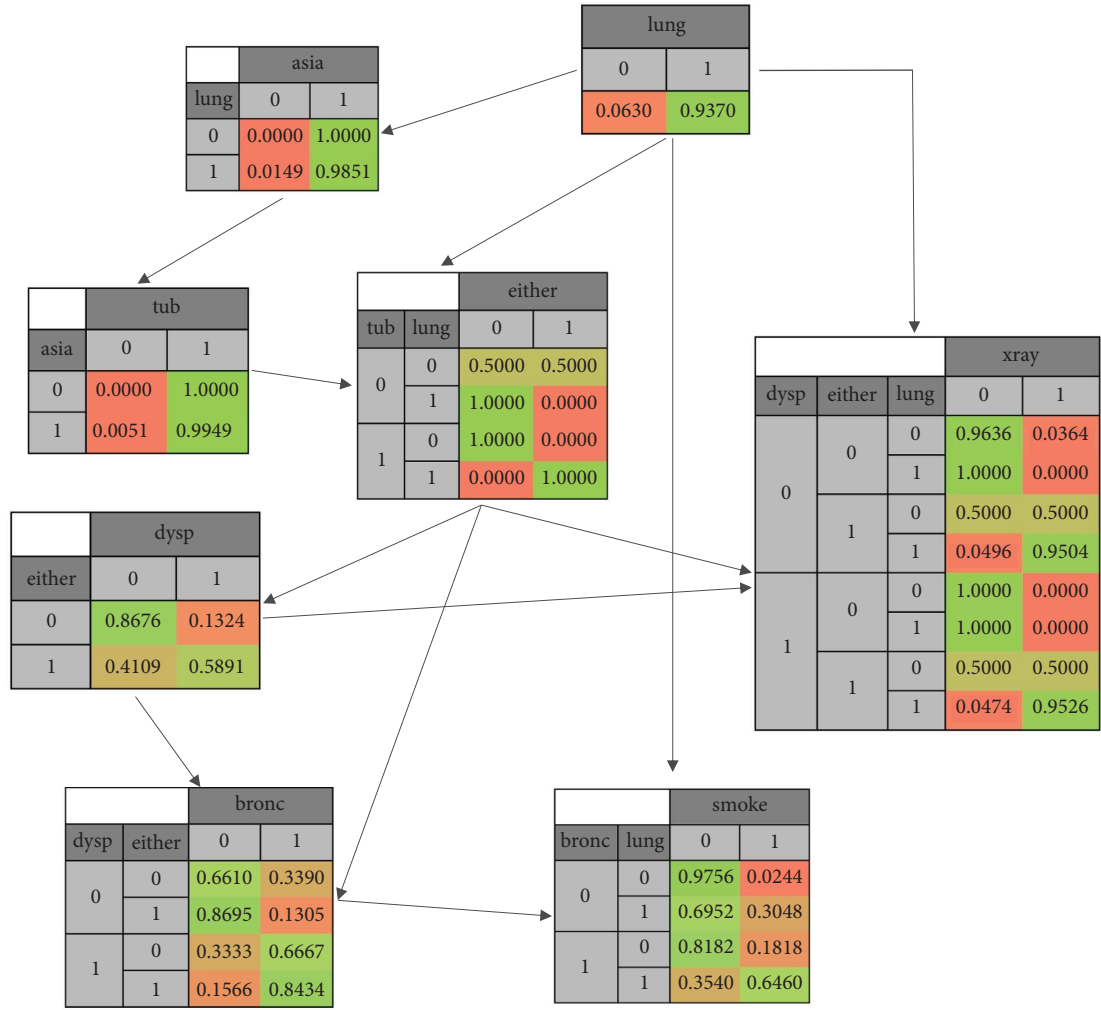


FIGURE 5: (a) The BN of using the improved scoring function. (b) The CPT of using the improved scoring function.

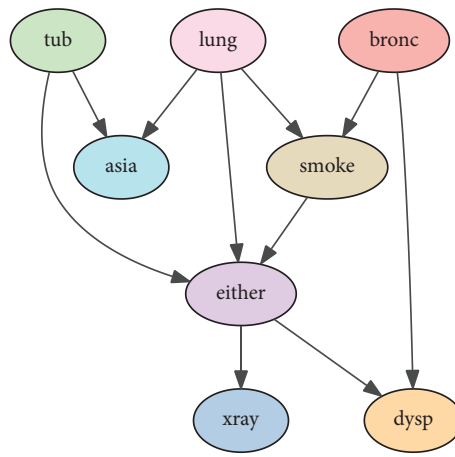


FIGURE 6: Continued.

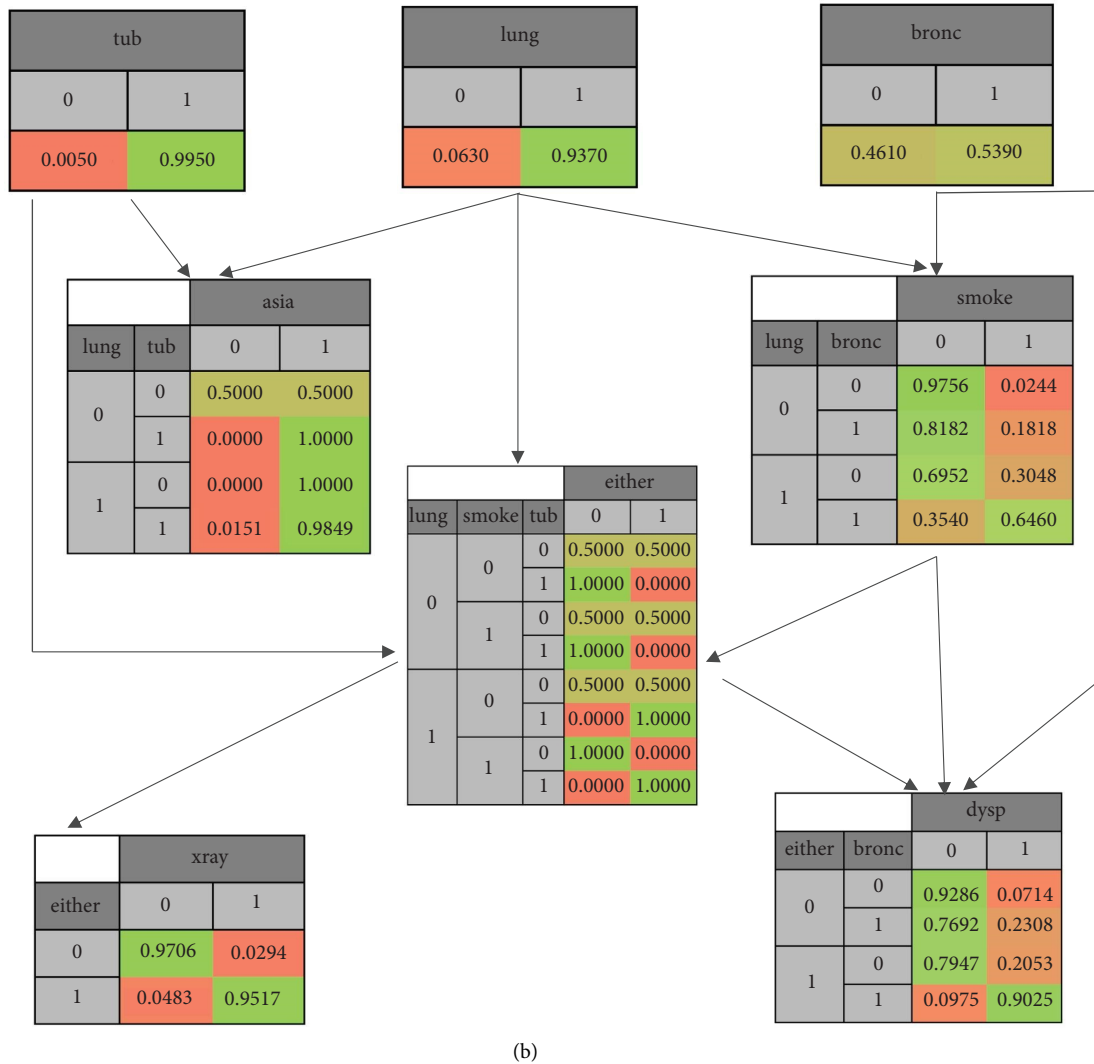


FIGURE 6: (a) Ensemble learning BN. (b) The CPT of the ensemble learning BN.

TABLE 4: Comparison of 10 results from the Asia network.

EF-BNSL		BDeu		
$F_1$	HD	$F_1$	HD	
0.47	4	0.28	15	
0.51	5	0.26	10	
0.48	6	0.24	16	
0.58	4	0.45	11	
0.48	5	0.33	16	
0.52	4	0.34	14	
0.49	5	0.36	14	
0.53	4	0.38	8	
0.46	6	0.38	9	
0.52	5	0.17	14	

no obvious progress in Alarm and Insurance networks. In the experiment, the number of edges in the white list is much smaller than the number of edges in the black list as the number of nodes increases, which leads to a much reduced

scoring function. In future research, different weights can be further set for the edges in the black and white list. In addition, when there are more network nodes, the network search space increases exponentially. To balance the search

TABLE 5: The comparison results of six standard networks.

Sample size	Asia			Sachs			Alarm			Insurance			Hailfinder			Hepar2							
	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD	EF-BNSL	BDeu	HD					
	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$	$F_1$					
1000	0.51	0.42	9.8	0.41	4.8	0.36	8.3	0.47	49.4	0.48	51.7	0.31	58.5	0.31	63.2	0.47	57.2	0.46	62.5	0.47	154.8	0.36	158.6
2000	0.53	0.43	9.2	0.49	5	0.42	7.2	0.51	45.5	0.50	47.6	0.36	53.2	0.46	55.5	0.51	61.4	0.53	65.2	0.51	128.9	0.45	130.2
5000	0.58	0.54	8.8	0.52	6	0.48	6.1	0.48	38.7	0.53	44.1	0.43	45.3	0.54	44.7	0.64	55.1	0.61	59.8	0.48	101.3	0.55	103.2



time and memory usage, the experiment adjusted down the weight of the black and white list penalty terms for large networks, which also affects the experimental results.

## 5. Conclusion and Future Research

In this paper, we combined the ideas of frequent item mining and ensemble learning into BN structure learning and proposed the EF-BNSL algorithm. We conducted experiments on six BNs of three types such as small, medium, and large. We used two evaluation metrics,  $F_1$  score and HD. The results show that the EF-BNSL algorithm can effectively improve  $F_1$  score, reduce HD, and learn the network structure that is closer to the real network.

In the experiment, as the number of nodes increases, there are more edges in the black list than in the white list. In fact, it is possible that the edges in the white list would be more important, so assigning different weights to the black and white list will be the next highlight of future research. In addition, when there are more network nodes, the network search space increases exponentially. Future research could consider using distributed computing to improve efficiency. Ensemble learning of different network structure learning algorithms, and making full use of the advantages of different algorithms for BN structure learning will be the focus of the future research.

## Data Availability

Six standard networks can be downloaded from the Bayesian Network Repository (<https://www.bnlearn.com/bnrepository/>).

## Conflicts of Interest

The authors declare that have no conflicts of interest.

## Authors' Contributions

Guoxin Cao contributed to the conception or design of the work. Guoxin Cao and Haomin Zhang contributed to the acquisition, analysis, or interpretation of the data and drafted the manuscript. All authors reviewed the manuscript.

## Acknowledgments

This project was supported by the National Natural Science Foundation of China (61763008, 71762008, and 62166015) and the Guangxi Science and Technology Planning Project (2018GXNSFAA294131 and 2018GXNSFAA050005).

## References

- [1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Massachusetts, USA, 1988.
- [2] C. Liang, M. Ghazel, O. Cazier, and E. M. El-Kourssi, "Risk analysis on level crossings using a causal Bayesian network based approach," *Transportation Research Procedia*, vol. 25, pp. 2167–2181, 2017.
- [3] A. Gharaei, S. A. Hoseini Shekarabi, and M. Karimi, "Optimal lot-sizing of an integrated EPQ model with partial backorders and re-workable products: an outer approximation," *International Journal of Systems Science: Operations & Logistics*, pp. 1–17, 2021.
- [4] Y. Liu and S. Jin, "Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets," *International Journal of Advanced Manufacturing Technology*, vol. 65, no. 9–12, pp. 1229–1237, 2013.
- [5] R. L. C. Souza, A. Ghasemi, A. Saif, and A. Gharaei, "Robust job-shop scheduling under deterministic and stochastic unavailability constraints due to preventive and corrective maintenance," *Computers & Industrial Engineering*, vol. 168, Article ID 108130, 2022.
- [6] A. Gharaei, A. Amjadian, and A. Shavandi, "An integrated reliable four-level supply chain with multi-stage products under shortage and stochastic constraints," *International Journal of Systems Science: Operations & Logistics*, pp. 1–22, 2021.
- [7] A. Amjadian and A. Gharaei, "An integrated reliable five-level closed-loop supply chain with multi-stage products under quality control and green policies: generalised outer approximation with exact penalty," *International Journal of Systems Science: Operations & Logistics*, vol. 9, no. 3, pp. 429–449, 2022.
- [8] A. A. Taleizadeh, A. Z. Safaei, and A. Bhattacharya, "Online peer-to-peer lending platform and supply chain finance decisions and strategies," *Annals of Operations Research*, vol. 315, pp. 1–31, 2022.
- [9] B. Nistal-Nuño, "Tutorial of the probabilistic methods Bayesian networks and influence diagrams applied to medicine," *Journal of Evidence-Based Medicine*, vol. 11, no. 2, pp. 112–124, 2018.
- [10] B. Drury, J. Valverde-Rebaza, M. F. Moura, and A de Andrade Lopes, "A survey of the applications of Bayesian networks in agriculture," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 29–42, 2017.
- [11] A. Gharaei, C. Diallo, and U. Venkatadri, "Optimal economic growing quantity for reproductive farmed animals under profitable by-products and carbon emission considerations," *Journal of Cleaner Production*, vol. 374, Article ID 133849, 2022.
- [12] A. Gharaei and E. Almehdawe, "Optimal sustainable order quantities for growing items," *Journal of Cleaner Production*, vol. 307, Article ID 127216, 2021.
- [13] A. Gharaei, A. Amjadian, A. Amjadian et al., "An integrated lot-sizing policy for the inventory management of constrained multi-level supply chains: null-space method," *International Journal of Systems Science: Operations & Logistics*, pp. 1–14, 2022.
- [14] D. M. Chickering, "Learning Bayesian networks is NP-complete," *Learning from Data*, pp. 121–130, Springer, New York, NY, UAS, 1996.
- [15] P. Spirtes and C. Meek, "Learning Bayesian networks with discrete variables from data," *KDD*, vol. 1, pp. 294–299, 1995.
- [16] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, Prediction, and Search*, MIT press, Cambridge, Massachusetts, USA, 2000.
- [17] X. Qi, X. Fan, Y. Gao, and Y. Liu, "Learning Bayesian network structures using weakest mutual-information-first strategy," *International Journal of Approximate Reasoning*, vol. 114, pp. 84–98, 2019.

- [18] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [19] C. Lee and P. Beek, "Metaheuristics for score-and-search Bayesian network structure learning," in *Proceedings of the Canadian Conference on Artificial Intelligence*, pp. 129–141, Springer, Edmonton, AB, Canada, May 2017.
- [20] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.
- [21] H. Akaike, "Information theory and an extension of the maximum likelihood principle," *Selected Papers of Hirotugu Akaike*, pp. 199–213, Springer, New York, NY, USA, 1998.
- [22] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978.
- [23] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.
- [24] J. R. Alcobé, "Incremental hill-climbing search applied to Bayesian network structure learning," in *Proceedings of the First International Workshop on Knowledge Discovery in Data Streams*, Hammamet, Tunisia, April 2004.
- [25] L. Zhang and J. Zhang, "Structure learning of BN based on improved genetic algorithm," *Computer Systems and Applications*, vol. 20, no. 9, pp. 68–72, 2011.
- [26] M. Singh and M. Valtorta, "Construction of Bayesian network structures from data: a brief survey and an efficient algorithm," *International Journal of Approximate Reasoning*, vol. 12, no. 2, pp. 111–131, 1995.
- [27] J. I. Alonso-Barba, L. delaOssa, J. A. Gámez, and J. M. Puerta, "Scaling up the Greedy Equivalence Search algorithm by constraining the search space of equivalence classes," *International Journal of Approximate Reasoning*, vol. 54, no. 4, pp. 429–451, 2013.
- [28] R. Eggeling, J. Viinikka, and A. Vuoksenmaa, "On structure priors for learning Bayesian networks," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1687–1695, PMLR, Okinawa, Japan, April 2019.
- [29] Z. Wang, X. Gao, Y. Yang, X. Tan, and D. Chen, "Learning Bayesian networks based on order graph with ancestral constraints," *Knowledge-Based Systems*, vol. 211, Article ID 106515, 2021.
- [30] H. Xiao, K. Yu, and H. Wang, "The method of learning Bayesian networks combining association rules with knowledge," *Microelectronics & Computer*, vol. 25, no. 12, p. 7072, 2008.
- [31] B. Sun, Y. Zhou, J. Wang, and W. Zhang, "A new PC-PSO algorithm for Bayesian network structure learning with structure priors," *Expert Systems with Applications*, vol. 184, Article ID 115237, 2021.
- [32] X. Li, S. Zhou, J. Zhang, Z. Zhou, and Q. Xiong, "BNSL-FIM: Bayesian network structure learning algorithm based on frequent item mining," *Animals: An Open Access Journal from MDPI*, vol. 11, no. 12, p. 3475, 2021.
- [33] S. Wang and B. Qin, "Bayesian network structure learning by ensemble learning and feedback strategy," *Journal of Computers*, vol. 44, no. 6, pp. 1051–1063, 2021.
- [34] H. Baradaran Rezaei, A. Amjadian, and M. V. Sebt, "An ensemble method of the machine learning to prognosticate the gastric cancer," *Annals of Operations Research*, pp. 1–42, 2022.
- [35] R. Askari, M. V. Sebt, and A. Amjadian, "A multi-product EPQ model for defective production and inspection with single machine, and operational constraints: stochastic programming approach," in *Proceedings of the International conference on logistics and supply chain management*, pp. 161–193, Springer, Tehran, Iran, December 2020.
- [36] M. Scutari, "Bayesian network repository," 2022, <https://www.bnlearn.com/bnrepository/>.

## Research Article

# A New Collaborative Filtering Algorithm Integrating Time and Multisimilarity

Qin Liu 

*Department of Information Science and Technology, East China University of Political Science and Law, Shanghai 200042, China*

Correspondence should be addressed to Qin Liu; [liuqin@ecupl.edu.cn](mailto:liuqin@ecupl.edu.cn)

Received 30 April 2022; Revised 27 May 2022; Accepted 10 August 2022; Published 30 August 2022

Academic Editor: Chaoqun Duan

Copyright © 2022 Qin Liu. 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.

Aiming at the problem of low recommendation accuracy of existing recommendation algorithms, an algorithm integrating time factors and multisimilarity is proposed to improve the impact of long-term data, user attention, and project popularity on the recommendation algorithm and the similarity of user attributes is introduced to improve the problem of cold start to a certain extent. Considering that the longer the time, the less likely it is to be selected again, time is introduced into the algorithm as a weight factor. When the behavior occurs, i.e., interest in the project, so as to judge the similarity between users, not just the score value, we normalize the popularity to avoid misjudgment of high scoring and popular items. Because new users do not have past score records, the problem of cold start can be solved by calculating the similarity of user attributes. Through the comparative experiment on Movielens100K dataset and Epinions dataset, the results show that the algorithm can improve the accuracy of recommendation and give users a better recommendation effect.

## 1. Introduction

In work and life, people rely more and more on computers and mobile devices. They face a large amount of data every day. At the same time, each user will also produce a large amount of data. How to select the information they need from a large amount of data depends on various factors; many users will choose the recommendation given by others, and the recommendation system also came into being. The recommendation system is applied in more and more fields, including commodity recommendation, learning resource recommendation, and case recommendation in intelligent justice [1–3]. Recommendation algorithms can be roughly divided into knowledge-based recommendation, collaborative filtering-based recommendation, and content-based recommendation. Among them, the recommendation algorithm based on collaborative filtering has always been the most classic and popular recommendation algorithm. Collaborative filtering recommendation algorithms are divided into two categories: collaborative filtering based on selection items and collaborative filtering based on participating users. They both predict preferences and generate

recommendations according to users' historical scoring data. The collaborative filtering recommendation algorithm has many problems, such as unreasonable similarity calculation, cold start, sparse data, and so on, which may lead to the decline of recommendation quality.

From different perspectives, many studies fully tap the potential information so that the collaborative filtering recommendation algorithm can be used to a wider range of scenarios. Literature [4] proposed a method to calculate the similarity of items utilized score and structural similarity, which can effectively work out the problem of poor effect of cold start. Reference [5] proposed a method of weighting items to obtain a user similarity calculation method considering the similarity weight of items. Literature [6] projected a method to calculate the similarity of items by combining the similarity of scores and the similarity of item attributes. Reference [7] proposed a similarity calculation method between users based on cosine similarity and fusion of relative differences in scores. In the nonpreference scoring system, the prediction error can be reduced and the recommendation quality can be improved. Literature [8] proposed this algorithm to cluster tags and generate topic tag

clusters, calculate the correlation between items and topics according to item labeling, and generate item  $g$  topic correlation matrix. At the same time, it is combined with item  $g$  score matrix to calculate the similarity between items, and collaborative filtering is used to complete the score prediction of target items, so as to realize personalized recommendation. The above methods are to enhance the accuracy of similarity calculation by fusing scores and other indicators. However, they did not consider the impact of the environment on user interest. In fact, there are many environmental factors affecting user interest and interest changes, such as time, weather, place, and mood; among them, the time factor is the main factor affecting user interest and interest changes. Therefore, this paper declares a collaborative filtering recommendation algorithm on the ground of time weight and multisimilarity. Multisimilarity includes the similarity of user attributes, user attention, and project popularity. The similarity of user attributes alleviates the problem of cold start to a certain extent, User attention and project popularity can reduce some significant “unfairness” in the existing scores from the perspective of users and projects, respectively.

## 2. Related Work

**2.1. User-Based Collaborative Filtering Algorithm.** The user-based collaborative filtering algorithm will involve user  $U$ , item  $I$ , and user rating matrix for items  $R_{ui}$ . The chief steps of the traditional user-based collaborative filtering algorithm are as follows: first, build the user’s score matrix, and mark the unknown score as 0, as shown in Table 1; the second step is to estimate the similarity between users by using the scoring matrix; in the third step, according to the similarity matrix, some users who are the most alike to the target user, that is, the users with the highest similarity are found to form a nearest neighbor set; the fourth step is to predict the score of the target user according to the score of the neighbor user and generate a recommendation set. Among them, the calculation of similarity between users is the important step of the algorithm.

**2.2. Calculation of User Similarity.** The traditional user similarity calculation formulas mainly include COS (cosine similarity), PCC (Pearson correlation coefficient), and ACOS (adjusted cosine similarity) [1, 8–11]. Formula (1) means the cosine similarity between user  $u$  and user  $v$ , Formula (2) represents the Pearson correlation coefficient, and Formula (3) represents the adjusted cosine similarity. Table 2 shows the meanings of symbols used in the formula.

$$\text{sim}(u, v)_{\text{COS}} = \frac{\sum_{c \in I_{u,v}} r_{u,c} r_{v,c}}{\sqrt{\sum_{c \in I_u} r_{u,c}^2} \sqrt{\sum_{c \in I_v} r_{v,c}^2}}, \quad (1)$$

$$\text{sim}(u, v)_{\text{PCC}} = \frac{\sum_{c \in I_{u,v}} (r_{u,c} - \bar{r}_u)(r_{v,c} - \bar{r}_v)}{\sqrt{\sum_{c \in I_{u,v}} (r_{u,c} - \bar{r}_u)^2} \sqrt{\sum_{c \in I_{u,v}} (r_{v,c} - \bar{r}_v)^2}}, \quad (2)$$

TABLE 1: User item scoring.

	$I_1$	$I_2$	$\dots$	$I_n$
$U_1$	$r_{1,1}$	$r_{1,2}$	$\dots$	$r_{1,n}$
$U_2$	$r_{2,1}$	$r_{2,2}$	$\dots$	$r_{2,n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$U_m$	$r_{m,1}$	$r_{m,2}$	$\dots$	$r_{m,n}$

TABLE 2: Meaning of symbols in formulas.

Symbol	Meaning
$r_{u,c}$	User $u$ ’s rating of item $C$
$I_u$	Collection of items with user $u$ rated scores
$I_{u,v}$	A collection of items that users $u$ and $v$ have jointly rated
$\bar{r}_u$	Average value of user $U$ ’s rating on all items

$$\text{sim}(u, v)_{\text{ACOS}} = \frac{\sum_{c \in I_{u,v}} (r_{u,c} - \bar{r}_u)(r_{v,c} - \bar{r}_v)}{\sqrt{\sum_{c \in I_u} (r_{u,c} - \bar{r}_u)^2} \sqrt{\sum_{c \in I_v} (r_{v,c} - \bar{r}_v)^2}}. \quad (3)$$

Formulas (2) and (3) take into account users’ personal scoring habits. Some users prefer to give high scores and some users give low scores. Therefore, Formulas (2) and (3) calculates users’ similarity after subtracting users’ average scores, which can improve the recommendation quality. However, for some nonuser preference scoring systems, the scores are given by specific standards, and Formulas (2) and (3) are difficult to be compared for the differences between users.

## 3. Collaborative Filtering Recommendation Algorithm Combining Time and Multisimilarity

**3.1. Similarity Calculation considering Time Factor.** At present, the research on the time factor of recommendation system mainly has three aspects [12]: first, obtain the information related to users through modeling so as to study the relationship between user interest and time and provide reasonable recommendations for users, but it is not easy to obtain the characteristics of user information by the modeling method, and it is not universal. The second aspect is to set the effective time limit for the score. Only the items within the effective time range will calculate their similarity; otherwise, it will be ignored, which greatly decreases calculation difficulty. However, it only considers the impact of recent data and denies the impact of long-term data on recommendation. The third aspect is to use the time weighting method and introduce the forgetting curve to obtain the time attenuation function.

In this paper, the time weight function [12] is introduced. As shown in Formula (4),  $t_{u,c}$  represents the degree of interest of user  $u$  in item  $c$  at a certain time, which can be expressed by score  $r_{u,c}$ .  $w(t_{u,c})$  is a monotonically increasing function, whose value increases with the increase in time  $t$ , but it will not exceed the upper limit 1. Considering that the longer the time is, the less likely it is to be selected again, time is introduced into the algorithm as a weight factor. We

enhance the similarity formula based on the modified cosine similarity (Formula (3)). After introducing the time weight, the specific calculation formula is shown as

$$w(t_{u,c}) = \frac{1}{1 + e^{(-t_{u,c})}}, \quad (4)$$

$$\text{sim}(u, v)_w = \frac{\sum_{c \in I_{uv}} (r_{u,c} * w(t_{u,c}) - \bar{r}_u)(r_{v,c} * w(t_{v,c}) - \bar{r}_v)}{\sqrt{\sum_{c \in I_u} (r_{u,c} * w(t_{u,c}) - \bar{r}_u)^2} \sqrt{\sum_{c \in I_v} (r_{v,c} * w(t_{v,c}) - \bar{r}_v)^2}} \quad (5)$$

**3.2. Similarity Calculation of User Attributes.** Because new users have no past score records, it is difficult for the actual program to predict and recommend products to these users. Generally, they will choose to recommend popular products to new users. In fact, the system can use some attribute information provided during user registration for recommendations so as to clear up the problem of cold start to a certain degree extent [13].

Extract the necessary information for new user registration, including gender, occupation, age, address, and income, in this paper, and the attributes are divided into tree (hierarchical) attributes and linear attributes. Linear attribute refers to the linear juxtaposition relationship between attribute values that does not distinguish between front and rear positions. For example, gender attribute, age, and income fields can be segmented and identified. For example, age can be divided into five categories: less than 18 years old, coded as one, 19–28 years old, coded as two, 29–40 years old, coded as three, 41–65 years old, coded as four, and over 66 years old, coded as five. Tree attribute refers to the parent-child relationship between attribute values in the hierarchical or tree structure. For example, there are many types of traditional occupations, such as occupation and address. Each occupation corresponds to at least one main industry. According to the industry classification of the national economy, the industry is divided into 20 categories, including the major category, medium category, and small category. Considering the potential semantic relationship between values, a classification semantic hierarchy tree based on domain knowledge is constructed. The leaf nodes of the tree are different attribute values, and the similarity between attribute values depends on their position in the tree structure.

The formula for calculating the similarity of linear attributes is

$$\text{sim}(u, v) = \frac{1}{1 + |u - v|}. \quad (6)$$

The formula for calculating the similarity of tree attributes is

$$\text{sim}(u, v) = 1 - \frac{L(u, v)}{H}, \quad (7)$$

where  $H$  is the height of the hierarchical tree and  $L(u, v)$  is the longest path length from the attribute node in the tree to the common node.

**3.3. User Attention Similarity.** The traditional similarity calculation is for the user's rating of the project. If the score similarity is high, the user similarity is high, and if the score similarity is low, the user similarity is low. In fact, this method ignores the behavior of the project, which can explain the similarity between users to a certain extent. Because usually only the items that users are interested in will produce behavior, and the score cannot completely explain whether users are interested in the field of the project [7]. For example, when a user buys an AI book, he cannot understand the content of the book due to his own knowledge structure, so he gives a low score, but this does not mean that he is not interested in AI books. Therefore, we can think that the behavior is interested in the project, so as to judge the similarity between users, not just the score value. The calculation formula of user attention similarity is

$$\text{sim}(u, v) = \frac{|I_{u,v}|}{|I_u| + |I_v| - |I_{u,v}|}, \quad (8)$$

where  $|I_{u,v}|$  is the number of common scoring items of users  $u$  and  $V$  and  $|I_u||I_v|$  is the number of scored items of users  $u$  and  $V$ .

**3.4. Project Popularity.** Project popularity is a factor that will be considered in many recommendation algorithms, because generally goods with high popularity are easier to be found by users, which will have greater influence on users than other goods. The existing algorithms generally set the weight of popularity to reduce the role of popular items in similarity calculation and final recommendation. However, it is found that this method is unfair to high scoring and popular projects. Therefore, before calculating the weight, the popularity can be normalized [13], as shown in formula (9).  $IP_i$  is the popularity of item  $I$ , which can be expressed by the frequency of item  $I$  in the data. Calculate the weight of each item according to the normalized popularity as

$$\text{nor}IP_i = \frac{IP_i - \min IP}{\max IP - \min IP}, \quad (9)$$

$$w_i = w_0 * \frac{\text{nor}IP_i}{\sum_{j=1}^N \text{nor}IP_j}. \quad (10)$$

**3.5. Improved Algorithm.** Through the above analysis, we propose a collaborative filtering recommendation algorithm (COS-MS) integrating time and multisimilarity. The basic steps are in Algorithm 1.

## 4. Experimental Analysis

**4.1. Dataset.** In this paper, Movielens100K dataset and Epinions dataset are used for experiments. The reason why the two datasets are used is that the sparsity of the two

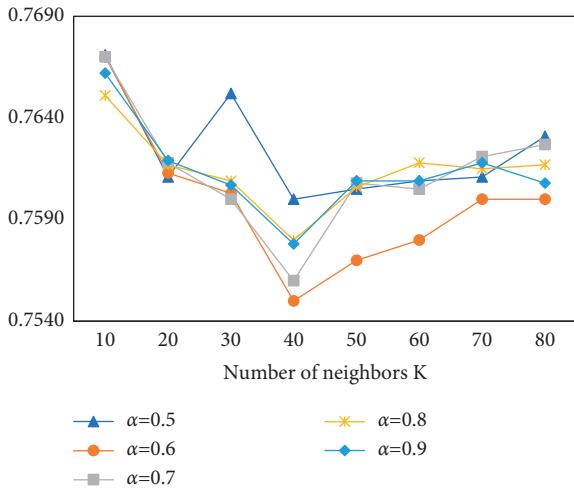


FIGURE 1: MAE changes with  $\alpha$  value.

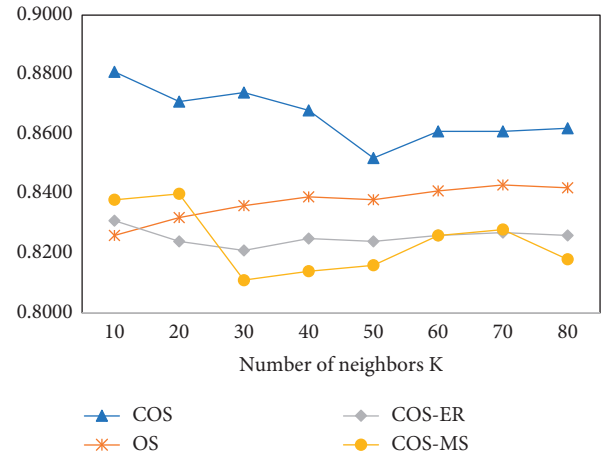


FIGURE 4: Comparison of MAE values of different algorithms based on Epinions dataset.

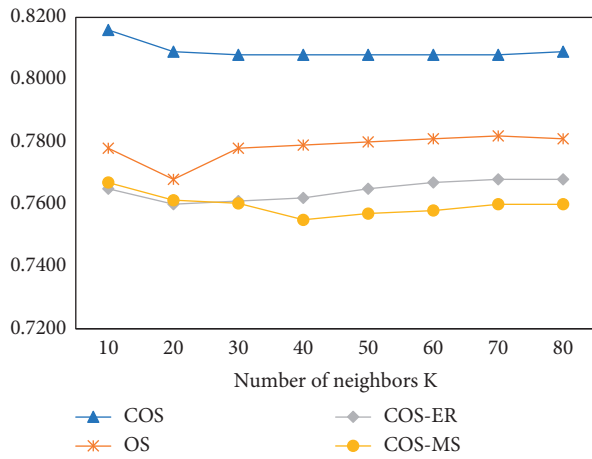


FIGURE 2: Comparison of MAE values of different algorithms based on Movielens100K dataset.

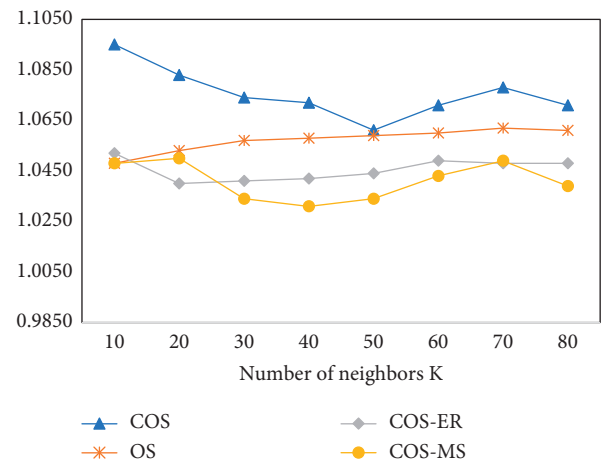


FIGURE 5: Comparison of RMSE values of different algorithms based on Epinions dataset.

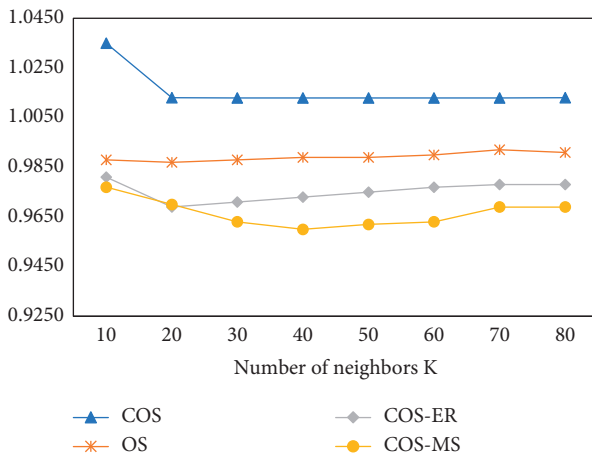


FIGURE 3: Comparison of RMSE values of different algorithms based on Movielens100K dataset.

datasets is different, which can better verify the universality of the algorithm.

Movielens100K dataset includes users' ratings of movies. 943 users rated 1682 movies, with a score of 1–5. Each user scored at least 20 movies, with a data density of about 6.3%. In addition, it also includes user attribute information and movie metadata information.

The Epinions dataset includes 664824 rating data (1–5 points) of 139738 items from 40163 users. The trust information between users includes 487181 pieces. The data sparsity is very low, so this paper preprocesses the data according to the method of literature [14]. Finally, the sparsity of the data is 98.22%.

We use Python for programming, and the experiment is carried out based on the Lenskit toolkit [15].The dataset is

**Input:** User item scoring matrix  $R$ , target user  $V$ , number of neighbor users  $K$ , number of recommended items  $n$   
**Output:**  $N$  items recommended to target user  $V$

- (1) Import user item scoring matrix data
- (2) For the common attention matrix, the attention similarity between users is calculated by Formula (5)
- (3) For user attributes, use Formulas (6) and (7) to calculate the similarity between user attributes
- (4) Through the common scoring items, the similarity of user attention is calculated by Formula (8)
- (5) The attention similarity calculated in step 2 (the weight is  $\alpha$ ), the attribute similarity calculated in step 3 (the weight is  $1 - \alpha$ ), the weight fusion method is used to weight the score similarity to obtain the user multisimilarity
- (6) According to the calculation result in step 5, extract  $K$  users' favorite  $m$  items ( $m > n$ , excluding the items already liked by the target user)
- (7) According to the popularity, use Formula (10) to calculate the weight  $W$  of the improved project popularity
- (8)  $M$  candidate items are multiplied by the weight  $W$ , and  $N$  recommended items are selected from high to low

ALGORITHM 1: Collaborative filtering recommendation algorithm integrating time and multisimilarity.

TABLE 3: Algorithm time efficiency (based on Movielens100K dataset).

Algorithm	Running time (unit: second)
COS	802.98
OS	1367.66
COS-ER	1211.23
COS-MS	2167.56

divided into training set and test set, with the proportion of 70% and 30%, respectively.

The experimental environment is 3.70 GHz CPU, with 32 GB Storage, windows 11 operating system.

**4.2. Evaluation Index.** Mean absolute error (MAE) and root mean squared error (RMSE) are the evaluation indicators for evaluating the accuracy of prediction scores, which are applied to calculate the difference between the predicted value and the actual value of the project. The smaller the MAE value (as shown in Formula (11)), where  $N$  represents the number of prediction scores) and the smaller RMSE value (as shown in Formula (12)) are, the higher the accuracy of the prediction value is [16]

$$\text{MAE} = \frac{\sum_{u \in U, j \in I} |P_{u,i} - R_{u,i}|}{N}, \quad (11)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{u \in U, j \in I} (P_{u,i} - R_{u,i})^2}{N}}. \quad (12)$$

### 4.3. Results and Analysis

**4.3.1. Determination of Relevant Parameters.** Because the data sparsity of Movielens100K dataset is relatively moderate, we adopt them to determine the relevant parameters in order to prevent over fitting. In the algorithm, the proportion of different angle similarity in the calculation is different. Considering in the actual scene, the scoring similarity (i.e. in Formula (5)) can generally reflect the similarity between users better than the attribute similarity

(i.e. in Formulas (6)–(8)). The weight of scoring similarity is set to  $\alpha$ . The proportion is higher than the weight of attribute similarity ( $1 - \alpha$ ). Therefore, the scoring weight  $\alpha$  increases in steps of 0.01, and the value range is [0.5, 0.9]. The number of neighbors  $K$  is 10, 20, 30, 40, 50, 60, 70, and 80, and the recommended number is 10. It can be found from Figure 1 that the value in the algorithm MAE is between 0.750 and 0.766; when  $\alpha$  is at 0.6, the value of MAE is relatively low and stable. Therefore, in this paper, the weight value  $\alpha$  is 0.6.

### 4.3.2. Comparative Analysis

**(1) Algorithm Error Rate.** For checking the effectiveness of this algorithm, the similarity COS-MS proposed in this paper is compared with COS, COS-ER [7], and OS [1] and other similarity calculation methods based on the above indicators. The number of neighbors  $K$  is 10, 20, 30, 40, 50, 60, 70, and 80. The MAE and RMSE values based on Movielens100K dataset are shown in Figures 2 and 3. The values of MAE and RMSE based on the Epinions dataset are shown in Figures 4 and 5.

From Figures 2 and 3, it can be found that the MAE and RMSE values of each algorithm based on Movielens100K dataset change with the change of  $K$  value. The algorithm COS-MS proposed in this paper has a certain betterment over COS and OS. When the number of neighbors is less than 30, the COS-ER algorithm is better than COS-MS. But when the number of neighbors gradually increases, the advantages of this algorithm are gradually reflected. This is mainly because this algorithm considers the multiangle similarity. When the number of neighbors is on the increase, the accuracy of relative recommendation will also improve.

Similarly, as can be seen from Figures 4 and 5, the MAE and RMSE values of each algorithm change with the change of  $K$  value. Compared with Figures 3 and 4, we can find that the values of MAE and RMSE have increased, indicating that when the data sparsity increases, the error will increase slightly. When the number of neighbors is greater than 30, the algorithm COS-MS proposed in this paper has certain improvement over other algorithm.

For different datasets, when the number of neighbors increases, the algorithm shows good accuracy, in other words, the values of MAE and RMSE decrease.

(2) *Algorithm Time Efficiency*. Table 3 lists the time required for each algorithm to compute the similarity based on Movielens100K dataset. It can be seen that the more factors considered in calculating the similarity, the higher the time complexity, and the longer the time required for calculation. The COS algorithm has the lowest time complexity, the OS and COS-ER algorithm have the similar time complexity, and the COS-MS algorithm has the highest time complexity and takes the longest time. COS is a relatively simple traditional calculation method, which takes the shortest time. OS and COS-ER have been modified on the basis of COS, and the amount of calculation is similar. Our algorithm combines many aspects of similarity calculation, and the time is relatively high. Therefore, for the calculation of large datasets, it is a requisite to study distributed calculation to enhance the recommendation efficiency. Due to space limitation, the running time based on Epinions dataset will not be described, which is similar to the conclusion shown in Table 3.

## 5. Concluding Remarks

In view of the existing algorithms that do not fully consider the impact of the environment on users' interest, this paper introduces the time factor to improve the influence of long-term data on recommendation. For the problem of cold start, new users are recommended through user attribute similarity. According to the idea that only the items that users are interested in will produce behavior, the improved algorithm pays more attention to users' attention and introduces the similarity of users' attention so as to reduce the impact of the score itself on the recommendation algorithm. Generally, products with high popularity are easier to be found by users. Different weights are set for different items to reduce the role of popular items in similarity calculation and final recommendation. Through the Movielens100K dataset and Epinions dataset as the experimental data, the experiment shows that this algorithm can effectively improve the accuracy of recommendation and give users a better recommendation effect.

Because the algorithm is slightly insufficient in time efficiency, the project team not only considers using distributed computing to improve efficiency when there is a large amount of data but also needs to optimize the algorithm steps to optimize the algorithm [17, 18]. At the same time, how to effectively model data with knowledge map and how to integrate deep learning technology to solve accurate recommendation under big data are the further works of our project team.

## Data Availability

The data that support the findings of this study are available from the author.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding this paper.

## Acknowledgments

This work was supported by the National Social Science Foundation (No.16BFX085).

## References

- [1] A. Gazdar and L. Hidri, "A new similarity measure for collaborative filtering based recommender systems," *Knowledge-Based Systems*, vol. 188, Article ID 105058, 2020.
- [2] W. U. Jian-xin and Z. Zhang, "Collaborative filtering recommendation algorithm based on user rating and similarity of explicit and implicit interest," *COMPUTER SCIENCE*, vol. 48, no. 5, pp. 147–154, 2021.
- [3] A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, and R. R. Shah, "Learning path personalization and recommendation methods: a survey of the state-of-the-art," *Expert Systems with Applications*, vol. 159, Article ID 113596, 2020.
- [4] Y. Jin-ming, J. Meng, and W. Qiu-feng, "Item collaborative filtering recommendation algorithm based on improved similarity measurement," *Computer Applications*, vol. 37, no. 5, pp. 1387–1391, 2017.
- [5] J. Luo and Z. Wen-qi, "User similarity calculation method considering item similarity weight," *Computer Engineering and Applications*, vol. 51, no. 8, pp. 123–127, 2015.
- [6] B. Wang, *Collaborative Filtering Recommendation Algorithm Combining Item Attribute Similarity and Rating Similarity*, Yanshan University, Qinhuangdao China, 2017.
- [7] W. Li-ping, F. U. Pan, and Q. yue, "Collaborative filtering algorithm combined with relative differences in scores," *Journal of Chinese Computer Systems*, vol. 43, no. 7, pp. 1388–1393, 2022.
- [8] L. I. H. yang and F. U. Y. qing, "Collaborative filtering recommendation algorithm based on tag clustering and item topic," *COMPUTER SCIENCE*, vol. 45, no. 4, pp. 247–251, 2018.
- [9] D. Wang, Y. Yih, and M. Ventresca, "Improving neighborhood-based collaborative filtering by using a hybrid similarity measurement," *Expert Systems with Applications*, vol. 160, Article ID 113651, 2020.
- [10] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [11] R. E. V. D. S. Rosa, F. A. S. Guimarães, R. D. S. Mendonça, and V. F. D. Lucena, "Improving prediction accuracy in neighborhood-based collaborative filtering by using local similarity," *IEEE Access*, vol. 8, pp. 142795–142809, 2020.
- [12] C. Dong, *Research on Improvement of Hybrid Algorithm Based on Content and Collaborative filtering*, Shanxi University of Finance & Economics, Taiyuan, China, 2021.
- [13] L. Deng, J. Huang, and C. Yue, "Collaborative filtering algorithm integrating item popularity and multi-similarity among users," *Journal of Chinese Computer Systems*, vol. 7, 2021.
- [14] S. Endle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, AUAI Press, Hildesheim, Germany, June 2012.
- [15] M. D. Ekstrand, "LensKit for python: Next-Generation Software for Recommender Systems experiments," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2999–3006, Ireland, October 2020.



- [16] Y. Wang, P. Wang, Z. Liu, and L. Y. Zhang, "A new item similarity based on  $\alpha$ -divergence for collaborative filtering in sparse data," *Expert Systems with Applications*, vol. 166, Article ID 114074, 2021.
- [17] L. Zhang, Z. Zhang, J. He, and Z. Zhang, "A User-Based Collaborative Filtering Recommendation System Based on Trust Mechanism and Time Weighting," in *Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 69–76, Tianjin, China, January 2020.
- [18] Y. Zhang, Z. Dong, and M. Xiang-Wu, "Research on personalized advertising recommendation system and their applications," *Chinese Journal of computer*, vol. 44, no. 3, pp. 531–563, 2021.

## Research Article

# An IGWOCNN Deep Method for Medical Education Quality Estimating

Lin Shi<sup>1</sup> and Lei Zheng<sup>2</sup> 

<sup>1</sup>City University of Macau, Faculty of Humanities and Social Sciences, Macau 999078, China

<sup>2</sup>Shen Zhen Institute of Technology, The School of Business, Shen Zhen 518116, China

Correspondence should be addressed to Lei Zheng; zhenglei@ssti.net.cn

Received 20 June 2022; Accepted 18 July 2022; Published 9 August 2022

Academic Editor: Chaoqun Duan

Copyright © 2022 Lin Shi and Lei Zheng. 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 deep learning and mining ability of big data are used to analyze the shortcomings in the teaching scheme, and the teaching scheme is optimized to improve the teaching ability. The convolution neural network optimized by improved grey wolf optimization is used to train the data so as to improve the efficiency of searching the optimal value of the algorithm and prevent the algorithm from tending to the local optimal value. In order to solve the shortcoming of grey wolf optimization, an improved grey wolf optimization, that is, grey wolf optimization with variable convergence factor, is used to optimize the convolution neural network. The grey wolf optimization with variable convergence factor is to balance the global search ability and local search ability of the algorithm. The testing results show that the quality estimating accuracy of convolutional neural networks optimized by improved grey wolf optimization is 100%, the quality estimating accuracy of convolutional neural networks optimized by grey wolf optimization is 93.33%, and the quality estimating accuracy of classical convolutional neural networks is 86.67%. We can conclude that the medical education quality estimating ability of convolutional neural network optimized by improved grey wolf optimization is the best among convolutional neural networks optimized by improved grey wolf optimization and classical convolutional neural networks.

## 1. Introduction

The traditional medical education method has been difficult to meet the existing educational needs. In order to promote the optimization of big data and deep learning algorithms to the education industry, the deep learning and mining ability of big data are used to analyze the shortcomings in the teaching scheme, and the teaching scheme is optimized to improve the teaching ability. In order to optimize the education scheme, it is necessary to analyze the teaching data in order to improve the educational effect of the teaching scheme. Therefore, this paper proposes a medical education data analysis method based on convolutional neural networks optimized by improved grey wolf optimization, optimizes the education information data by using an artificial intelligence algorithm, and analyzes the education status through the characteristics of different levels of data. It is

necessary to analyze the medical education data in order to realize the multilevel training program of medical education. This study uses the students' education and teaching materials, the relevant data generated by online learning and the students' examination results and evaluation data to analyze the learning degree data of students, and analyzes the requirements of relevant enterprises for the post.

Convolutional neural network is a popular deep learning method [1–3]. The difference between convolutional neural networks and traditional neural networks is that a convolutional neural network includes a feature extractor composed of a convolution layer and a pooling layer. The learning rate of convolutional neural networks is often based on human experience, which will lead to over-fitting or under-fitting of the model [4–6]. Therefore, a method with a simple structure, fast convergence speed, and easy implementation is needed to optimize the learning rate of a convolutional neural network.

The convolution neural network optimized by improved grey wolf optimization (IGWOCNN) is used to train the data, so as to improve the efficiency of searching for the optimal value of the algorithm and prevent the algorithm from tending to the local optimal value. Grey wolf optimization is a population intelligent optimization algorithm based on the social order of grey wolves, inspired by the activity of grey wolf hunting prey [7–11], which has the characteristics of strong convergence, few parameters, and easy implementation. In order to solve the shortcoming of grey wolf optimization, an improved grey wolf optimization, that is, grey wolf optimization with variable convergence factor, is used to optimize the convolution neural network. The grey wolf optimization with variable convergence factor is to balance the global search ability and local search ability of the algorithm. Thus, the improved grey wolf optimization is used to optimize the convolution neural network. The quality estimating accuracies of IGWOCNN, convolutional neural network optimized by grey wolf optimization (GWOCNN), and classical convolutional neural network (CNN) are shown in the testing results. From results of the test, we can conclude that the medical education quality estimation ability of IGWOCNN is the best among IGWOCNN, GWOCNN, and classical CNN.

## 2. The Optimization of Convolutional Neural Network Based on Improved Grey Wolf Optimization

A convolutional neural network is a popular deep learning model, which is a kind of feedforward neural network with a deep structure including convolution calculation. The difference between convolutional neural networks and traditional neural networks is that a convolutional neural network includes a feature extractor composed of a convolution layer and a pooling layer. In the convolution layer of a convolution neural network, a neuron is only connected with some adjacent neurons. The neurons of the same feature map share the same weight, named the convolution kernel [12, 13]. The convolution kernel is generally initialized in the form of a random decimal matrix, which will learn to obtain reasonable weights in the process of network training. The direct benefit of the convolution kernel is to reduce the connection between network layers and reduce the risk of over-fitting. A convolutional neural network usually includes four layers: a convolution layer, a pooling layer, a full connection layer, and a classification layer.

Each convolution layer of a convolutional neural network is composed of several convolution units, and the purpose of the first layer is to extract more features from the lower-level network. The pooling layer usually obtains features with large dimensions after the convolution layer. The fully connected layer combines all local features into global features to calculate the score of each last category. The classification layer outputs the probability of the corresponding category of medical education quality.

The learning rate of convolutional neural networks is often based on human experience, which will lead to over-fitting or under-fitting of the model. Therefore, an optimization method with a simple structure, fast

convergence speed, and easy implementation is needed to optimize the learning rate of a convolutional neural network.

This study analyzes the medical education data through the IGWOCNN algorithm and outputs the medical education quality information, so as to evaluate the quality of education, and complete the education scheme required by the students.

Grey wolf optimization is used to optimize the learning rate of convolutional neural networks because of its fast convergence speed and easy implementation. Grey wolf optimization based on the social order of grey wolves, inspired by the activity of grey wolf hunting prey [14, 15], has the characteristics of strong convergence, few parameters, and easy implementation. Grey wolves belong to the social canine family and strictly abide by a hierarchy of social dominance [16]. The first level of social hierarchy: the first wolf of social hierarchy is  $\alpha$  wolf, which is mainly responsible for making decisions on predation, habitat, work and rest time, and other activities. Other wolves need to obey the orders of the  $\alpha$  wolf. The second level of social hierarchy is  $\beta$  wolf, which obeys the  $\alpha$  wolf and assists the  $\alpha$  wolf in making decisions. The third level of social hierarchy is  $\delta$  wolf, which obeys  $\alpha$  and  $\beta$  wolves and dominates the remaining levels of wolves [17, 18]. In order to balance the global search ability and local search ability of the algorithm, the improved GWO, that is, grey wolf optimization with variable convergence factor, is used to optimize the convolution neural network, which includes the steps of the grey wolf's social hierarchy, encircling, hunting, and attacking prey, which are described as follows.

*2.1. Grey Wolf's Social Hierarchy.* Calculate the fitness of each individual in the population. Mark the three grey wolves with the best fitness as  $\alpha$ ,  $\beta$ , and  $\delta$ .

*2.2. Encircling.* Encircling the prey, the grey wolf will gradually approach the prey and surround it when it ropes the prey. The mathematical model of this behavior is as follows:

$$\begin{aligned} D &= \left| C \cdot X_p(t) - X(t) \right|, \\ X(t+1) &= \left[ X_p(t) - A \cdot D \right], \\ A &= 2a \cdot r_1 - a, \\ C &= 2r_2, \end{aligned} \quad (1)$$

where  $t$  is the current number of iterations,  $A$  and  $C$  are synergy coefficient vectors,  $X_p(t)$  is the position vector of the current prey, and  $X(t)$  is the position vector of the current grey wolf. In the whole iterative process,  $a$  decreases linearly from 2 to 0, and  $r_1$  and  $r_2$  are random vectors in  $[0, 1]$ ;  $a$  is the variable convergence factor:

$$a = 2 \left( 1 - \frac{t}{t_{\max}} \right), \quad (2)$$

where  $t_{\max}$  is the maximum iteration.

**2.3. Hunting.** Grey wolves have the ability to identify the location of potential prey. The search process is mainly completed under the guidance of  $\alpha$ ,  $\beta$ , and  $\delta$  grey wolves. However, the solution space characteristics of many problems are unknown, and the grey wolf cannot determine the exact location of prey (optimal solution). The mathematical model of the search behavior of a grey wolf can be expressed as follows:

$$\begin{cases} D_\alpha = |C_1 \cdot X_\alpha - X|, \\ D_\beta = |C_2 \cdot X_\beta - X|, \\ D_\delta = |C_3 \cdot X_\delta - X|, \end{cases} \quad (3)$$

$$\begin{cases} X_1 = |X_\alpha - A_1 \cdot D_\alpha|, \\ X_2 = |X_\beta - A_2 \cdot D_\beta|, \\ X_3 = |X_\delta - A_3 \cdot D_\delta|, \end{cases} \quad (4)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}, \quad (5)$$

where  $X\{\alpha\}$ ,  $X\{\beta\}$ , and  $X\{\delta\}$  are, respectively, the position vectors of  $\alpha$ ,  $\beta$ , and  $\delta$ , and in the current population;  $D\{\alpha\}$ ,  $D\{\beta\}$ , and  $D\{\delta\}$  are the distance between the current candidate grey wolf and the best three wolves, respectively.

**2.4. Attacking Prey.** In the process of constructing the attack prey model, the decrease of a value will cause the value of  $a$  to fluctuate. Searching for prey, grey wolves mainly rely on the information of  $\alpha$ ,  $\beta$ , and  $\delta$  to find the prey.

The steps for optimizing the learning rate of convolutional neural networks are as follows:

Step 1: the range of the learning rate of convolutional neural network is initialized, and the GWO parameters are set. The size of the grey wolf population is 20, and the grey wolf population is randomly generated; the individual position of each grey wolf group is indicated, and the maximum number of iterations is set to 100.

Step 2: input the individual position to the convolutional neural network model to obtain the fitness value of the current grey wolf individual.

Step 3: according to the current fitness value, the grey wolf population is divided into  $\alpha$ ,  $\beta$ , and  $\delta$ . Encircling prey, the grey wolf will gradually approach the prey and surround it when it ropes the prey.

Step 4: the search process is mainly completed under the guidance of  $\alpha$ ,  $\beta$ , and  $\delta$  grey wolves, and the position of each individual in the wolf group is updated according to Eqs. (3)–(5).  $\alpha$ ,  $\beta$ , and  $\delta$  are selected from the current wolves by using the fitness value.

Step 5: searching for prey grey wolves mainly relies on the information of  $\alpha$ ,  $\beta$ , and  $\delta$  to find prey.

Step 6: if the maximum number of iterations is reached, the iteration is terminated; otherwise, the algorithm returns to Step 2.

Step 7: the optimized learning rate of convolutional neural network is employed, and the optimized convolutional neural network is employed.

### 3. Experimental Study of Medical Education Quality Estimating Method Based on IGWOCNN

In this experiment, we collected 15 medical education quality data as our testing samples to study the feasibility of the medical education data analysis method based on IGWOCNN. Figure 1 gives the flowchart of the medical education data analysis based on IGWOCNN. The flow of the medical education data analysis is composed of data source, data acquisition, depth analysis, and medical education quality. The data source are the influencing features of medical education quality, which include the students' education and teaching materials, relevant data generated by online learning, and students' examination results and evaluation data, as well as the requirements of relevant enterprises for the post, which is given in Table 1. Data acquisition includes data extraction and conversion, digitally extract these data, and storing them in the database. The IGWOCNN algorithm is used as a deep analysis method. Input the students' education and teaching materials, relevant data generated by online learning, and students' examination results and evaluation data, as well as the requirements of relevant enterprises for the post, and calculate the teaching quality value by using the IGWOCNN algorithm, so as to evaluate the teaching quality and optimize the teaching scheme. The values of medical education quality are 1–9, and the bigger the value is, the better the medical education quality is.

In order to show the medical education quality estimating ability of IGWOCNN, the comparison of the estimating values of medical education quality among IGWOCNN, GWOCNN, and classical convolutional neural networks are given in this paper. As shown in Figure 2, the quality values of all the testing samples are correct in the medical education quality estimation values of 15 testing samples based on IGWOCNN. As shown in Figure 3, the quality value of only one testing sample is incorrect in the medical education quality estimation values of 15 testing samples based on GWOCNN. As shown in Figure 4, the quality values of two testing samples are incorrect in the medical education quality estimation values of 15 testing samples based on classical CNN. As shown in Table 2, the medical education quality estimating accuracy of IGWOCNN is 100%, the medical education quality estimating accuracy of GWOCNN is 93.33%, and the medical education quality estimating accuracy of classical CNN is 86.67%. Therefore, we can conclude that the medical education quality estimating ability of IGWOCNN is best among IGWOCNN, GWOCNN, and classical CNN.

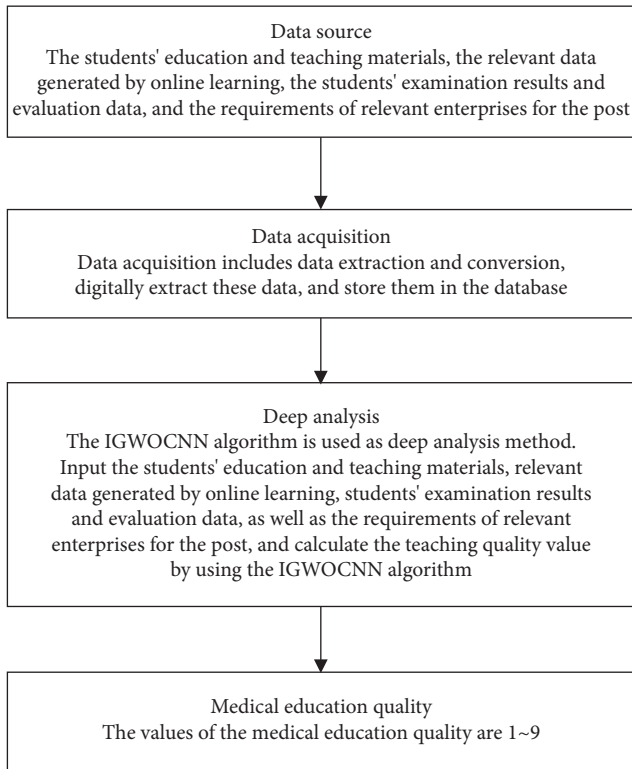


FIGURE 1: The flowchart of the medical education data analysis based on IGWOCNN.

TABLE 1: The influencing features of medical education quality.

No.	The influencing features of medical education quality
1	The students' education and teaching materials
2	The relevant data generated by online learning
3	The students' examination results and evaluation data
4	The requirements of relevant enterprises for the post

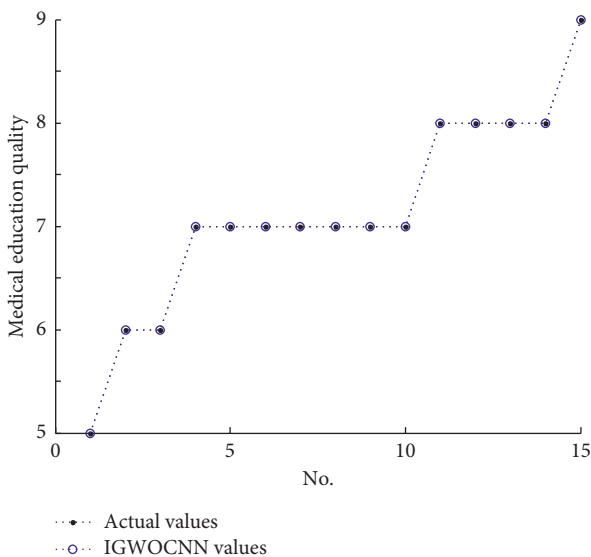


FIGURE 2: The medical education quality estimating values of 15 testing samples based on IGWOCNN.

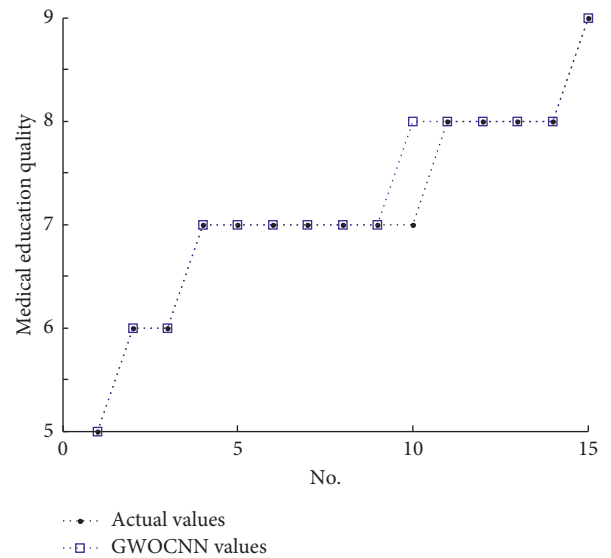


FIGURE 3: The medical education quality estimating values of 15 testing samples based on GWOCNN.

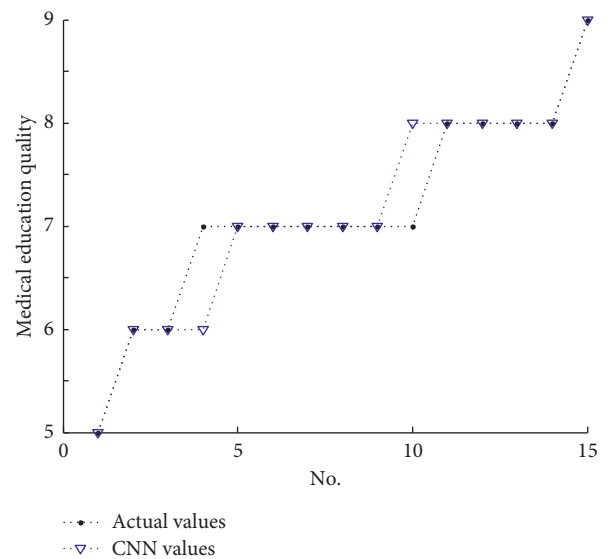


FIGURE 4: The medical education quality estimating values of 15 testing samples based on classical CNN.

TABLE 2: The medical education quality estimating accuracies of IGWOCNN, GWOCNN, and classical CNN.

Neural networks	Quality estimating accuracy (%)
IGWOCNN	100
GWOCNN	93.33
Classical CNN	86.67

#### 4. Conclusion

This study presents the convolution neural network optimized by improved grey wolf optimization for medical education quality estimating so as to complete the education scheme required by the students. In this paper, the improved

grey wolf optimization algorithm, that is, grey wolf optimization with variable convergence factor, is used to optimize the convolution neural network so as to improve the efficiency of searching for the optimal value of the algorithm and prevent the algorithm from tending to the local optimal value. The medical education quality estimation accuracy of IGWOCNN is higher than that of GWOCNN and classical CNN. In conclusion, we can conclude that the medical education quality estimation algorithm based on IGWOCNN can run more accurately.

### Data Availability

The dataset used to support the findings of the study can be accessed upon request.

### Conflicts of Interest

The authors declare no conflicts of interest.

### Acknowledgments

This study was supported by the Innovative Research Group Project of National Natural Science Foundation of China (41721003).

### References

- [1] A. Ghanbari and M. Modarressi, "Energy-efficient acceleration of convolutional neural networks using computation reuse," *Journal of Systems Architecture*, vol. 126, Article ID 102490, 2022.
- [2] R. D. Gottapu, C. Dagli, and B. Ali, "Entity resolution using convolutional neural network," *Procedia Computer Science*, vol. 95, pp. 153–158, 2016.
- [3] A. Ferreira and G. Giraldo, "Convolutional Neural Network approaches to granite tiles classification," *Expert Systems with Applications*, vol. 84, pp. 1–11, 2017.
- [4] M. Sargül, B. M. Ozyildirim, and M. Avci, "Differential convolutional neural network," *Neural Networks*, vol. 116, pp. 279–287, 2019.
- [5] S. Ghosh, N. Das, and M. Nasipuri, "Reshaping inputs for convolutional neural network: some common and uncommon methods," *Pattern Recognition*, vol. 93, pp. 79–94, 2019.
- [6] S. Izadi, M. Ahmadi, and R. Nikbazm, "Network traffic classification using convolutional neural network and ant-lion optimization," *Computers & Electrical Engineering*, vol. 101, Article ID 108024, 2022.
- [7] M. Ali, M. A. El-Hameed, and M. A. Farahat, "Effective parameters' identification for polymer electrolyte membrane fuel cell models using grey wolf optimizer," *Renewable Energy*, vol. 111, pp. 455–462, 2017.
- [8] K. Panwar and K. Deep, "Transformation operators based grey wolf optimizer for travelling salesman problem," *Journal of Computational Science*, vol. 55, Article ID 101454, 2021.
- [9] G. Siva shankar and K. Manigandan, "Diagnosis of diabetes diseases using optimized fuzzy rule set by grey wolf optimization," *Pattern Recognition Letters*, vol. 125, pp. 432–438, 2019.
- [10] A. K. Tripathi, K. Sharma, and M. Bala, "A novel clustering method using enhanced grey wolf optimizer and MapReduce," *Big Data Research*, vol. 14, pp. 93–100, 2018.
- [11] B. Zhao, Yi Ren, D. Gao, L. Xu, and Y. Zhang, "Energy utilization efficiency evaluation model of refining unit Based on Contourlet neural network optimized by improved grey optimization algorithm," *Energy*, vol. 185, pp. 1032–1044, 2019.
- [12] M. Kozłowski, P. Górecki, and P. M. Szczypiński, "Varietal classification of barley by convolutional neural networks," *Biosystems Engineering*, vol. 184, pp. 155–165, 2019.
- [13] R. Rasti, M. Teshnehlab, and S. L. Phung, "Breast cancer diagnosis in DCE-MRI using mixture ensemble of convolutional neural networks," *Pattern Recognition*, vol. 72, pp. 381–390, 2017.
- [14] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of interconnected power system using grey wolf optimization," *Swarm and Evolutionary Computation*, vol. 27, pp. 97–115, 2016.
- [15] S. Zapotecas-Martínez, A. García-Nájera, and A. López-Jaimes, "Multi-objective grey wolf optimizer based on decomposition," *Expert Systems with Applications*, vol. 120, pp. 357–371, 2019.
- [16] P. Siddavaatam and R. Sedaghat, "Grey Wolf Optimizer Driven design space exploration: a novel framework for multi-objective trade-off in architectural synthesis," *Swarm and Evolutionary Computation*, vol. 49, pp. 44–61, 2019.
- [17] M. Pradhan, P. K. Roy, and T. Pal, "Grey wolf optimization applied to economic load dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. 83, pp. 325–334, 2016.
- [18] F. B. Ozsoydan, "Effects of dominant wolves in grey wolf optimization algorithm," *Applied Soft Computing*, vol. 83, Article ID 105658, 2019.