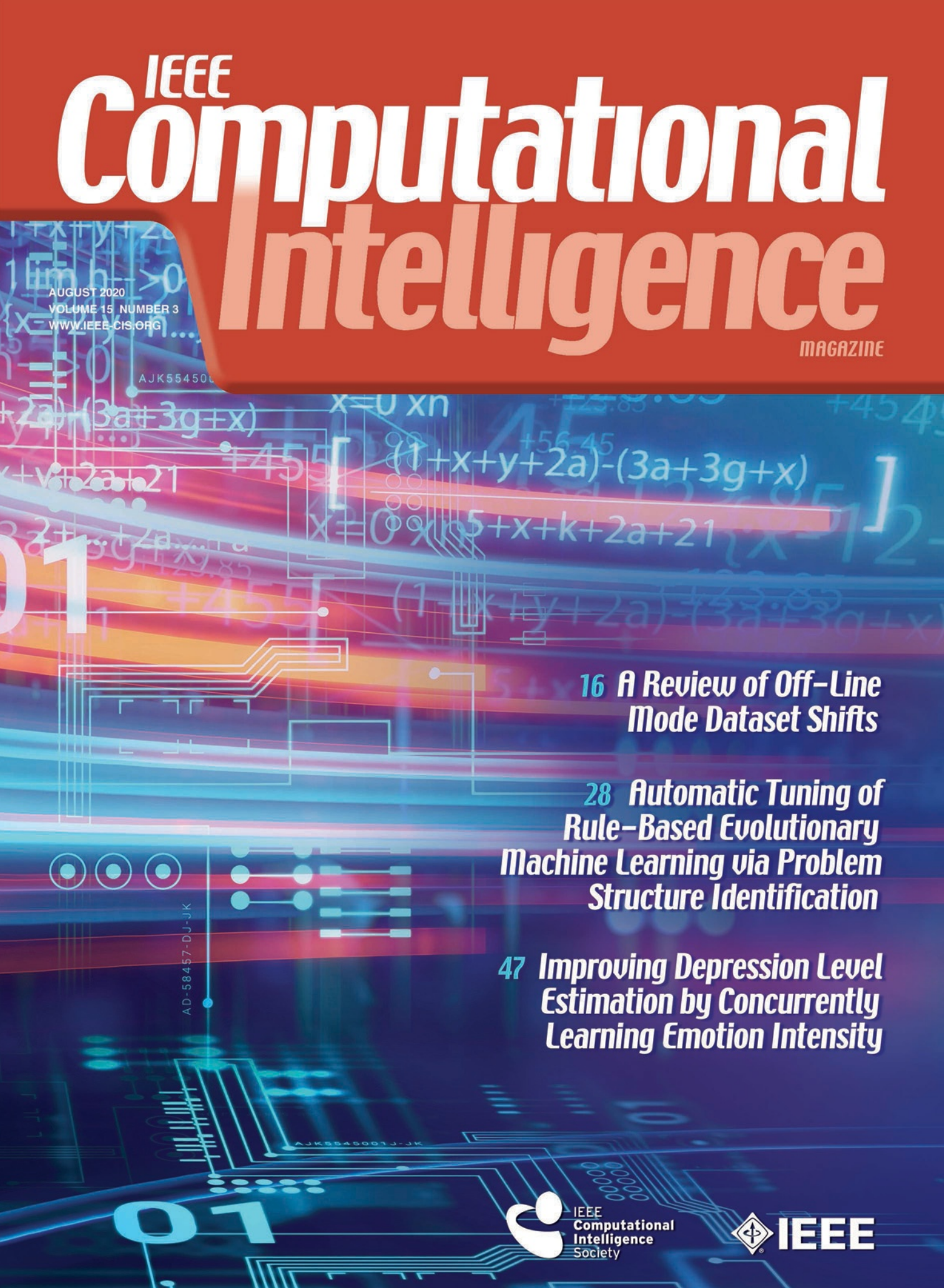
IEEE

Computational Intelligence

MAGAZINE

AUGUST 2020
VOLUME 15 NUMBER 3
WWW.IEEE-CIS.ORG



IEEE Computational Intelligence Society

IEEE

IEEE Transactions on Artificial Intelligence

Call for Papers

Scope

The IEEE Transactions on Artificial Intelligence (TAI) is a multidisciplinary journal publishing papers on theories and methodologies of Artificial Intelligence. Applications of Artificial Intelligence are also considered.

Topics covered by IEEE TAI include, but not limited to, Agent-based Systems, Augmented Intelligence, Autonomic Computing, Constraint Systems, Explainable AI, Knowledge-Based Systems, Learning Theories, Planning, Reasoning, Search, Natural Language Processing, and Applications. Technical papers addressing contemporary topics in AI such as Ethics and Social Implications are welcomed.

Invitation

The journal invites impactful Artificial Intelligence research, survey articles, and applications. Submit your manuscript at the IEEE TAI Manuscript Central website at https://mc.manuscriptcentral.com/tai-ieee. Potential authors should consult the Information to Authors Document at https://cis.ieee.org/publications/ieee-transactions-on-artificial-intelligence/information-for-authors-tai. Further questions can be directed to the Founding Editor-in-Chief at ieee.tai.eic@gmail.com

Founding Editor-in-Chief

Hussein Abbass, University of New South Wales, Canberra, Australia

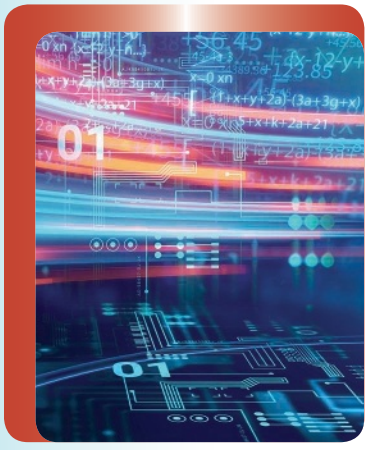
Associate Editors

- Amal El Fallah Seghrouchni, Sorbonne University, France
- Catherine Huang, McAfee, USA
- Christian Wagner, University of Nottingham, UK
- Dongbin Zhao, University of Chinese Academy of Sciences, China
- Fiora Pirri, University of Rome, Italy
- Gary Yen, Oklahoma State University, USA
- Guilherme DeSouza, University of Missouri, USA
- Haibo He, University of Rhode Island, USA
- Hao Luo, Harbin Institute of Technology, China
- Johan Suykens, Katholieke Universiteit Leuven, Belgium
- Kay Chen Tan, City University of Hong Kong, Hong Kong
- Lirong Xia, Rensselaer Polytechnic Institute , USA
- Matthew Garratt, University of New South Wales, Australia
- Michael Wooldridge, University of Oxford, UK
- Pau-Choo Chung, National Cheng Kung University, Taiwan
- Peter Stuckey, Monash University, Australia
- Ran Cheng, Southern University of Science and Technology, China
- Sanaz Mostaghim, Otto von Guericke University Magdeburg, Germany
- Simon Yang, University of Guelph, Canada
- Supratik Mukhopadhyay, Louisiana State University, USA
- Weizhong Yan, General Electric Global Research Center, USA
- Yo-Ping Huang, National Taipei University of Technology, Taiwan
- Pablo Estevez , University of Chile, Chile
- Pascal Van Hentenryck, Georgia Tech, USA

Digital Object Identifier 10.1109/MCI.2020.2998306

IEEE Computational Intelligence MAGAZINE

Volume 15 Number 3 ❑ August 2020
www.ieee-cis.org



on the cover
©ISTOCKPHOTO.COM/MONSITJ

Features

Columns

IEEE Computational Intelligence Magazine (ISSN 1556-603X) is published quarterly by The Institute of Electrical and Electronics Engineers, Inc. **Headquarters:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997, U.S.A. +1 212 419 7900. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society, or its members. The magazine is a membership benefit of the IEEE Computational Intelligence Society, and subscriptions are included in Society fee. Replacement copies for members are available for US$20 (one copy only). Nonmembers can purchase individual copies for US$213.00. Nonmember subscription prices are available on request. Periodicals postage paid at New York, NY and at additional mailing offices. Postmaster: Send address changes to IEEE Computational Intelligence Magazine, IEEE, 445 Hoes Lane, Piscataway, NJ 08854-1331 U.S.A. PRINTED IN U.S.A. Canadian GST #125634188.

Departments

Digital Object Identifier 10.1109/MCI.2020.2997121

Digital Object Identifier 10.1109/MCI.2020.2997122

Chuan-Kang Ting
National Tsing Hua University, TAIWAN

Editor's Remarks

Every Cloud Has a Silver Lining

The COVID-19 pandemic broke out at the beginning of 2020. As of mid-May, there have been more than 4.65M cases reported and 312K deaths. Governments worldwide implement control measures, such as lockdowns, travel restrictions, school closure, and social distancing, in order to prevent further spread of COVID-19. This disease not only threatens people's lives but also impacts economic as well as academic activities. On the other hand, this difficult time gives us an opportunity to think about the important things in our life and ponder the value of our research—has it really contributed to human well-being?

The CIS has launched the *COVID-19 initiative* to expedite and expand dissemination of research results on computational intelligence helping combat COVID-19. All the IEEE CIS Transactions and Magazine will speed up the review process of the articles that focus on COVID-19. Under this initiative, the IEEE CIM has set up a fast-track special issue on "Computational Intelligence for Combating COVID-19" to present the latest research and development in applying computational intelligence technologies to fight COVID-19.

In this issue, we interview Hussein Abbass from the University of New South Wales, Australia. The founding Editor-in-Chief of IEEE Transactions on Artificial Intelligence talks about this new publication, his life and favorites. The *Features* includes three articles on computational intelligence and machine learning. The first article reviews the characteristics and causes of dataset shifts, and presents a survey on machine learning techniques for handling off-line mode dataset shifts problems. The second article addresses a desirable feature, namely automatic tuning, for evolutionary machine learning. The third article leverages the relation between depression level and emotion intensity. Three multi-task architectures are employed for concurrently learning the two measures and gain promising performance. In the *Columns*, the Research Frontier article introduces a strength index to the softmax policy, so as to adjust and assess the strength of Monte-Carlo tree search. This approach is applied to Go programs and achieves state-of-the-art results.

We hope you will enjoy the articles in this issue. I would also like to encourage you to contact me at ckting@pme.nthu.edu.tw with your suggestions and comments. Hope the pandemic will be under control soon and look forward to seeing you at SSCI 2020.

Stay safe and healthy!

Digital Object Identifier 10.1109/MCI.2020.2998226
Date of current version: 15 July 2020

President's Message

Bernadette Bouchon-Meunier
CNRS – Sorbonne Université, FRANCE

Beyond the COVID-19 Pandemic

At the time when I write this message, the world is swamped by the COVID-19 pandemic and I think of all of you, faced with the danger, the risk, the concern for your loved ones, the isolation due to a lockdown. Your work in online teaching and tutoring is essential in these difficult times. As your field of excellence is related to computational intelligence, I am sure that many of you are involved in Coronavirus research, either as data scientists or more specifically as specialists of bioinformatics and computational biology. I am very proud of you and the IEEE Computational Intelligence Society will do what it can to support you. I sincerely hope that the situation will be better for you when you will read these lines.

I would like to focus on the so-called *COVID-19 Initiative* launched by the CIS to encourage all researchers working on any aspect of the fight against COVID-19 to submit their papers to the most appropriate IEEE CIS-sponsored or co-sponsored Transactions. The CIS Editors-in-Chief will *expedite*, to the extent possible, the processing of all articles submitted on the special COVID-19 track proposed by all these journals. It is very important to note that all accepted papers will be published, free-of-charge to authors and readers, with *free access* for one year from the date of the publication, so that the results can be used by other researchers and the community at large, in an effort to support researchers who dedicate their time to this humanitarian task and to contribute to the global effort to stop the pandemic. Many thanks to all the Editors-in-Chief for their instant and unreserved contribution to this initiative.

The CIS conferences will move to a virtual form during the period when no one can travel normally. It is of course not as friendly as physical meetings, but I can assure you that every effort will be made to allow with the discussion with the speakers and the networking. I am very grateful to all conference organizers for making this possible, as well as to all the authors, and in particular all the keynote and tutorial speakers who agreed to make their presentations remotely, despite the technical difficulties. I even dream that there will be more of you attending these conferences than expected, as they avoid all travel expenses and offer you a unique opportunity to access their content from home.

The meetings that should be held on the occasion of CIS conferences, such as the meetings of Women in Computational Intelligence and students or the chapter forum, will be replaced by virtual meetings. We are counting on you to explore this new form of discussion with us. I am also thankful to the CIS technical committee chairs who lead the activities of their groups in online meetings.

On a more optimistic note, I want to believe that we will all be able to meet physically in at least one of the CIS conferences this year. In particular, I hope to see you in Canberra for the very rich program of the IEEE SSCI 2020 conference on 1-4 December and its satellite events.

Take care and see you soon!

Digital Object Identifier 10.1109/MCI.2020.2998227
Date of current version: 15 July 2020

Digital Object Identifier 10.1109/MCI.2020.2997123

Career Profile

CIM Editorial Office

Interview with Founding Editor-in-Chief of IEEE Transactions on Artificial Intelligence

The Founding Editor-in-Chief of IEEE Transactions on Artificial Intelligence, Prof. Hussein Abbass, talks about this new publication, his life and favorites.

1. What events in your life most likely placed you on the path that led you to where you are today?

A few events in my life led me to where I am today. I will share below a few of them.

The first event took place during the last year of my first undergraduate Bachelor degree. One of my mid-year exams contained a linear programming question. After the exam, my friends and I used to discuss our answers. All my friends and tutors told me that my solution was incorrect. They claimed that the question had been taken from a booklet they studied before the exam. I did not have a copy of that booklet, so I kept digging.

It was close to impossible to talk to the professor in those days about that question, especially after the exam. I was very frustrated, impatient and curious at the same time. I waited for two weeks (the mid-year break) and another two weeks until I managed to finally catch the professor while he was walking to his car one night after the lecture. I explained to him my logic and told him that all my tutors and friends claimed I had provided the wrong solution. I continued talking to him while we were walking and begged him to quickly look at my detailed solution. He stopped when he reached his car door, looked at my solution, and exclaimed 'you are right, go home'! I was still puzzled! I needed more information.

In the following lecture after our conversation, he announced that he needed to re-mark the exam and that the answer in the printed booklet was wrong. He explained that one inequality was mistakenly printed with the opposite sign, resulting in the wrong solution, hence making my solution the only correct solution to the problem!

I obtained a full mark in the course. What I did not realize at the time was that, at that very moment I had found my passion. I switched from someone who used to study with little ambition that has no interest to continue beyond graduating and getting a job and seeing how life will play out, to someone who was filled with internal energy. That internal spark and fire have kept me going up to now. My life turned into mathematics, algorithms, technologies and cognition. While my field of research took a few turns, at every turn, I had the same passion and energy that turned me on to be a proactive and an autonomous person who does not stop at a 'no' or at 'what people think'.

The action of that professor taught me what professionalism and ethics are all about. To remark the exam for a few hundred students is not a simple task for anyone. However, the cost he paid to be fair is far less than the cost of guilt and being unfair that he would have lived with if he did not remark the exam. His ethics taught me a lesson for life.

The second event was after I completed my pre-master course work on Operations Research (OR). A young lecturer returned from her PhD and post-doc at Imperial College London, where she worked on Concurrency in Logic Programming, a topic I knew nothing about at the time. One thing I was always keen to do is to study different ways to solve problems. I used to believe that mathematical optimality is sufficient to convince people to use a solution; a fallacy that I spent the following 27 years of my life learning to address by contrasting descriptive and normative approaches.

She introduced me to the area of Logic and Artificial Intelligence (AI). I saw in that a different way to solve problems and started my journey into Constraint Logic Programming (CLP), a formalism that replaces the numeric calculus based solvers that I was used to in OR, with symbolic logic-based solvers. The moment I was exposed to Warren Abstract Machines, I spent a great deal of time to understand the fundamentals of this new world in which I was not acquainted.

I designed the first CLP system that can handle conflicting objectives and developed skills in writing a compiler for a language that sits on top of my multi-objective CLP system. I developed passion for compilers and continued with CLP for many years. I went on to complete my PhD qualifying exam and started a PhD. Towards the last third of that PhD, I was awarded a scholarship

Digital Object Identifier 10.1109/MCI.2020.2998228
Date of current version: 15 July 2020

to do a second Master degree on non-symbolic AI at the University of Edinburgh. That was my first encounter with Connectionism and Genetic Algorithms, but I also augmented my courses with others in knowledge representation, natural language processing and automated reasoning due to my passion for symbolism. My Master thesis combined symbolic and non-symbolic AI to solve a problem in heat exchanger networks, to which I was exposed after taking a course from the department of chemical engineering. My thesis was awarded the best AI thesis of the year at the University of Edinburgh. During that Master, I met a few people who truly influenced my life, including my wife, who kept growing my passion for linguistics and the philosophy of language. By the end of that Master, my brain was buzzing with new knowledge and information that kept influencing my way of thinking. I became a believer in the importance of using the right technique for the right problem and avoiding biases when applying the same technique to every problem. I started building passion for philosophy and language, although I did not enjoy any of these topics before.

The third critical event in my life which shaped my thinking and made me who I am today was an offer I received before I needed to return to resume my lecturer position and continue my first uncompleted PhD at Cairo University. I was awarded a PhD scholarship from Queensland University of Technology (QUT), Brisbane, Australia to work on data mining for the dairy industry. The mere idea of quitting a PhD right at the writing stage to start a new one, and moving all the way to Australia would sound inconceivable to many. The risk and sacrifice were significant. Had I not taken that risk, my life today would have been very different and most likely, I would not have been where I am currently.

With continuous fire in my belly and a keen interest in improving myself, I took the plunge and started a new PhD from scratch on a topic that required from me to study an area I knew nothing about; that is, animal genetics which was essential for the problem domain where I needed to apply AI. During that PhD, I learnt about rule extraction from neural networks and invented dynamic decision trees, where nodes have memories encoding Elman-type recurrent equations. However, my scholarship was supported from industry and I needed to also design data mining and multi-stage multi-objective algorithms for the optimization of genetic materials of dairy cattle, which I did.

Hussein Abbass with his family: Eleni, Adam and Zach.

At the end of that PhD, I was ready with a portfolio of mathematical and algorithmic skills, and experience in solving a wide range of problems across so many diverse domains, from finance, variety of engineering problems to animal genetics, and more! This was the starting point for the twenty years that followed and that brought me to where I am now to become the Founding Editor-in-Chief of the IEEE Transactions on Artificial Intelligence.

2. Can you tell us a little bit abou1t your research work that brought you to this appointment?

Over the last twenty years, and after completing my PhD at QUT, I continued to work with multi-objective problems, both for optimization and machine learning. I quickly realized the value of, and invented Pareto-based neural ensemble learning. I then had two major application themes that funded the first 12 years post my PhD. One was a set of technical projects in air traffic control and robotics with well-structured problems. The other set was sitting at the opposite end, where problems are wicked and ill-structured. I worked on developing computational environments and new algorithms for both. Optimization, simulation, machine learning and knowledge based systems were the natural choice for the first theme. The second theme required me to find ways to bring AI to wicked problems that mostly require understanding of complex systems, system thinking, and game theory. This is when I proposed the Computational Red Teaming (CRT) environment that I became known for. CRT brought AI to wicked problems to support long-term planning decisions, concept evaluations, strategy development, which then became a common technology in cyber security.

In the last 8 years of my career, I started to bring the human to AI and bring AI to the human. The skills I developed after my first degree led me naturally to appreciate cognitive engineering and design technical systems that connect the human brain and psychophysiological data with AI. I developed new architectures, such as cognitive-cyber symbiosis, for seamless dynamic coupling of humans and distributed autonomous systems.

Hussein Abbass at Eurocontrol Experimental Centre, Brétigny, France, injecting gel into an EEG electrode while preparing a participant before an augmented cognition experiment.

3. Where will your research take you from here?

Currently, I am focusing on distributed AI. With a wonderful team of students, post-docs and collaborators, we are working on a variety of topics including explainable distributed AI, smart swarm control algorithms using shepherding concepts, trusted AI algorithms for distributed contextual and situation awareness, autonomous AI testing, brain biometrics, and trusted AI algorithms for autonomous human performance assessment and dynamic allocation of automation functions between humans and distributed autonomous systems. I use neural networks, classifier systems, evolutionary computation, and sometimes fuzzy logic in my research. However, my early research career is still influencing my research program, where I am bringing predicate calculus, CLP and formal methods for trust assurance and explainable AI.

4. AI has accomplished remarkable achievements and showed its benefits to human life; on the other hand, people are concerned about its potential risks. Can you share your thoughts about the development and future of AI?

AI in the future will not be the AI we have known in the past, which has been a mere attempt to replicate human intelligence. Today, our world is being digitized. From the computational power of today's computers, to the connectivity offered by Industry 4.0, the future of AI will focus on distributed AI that is designed with its social responsibilities in mind, and consider the importance to blend within the human social system.

Every technology comes with risks. It is important to be wisely concerned in order to develop a better understanding of the risks and attempt to mitigate them. However, we should not be alarmed or develop risk-aversion that denies us the many positive opportunities AI brings. There is abundant evidence to suggest that AI as a technology will continue to grow. The momentum has reached a tipping point that it is inconceivable to even think we should or can try stopping it.

I feel saddened when I see some people spending their energy to scare people away from AI. I prefer to spend my energy on educating people to be good designers of ethical and beneficial AI, to embrace AI, and to put more emphasis than ever on ethics and the importance of trust. My philosophy is to take students on a journey to develop innovative AI systems to make them understand the challenges, give them a taste of the realm of the possible, and get them to truly appreciate the complexity and opportunities of AI systems. If we all channel our energy into teaching ethical AI, the gain will far exceed, and mitigate, the potential risks of these technologies.

5. The IEEE Transactions on Artificial Intelligence (TAI) is dedicated to all aspects of AI. How do you define your role as the Founding Editor-in-Chief of this new publication?

I see my role as threefold. In my strategic role, my aim is to ensure that when the readers of the journal read papers published in TAI, they read quality research that advances the body of knowledge in AI; and they see the richness of AI research and how it contributes to the society. In my executive role, my aim is to give a fair go to every author and improve the quality of accepted papers. In my tactical role, I want to guarantee smooth workflow in the review system.

6. What would be your priorities for the first year?

Setting the standards for TAI for future years.

7. What is your vision for TAI in five years from now?

TAI will be the go-to journal for both the public and the technical readers to discover and understand recent AI developments and applications.

8. What advices would you like to offer to the interested authors to submit their research work?

Follow the scientific method. Write to influence and generate impact. Be clear on the contribution and novelty. Your literature is your evidence of the scientific gap your paper is addressing. Mathematics exists for people to understand; it does not exist to obscure the presentation of a topic. Be upfront with the assumptions of your proposal. No algorithm will be the best on all classes of problems, explain clearly when your algorithm will work and when it won't. Edit your paper before submission. Read the Information for Authors and follow the guidelines there. Do not submit a paper that you would

Nikhil Pal, Kalyanmoy Deb, Yew Soon Ong, Hussein Abbass, Garrison Greenwood, Jun Wang and Akira Oyama at the Cognitive Engineering Facilities of Hussein Trusted Autonomy Laboratory in ACALCI 2016.

reject yourself if you get it as a reviewer. Good research takes time, spend the time, it will save you a lot of time in your own life. At the core of science, ethics and philosophy sit; be ethical, ask the right questions, and ask the questions right.

Five Minutes with Prof. Hussein Abbass

9. What was your service pathway in the Computational Intelligence Society?

I have been a member of IEEE CIS since its inception (was an IEEE member before that). I was the founding chair of the Artificial Life and Complex Adaptive Systems task force in 2003, which still exists to today. I was the vice general chair for IEEE CEC 2003, the general chair for IEEE WCCI 2012, and then became the chair of the Emerging Technologies Technical Committee for 2013-2014. I was an Associate Editor for IEEE CIM and IEEE Trans on Computational Social Systems, and have been an Associate Editor for IEEE Trans on Cybernetics, TEVC, TCDS, and TNNLS. I volunteered on many IEEE CIS committees and was elected as the vice-president for technical activities for two terms (2016-2019).

10. What is your typical working day?

Breakfast, spend a couple of hours in my home office, go to work, return between 1700-1900 depending on whether I need to pick up my kids for after-school co-curricular activities or not, dinner, possibly watch TV on the rare occasions when I have time, put the kids to sleep, work in my home office sometimes till 3am, then sleep.

Profile: Hussein Abbass

Professional qualifications:

- ❏ PhD Comp. Sc., QUT, Australia, 2000; MSc AI, University of Edinburgh, UK, 1997; MSc OR and CLP, Cairo University, 1995; Post-graduate Diploma, Cairo University, Egypt, 1992; Bachelor, Cairo University, Egypt, 1990.

Current position:

- ❏ Professor, School of Engineering and Information Technology, University of New South Wales, Canberra, Australia.

Institutions or companies where you have taught/conducted research:

- ❏ University of New South Wales Canberra; National University of Singapore; Imperial College London; University of Illinois Urbana Champaign; University of Edinburgh; Cairo University

Most notable award/recognition:

- ❏ Fellow, IEEE, Australian Computer Society, UK Operations Research Society, and Australian Institute of Managers and Leaders
- ❏ Founding Editor-in-Chief, IEEE Transactions on Artificial Intelligence

11. What is your ideal weekend?

Drive to Sydney with my family, stay and walk by the beach, have a room with view to the ocean, walk, have a great chocolate and possibly a nice cake with a Latte with an extra shot, eat my favorite food, watch the fireworks at night, walk and sleep.

12. Give one interesting fact about yourself

I enjoy cooking.

13. What are you reading, watching, or listening to at the moment?

I listen a lot to the Voice, both the voice kids and adults. I like classical music but only when I am alone. What I enjoy most is watching and listening to people; life is the university with the longest degree we will enroll in.

14. Person you would most like to meet – past or present, real or fictional?

I enjoy Chomsky, not because I agree with everything he said, but I find his style and depth of thinking entertaining. Marvin Minsky makes me think that we can make anything artificial. The two past persons I would love to meet again are my dad and mom, just to say thank you.

15. Can you share with us one success story that will motivate young members and provide useful guidelines for their careers?

Read my answer to the first question. These are my guidelines to everyone, young or not. Finish what you start even if you discover you do not like it any more. You can't guarantee perfection but you can guarantee that you did your due diligence by doing the best you could do. Face your fears in science, technology or humanity; all of these fields were invented by humans like us and they were not smarter than any of us; they just had the passion, they tried and persisted. Do not sacrifice your principles and your loved ones—you can recover wealth, publications, and anything in your life, but you can't recover your principles and your loved ones. Strike the right balance in your life by being happy with what you have to maintain self-satisfaction, and be ambitious to maintain self-motivation. Maintain self-respect, if you can sleep at night respecting who you are, you are a happy human being.

Longbing Cao
IEEE TF-DSAA Chair, University of Technology Sydney, AUSTRALIA

Marley M.B.R. Vellasco
IEEE CIS VP-Conferences, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), BRAZIL

IEEE DSAA–The IEEE Flagship Conference in Data Science

The Era of Data Science and Analytics

The twenty-first century has ushered in a new age that is coined as *data science* and *big data analytics*. Data-driven scientific discovery is regarded as the fourth science paradigm. Data science has been a core driver of the new-generation science, technologies and economy, and is driving new researches, innovation, profession, applications and education across both disciplines and business domains. There are many scientific and technical challenges associated with big data, ranging from data capture, creation, storage, search, sharing, modeling, representation, analysis, learning, visualization, explanation, and decision-making. Among the many data characteristics and complexities to be addressed, we mention here the hybridization of heterogeneous, multisource, hierarchical, interactive, dynamic, multidimensional, and quality-poor data mixed with real-time business operations, strategic planning, decision-making, value creation, and future developments. Another important agenda in the data science community is to address the misinformation and pitfalls and promote a deep understanding of the data science nature and reality.

Accordingly, the field of data sciences and big data analytics have been evolving from statistics since half century ago to broad areas including but not limited to data and signal analytics, knowledge discovery, information retrieval, machine learning, statistics, optimization, computing, and data management. By synergizing the three big areas—statistics, informatics and computing, data science has been spreading to essential and specific areas such as (1) data intelligence and complexity analysis, (2) representation, modeling, analytics, mining and learning including statistical and deep learning, (3) computational intelligence including neural networks, evolutionary computing, fuzzy systems, (4) neuroscience and linguistics, (5) behavioral science and social and economic computing, (6) uncertainty and optimization, (7) system and modeling infrastructures and architectures, (8) networking and interoperation, (9) social issues including privacy, security, trust, value and impact, (10) enterprises, services, applications, solutions and systems, and (11) simulation, visualization and explanation.

IEEE DSAA—Transdisciplinary Data Science Led by IEEE

The IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA) (see more about DSAA at dsaa2020.dsaa.co) aims to be a premier forum for addressing the above ever increasing and important demand, data volume and complexities, and the associated business problems, opportunities, decisions and values in a translational and transdisciplinary approach. DSAA was launched in 2014 in Shanghai chaired by Prof. Masaru Kitsuregawa and Prof. Philip S Yu as a major IEEE initiative in data science and big data, received support from the IEEE Big Data Initiative. Since 2015, DSAA has been financially sponsored by the IEEE Computational Intelligence Society (CIS), and also technically sponsored by ACM through SIGKDD, the American Statistical Association (ASA), and the China Computer Foundation (CCF). DSAA has been successfully rotated in Asia, Europe and America with DSAA'2019 held in Washington chaired by Prof. Philip S Yu and Prof. Richard De Veaux, DSAA'2018 in Turin chaired by Dr. Francesco Bonchi and Prof. Foster Provost, DSAA'2017 in Tokyo Chaired by Prof. Fosca Giannotti, Prof. Tomoyuki Higuchi and Prof. Motoda Motoda, DSAA'2016 in Montreal chaired by Prof. Stan Matwin and Prof. Osmar R. Zaiane, DSAA'2015 in Paris chaired by Prof. Longbing Cao and Prof. Eric Gaussier. DSAA'2020 will be held in Sydney on 6–9 Oct. 2020 and chaired by Prof. Geoff Webb and Dr. Usama Fayyad.

IEEE DSAA, technically managed by the IEEE CIS Task Force on Data Science and Advanced Analytics (TF-DSAA), has taken a strong transdisciplinary approach. The annual DSAA provides a premier data science forum that brings together researchers, industry and government practitioners, as well as developers and users in statistics, computing science, informatics and intelligence science for the exchange of the latest theoretical developments in data science and analytics and the best practice for a wide range of applications. DSAA also features its cross-domain interactions and gap-bridging between academia and business for innovative industry and government data science and analytics.

Digital Object Identifier 10.1109/MCI.2020.2998229
Date of current version: 15 July 2020

Synergizing statistics (via ASA), computing and informatics/intelligence sciences (IEEE and ACM), DSAA sets up a high standard for its organizing committee, keynote speeches, submissions to main conference and special sessions. Leading researchers who have delivered keynote speeches at DSAA including physician Prof. Kyle Cranmer, machine learning expert Dr. Christopher Bishop, statisticians Prof. Michael I. Jordan, Prof. David Donoho, Prof. Serge Abiteboul and Prof. Bin Yu, robotics expert Prof. Hiroaki Kitano, deep learning founder Prof. Yoshua Bengio, and business data science leader Dr. Usama Fayyad. DSAA has a highly competitive rate for paper acceptance. DSAA has been widely recognized as a dedicated flagship in data science and analytics, such as by the Google Metrics[1] and the conference ranking made by the China Computer Foundation[2] as an influential event in the area.

The conference invites submission of papers describing innovative research on all aspects of data science and advanced analytics as well as application-oriented case studies that make significant, original, and reproducible contributions to improving the practice of data science and analytics in real-world scenarios. Visionary opinions, reviews and surveys are also welcome.

[1]https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_dataminganalysis
[2]https://www.ccf.org.cn/c/2019-04-25/663625.shtml

DSAA has widely involved industry, government and non-profit organizations. DSAA received sponsorship and support from multinational vendors and organizations such as Baidu, Google, Huawei, Infosys, SAP and TCS.

IEEE DSAA Tracks and Activities

DSAA solicits both theoretical and practical works on data science and advanced analytics through two main tracks: Research and Applications, in addition to a series of Special Sessions, Student Poster sessions, Industry Poster sessions, and an Industry Day, which form the essential features of DSAA. DSAA also facilitates its unique Trends and Controversies session, Invited Industry Talks session, Panel discussion, and four keynote speeches

from statistics, mathematics, informatics, computing, and business on data science research and applications. Both traditional and hands-on tutorials are offered in DSAA, in addition to additional data science schools and early-career researcher forums. As the only initiative in IEEE, ACM and ASA, the DSAA Next-generation Data Scientist Award (NGDS Award) calls for the nominations of data science role models, contributing to training and fostering next-generation data scientists. DSAA'2020 will also host a journal track on Data Science and AI in FinTech.

The **Research Track** solicits the latest, original and significant contributions related to foundations and theoretical developments of Data Science and Advanced Analytics. Topics of interest include but are not limited to:

- ❏ Data science foundations and theories
- ❏ Mathematics and statistics for data science and analytics
- ❏ Understanding data characteristics and complexities
- ❏ Data quality and misinformation
- ❏ Models, algorithms, and methods
- ❏ Optimization, inference, and regularization
- ❏ Infrastructures, and systems
- ❏ Evaluation, explanation, visualization, and presentation
- ❏ Survey and review

The **Application Track** solicits original, impactful and actionable application results of Data Science and Advanced Analytics across various disciplines and domains, including business, government, healthcare and medical science, physical sciences, and social sciences. Submissions address a real problem on real-life data that is reproducible ideally through a public git repository, providing inspiring results to policy-makers, end-users or practitioners or highlighting new practical challenges for researchers. Topics of interest include but are not limited to:

- ❏ Domain-driven data science and analytics practice
- ❏ Real-world applications and case studies
- ❏ Operationalizable infrastructures, platforms and tools
- ❏ Deployment, management, and decision-making
- ❏ System and software demonstrations
- ❏ Social and economic impact modeling
- ❏ Ethics, social issues, privacy, trust, and bias
- ❏ Reflections and lessons for better practice

DSAA **Special Sessions** substantially upgrade traditional workshops to encourage emerging topics in data science while maintain rigorous selection criteria, with accepted papers included in the main conference proceedings. Typical topics organized in the DSAA special sessions consist of mathematics and statistics for data science, data quality issues, data science for finance, health and medical data science, data science for cyber-physical systems, environmental and geo-spatial data analytics, misinformation and fake news, and social issues including privacy, security and trust.

Calls for Participating in IEEE DSAA Conferences

More information about **DSAA conferences** is available at www.dsaa.co. Specific information about **IEEE DSAA'2020** which will be held on 6–9 Oct. 2020 in Sydney Australia is available at dsaa2020.dsaa.co, and **IEEE DSAA'2021** will be held on 6–9 Oct. 2021 in Porto, Portugal. The **Call for hosting IEEE DSAA'2022 and DSAA'2023 proposals** is available at www.dsaa.co.

Publication Spotlight

Haibo He, Jon Garibaldi, Kay Chen Tan, Julian Togelius, Yaochu Jin, and Yew Soon Ong

CIS Publication Spotlight

IEEE Transactions on Neural Networks and Learning Systems

Neuromemristive Circuits for Edge Computing: A Review, by O. Krestinskaya, A. P. James and L. O. Chua, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 1, January 2020, pp. 4-23.

Digital Object Identifier: 10.1109/TNNLS.2019.2899262

"The volume, veracity, variability, and velocity of data produced from the ever increasing network of sensors connected to Internet pose challenges for power management, scalability, and sustainability of cloud computing infrastructure. Increasing the data processing capability of edge computing devices at lower power requirements can reduce several overheads for cloud computing solutions. This paper provides the review of neuromorphic CMOS-memristive architectures that can be integrated into edge computing devices. We discuss why the neuromorphic architectures are useful for edge devices and show the advantages, drawbacks, and open problems in the field of neuromemristive circuits for edge computing."

Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks, by B. Petro, N. Kasabov and R. M. Kiss, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 2, February 2020, pp. 358-370.

Digital Object Identifier: 10.1109/TNNLS.2019.2906158

"Spiking neural networks (SNNs) receive trains of spiking events as inputs. In order to design efficient SNN systems, real-valued signals must be optimally encoded into spike trains so that the task-relevant information is retained. This paper provides a systematic quantitative and qualitative analysis and guidelines for optimal temporal encoding. It proposes a methodology of a three-step encoding workflow: method selection by signal characteristics, parameter optimization by error metrics between original and reconstructed signals, and validation by comparison of the original signal and the encoded spike train. Four encoding methods are analyzed: one stimulus estimation [Ben's Spiker algorithm (BSA)] and three temporal contrast [threshold-based, step-forward (SW), and moving-window (MW)] encodings. A short theoretical analysis is provided, and the extended quantitative analysis is carried out applying four types of test signals: step-wise signal, smooth (sinusoid) signal with added noise, trended smooth signal, and event-like smooth signal. Various time-domain and frequency spectrum properties are explored, and a comparison is provided. BSA, the only method providing unipolar spikes, was shown to be ineffective for step-wise signals, but it can follow smoothly changing signals if filter coefficients are scaled appropriately. Producing bipolar (positive and negative) spike trains, SW encoding was most effective for all types of signals as it proved to be robust and easy to optimize. Signal-to-noise ratio (SNR) can be recommended as the error metric for parameter optimization. Currently, only a visual check is available for final validation."

IEEE Transactions on Fuzzy Systems

CFM-BD: A Distributed Rule Induction Algorithm for Building Compact Fuzzy Models in Big Data Classification Problems, by M. Elkano, J. Antonio Sanz, E. Barrenechea, H. Bustince, and M. Galar, *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 1, January 2020, pp. 163-177.

Digital Object Identifier 10.1109/MCI.2020.2998230
Date of current version: 15 July 2020

Digital Object Identifier: 10.1109/TFUZZ.2019.2900856

"Interpretability has always been a major concern for fuzzy rule-based classifiers. The usage of human-readable models allows them to explain the reasoning behind their predictions and decisions. However, when it comes to Big Data classification problems, fuzzy rule based classifiers have not been able to maintain the good tradeoff between accuracy and interpretability that has characterized these techniques in non-Big-Data environments. The most accurate methods build models composed of a large number of rules and fuzzy sets that are too complex, while those approaches focusing on interpretability do not provide state-of-the-art discrimination capabilities. In this paper, we propose a new distributed learning algorithm named CFM-BD to construct accurate and compact fuzzy rule-based classification systems for Big Data. This method has been specifically designed from scratch for Big Data problems and does not adapt or extend any existing algorithm. The proposed learning process consists of three stages: Preprocessing based on the probability integral transform theorem; rule induction inspired by CHI-BD and Apriori algorithms; and rule selection by means of a global evolutionary optimization. We conducted a complete empirical study to test the performance of our approach in terms of accuracy, complexity, and runtime. The results obtained were compared and contrasted with four state-of-the-art fuzzy classifiers for Big Data (FBDT, FMDT, Chi-Spark-RS, and CHI-BD). According to this study, CFM-BD is able to provide competitive discrimination capabilities using significantly simpler models composed of a few rules of less than three antecedents, employing five linguistic labels for all variables."

A Novel Classification Method From the Perspective of Fuzzy Social Networks Based on Physical and Implicit Style Features of Data, by S. Gu, Y. Nojima, H. Ishibuchi, and S. Wang, *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 2, February 2020, pp. 361-375.

Digital Object Identifier: 10.1109/TFUZZ.2019.2906855

"Many practical scenarios have demanded that we should classify unlabeled data more accurately based on both physical features (e.g., color, distance, or similarity) and implicit style features of data. As most extant classification algorithms classify unlabeled data based only on their physical features, they become weak in achieving expected classification results for many scenarios. To work around this drawback in this paper, a novel classification method (FuCM) from the perspective of fuzzy social network based on both physical and implicit style features of data is proposed. Based on the proposed fuzzy social network and its dynamics about fuzzy influences of nodes, FuCM comprises two stages. In its training stage, after the fuzzy social network has been built, it learns the topological structure, reflecting physical features and implicit style features of data by carrying out fuzzy influence dynamics in the built network. In its prediction stage, both physical and implicit style features of data are effectively integrated to yield the double structure efficiency characterized by fuzzy influences of nodes. FuCM classifies unlabeled data according to the strongest connection measure based on the proposed double structure efficiency. FuCM does not assume that both data distribution and the classification by physical features or by both physical and implicit style features of data must be known in advance. Thus, it is a novel unified classification framework in this sense. In contrast to all the nine comparative methods, FuCM experimentally demonstrates its comparable classification performance on most synthetic, UCI and KEEL datasets, which can be well classified based only on physical features of data. Furthermore, it displays distinctive superiority on five case studies where satisfactory classification certainly depends on both physical and implicit style features."

IEEE Transactions on Evolutionary Computation

An Experimental Method to Estimate Running Time of Evolutionary Algorithms for Continuous Optimization, by H. Huang, J. Su, Y. Zhang, and Z. Hao, *IEEE Transactions on Evolutionary Computation*, Vol. 24, No. 2, April 2020, pp. 275-289.

Digital Object Identifier: 10.1109/TEVC.2019.2921547

"Running time analysis is a fundamental problem of critical importance in evolutionary computation. However, the analysis results have rarely been applied to advanced evolutionary algorithms (EAs) in practice, let alone their variants for continuous optimization. In this paper, an experimental method is proposed for analyzing the running time of EAs that are widely used for solving continuous optimization problems. Based on Glivenko–Cantelli theorem, the proposed method simulates the distribution of gain, which is introduced by average gain model to characterize progress during the optimization process. Data fitting techniques are subsequently adopted to obtain a desired function for further analyses. To verify the validity of the proposed method, experiments were conducted to estimate the upper bounds on expected first hitting time of various evolutionary strategies, such as evolution strategy, standard evolution strategy, covariance matrix adaptation evolution strategy, and its

improved variants. The results suggest that all estimated upper bounds are correct. Backed up by the proposed method, state-of-the-art EAs for continuous optimization will have identical results about the running time as simplified schemes, which will bridge the gap between theoretical foundation and applications of evolutionary computation."

IEEE Transactions on Games

Procedural Puzzle Generation: A Survey, by B. De Kegel and M. Haahr, *IEEE Transactions on Games*, Vol. 12, No. 1, March 2020, pp. 21-40.

Digital Object Identifier: 10.1109/TG.2019.2917792

"Procedural content generation (PCG) for games has existed since the 1980s and is becoming increasingly important for creating game worlds, backstory, and characters across many genres, in particular, open-world games, such as Minecraft (2011) and No Man's Sky (2016). A particular challenge faced by such games is that the content and/or gameplay may become repetitive. Puzzles constitute an effective technique for improving gameplay by offering players interesting problems to solve, but the use of PCG for generating puzzles has been limited compared with its use for other game elements, and efforts have focused mainly on games that are strictly puzzle games, rather than creating puzzles to be incorporated into other genres. Nevertheless, a significant body of work exists, which allows puzzles of different types to be generated algorithmically, and there is scope for much more research into this area. This paper presents a detailed survey of existing work in PCG for puzzles, reviewing 32 methods within 11 categories of puzzles. For the purpose of analysis, this paper identifies a total of seven salient characteristics related to the methods, which are used to show commonalities and differences between techniques and to chart promising areas for future research."

IEEE Transactions on Cognitive and Developmental Systems

DeepFeat: A Bottom-Up and Top-Down Saliency Model Based on Deep Features of Convolutional Neural Networks, by A. Mahdi, J. Qin, and G. Crosby, *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 12, No. 1, March 2020, pp. 54-63.

Digital Object Identifier: 10.1109/TCDS.2019.2894561

"A deep feature-based saliency model (DeepFeat) is developed to leverage understanding of the prediction of human fixations. Conventional saliency models often predict the human visual attention relying on few image cues. Although such models predict fixations on a variety of image complexities, their approaches are limited to the incorporated features. In this paper, the authors aim to utilize the deep features of convolutional neural networks by combining bottom-up (BU) and top-down (TD) saliency maps. The proposed framework is applied on deep features of three popular deep convolutional neural networks (DCNNs). The authors exploit four evaluation metrics to evaluate the correspondence between the proposed saliency model and the ground-truth fixations over two datasets. The results demonstrate that the deep features of pretrained DCNNs over the ImageNet dataset are strong predictors of the human fixations. The incorporation of BU and TD saliency maps outperforms the individual BU or TD implementations. Moreover, in comparison to nine saliency models, including four state-of-the-art and five conventional saliency models, their proposed DeepFeat model outperforms the conventional saliency models over all four evaluation metrics."

IEEE Transactions on Emerging Topics in Computational Intelligence

Complex-Valued Neural Networks With Nonparametric Activation Functions, by S. Scardapane, S. V. Vaerenbergh, A. Hussain, and A. Uncini, *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 4, No. 2, April 2020, pp. 140-150.

Digital Object Identifier: 10.1109/TETCI.2018.2872600

"Complex-valued neural networks (CVNNs) are a powerful modeling tool for domains where data can be naturally interpreted in terms of complex numbers. However, several analytical properties of the complex domain (such as holomorphicity) make the design of CVNNs a more challenging task than their real counterpart. In this paper, we consider the problem of flexible activation functions (AFs) in the complex domain, i.e., AFs endowed with sufficient degrees of freedom to adapt their shape given the training data. While this problem has received considerable attention in the real case, very limited literature exists for CVNNs, where most activation functions are generally developed in a split fashion (i.e., by considering the real and imaginary parts of the activation separately) or with simple phase-amplitude techniques. Leveraging over the recently proposed kernel activation functions, and related advances in the design of complex-valued kernels, we propose the first fully complex, nonparametric activation function for CVNNs, which is based on a kernel expansion with a fixed dictionary that can be implemented efficiently on vectorized hardware. Several experiments on common use cases, including prediction and channel equalization, validate our proposal when compared to real-valued neural networks and CVNNs with fixed activation functions."

IEEE Collabratec™

Bright Minds. Bright Ideas.

Introducing IEEE Collabratec™

The premier networking and collaboration site for technology professionals around the world.

IEEE Collabratec is a new, integrated online community where IEEE members, researchers, authors, and technology professionals with similar fields of interest can **network** and **collaborate**, as well as **create** and manage content.

Featuring a suite of powerful online networking and collaboration tools, IEEE Collabratec allows you to connect according to geographic location, technical interests, or career pursuits.

You can also create and share a professional identity that showcases key accomplishments and participate in groups focused around mutual interests, actively learning from and contributing to knowledgeable communities. All in one place!

Network.
Collaborate.
Create.

Learn about IEEE Collabratec at
ieee-collabratec.ieee.org

IEEE

©ISTOCKPHOTO.COM/LEGART

A Review of Off-Line Mode Dataset Shifts

Carla C. Takahashi

Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, BRAZIL, Loggi Tecnologia, São Paulo, BRAZIL

Antônio P. Braga

Department of Electronic Engineering, Federal University of Minas Gerais, Belo Horizonte, BRAZIL

***Abstract*—**Dataset shifts are present in many real-world applications, since data generation is not always fully controlled and is subject to noise, degradation, and other natural variations. In machine learning, the lack of regularity in data can degrade performance by breaching error constraints. Different methods have been proposed to solve shifting problems; however, shifts in off-line learning mode are not as well examined. Off-line shifts consist of problems where drifts occur only with unlabeled data. Most methods aimed at dataset shifts consider that new labeled data can be received after training, which is not always the case. Here, a review on dataset shift characteristics and causes is presented as a tool for the analysis and implementation of machine learning methods targeting off-line mode dataset shift problems. In this context, a relationship between statistical learning risk functions and error degradation due to variation in data distribution was straightforwardly derived. Moreover, this paper provides a consistent survey of recent popular machine learning methods that address off-line mode dataset shift problems, focusing on the main characteristics of unlabeled data shifts.

Digital Object Identifier 10.1109/MCI.2020.2998231
Date of current version: 15 July 2020

Corresponding Author: Carla C. Takahashi (ctakahashi@ufmg.br)

1556-603X/20©2020IEEE

I. Introduction

Most machine learning algorithms are based on the principle that a given learning set $\tau = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ is representative of the underlying process to be modeled. Thus, the algorithms do not change once they are sampled. The inductive principle, often based on empirical risk minimization, assumes that a model $f(\mathbf{x}, \mathbf{w})$ can be obtained from τ. Sampled data is expected to embody all the information needed to obtain the vector of parameters $\mathbf{w}$ that characterizes $f(\mathbf{x}, \mathbf{w})$, which is expected to be valid in the domain $\mathbf{x} \in X$. In current problems, this fundamental principle of statistical inference may not be valid as a general assumption. *Dataset shift* may occur when the generator of sampled data undergoes a change, or drift, causing new samples to be given by a different function. Thus, the density distribution of the labeled training data, $P_{\text{train}}(\mathbf{x}_{\text{train}}, y_{\text{train}})$ might differ from the density of the test data $P_{\text{test}}(\mathbf{x}_{\text{test}}, y_{\text{test}})$.

$$P_{\text{train}}(\mathbf{x}_{\text{train}}, y_{\text{train}}) \neq P_{\text{test}}(\mathbf{x}_{\text{test}}, y_{\text{test}}) \tag{1}$$

where the training set comprises pairs $(\mathbf{x}_{\text{train,labelled}}, y_{\text{train,labelled}})$, and the test set is formed only by $(\mathbf{x}_{\text{test,unlabelled}})$, since y_{test} is not available at test time and will not be considered here.

While drifts are found in online data due to seasonality or periodicity effects, they also appear from changes in user preferences, hardware or software faults, aging effects of sensors, and more. However, discrepancies between training and test data are also observed in off-line systems [1], [2], in which new labeled cases are unavailable. Off-line shifts occur when a model is created to work in a specific process but is estimated using data from a different source. This happens when industrial plant models are estimated according to data from a similar system before the implementation of the actual factory, or in mobile ad-hoc networks (MANETs) with configurations based on simulations [3]. Another problem occurs in medical diagnoses, which tend to be biased since training data often has a greater proportion of diseased individuals than the overall population. In data streams where labeled data is only available initially and later the stream is exclusively unlabeled, the shift problem is known as initially labeled nonstationary streaming (ILNS) [4]. In the literature, several approaches attempt to account for uncertainties and differences between training and test data, such as transfer learning, domain adaptation, semi-supervised learning, and transductive learning. However, only a limited number of methods can address off-line shifts.

Dataset shift is complex and can be interpreted in various ways; thus, this paper complements reviews in related subjects, such as [1], [2], [4]–[8]. The works in [9], [10] provide a foundational understanding of the subject. Cross-validation on Dataset Shift problems is evaluated by [11]. Here, we focus on shifts that occur off-line in unlabeled data when new target values are not available. Also, since comprehensive analysis of real-world problems does not exist in the literature [8], we present real-world cases with specific shifts in multidisciplinary fields to evidence the importance of the subject.

II. Degradation of the Empirical Risk in Risk-Based Methods

Inductive systems for off-line learning, which are based on the principle of empirical risk minimization, are often affected by Dataset Shift. Learning methods usually require a generator of input vectors $\mathbf{x}$, which follows a given probability distribution function $P(\mathbf{x})$, and for supervised problems, an output $y = f_g(\mathbf{x})$ sampled according to a conditional distribution function $P(y \mid \mathbf{x})$ to estimate a model $f(\mathbf{x}, \mathbf{w})$. This learning method is achieved by an algorithm and a machine that implements the set of functions $f(\mathbf{x}, \mathbf{w})$, with $\mathbf{w} \in W$, for instance, an artificial neural network with backpropagation learning. The training dataset should be large enough to guarantee representativeness, but this is not always the case [12], [13].

To assess the statistical validity of a given model, risk and loss functions are used to measure differences between a system's response and the estimations provided by the learning machine. Considering a loss function $\mathcal{L}$, the expected risk would be given as [14]:

$$R_{\text{exp}}(\mathbf{w}) = \iint \mathcal{L}(y, f(\mathbf{x}, \mathbf{w}))\, dP(\mathbf{x}, y) \tag{2}$$

where $\mathcal{L}$ is the loss function and $P(\mathbf{x}, y)$ is the joint distribution of $\mathbf{x}$ and y.

In that this joint distribution is usually unknown and only a limited set of input/output pairs $(\mathbf{x}, y)$ is available for learning, an empirical risk function, $R_{\text{emp}}(\mathbf{w})$, is adopted in eq. 3.

$$R_{\text{emp}}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, f(\mathbf{x}_i, \mathbf{w})) \tag{3}$$

where N is the number of instances of the training dataset.

Model selection is performed with empirical risk minimization considering the training set distribution. The joint error, then, is equal to the expectation of the loss function between the true targets and the selected model output. Given that $f(\mathbf{x}, \mathbf{w})$ regards the joint distribution $P(\mathbf{x}, y)$ at learning time, the parameters that define $P(\mathbf{x}, y)$ are expected to be the same for both training and test data to guarantee convergence constraints. Differences in distributions might increase the estimation error. For instance, the empirical risk for a quadratic loss function is $R_{\text{emp}}(\mathbf{w}) = (1/N)\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2$, and its estimation error is shown in eq. 4 [15].

$$E\left[(y - f(\mathbf{x}, \mathbf{w}))^2\right] = E\left[(y - E[y|\mathbf{x}])^2\right] + E\left[(f(\mathbf{x}, \mathbf{w}) - E[y|\mathbf{x}])^2\right] \tag{4}$$

where $E\left[(y - E[y|\mathbf{x}])^2\right]$ is the sampling error and $E\left[(f(\mathbf{x}, \mathbf{w}) - E[y|\mathbf{x}])^2\right]$ is the approximation error.

If a shift occurs in $P(y, \mathbf{x})$, the expected value $E[y|\mathbf{x}]$ also shifts, which leads to degradation in sampling and approximation errors, given by eq. 4. This situation will happen when the training probability density function (PDF) differs from the test PDF, $P_{\text{train}}(y, \mathbf{x}) \neq P_{\text{test}}(y, \mathbf{x})$. Thus, methods based on the assumption that training and test data are independent and identically distributed might be vulnerable to Dataset Shifts [9], [10].

III. Types of Dataset Shifts

A. Covariate Shift

A *covariate shift* is observed when future values of covariates $\mathbf{x}$ differ from past observations [16], [17]. Covariate shifts occur only when the distribution of $\mathbf{x}$ suffers a drift and all other probabilities remain unchanged, thus, $P_{\text{train}}(y|\mathbf{x}) = P_{\text{test}}(y|\mathbf{x})$ but $P_{\text{train}}(\mathbf{x}) \neq P_{\text{test}}(\mathbf{x})$. Therefore, the relation between the target y and the covariate x remains the same, but the covariate distribution $P(\mathbf{x})$ changes, as shown in the histograms in Fig. 1(a). A purely discriminative solution for classification problems with covariate shift can be implemented by searching for all model parameters according to an integrated optimization approach [18]. Such a solution shows that exponential models, such as Gaussian kernels, lead to a convex setting, which allows for simplified optimization with Newton gradient descent methods, resulting in a kernel logistic regression and an exponential model classifier.

In eq. 2, the risk changes with covariate shift as the integration of the loss function $\mathcal{L}$ is performed with respect to a different input space. If the model risk is calculated over m different datasets, then it can be understood as a random variable $\mathbf{R}$. Considering the Central Limit Theorem and that $\mathbf{R}$ has a quadratic loss function, the risk mean and variance are given by eq. 5 and 6.

$$E[\mathbf{R}] = E\left[E\left[(y - f(\mathbf{x}, \mathbf{w}))^2\right]\right] = \frac{m-1}{m}\mu_2 \tag{5}$$

$$\text{var}(\mathbf{R}) = \text{var}\left[E\left[(y - f(\mathbf{x}, \mathbf{w}))^2\right]\right] = \frac{(m-1)^2}{m^3}\mu_4 - \frac{(m-1)(m-3)}{m^3}\mu_2^2 \tag{6}$$

where

$$\mu_n = \int (y - f(\mathbf{x}, \mathbf{w}))^n \, dP(\mathbf{x}, y) \tag{7}$$

and $P(x, y) = P(y|\mathbf{x})\,P(\mathbf{x})$ is the joint probability density. Thus, the expected value for risk might change and its variance can increase, resulting in a larger error due to dependencies on $P(\mathbf{x})$.

A typical example associated with covariate shift is the brain-computer interface (BCI), which is based on electroencephalograms (EEG). In EEGs, electrode placement, attention level, user fatigue, and other factors influence brain activity scanning, which causes signals to be highly variable [20]–[22] Due to the complexity of training protocols and pre-processing procedures, BCI systems are often trained off-line. Moreover, adaptive classifiers and transfer learning tend to perform well [23]. The example in Fig. 1 is from the Dataset IVc of the BCI Competition III [19], [24]. Training and test sets were sampled at different times, consisting of discriminating between the motor imagery of left hand movement and right foot movement using EEG signals. Ensemble-based classifiers used to solve this problem tend to outperform single models when they are segmented into local models according to the covariate input space [23], [25].

B. Prior Probability Shift

Prior probability shifts occur when the target distribution $P(y)$ differs between training and test data, affecting models that have assumptions of causal relationships in data. This is relevant in models that infer conditional probability $P(y|\mathbf{x})$ through $P(\mathbf{x}|y)\,P(y)$, such as naive Bayes classifiers, which consider the conditional probability of the training set to classify the test data. If covariates x depend on targets y and the target distribution $P(y)$ shifts, the results given by the model might differ from expected values [9], [10], [26], [27]. In classification

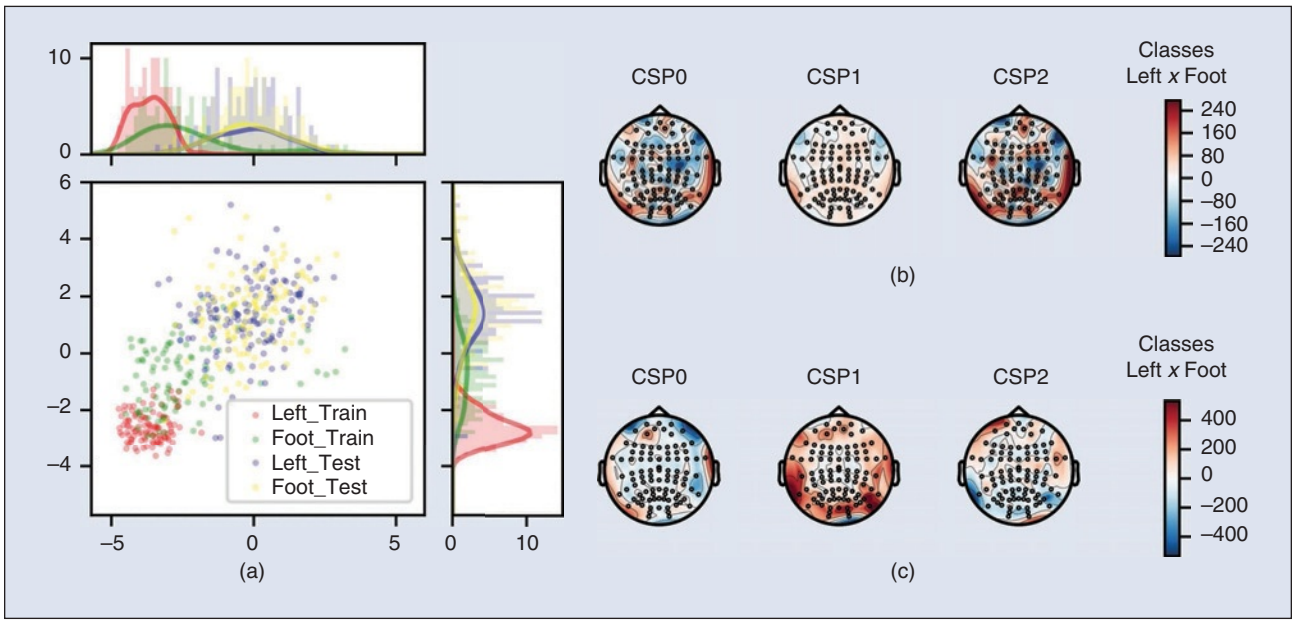


FIGURE 1 The spatial distribution of brain activity represented with common spatial patterns (CSP) of each class differs greatly between training and test data [19]. (a) Spatial distribution of the two principal components for both classes in training and test sets. (b) Training set spatial representation of brain activity. (c) Test set spatial representation of brain activity.

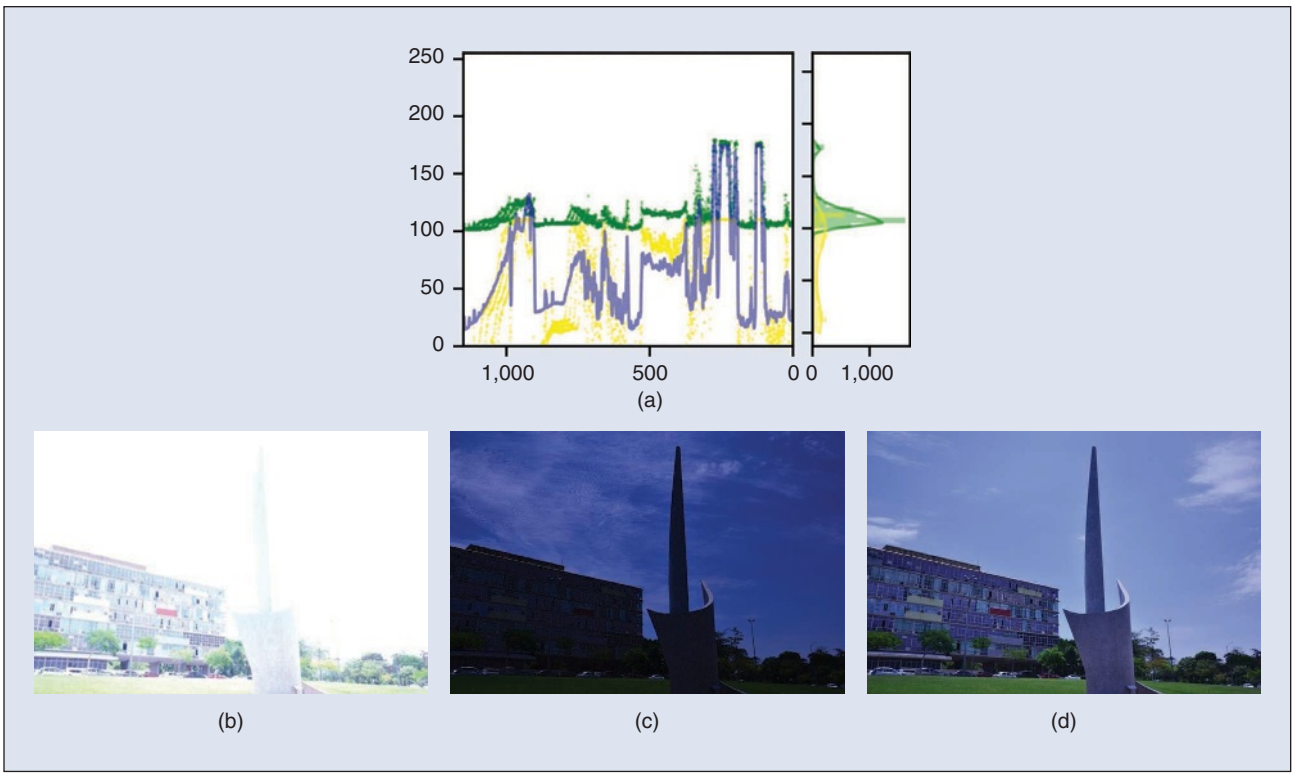


FIGURE 2 The information distribution depends on the target value, characterizing the prior probability shift. Here, training data is represented by (b), and test data by (c). (a) Normalized luminance of a section of the image (saturated values were discarded). The blue line is the average of the real image (d), the training data is yellow, and the test data is green. (b) Overexposed training image. (c) Underexposed test image. (d) Real image with correct exposure.

problems, prior probability shifts can appear when the class balance in the training set differs from the test set [7].

The estimation of luminance across an image can be used to establish background baselines to detect new objects on the scene. Histograms in Fig. 2(a) illustrate a case in which the response variable differs between training and test images. Image capture systems often face overexposure or light saturation problems, which cause bright or dark spots in the image with low or even no information. In any light sensing system, the sensors are subject to saturation, where the scene might have more information, but it cannot be translated by the sensors. A prior probability shift occurs here, since information between test and training differ according to the luminance output, as shown in Fig. 2. Practical examples are remote sensing applications with UAVs and image classification, where drift might occur due to various factors [28]–[33].

C. Concept Shift

Concept is an abstract interpretation of information that is learned by a machine, such as the relation between a given covariate and its class. Thus, in terms of probability density distributions, concepts are related to conditional probabilities [34]–[36]. In *concept shift* problems, both prior probability and $P(\mathbf{x})$ are usually unaltered; however, shifts occur in the relationship between covariates and targets. In this context, $P_{\text{train}}(y|\mathbf{x}) \neq P_{\text{test}}(y|\mathbf{x})$ implies changes in data generators, thus, $P_{\text{train}}(y, \mathbf{x}) \neq P_{\text{test}}(y, \mathbf{x})$. Concept drifts are usually categorized in two types [1], as shown in Fig. 3, defined as:

- ❏ Real concept drift: The conditional probability $P(y|\mathbf{x})$ of the output given that the input differs between training and test data; therefore, a shift occurs in the process that generates data.

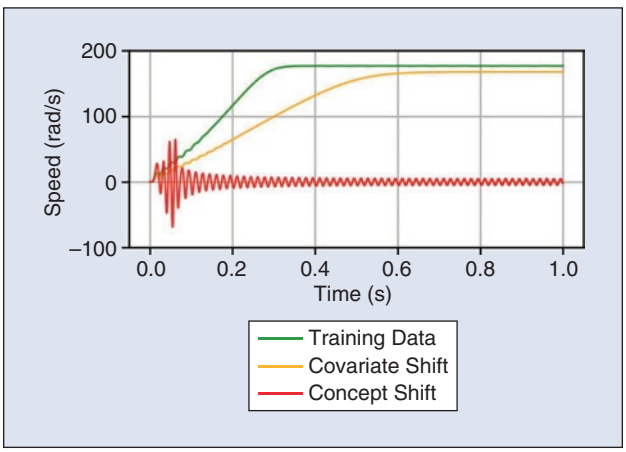


FIGURE 3 Graph showing the speed of a three-phase electric motor. The training data is the motor speed for a controlled training set point. The orange data is the motor working with a covariate shift, that is, a shifted set point. The red plot is the output of the system for the same set point of the training data, but the motor has a phase fault short-circuit.

❑ Virtual concept drift: The joint probability $P(y, \mathbf{x})$ differs between training and test data because $P_{\text{train}}(\mathbf{x}) \neq P_{\text{test}}(\mathbf{x})$, which is a covariate shift.

Smart industries have contributed to an increase in the number of sensors in machines, which allows self-sensing and integration with data analytics tools [37]. Smart monitoring and control of machinery tend to become increasingly important with the development of Industry 4.0. In this context, the example in Fig. 3 shows how the speed of motors varies according to a different operation point or phase fault. In this case, dataset shift is considered when the motor control system should react to different working conditions and possible faults.

IV. Causes of Dataset Shift

A. Sample Selection Bias

A common cause of dataset shift is the selection of uniform, or biased, training sets. Consider that the training data is chosen according to a sampling decision variable s and its probability density is given by eq. 8. In this equation, the decision variable is dependent on targets y and covariates x, which causes a bias in the data used for training and, therefore, also on the probability density P_{train}. Meanwhile, test data and its probability density P_{test} are not subject to this same bias, which reflects a shift between training and test data, as represented in eq. 8 to 10 [9], [10].

$$\begin{aligned} P_{\text{train}}(y, \mathbf{x}) &= P(y, \mathbf{x} | s = 1) \\ &= P(s = 1 | y, \mathbf{x})\, P(y | \mathbf{x})\, P(\mathbf{x}). \end{aligned} \tag{8}$$

$$\begin{aligned} P_{\text{test}}(y, \mathbf{x}) &= P(y, \mathbf{x}) \\ &= P(y | \mathbf{x})\, P(\mathbf{x}). \end{aligned} \tag{9}$$

$$P_{\text{train}}(y, \mathbf{x}) \neq P_{\text{test}}(y, \mathbf{x}). \tag{10}$$

In this case, the sample is selected when $s = 1$ and discarded when $s = 0$. The types of shift that might occur in this scenario are: (1) covariate shift when $P_{\text{train}}(s = 1|\mathbf{x})$; (2) prior pr shift when $P_{\text{train}}(s = 1|y)$; and (3) any type when $P_{\text{train}}(s = 1|y, \mathbf{x})$, which characterizes missing not at random (MNAR) sampling systems [9], [10].

Bias in estimators may be induced by unequal selection probabilities at any stage of sampling. Consistent estimators can be obtained by weighting the model estimation with reciprocals of selection probabilities at each stage [38]. Bias problems are often seen in off-line scenarios, where labeled data is difficult to obtain, such as diagnostic and clinical studies [39]–[42]. In these cases, training data can be biased because exams are often performed only on diseased subjects and healthy subjects are undersampled. Furthermore, prior probability may differ between training and test data if the criteria used to choose patients changes.

Differences in balance between training and test sets in classification problems are a prior probability shift scenario. Thus, methods to solve this shift can be used in imbalanced datasets problems alongside with rebalancing strategies [26], [27]. For example, class distribution estimation can be solved by quantification learning methods [7]. However, appropriate balancing strategies should be used to guarantee the model quality. For instance, in Fig. 4, the hepatitis dataset [43] was rebalanced with an inadequate MNAR approach and resulted in a poor model. To minimize possible dataset shifts, classes can be balanced through a partition of the dataset using "Distribution optimally balanced stratified cross-validation" [41]. Likewise, intrinsic characteristics of the data can be used to rebalance data [44].

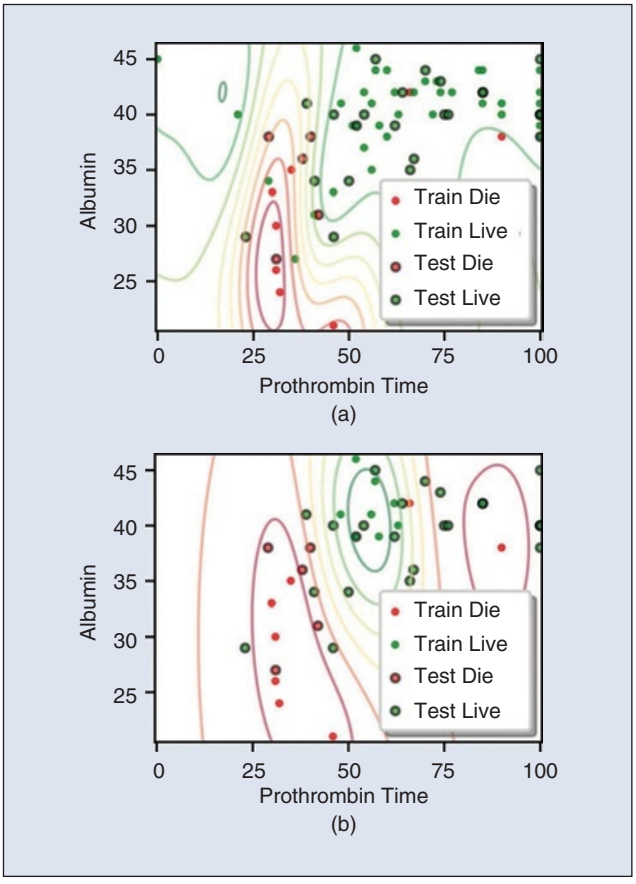


FIGURE 4 Projection of the features albumin and prothrombin time from the hepatitis dataset. Class Live (green) has significantly more instances than class Die (red) in (a). The undersampling process that was dependent on values of other features, such as malaise, age, and histology, resulted in the misspecified models in (b), with the class contours significantly different. (a) Original imbalanced datasets with a Gaussian SVM classifier, with 85% test accuracy (black border dots). (b) Undersampled, rebalanced data with a Gaussian SVM classifier, with 63% test accuracy.

B. Domain Shift

An example of *domain shift* is illustrated in Fig. 5(a), where the covariate x is not the actual latent variable x_0, but instead is an observation given by a function $x = f(x_0)$. When the function $f(x_0)$ is altered, the covariate x perceived by the model is different even if the latent variable x_0 remains the same. Thus, domain shift characterizes changes in the measurement system, metrics, or even the description of the data generator, such as currency devaluation in pricing predictions or the visual classification of images with different lighting. In image classification, inputs are pictures from real-world scenes and capture methods are

observation functions of the scenes; hence different lighting settings could result in shifts, as in Fig. 5(b). Photographs taken of the same scene can look quite different, depending on shutter speed, aperture, and illumination settings [45].

C. Source Component Shift

The source of any real-world data is subject to variations. Moreover, data is often generated from multiple sources, and each of them is prone to disturbances. In this context, the overall distribution of the covariates can easily differ between training and test data; thus, model selection becomes difficult. Three kinds of shift may appear in source components: (1) *Mixture component shift* may appear in data generated by several sources, while the origin source of any instance is unknown, so prior probability shift occurs when there is a behavioral change of a source; (2) *Mixing component shift* is similar to the previous case, however, observed data is aggregated so individual instances are not observed and only mean values are available; and (3) *Factor component shift* occurs in problems in which the probability of the data is influenced by factors that can be decomposed into (A) form, which is the shape of a distribution, that is, a Gaussian curve; and (B) strength, which is the intensity and spread of a curve, such as the maximum value of a Gaussian curve or its standard deviation. This kind of shift happens when the form of the factor remains constant but its strength changes between training and test data [10].

The EEG problem is a typical example of the source component shift. Since brain activity is extremely complex and noisy, several electrodes need to be used, which causes data dimensionality to be high. Therefore, filters and feature extraction methods are commonly used [46], which results in shifts due to data aggregation. Shifts also occur in the EEG procedure itself, since each electrode measures average stimuli in a brain region. Thus, small deviations in electrode placement might lead to dataset shifts.

V. Dataset Shift Time-Space Patterns

One of the most common assumptions in this scenario is that the machine can receive labeled and unlabeled data at any moment in time. Thus, it is straightforward to define drifting characteristics of temporal nature with a data PDF dependent of time $P(y, \mathbf{x}, t) = P((y|\mathbf{x}), t)\, P(\mathbf{x}, t)$. Shifts are classified in time-space patterns according to a combination of their transitory characteristics' duration and transience, according to Fig. 6.

The duration of shifts can be defined as the time taken for data to leave an initial stable state A and reach the final stable state B [47], if data leaves state A in $t = s$ and reaches B in $t = e$, then the duration of the drift is:

$$D = e - s. \tag{11}$$

The traditional two types of drift, abrupt and gradual, can be defined according to a threshold δ that depends on the application of the following rule, as presented in [47]:

$$\text{Type} = \begin{cases} \text{Abrupt} & \text{if } D \leq \delta, \\ \text{Gradual} & \text{if } D > \delta. \end{cases} \tag{12}$$

Data timestamps are often not available on off-line learning systems, so it might not be possible to make assumptions over shift progressions, persistence, or transience. Likewise, context sequences may not be known [2], [4].

In distributed systems, time dependencies can be replaced by spatial ones, that is, instead of analyzing data according to a temporal variable, a positional variable is used. This scenario is becoming more relevant with smart sensor networks and smart distributed systems in general, but also appears in behavioral geography, computational social science, and market research.

A. Abrupt Shift

Abrupt shift occurs when data behavior suddenly changes, such as a fault from a sensor. In this case, the probability distribution of the data $P(y, \mathbf{x}, t)$ is different from $P(y, \mathbf{x}, t+1)$. In linear systems theory, this would be equivalent to a step disturbance.

An example of an abrupt shift is the implementation of a system trained with a similar case or simulation, which is often the case in mobile, ad-hoc networks [3]. Abrupt shifts also occur in other system identification and control problems, such as a

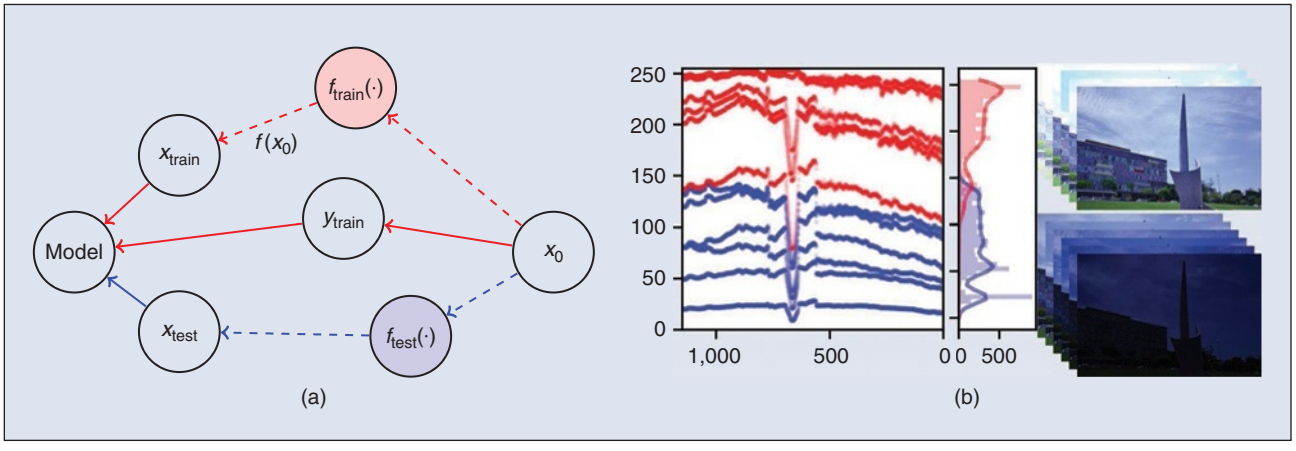


FIGURE 5 Representation of an image capture problem where a domain shift occurs due to variation in camera settings. The training data is in red and the test is in blue. (a) Diagram representing domain shift causal model. (b) Average luminance of the pictures columnwise.

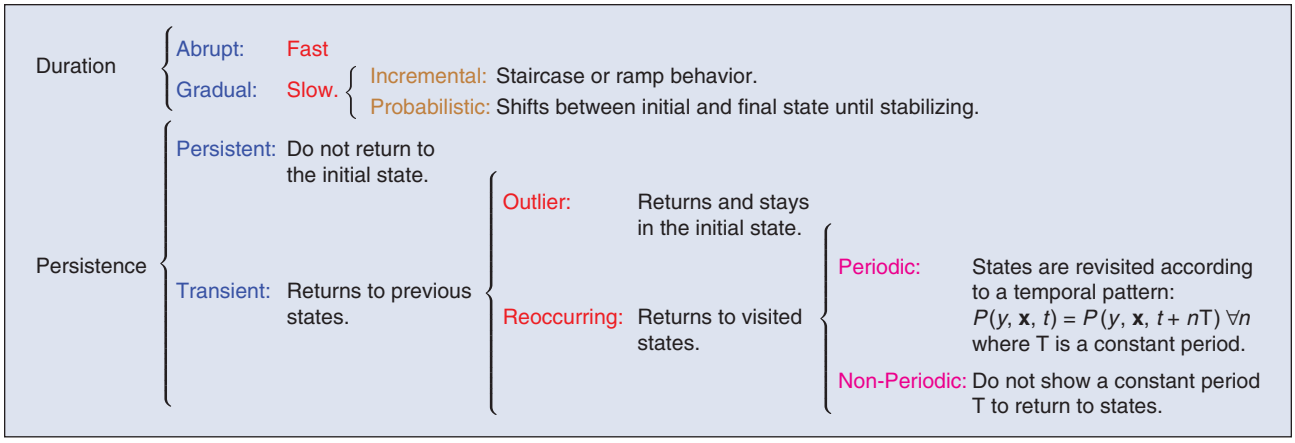


FIGURE 6 Dataset shift patterns according to their speed and persistence.

learning machine for an audio system trained with data collected in an indoor environment but tested and applied outdoors, as shown in Fig. 7.

B. Gradual Shift

Gradual shifts are subtle and can be perceived as a trend in data. In real-world scenarios, gradual shifts might be caused by the aging of sensors or from thermal effects. Specifically, an incremental shift is a subset of gradual shifts which occurs when a trend appears in the data as a "staircase." Each step is an abrupt change, but overall, there is a gradual shift trend. However, gradual shifts might not always be steady, whereas probabilistic shifts occur similar to atomic transitions in quantum mechanics. In this case, data is initially in a stable state A but starts gradually oscillating between states A and B, spending less time in state A and more in B, until the data stabilize in B.

A practical problem would be the identification of people in pictures from multiple stages of their lives, as in Fig. 8, but with a limited set of labeled pictures. This could be applied for face identification in social media pictures or even in missing children cases [48], [49].

Similarly, if data is obtained from multiple sources, targets might differ for the same covariate, depending on the source that provided it. For instance, in market research problems, consumer patterns change depending on their living area or niches [51]. Therefore, shifts occur geographically when using a model trained for a specific market across the others. Another example is the TeleECG of the Federal University of Minas Gerais [52], which is part of a semi-automatic health program. In this system, electrocardiogram exams are made in hundreds of remote locations and diagnostics evaluation is centralized with automatic triage. In that several exams are made simultaneously, and conditions may change due to the ability of nurse technicians or the condition of equipment, shifts may appear both in time and location.

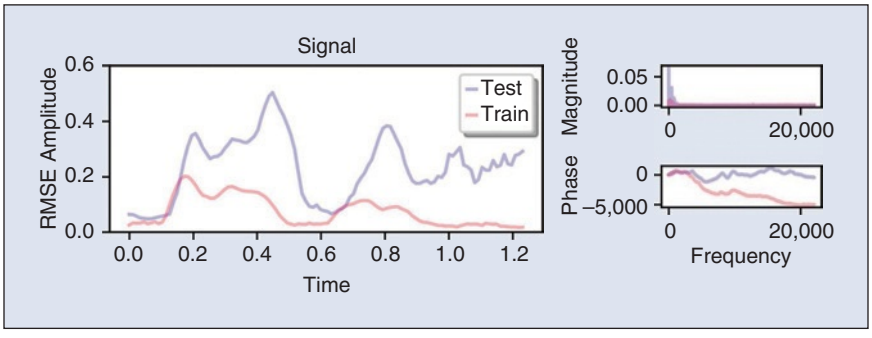


FIGURE 7 Example of audio data collected indoors (red) and outdoors (blue), where street noise had an extensive impact on the overall data. A machine trained with the indoor data should compensate for possible background noise.

VI. Learning Strategies for Shifting Datasets

Most learning strategies for dataset shift are based on two approaches: active and passive [4]. Essentially, active methods detect shifts and adapt learning according to changes in data, thus models can be optimized for data when necessary. Meanwhile, passive approaches are robust overall to shifts that may occur, and models are made with an underlying assumption that data will change. However, other possibilities to solve dataset shift problems exist, such as the methodology in [53], which extracts the rules that govern the shifts and creates classification models according to them. This section presents some methods capable of accounting for shifts in off-line scenarios.

A. Shift Detectors

Shift detectors are one of the main elements of an active approach, since they observe the data and, through tests or other comparison methodologies, indicate whether a change in data has occurred. Among shift detection methods, statistical hypothesis testing of differences in multivariate data can be implemented in several ways, such as adaptations of Kolmogorov–Smirnov, Wilcoxon, or Fisher's exact tests [54], [55]. Similarity and dissimilarity measures, such as Kullback–Leibler, Jensen–Shannon divergencies, or Jaccard distance measures [56], are also straightforward approaches.

However, multivariate statistical testing can be computationally expensive, so dimensionality reduction along with maximum mean discrepancy tests, Kolmogorov–Smirnov tests with Bonferroni correction, and chi-squared tests are possible approaches [57]. Still, most of these statistical tests depend on the underlying distribution. In addition, in the case of the drift causing small observable changes by the statistical method, the detector tends to perform poorly [58]. A semi-supervised approach monitors changes in the classifier's confidence to detect shift. The approach proposed to detect multi-class novelty and concept shifts using dynamic windows to mitigate the trade-off between performance during stable periods and delayed detection [59], [60]. The method suffered from low execution speed, so dynamic programming and selective executions of the detection module were implemented as an improvement [59]. The detection and classification framework proposed is an ensemble of k-nearest neighbors (k-NN) classifiers. Each test instance is classified, and the classifier's confidence score is stored in a vector. Then, a sliding window approach is used to detect significant changes. If a change is detected, the framework determines a new chunk boundary of scores and updates the classifier. The dynamic implementation improves the framework speed; however, the method still estimates distributions and implements a recursive calculation of change detection scores. Despite the sporadic calls of the detector, burdensome calculations may cause the method to be prohibitively slow in streaming scenarios.

A series of methods employing exponentially weighted moving average (EWMA) for the detection of some dataset shift problems, particularly covariate shift, have been proposed in the literature [61], [62]. These methods use statistical process control charts to detect shifts in the input covariate. The shift detection based on EWMA works in two stages: (1) a control chart is used to detect the dataset shift in the data stream, and (2) a statistical hypothesis test is used retrospectively in the testing phase to validate the shift detected by the first stage. During the test phase, input data is continuously monitored by the EWMA chart and, when control limits are exceeded, the method considers that a shift has occurred. For example, in applications with EEG-based BCIs, detection based on EWMA was successful in identifying covariate shifts in motor imagery-based EEG with principal component analysis [20], [21].

In unlabeled data streams, a possible approach is to use unsupervised anomaly detection to define transitions of concepts [63]. However, this approach requires meticulous work as search window sizes would need to be well tuned depending on the speed pattern and intensity of the shift, which could make this approach unfeasible. The Plover [64] algorithm facilitates the detection of drift in unsupervised streams using statistical moments to measure data stability, but it requires independent and identically distributed (i.i.d.) input data. For unlabeled data streams, a semi-supervised diversity measure was proposed as a drift detection method [65], which evaluates the diversity of a pair of classifiers. Drifts can be detected when the disagreement between them increases, yet this approach is heavily dependent on the distance between decision surfaces.

From a data-mining standpoint, prior probability shifts can be tracked and estimated with parametric distribution estimation, with the assumption that data comprises mixtures of distributions [26], [27]. Still, this is a strong assumption with few usable cases. From this perspective, "utilities," which are assessed by the frequency an instance occurs in the dataset, can be extracted from a data stream and tracked to detect drifts [66]. This method is similar to an unsupervised detection of drifts in a stream based on the importance of each pattern within a window. Importance measures, however, are difficult to define, and poor choices may hinder the model.

More recently, regional concept drift detection strategies were proposed to detect drifts that occur in local regions but do not cause significant overall changes in data [67], [68]. A local drift degree (LDD) measures regional drifts by determining whether the local density discrepancies of two i.i.d. distributions are similar [69]. Similarly, a shift detector based on nearest neighbor-based density variation (NN-DVI) was proposed according to a k-nearest neighbor-based space-partitioning schema (NNPS) [68]. This schema improves the sensitivity of regional drift detection by transforming unmeasurable discrete data instances into a set of shared subspaces for density estimation. Due to the local metrics based on windows, they are slower compared to other shift detectors, with marginal improvements depending on shift characteristics.

B. Passive Approaches

Passive approaches assume that data will change; thus, the model itself adapts for every new data it receives irrespective of whether a drift occurred. Despite being generally robust to gradual or incremental shifts, passive methods in off-line cases might perform differently depending on the type and constraints of a shift. Some of the main passive approaches for off-line problems are based on transductive and semi-supervised learning models, described in Subsection VI.C.

The use of intrinsic local mode functions in a model can be used to adapt to shifts in time [70]. This method is based on the empirical decomposition of the data, allowing for a well-behaved Hilbert transform for each function. As it concerns local time scales of data, the model is redefined at each instant,

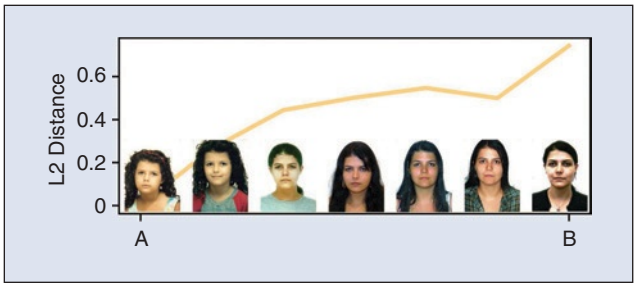


FIGURE 8 Incremental shift represented by pictures of a person in her infancy (initial state *A*), childhood, teenage years, and adulthood (final state *B*). L2 distance to state *A* was calculated with OpenFace [50].

Despite being generally robust to gradual or incremental shifts, passive methods in off-line cases might perform differently depending on the type and constraints of a shift.

which can be considered a passive adaptive approach. Another early strategy was to modify data windows to minimize the estimated generalization error [71], which can be used to detect drifts, to some extent, according to the window size. However, it is a passive adaptive approach since window sizes are adjusted with every batch input. This method is similar to cross-validation, in that several models with many possible window sizes are trained, then the size with a lower estimated error is chosen. Through simple analysis of the problem, the parameters of multinomial random variables can be estimated for the first and second moments according to principles of stochastic learning [72].

Several methods aim to solve dataset shift problems by weighting data-intrinsic characteristics, such as their estimated probability densities. Usually these methods can be used in off-line scenarios as they incorporate information from unlabeled data into the learning model by comparing it to the known input. The maximum weighted log-likelihood estimate (MWLE) is a method to deal with covariate shifts that improves the maximum likelihood estimate (MLE) by introducing a weighted function of the covariate in the log-likelihood function [17]. The MWLE is obtained by maximizing the weighted log-likelihood function; however, different choices of the weight function affect the expected loss for moderate sample sizes by compromising the bias and variance of MWLE. Thus, this method is adequate when the PDF of observed samples do not correspond to the total population.

In binary classification problems, covariate shifts can cause bias in cross-validation, which can be solved through density ratio, or importance-weighted cross-validation methods [73]. In this case, bias is mitigated according to the ratio between classes. Otherwise, direct importance can be estimated by minimizing Kullback–Leibler divergence between the true test input density and its estimate [74].

C. Transductive and Semi-Supervised Learning

Transductive and semi-supervised learning (SSL) approaches use unlabeled data in combination with labeled datasets to improve classification performance. These methods are appropriate to deal with a variety of shifts, since they are capable of using test data to adapt the model. In this context, SSL methods can retrieve information that might have changed between the training and test sets. SSL approaches usually are inductive modeling strategies that incorporate both labeled and unlabeled data during training. In comparison, transductive models classify and perform regression of data without an induced general model by considering both training and test data to compute every output. Moreover, the test error in transduction is not affected by differences between training and test sets since the joint error in eq. 4 is particular to induced models. Specifically, prior probability shifts and class imbalance problems have promising transductive solutions. For instance, gene expression profile-based cancer classification, which is an imbalanced diagnostic problem, had encouraging performance with such an approach [69].

However, transductive approaches tend to be much slower than inductive methods, which might be prohibitive to several applications. To improve the speed of transductive approaches, some algorithms propose hybrid methodologies. For instance, TRANSE is an ensemble algorithm with transductive learning that solves the drift problem by assuming that test data is a sample of an unknown distribution [75]. Transductive approaches are also applied alongside other methodologies to adapt domains [76] or extract aspects of novel domains [77]. Approaches that incorporate support vector machines (SVM) with transductive and SSL strategies, such as the transductive support vector machine (TSVM), have been proposed [78]. For instance, a procedure for binary TSVM was used in remote-sensing classification with ill-posed data and small-sample sizes, which resulted in both high classification accuracy and good stability [79].

SDA is a semi-supervised discriminant analysis proposed for image classification based on linear discriminant analysis (LDA) [45]. The resulting SDA classifier maximizes inter-class and minimizes intra-class covariances, but for small training sample sizes, the covariance matrix of each class may not be accurately estimated. Thus, an SSL approach was proposed to include the knowledge from intrinsic geometric structures of unlabeled data. The COMPOSE framework, aimed at ILNS problems [80], uses both labeled and unlabeled data to assign new labels with an SSL approach. It uses a multi-modal density estimation to create an envelope around data. Then, it draws labeled samples from the most significant part of each class, which are used in streams of unlabeled data. An application of off-line shifts is voice recording, which can be modeled as a covariate shift. Importance-weighted kernel logistic regression (IWKLR) was combined in an SSL method with importance-weighted cross-validation (IWCV) to solve speech classification, achieving good results for text-independent scenarios [81].

A solution for EEG-based BCI identification was proposed with a method that redefines the knowledge base according to the distance between labeled and unlabeled data points [20], [21]. This system is updated according to transductive strategies whenever the covariate shift detector indicates a drift. Learning is based on probabilistic k-NN and defines a growing knowledge base that is used to adapt inductive classifiers.

D. Transfer Learning and Domain Adaptation

Dataset shift and transfer learning are related as they are both subject to model training with limited settings of data (the source domain), yet they are intended to perform predictions for related settings, or target domains [82]. Another possible approach is the pre-processing of data, which allows traditional

learning algorithms to track data shifts. FIELDS is a computational framework for extending an incomplete, labeled data stream. The framework transforms the original data stream with a few labeled data into a new stream that incorporates the concept drift [83]. Strictly, this is a transfer learning strategy since it learns and encodes invariances from a limited set of labeled data. FIELDS can be understood as data preprocessing, which can be integrated into different machine learning methodologies. The method, however, intrinsically introduces delay and can make learning and predictions slower.

Domain adaptation consists of adjusting classifiers created in an initial context (source domain) to a new one (target domain) without the need for data from the new context [84]–[86]. However, few domain adaptation strategies function without labeled data from the target domain [87]. In this context, feature augmentation methods [88], unsupervised feature transforms [89, 90], unsupervised domain adaptation [91], [92], or transductive predictive adaptation [76] methods can solve off-line problems. For instance, transductive approaches can be used to extract unlabeled aspects from novel domains in sentiment analysis [77].

More specifically, unsupervised domain adaptation (UDA) methods have a clear relation to off-line dataset shifts, as they attempt to recognize new patterns from the target domains even if only unlabeled samples are presented. F-HeUDA is a learning model that addresses heterogeneous domain adaptation (HeUDA) using fuzzy geometry and equivalence relations to transfer knowledge from big datasets to small ones [92]. However, the results shown are only marginally better than the benchmark. Due to the number of iterations required for the method to converge and the fuzzy geometry calculation, F-HeUDA is slower than traditional approaches.

When multiple datasets are used, bias of the training data can be minimized using domain adaptation strategies that discriminate data associated with individual datasets. Khosla et al. [93] proposed a discriminative framework aimed at multiple datasets for image classification. The method defines visual global weights that are common to all datasets and biased vector weights, which are associated with individual datasets. Visual global weights are retrieved by removing bias weights from each dataset, and the model is trained with unbiased datasets. In dataset shift problems, this method can be used to remove divergent bias in training and test sets.

Moreover, the performance of existing classifiers can improve with domain adaptation by including features from a new dataset. This was successfully done with a genetic programming-based algorithm. A classification problem of biological laboratory data for cancer diagnostics was solved with the incorporation of data from a different laboratory that adopted the same protocols [94].

E. Ensemble Learning

Model misspecification often occurs due to data having multimodal behavior throughout the input domain, leading to drifts when training and test sets are biased towards particular modes. Ensemble learning methods define multiple classifiers [95], [96], which allow different definitions of the model across the data domain. Therefore, ensemble methods are capable of learning drifting concepts with similar accuracy to base algorithms that learn each concept individually [97]. Moreover, higher diversity ensembles show lower test errors shortly after the drift. Afterward, however, diversity becomes less important and prevents fast recoveries in the long term [98], [99].

Ensemble SSL approaches can be used to solve non-stationary environments. The weights of an ensemble classifier can be updated when the unlabeled data is obtained from a distribution that is not already known [100]. Similarly, ensemble approaches can be integrated with transductive learning [75]. These strategies are appropriate for off-line learning because they use information from unlabeled data to adapt previously trained models. Alternatively, labeled data can be used to train a classifier while unlabeled data is used in clustering. In this context, an ensemble voting system uses both classifiers and clusters to make decisions [101].

A typical off-line dataset shift problem is training a model in simulated environments with mobile ad-hoc networks (MANETs) before the real system implementation. Training data is often unfeasible to obtain from a real system; hence, a model is trained with simulated data and later applied to the real system [3]. In this case, a weak detector of malicious nodes that incorporates behavioral features can be used to adjust the decision to local parameters.

VII. Discussions and Final Remarks

The literature shows that dataset shift comprises a broad range of problems that can be divided into overlapping categories depending on density distribution, transience, duration, and cause.

Machine learning methods tend to present straightforward solutions to specific problems. For example, an ensemble method can be used to solve concept shift problems in non-stationary environments [102]. Covariate shifts caused by source component shifts and non-stationary environments can be solved through adaptive transductive learning approaches [21]. Thus, characteristics of shifts and methodologies should be considered to solve dataset shifts, as solutions are particular to each problem. With that objective, the focus of this paper is on methods and definitions targeting off-line shifts, resulting in the following findings:

- ❏ Most methods consider scenarios where new labels will be available when the shift occurs; thus, strategies and frameworks capable of dealing with off-line shifts are limited.
- ❏ Active detection of shifts faces specific challenges, such as local shift detection, to increase detection sensibility. Detection methods for off-line shifts tend to depend on data density estimation or multiple hypothesis testing, which causes the methods to be slow.
- ❏ Transfer learning and domain adaptation have not been popular approaches for off-line dataset shifts in recent years, despite their close relation to domain shifts. Regardless, interest in unsupervised domain adaptation approaches has increased.

- ❏ Ensemble, transductive, and semi-supervised approaches, including unsupervised domain adaptation methods, are increasingly applied in off-line shift scenarios.

Finally, the following topics are indicative of the research on off-line dataset shifts:

- ❏ Adaptive, semi-supervised, and unsupervised domain adaptation approaches are promising as they are intrinsically able to solve off-line shifts.
- ❏ Off-line shifts, in general, have not been broadly addressed in the literature. In this sense, description and analysis of real-world problems are promising research topics. In particular, off-line shift detection methods are gaining attention.

Acknowledgment

The authors would like to thank the electrical engineering graduate program of the Federal University of Minas Gerais (UFMG). This work has been supported by the Brazilian agencies Coordination for the Improvement of Higher Education Personnel (CAPES) and National Council for Scientific and Technological Development (CNPq).

References

[1] J. a Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, Mar. 2014. doi: 10.1145/2523813.

[2] I. Žliobaitė, M. Pechenizkiy, and J. Gama, *An Overview of Concept Drift Applications*. NewYork: Springer-Verlag, 2016, pp. 91–114.

[3] B. Gao, T. Maekawa, D. Amagata, and T. Hara, "Environment-adaptive malicious node detection in MANETs with ensemble learning," in *Proc. IEEE 38th Int. Conf. Distributed Computing Systems (ICDCS)*, July 2018, pp. 556–566. doi: 10.1109/ICDCS.2018.00061.

[4] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comp. Int. Mag.*, vol. 10, no. 4, pp. 12–25, Nov. 2015. doi: 10.1109/MCI.2015.2471196.

[5] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4802–4821, Oct. 2018. doi: 10.1109/TNNLS.2017.2771290.

[6] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: An overview," *Progr. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, Apr. 2012. doi: 10.1007/s13748-011-0008-0.

[7] P. González, A. Castaño, N. V. Chawla, and J. J. D. Coz, "A review on quantification learning," *ACM Comput. Surv.*, vol. 50, no. 5, p. 74, Sept. 2017. doi: 10.1145/3117807.

[8] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019. doi: 10.1109/TKDE.2018.2876857

[9] J. J. G. J. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, Jan. 2012. doi: 10.1016/j.patcog.2011.06.019.

[10] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA: MIT Press, Feb. 2009.

[11] J. G. Moreno-Torres, J. A. Saez, and F. Herrera, "Study on the impact of partition-induced dataset shift on *k*-fold cross-validation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1304–1312, Feb. 2012. doi: 10.1109/TNNLS.2012.2199516.

[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, June 1995.

[13] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[14] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Advances Neural Information Processing Systems 4 (NIPS 1991)*, Dec. 1991, vol. 4, pp. 831–838.

[15] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, Jan. 1992. doi: 10.1162/neco.1992.4.1.1.

[16] M. Sugiyama and A. J. Storkey, "Mixture regression for covariate shift," in *Proc. Advances Neural Information Processing Systems 19 (NIPS)*, Dec. 2007, pp. 1337–1344.

[17] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Plan. Inf.*, vol. 90, no. 2, pp. 227–244, Oct. 2000. doi: 10.1016/S0378-3758(00)00115-4.

[18] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, Sept. 2009.

[19] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, "Application of covariate shift adaptation techniques in brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1318–1324, June 2010. doi: 10.1109/TBME.2009.2039997.

[20] H. Raza, G. Prasad, Y. Li, and H. Cecotti, "Covariate shift-adaptation using a transductive learning model for handling non-stationarity in EEG based brain-computer interfaces," in *Proc. IEEE Int. Conf. Bioinformatics Biomedicine*, Nov. 2014, pp. 230–236. doi: 10.1109/BIBM.2014.6999160.

[21] H. Raza, H. Cecotti, Y. Li, and G. Prasad, "Adaptive learning with covariate shift-detection for motor imagery-based brain-computer interface," *Soft Comput.*, vol. 20, no. 8, pp. 3085–3096, Aug. 2016. doi: 10.1007/s00500-015-1937-5.

[22] H. Raza, D. Rathee, S.-M. Zhou, H. Cecotti, and G. Prasad, "Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface," *Neurocomputing*, vol. 343, pp. 154–166, May 2019. doi: 10.1016/j.neucom.2018.04.087.

[23] F. Lotte et al., "A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update," *J. Neural Eng.*, vol. 15, no. 3, p. 031005, Apr. 2018. doi: 10.1088/1741-2552/aab2f2.

[24] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Muller, "Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 993–1002, June 2004. doi: 10.1109/TBME.2004.827088.

[25] S.-H. Park, D. Lee, and S.-G. Lee, "Filter bank regularized common spatial pattern ensemble for small sample motor imagery classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 2, pp. 498–505, Feb. 2018. doi: 10.1109/TNSRE.2017.2757519.

[26] R. Alaíz-Rodríguez, A. Guerrero-Curieses, and J. Cid-Sueiro, "Class and subclass probability re-estimation to adapt a classifier in the presence of concept drift," *Neurocomputing*, vol. 74, no. 16, pp. 2614–2623, Sept. 2011. doi: 10.1016/j.neucom.2011.03.019.

[27] V. Hofer and G. Krempl, "Drift mining in data: A framework for addressing drift in classification," *Comput. Statist. Data Anal.*, vol. 57, no. 1, pp. 377–391, Jan. 2013. doi: 10.1016/j.csda.2012.07.007.

[28] T. Ahamed, L. Tian, Y. Zhang, and K. Ting, "A review of remote sensing methods for biomass feedstock production," *Biomass Bioenergy*, vol. 35, no. 7, pp. 2455–2469, July 2011. doi: 10.1016/j.biombioe.2011.02.028.

[29] D. Tuia, E. Pasolli, and W. J. Emery, "Dataset shift adaptation with active queries," in *Proc. Joint Urban Remote Sensing Event*, Apr. 2011, pp. 121–124. doi: 10.1109/JURSE.2011.5764734.

[30] D. Tuia, E. Pasolli, and W. Emery, "Using active learning to adapt remote sensing image classifiers," *Remote Sens. Environ.*, vol. 115, no. 9, pp. 2232–2242, Sept. 2011. doi: 10.1016/j.rse.2011.04.022.

[31] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Can. J. Forest Res.*, vol. 45, no. 7, pp. 783–792, Mar. 2015. doi: 10.1139/cjfr-2014-0347.

[32] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens.*, vol. 4, no. 2, pp. 41–57, June 2016. doi: 10.1109/MGRS.2016.2548504.

[33] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2018, pp. 3339–3348. doi: 10.1109/CVPR.2018.00352.

[34] S. J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A case-based technique for tracking concept drift in spam filtering," *Knowl.-Based Syst.*, vol. 18, no. 4–5, pp. 187–195, Aug. 2005. doi: 10.1016/j.knosys.2004.10.002.

[35] G. Ditzler and R. Polikar, "An ensemble based incremental learning framework for concept drift and class imbalance," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, July 2010, pp. 1–8. doi: 10.1109/IJCNN.2010.5596764.

[36] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011. doi: 10.1109/TNN.2011.2160459.

[37] P. Zheng et al., "Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives," *Frontiers Mech. Eng.*, vol. 13, no. 2, pp. 137–150, June 2018. doi: 10.1007/s11465-018-0499-5.

[38] D. Pfeffermann, C. J. Skinner, D. J. Holmes, H. Goldstein, and J. Rasbash, "Weighting for unequal selection probabilities in multilevel models," *J. Roy. Statist. Soc.*, vol. 60, no. 1, pp. 23–40, Jan. 1998. doi: 10.1111/1467-9868.00106.

[39] G. Tripepi, K. J. Jager, F. W. Dekker, and C. Zoccali, "Selection bias and information bias in clinical research," *Nephron Clin. Pract.*, vol. 115, no. 2, pp. 94–99, June 2010. doi: 10.1159/000312871.

[40] A. Fernandez, S. Garcia, and F. Herrera, "Addressing the classification with imbalanced data: Open problems and new challenges on class distribution," in *Proc. 6th Int. Conf. Hybrid Artificial Intelligent Systems*, May 2011, vol. 6678, pp. 1–10. doi: 10.1007/978-3-642-21219-2_1.

[41] V. López, A. Fernández, and F. Herrera, "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," *Inf. Sci.*, vol. 257, pp. 1–13, Feb. 2014. doi: 10.1016/j.ins.2013.09.038.

[42] B. Turhan, "On the dataset shift problem in software engineering prediction models," *Empirical Softw. Eng.*, vol. 17, no. 1-2, pp. 62–74, Feb. 2012. doi: 10.1007/s10664-011-9182-8.

[43] D. Dua and C. Graff, "UCI machine learning repository," 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[44] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013. doi: 10.1016/j.ins.2013.07.007.

[45] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE 11th Int. Conf. Computer Vision*, Oct. 2007, pp. 1–7.

[46] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, A review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, Jan. 2012. doi: 10.3390/s120201211.

[47] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 964–994, July 2016. doi: 10.1007/s10618-015-0448-4.
[48] N. Ramanathan, R. Chellappa, and S. Biswas, "Computational methods for modeling facial aging: A survey," *J. Vis. Lang. Comput.*, vol. 20, no. 3, pp. 131–144, June 2009. doi: 10.1016/j.jvlc.2009.01.011.
[49] M. A. Taister, S. D. Holliday, and H. Borrmman, "Comments on facial aging in law enforcement investigation," *Forensic Sci. Commun.*, vol. 2, no. 2, Apr. 2000.
[50] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU School of Comput. Sci., Tech. Rep. CMU-CS-16-118, 2016.
[51] S. Berry, J. Levinsohn, and A. Pakes, "Differentiated products demand systems from a combination of micro and macro data: New car market," *J. Politic. Econ.*, vol. 112, no. 1, pp. 68–105, Feb. 2004. doi: 10.1086/379939.
[52] D. M. F. Palhares et al., "Normal limits of the electrocardiogram derived from a large database of Brazilian primary care patients," *BMC Card. Disord.*, vol. 17, no. 1, p. 152, June 2017.
[53] C. J. Tsai, C. I. Lee, and W. P. Yang, "Mining decision rules on data streams in the presence of concept drifts," *Expert Sys. Appl.*, vol. 36, no. 2, pp. 1164–1178, Mar. 2009. doi: 10.1016/j.eswa.2007.11.034.
[54] Z. Lipton, Y.-X. Wang, and A. Smola, "Detecting and correcting for label shift with black box predictors," in *Proc. 35th Int. Conf. Machine Learn.*, July 2018, vol. 80, pp. 3122–3130.
[55] D. R. de Lima Cabral and R. S. M. de Barros, "Concept drift detection based on fisher's exact test," *Inf. Sci.*, vol. 442–443, pp. 220–234, May 2018. doi: 10.1016/j.ins.2018.02.054.
[56] M. A. A. Abdualrhman and M. C. Padma, "Cd2a: Concept drift detection approach toward imbalanced data stream," in *Proc. Int. Conf. Emerging Research Electronics, Computer Science and Technology*, Aug. 2019, pp. 597–612. doi: 10.1007/978-981-13-5802-9_54.
[57] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," in *Proc. Advances Neural Information Processing Systems 32 (NIPS)*, Dec. 2019, pp. 1394–1406.
[58] A. Dries and U. Rckert, "Adaptive concept drift detection," *Statist. Anal. Data Mining*, vol. 2, no. 5–6, pp. 311–327, Nov. 2009. doi: 10.1002/sam.10054.
[59] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," in *Proc. 30th AAAI Conf. Artificial Intelligence.*, Feb. 2016, pp. 1652–1658.
[60] A. Haque, L. Khan, M. Baron, B. Thuraisingham, and C. Aggarwal, "Efficient handling of concept drift and concept evolution over stream data," in *Proc. IEEE 32nd Int. Conf. Data Eng*ineering, May 2016, pp. 481–492. doi: 10.1109/ICDE.2016.7498264.
[61] H. Raza, G. Prasad, and Y. Li, "Dataset shift detection in non-stationary environments using EWMA charts," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Oct. 2013, pp. 3151–3156. doi: 10.1109/SMC.2013.537.
[62] H. Raza, G. Prasad, and Y. Li, "EWMA model based shift-detection methods for detecting covariate shifts in non-stationary environments," *Pattern Recognit.*, vol. 48, no. 3, pp. 659–669, Mar. 2015. doi: 10.1016/j.patcog.2014.07.028.
[63] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, Nov. 2017. doi: 10.1016/j.neucom.2017.04.070.
[64] R. F. de Mello, Y. Vaz, C. H. Grossi, and A. Bifet, "On learning guarantees to unsupervised concept drift detection on data streams," *Expert Syst. Appl.*, vol. 117, pp. 90–102, Mar. 2019. doi: 10.1016/j.eswa.2018.08.054.
[65] O. Mahdi, E. Pardede, N. Ali, and J. Cao, "Fast reaction to sudden concept drift in the absence of class labels," *Appl. Sci.*, vol. 10, no. 2, p. 606, Jan. 2020. doi: 10.3390/app10020606.
[66] Q.-H. Duong, H. Ramampiaro, K. Nrvg, P. Fournier-Viger, and T.-L. Dam, "High utility drift detection in quantitative data streams," *Knowl.-Based Syst.*, vol. 157, pp. 34–51, Oct. 2018. doi: 10.1016/j.knosys.2018.05.014.
[67] A. Liu, Y. Song, G. Zhang, and J. Lu, "Regional concept drift detection and density synchronized drift adaptation," in *Proc. 26 Int. Joint Conf. Artificial Intelligence.*, Aug. 2017, pp. 2280–2286.
[68] A. Liu, J. Lu, F. Liu, and G. Zhang, "Accumulating regional density dissimilarity for concept drift detection in data streams," *Pattern Recognit.*, vol. 76, pp. 256–272, Apr. 2018. doi: 10.1016/j.patcog.2017.11.009.
[69] D. Li, L. Wang, J. Wang, Z. Xue, and S. T. C. Wong, "Transductive local fisher discriminant analysis for gene expression profile-based cancer classification," in *Proc. IEEE EMBS Int. Conf. Biomed. Health Inf. (BHI)*, Feb. 2017, pp. 49–52. doi: 10.1109/BHI.2017.7897202.
[70] N. E. Huang et al., "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, Mar. 1998. doi: 10.1098/rspa.1998.0193.
[71] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," in *Proc. 17th Int. Conf. Machine Learning*, June 2000, pp. 487–494.
[72] B. J. Oommen and L. Rueda, "Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments," *Pattern Recognit.*, vol. 39, no. 3, pp. 328–341, Mar. 2006. doi: 10.1016/j.patcog.2005.09.007.
[73] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, Dec. 2007.
[74] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. Advances Neural Information Processing Systems 20 (NIPS)*, Dec. 2008, pp. 1433–1440.
[75] G. Ditzler, G. Rosen, and R. Polikar, "Transductive learning algorithms for nonstationary environments," in *Proc. 2012 Int. Joint Conf. Neural Net*works, July 2012, pp. 10–15.
[76] S. Clinchant, B. Chidlovskii, and G. Csurka, "Transductive adaptation of black box predictions," in *Proc. 54th Annu. Meeting Association Computational Ling*uistics, Aug. 2016, vol. 2, pp. 326–331.
[77] R. M. Marcacini, R. G. Rossi, I. P. Matsuno, and S. O. Rezende, "Cross-domain aspect extraction for sentiment analysis: A transductive learning approach," *Decis. Support Syst.*, vol. 114, pp. 70–80, Oct. 2018. doi: 10.1016/j.dss.2018.08.009.
[78] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Advances Neural Information Processing Systems 11 (NIPS)*, Dec. 1998, pp. 368–374.
[79] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive svm for semisupervised classification of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3372, Nov. 2006. doi: 10.1109/TGRS.2006.877950.
[80] K. B. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 12–26, Jan. 2014. doi: 10.1109/TNNLS.2013.2277712.
[81] M. Yamada, M. Sugiyama, and T. Matsui, "Semi-supervised speaker identification under covariate shift," *Signal Process.*, vol. 90, no. 8, pp. 2353–2361, Aug. 2010. doi: 10.1016/j.sigpro.2009.06.001.
[82] S. S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. doi: 10.1109/TKDE.2009.191.
[83] D. H. Widyantoro and J. Yen, "Relevant data expansion for learning concept drift from sparsely labeled data," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 401–412, Mar. 2005. doi: 10.1109/TKDE.2005.48.
[84] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2011, pp. 1785–1792.
[85] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia, "Semisupervised transfer component analysis for domain adaptation in remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3550–3564, July 2015. doi: 10.1109/TGRS.2014.2377785.
[86] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Proc. 30th Int. Conf. Machine Learning*, June 2013.
[87] G. Csurka, Domain adaptation for visual applications: A comprehensive survey. 2017. [Online]. Available: arXiv:1702.05374
[88] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2288–2302, Nov. 2014. doi: 10.1109/TPAMI.2013.249.
[89] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp. 1410–1417. doi: 10.1109/CVPR.2014.183.
[90] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, June 2014, pp. 2481–2488. doi: 10.1109/CVPR.2014.318.
[91] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, July 2017.
[92] F. Liu, J. Lu, and G. Zhang, "Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations," *IEEE Trans Fuzzy Syst.*, vol. 26, no. 6, pp. 3555–3568, Dec. 2018. doi: 10.1109/TFUZZ.2018.2836364.
[93] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *Proc. 12th European Conf. Computer Vision*, Oct. 2012, vol. 7572, pp. 158–171.
[94] J. G. Moreno-Torres, X. Llorà, D. E. Goldberg, and R. Bhargava, "Repairing fractures between data using genetic programming-based feature extraction: A case study in cancer diagnosis," *Inf. Sci.*, vol. 222, pp. 805–823, 2013. doi: 10.1016/j.ins.2010.09.018.
[95] L. I. Kuncheva, "Classifier ensembles for changing environment," in *Proc. Int. Workshop Multiple Classifier Systems*, June 2004, pp. 1–15. doi: 10.1007/978-3-540-25966-4_1.
[96] R. Polikar, *Ensemble Learning.* Boston, MA: Springer-Verlag, 2012, pp. 1–34.
[97] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: A new ensemble method for tracking concept drift," in *Proc. 3rd IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 123–130. doi: 10.1109/ICDM.2003.1250911.
[98] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010. doi: 10.1109/TKDE.2009.156.
[99] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 619–633, Apr. 2012. doi: 10.1109/TKDE.2011.58.
[100] G. Ditzler and R. Polikar, "Semi-supervised learning in nonstationary environments," in *Proc. Int. Joint Conf. Neural Networks*, July 2011, pp. 2741–2748.
[101] P. Zhang, X. Zhu, J. Tan, and L. Guo, "Classifier and cluster ensembles for mining concept drifting data streams," in *Proc. IEEE 10th Int. Conf. Data Mining*, Dec. 2010, pp. 1175–1180. doi: 10.1109/ICDM.2010.125.
[102] S. K. Siahroudi, P. Z. Moodi, and H. Beigy, "Detection of evolving concepts in non-stationary data streams: A multiple kernel learning approach," *Expert Sys. Appl.*, vol. 91, pp. 187–197, Jan. 2018. doi: 10.1016/j.eswa.2017.08.033.

Automatic Tuning of Rule-Based Evolutionary Machine Learning via Problem Structure Identification

Maria A. Franco, Natalio Krasnogor and Jaume Bacardit
The Interdisciplinary Computing and Complex BioSystems (ICOS) research group, School of Computing, Newcastle University, UK

Abstract—The success of any machine learning technique depends on the correct setting of its parameters and, when it comes to large-scale datasets, hand-tuning these parameters becomes impractical. However, very large-datasets can be pre-processed in order to distil information that could help in appropriately setting various systems parameters. In turn, this makes sophisticated machine learning methods easier to use to end-users. Thus, by modelling the performance of machine learning algorithms as a function of the structure inherent in very large datasets one could, in principle, detect "hotspots" in the parameters' space and thus, auto-tune machine learning algorithms for better dataset-specific performance. In this work we present a parameter setting mechanism for a rule-based evolutionary machine learning system that is capable of finding the adequate parameter value for a wide variety of synthetic classification problems with binary attributes and with/without added noise. Moreover, in the final validation stage our automated mechanism is able to reduce the computational time of preliminary experiments up to 71% for a challenging real-world bioinformatics dataset.

Digital Object Identifier 10.1109/MCI.2020.2998232
Date of current version: 15 July 2020

Corresponding Author: Jaume Bacardit (jaume.bacardit@newcastle.ac.uk)

1556-603X/20©2020IEEE

1. Introduction

Many machine learning techniques are tied to a series of hyper-parameters and/or selection of sub-components that need to be tuned. This challenge can broadly be defined as the *algorithm configuration problem*. When handling large-scale datasets, finding the adequate set of hyper-parameter values becomes a very expensive experimental process. Therefore, automatic hyper-parameter setting approaches are necessary in order to avoid a time-consuming preliminary experimentation stage, reduce the number of hyper-parameters that need to be set, and make these techniques more accessible to end-users. The creation of automated methods for the hyper-parameter tuning of machine learning algorithms but also of metaheuristic optimization algorithms has been a very active area of research in recent years [1]–[4].

This paper focuses on the specific context of evolutionary machine learning (EML). EML algorithms have shown to be very competitive methods for machine learning [5], [6] and have been applied to a very broad variety of real-world problems [7]–[13]. Hyper-parameter setting is a widespread problem in this field, since these systems use a genetic algorithm (GA) and fitness functions that often involve many hyper-parameters. Besides the generic approaches for hyper-parameter tuning mentioned above, there is a variety of methods that are specific to evolutionary computation [14]. Several techniques have been used such as reinforcement learning [15], or self-adaptive approaches [16]–[21], among others. Moreover, even though theory exists about EML systems [22]–[24] that explains how these systems should be parameterized, only recently these theoretical works have started to be applied to fully tune these algorithms using a few synthetic problems as test scenarios [25], [26].

This paper proposes a heuristic procedure for the application of theoretical models of EML to automatically estimate the structure of classification problems with binary attributes and, from such estimates, set hyper-parameters accordingly. This work focuses on a specific EML algorithm called BioHEL [27]. This system has shown competent performance in very complex real-world domains [7]–[10], [28], [29]. One of the main characteristics of this system is its fitness function, which tries to balance the accuracy, complexity and coverage of the rules in the solution. The key element of this fitness function is the *coverage breakpoint* hyper-parameter, which determines how many instances in the dataset a rule should cover to be considered good enough. Previous studies have shown that learning can be facilitated when this hyper-parameter is set correctly, in contrast to an incorrect setting which can push the system towards overgeneralization [30]. Moreover, this hyper-parameter is highly problem dependant and finding its adequate value requires an extensive preliminary experimentation.

However, often the data holds the key of how to parameterize the system correctly. For example, as it was shown by [30] it is possible to set the adequate coverage breakpoint for classification problems with binary attributes (for the sake of compactness we will refer to these as *binary problems* in the rest of the paper) in BioHEL if the structure of the problem is known. But is it possible to define a generalization about the structure of binary problems? A useful framework to model binary problems is the use of k-Disjunctive Normal Form (k-DNF) formulas, which are binary formulas that have r disjunctive terms and each one of these terms has k relevant variables of attributes expressed from the d possible ones. Many boolean benchmarks can be expressed as k-DNF formulas, and even complex real-world problems. Moreover, based on this structure (k and r) it is possible to create models that explain the behavior of the BioHEL system [31]. Nevertheless, determining these values (k and r) is not straightforward and calculating their exact value would involve applying data mining over the problem which is an extra computational cost that we wish to avoid.

In this work we introduce an automated heuristic approach to determine the structure of an unknown binary problem (k and r) at runtime. This heuristic combines observations from the data and from a sample of randomly initialized rules evaluated against this data, to retro-feed theoretical models of the behavior of the system [30], [31] and classify the problems into groups called *kr-groups* with a particular k and r associated.

In this particular work we show how this methodology can help adapt the coverage breakpoint hyper-parameter of BioHEL. Our experiments show how this mechanism can characterize challenging binary problems with and without noise. Moreover, we show how this mechanism can help adapt hyper-parameters of the system to solve a real-world protein structure prediction problem and reduce the total experimental time up to 71%, hence saving a lot of computational time and human effort.

2. Related Work

The automatic configuration and tuning is a very challenging process in evolutionary algorithms (EAs), where it has been studied in great length. In this section we first describe, from a general EA perspective, the different approaches to hyper-parameter control (and some examples of each). Afterwards, we focus on specific examples of hyper-parameter control in EML systems, which is the particular context of BioHEL. In terms of nomenclature we use a variety of terms (tuning, adjustment, control, configuration) because any one of these individual terms does not capture all the relevant literature in this area.

There are different types of hyper-parameter control algorithms. [14] presented a classification for hyper-parameter control where the different techniques can be classified depending on what is changing (representation, mutation or crossover rates, selection mechanisms, etc.), or depending on how the change is made (deterministic, adaptive or self-adaptive). The *deterministic* techniques are the ones that adjust hyper-parameter without any feedback from the algorithm being controlled. The *adaptive* techniques are the ones that use some sort of feedback from the search process (i.e. during the optimization/learning process). The *self-adaptive* techniques are the ones which evolve the hyper-parameters along with the rest of features of the problem. Using this classification our approach can

be classified as a deterministic one, the methods described below applying the 1/5 rule for mutation rate adjustment [32], [33] would be adaptive, and [34] would be an example of the self-adaptive approach.

In the area of EAs, the earliest automatic hyper-parameter setting approaches were presented by [32] and [33], in which the mutation rate was adapted according to the *1/5th rule*. If the rate of successful mutation was over 1/5 the mutation rate was increased and if it was below this value it was decreased. Other early approaches involve the adaptation of crossover rate depending on how good were the resulting offsprings [35]. Regarding operator selection within evolutionary algorithms, [34] presented a very simple self-adaptive crossover-selection method. One extra bit in the classifier encoding represented the crossover that should be applied. Other approaches [36], [37] use rules to modify the hyper-parameters of local search operators (crossover and mutation).

Self-adaptive approaches have also been used in memetic algorithms to adapt the local search operators depending on the stage on which the search process is [18]–[21].

Reinforcement learning (RL) has also been used to adapt hyper-parameters in EAs to identify the appropriate step size for the *1/5th rule* when adapting the mutation rate [38]. Furthermore, other more complex approaches [39] use RL to adapt the hyper-parameters of the GA considering not only the quality of the solutions, but also the cost incurred by the selected search operators.

There are several examples of self-adaptive mechanisms in the EML context. The Zeroth Level Classifier System (ZCS) [40] was extended with self-adaptive mutation in [41], to give an independent mutation rate to each classifier. Afterwards, this work was extended by self-adapting all the hyper-parameters in ZCS (mutation, learning rate, tax rate and discount factor) at the same time [42]. While in stationary environments the results are as good as the ones obtained with fixed hyper-parameters, in the more dynamical ones the self-adaptation improves the performance of the system. Self-adaptive mechanisms for both the mutation and the learning rate in XCS [43] were investigated in [44]. The self-adaptive mutation rate solved some generalization problems when XCS was applied to long rule chain environments. Nevertheless, the performance was still sub-optimal in this case. Also, the system showed worse performance when trying to adapt the learning rate. Finally, Self-adaptive mutation has also been applied to the XCSF [45] using hyper-ellipsoidal condition structures [46], in which each part of the knowledge representation had its own self-adapted mutation rate.

More generally, the automated tuning of machine learning algorithms and pipelines is a very active field of research nowadays, broadly called *AutoML*. In broad terms we can consider that these methods apply a *wrapper approach*. They run the underlying algorithm/analysis pipeline on samples of the data (e.g. using cross-validation) to estimate the predictive performance of a given configuration, which is very different from our approach, in which we never run the full BioHEL algorithm during the hyper-parameter tuning process. A variety of strategies exist to search for the optimal configurations. For instance, TPOT [1] uses genetic programming to evolve trees that represent complete machine learning pipelines, including data cleaning, feature selection/construction and the selection and tuning of machine learning algorithms. ML-Plan [2] defines the algorithm tuning process as a planning task, and uses hierarchical planning networks [47] to identify the optimal plan, i.e. algorithm selection and tuning. A different search strategy is employed by Auto-sklearn [4], which uses Bayesian optimization for the *AutoML* task. As a final example we would like to mention *irace* [3] which, while intended to tune optimization algorithms, uses an equivalent wrapper principle to the *AutoML* approaches. It uses an extension of the racing algorithm [48] to perform the tuning process. In the classic racing algorithm for machine learning model selection, a set of candidate models is evaluated sample by sample. In each step the methods that perform significantly worse are discarded. The process continues until a certain evaluation budget is reached or the set of remaining models is small enough.

Finally, another existing approach for algorithm selection (rather than tuning) is based on complexity measures defined to capture dataset difficulty in a supervised machine learning context [49]. These measures have been used to study the domains of competence of the XCS algorithm [50], as well as to recommend machine learning algorithms based on a meta-learning approach [51].

3. The BioHEL System

BioHEL [27] is an EML algorithm designed to handle large-scale datasets [52]–[54], [28], [29], [7], [8], [10]. BioHEL learns a set of rules following the Iterative Rule Learning paradigm first used in EML in the SIA system [55]. This learning paradigm generates ordered rule sets in which the rules in the solution are learnt sequentially, using a generational GA to learn individual rule. Hence, each individual of the GA population is a rule. Once a rule (the best individual in the final GA population) is learnt, it is added to the rule set and the training set is filtered by removing all the examples covered by this rule. The iterative process generally stops when the whole training set is covered. However, the stopping criteria changes when using a default rule as detailed in Section 3.1. In the rest of the section we only describe the aspects about BioHEL that are relevant to this work. For a full description of the algorithm, please see [27].

3.1. Representation

BioHEL uses the Attribute List Knowledge Representation (ALKR) [56], a sparse representation designed to handle high-dimensional problems. In this encoding, each rule represents only its relevant attributes, reducing considerably the cost of the match operations (as all the irrelevant attributes for that rule are not present). The relevant attributes can vary across rules, and are discovered during the learning process. ALKR uses hyper-rectangles [57] to represent continuous attributes and the GABIL representation [58] for nominal attributes.

Since in this paper we focus on problems with discrete (binary) attributes, it is necessary to explain the GABIL representation in greater detail. In this representation the attributes are expressed by binary strings of fixed length. The length corresponds to the number of possible values the attribute can have. For example, in a problem with three attributes (F_1, F_2 and F_3) if the attribute F_1 may take the values (A, B, C), F_2 the values (O, P), and F_3 the values (W, Z, X, Y) a possible condition string for each one of the attributes would look like:

F_1	F_2	F_3
100	01	1101

Each attribute is read as a disjunctive clause between all the values that have their bit on. For example, this condition can be interpreted as F_1 *is A and* F_2 *is P and* F_3 *is* {*W or Z or Y*}.

To initialize a rule in ALKR first a subset of the problem's attributes is randomly chosen to be included in the rule. The size of the subset is controlled by the *ExpAtts* hyper-parameter. Afterwards, an instance from the training set is sampled, and the bits corresponding to the instance's values in the GABIL predicate are set to 1. The rest of bits in the predicate are also set to 1 with probability p.

This representation uses an explicit default rule mechanism [59], which consists of a rule that covers all the examples left in the training set and assigns them to a user predefined class. Since the default class is not used in the evolved rules, this mechanism generates more compact rule sets. As a result of using a default rule, the stopping criteria of the iterative rule learning process is changed. In BioHEL, the rule learning process stops whenever it is not possible to generate a rule that has accuracy higher than the default rule.

3.2. Fitness Function

The BioHEL's fitness function is based on the Minimum Description Length principle [60]. This fitness function is designed to promote accurate, general and compact rules by integrating three metrics into the fitness formula: accuracy, coverage and complexity. This fitness function has two terms as shown in Equation (1).

$$F(a) = TL(a) \cdot W + EL(a). \tag{1}$$

$TL(a)$ (theory length) corresponds to the complexity of rule a, $EL(a)$ (exceptions length) corresponds to the accuracy and coverage of rule a and W is a weight that adjusts the relation between the previous terms. This hyper-parameter W is adjusted automatically using a heuristic defined by [59].

The definition of $TL(a)$ depends on the employed knowledge representation. For the GABIL representation it is defined as follows:

$$TL(a) = \frac{\sum_{i=1}^{NA} n_i / v_i}{NA_a}$$

where NA is the number of attributes in the problem, NA_a is the number of attributes explicitly represented by rule a, n_i is the number of values set to 0 and v_i is the number of possible values in the GABIL string for the i-th attribute represented in rule a. A simple interpretation of $TL(a)$ is that is computing the percentage of bits of the GABIL's encoding of a rule set to 0, with the added caveat that we use the ALKR sparse rule encoding in which only a fraction of the attributes is represented, and only the represented attributes contribute to the formula. In this formula lower is better, so any non-represented attribute contributes 0 to the formula. Hence $TL(a)$ promotes rules in which (a) few attributes are represented and (b) few values within each attribute are set to 0.

Furthermore, $EL(a)$ is defined as:

$$\begin{aligned}
EL(a) &= 2 - ACC(a) - COV(a) \\
ACC(a) &= \frac{correctlyClassified(a)}{matched(a)} \\
COV(a) &= \begin{cases} 0 & \text{if } RC(a) < CB(c(a))/3 \\ MidCov(a) & \text{if } RC(a) < CB(c(a)) \\ HighCov(a) & \text{if } RC(a) \geq CB(c(a)) \end{cases} \\
MidCov(a) &= CR \cdot \frac{RC(a)}{CB(c(a))} \\
HighCov(a) &= CR + \frac{(1 - CR) \cdot (RC(a) - CB(c(a)))}{1 - RC(a)} \\
RC(a) &= \frac{correctlyClassified(a)}{|\, T_{c(a)} \,|} \\
CB(c(a)) &= CB \cdot \frac{|\, T \,|}{|\, T_c \,|}.
\end{aligned} \tag{2}$$

In these formulas, $ACC(a)$ corresponds to the accuracy of the rule and $COV(a)$ is the term that needs to promote general rules which is the key of BioHEL's fitness function. As shown in Equation (2), the value of $COV(a)$ depends on $RC(a)$ (recall; the ratio between the number of examples correctly classified by the rule over the total number of examples in the training set belonging to the same class as a) and $CB(c(a))$ (the percentage of examples of its class that any rule should cover to be considered a "good rule"). CB corresponds to the hyper-parameter known as *coverage breakpoint*. This hyper-parameter is first set globally and afterwards it is adjusted for every class in the problem based on the class distribution. This is a very problem dependent hyper-parameter which affects the performance of the system, as it was shown by [30] and the main target of the automated hyper-parameter setting of this paper.

Moreover, CR (coverage ratio) corresponds to the percentage of "reward" awarded to a rule with a higher coverage than the coverage breakpoint. Figure 1 shows the value of the coverage term $COV(a)$, depending on the coverage of the rule.

4. *k*-DNF Functions

k-Disjunctive Normal Form (*k*-DNF) functions [61] are a broad family of boolean functions which can be a useful tool to characterize the structure of binary problems. These functions

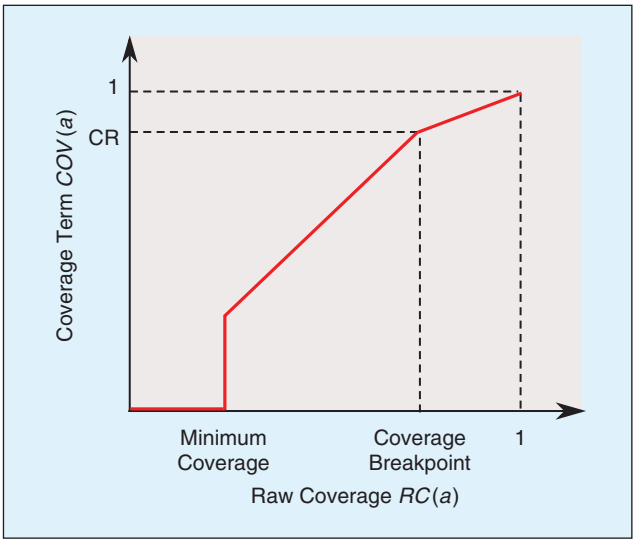


FIGURE 1 Coverage term *COV*(*a*) according to rule coverage. The minimum coverage corresponds to one third of the coverage breakpoint.

have been shown to be very useful to benchmark machine learning algorithms [62], [30], [63].

Given a space of d attributes or variables a k-DNF function is a boolean formula that presents the following form:

$$T_1 \vee T_2 \vee \cdots \vee T_r$$

where r is the number of disjunctive terms and each term T_x represents the conjunction of k boolean variables out of the d possible options $(x_1, x_2, \ldots, x_d)$, where some of the variables might have the *not* function ($\neg$) applied to them. Equation (3) represents an example of a k-DNF function for a space of 10 representable attributes (d), 2 terms (r) and 4 represented attributes (k).

$$(x_1 \wedge x_5 \wedge \neg x_7 \wedge x_{10}) \vee (x_1 \wedge \neg x_3 \wedge \neg x_6 \wedge x_7). \quad (3)$$

To construct a machine learning problem from a k-DNF boolean formula we generate all the possible 2^d instances (binary strings of size d). Afterwards, if the formula holds for a particular instance (one or more terms are true), class 1 (i.e. the positive class) is assigned to it. Otherwise class 0 (the negative class, to be covered by the default rule) is assigned. Considering this, the optimal rules that correspond to the solution of the previous problem in Equation (3)[1] would be:

```
1###1#0##1: 1
1#0##01###: 1
Default Rule: 0
```

Since most EML systems learn a set of rules as the solution of a problem, k-DNF formulas are really useful to evaluate them, as the systems are expected to learn one rule for each of the r terms in the problem and correct rules should represent at least the k relevant attributes in these terms. From this point onwards, every time we talk about "rules" we are referring to the solution of the problem and when we talk about "terms" we refer to the problem itself.

Many well-known boolean benchmarks can be represented by a function in k-DNF [30]. For instance, the rules in the 6-bit multiplexer have a k of 3 (two address bits and one data bit), the rules in the 20-bit multiplexer have a k of 5, a k of 6 is present in the 37-bit multiplexer and so on. The 18-bit hybrid Parity-Multiplexer problem [22] has a k of 9, as this problem is composed by a 6-bit multiplexer where each of these "bits" is the result of a 3-bit parity problem.

The difficulty of the k-DNF problems is closely related to the class imbalance. The class imbalance of a k-DNF problem can be estimated by calculating the probability of finding a negative example in the dataset given specific values for k and r.

$$P(neg) = (1 - 2^{-k})^r. \quad (4)$$

Considering each term covers a percentage of 2^{-k} of the training set, this formula states that the probability of having a negative example is equal to the probability of the example not being covered by any of the r terms in the problem. This formula holds under the assumption that the k attributes of the r terms are randomly picked and hence there is no large amount of overlap between rules.

Figure 2 shows the corresponding probability distribution for $P(neg)$. Here we can see that, depending on k and r, scenarios with very high class imbalance can be possible. When k is low each term covers a large proportion of the training set, and hence, just with a few terms (r), most of the examples will be positive. In these situations we encounter a known source of difficulty for EML systems: term overlap [31], [64]. On the other hand, a high k and low r create problems with very few positive examples, as each term is very specific and overlap is unlikely. These cases, essentially become scenarios of trying to find the needle in the haystack. Since the class imbalance makes the problem more difficult, the red area represents the problems that are easier to solve. For more information about the generation of artificial k-DNF problems please refer to [30].

5. Automatic Hyper-Parameter Setting

Considering the importance of the coverage breakpoint hyper-parameter in the performance of the BioHEL system, it seems necessary to adjust this hyper-parameter automatically for several reasons:

- **Runtime.** Since BioHEL is a system mainly oriented to solve large scale datasets, finding the correct setting for problem dependant hyper-parameters such as the coverage breakpoint involves a time-consuming preliminary experimentation stage. The automatic setup of this hyper-parameter can avoid preliminary experimentation and reduce the total experimental time.

[1] For simplicity we present the rules in ternary representation, where # means the attribute is irrelevant, and 0 or 1 represent the value the attribute should take to make the predicate hold. However, BioHEL uses GABIL to represent binary and discrete attributes as shown in Section 3.1.

❑ **Usability.** In many cases, end-users avoid exhaustive experimentation and settle for naive configurations that do not produce the best results. This improvement could make the system easier to use to an end-user and could also find better solutions for problems where the adequate coverage breakpoint has not been determined yet.

According to [30] it is possible to determine the coverage breakpoint if the characteristics of the problem (k-DNF formula) are known. These characteristics are the number of attributes expressed in the terms k and the total number of terms in the formula r. According to this work if the number of attributes in the terms is k the adequate coverage breakpoint to solve the problem is 2^{-k}. To ensure learning, the coverage breakpoint should be equal or smaller than this value. This translates the problem of finding the adequate coverage breakpoint to finding k.

But how it is possible to identify the structure of the problem by observing the data? To do this it is necessary to develop models based on k and r that explain characteristics of the data and behaviors of the system when working this data. For example, the model for the number of negative examples in Equation (4) finds a correlation between k and r and the proportion of negative examples in the problem. Moreover, the probability of obtaining good individuals, modelled for BioHEL using the ALKR representation and the GABIL encoding [31], also provides a relationship between a characteristic of the initial population and the variables k and r. This means that by observing these characteristics we can have estimates of the k and r of the problem.

However, these models only indicate a relationship between k and r. Many different values for k and r can satisfy the equation and, independently, each model does not provide information useful enough to identify these problem characteristics. However, by combining the results of different models together it is possible to determine the values for k and r that are more likely to match the characteristics of the problem.

In this paper we present a heuristic approach to determine the k and r of a given unknown boolean problem at runtime. This approach works by classifying the problems based on observations made over the data and randomly sampled individuals. Using this information the problems are classified into groups with a particular k and r associated, which we will call *kr-groups*.

The classification is done using a voting system. The space of k and r is divided uniformly in *kr-groups* which have an associated expected value and standard deviation boundaries for each one of the characteristics we measure. When a problem presents a characteristic that falls into the standard deviation boundaries of a particular *kr-group* the group gets awarded points. At the end, the *kr-group* that obtained more votes is considered the winner. Figure 3 illustrates how the heuristic works.

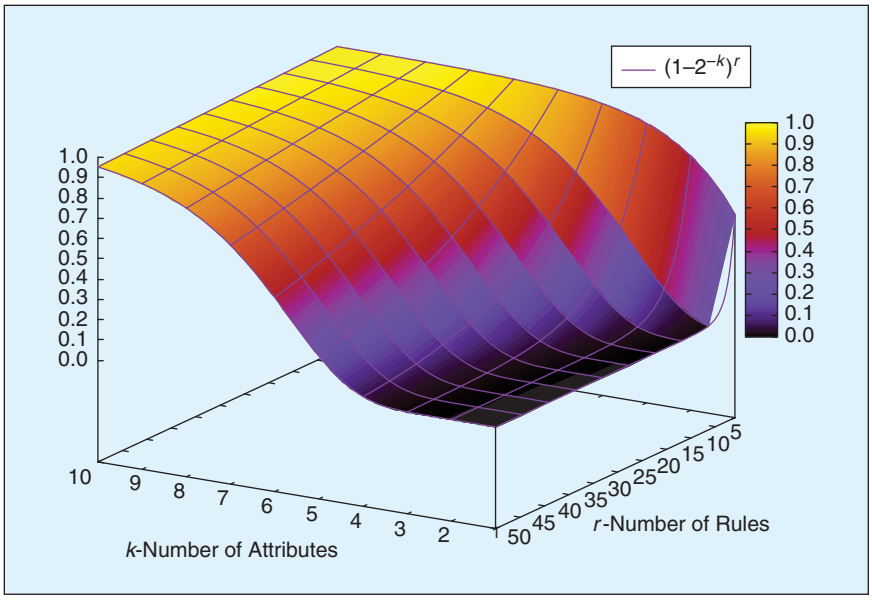


FIGURE 2 Probability of having a negative example in a k-DNF function according to the number of attributes k and the number of terms r.

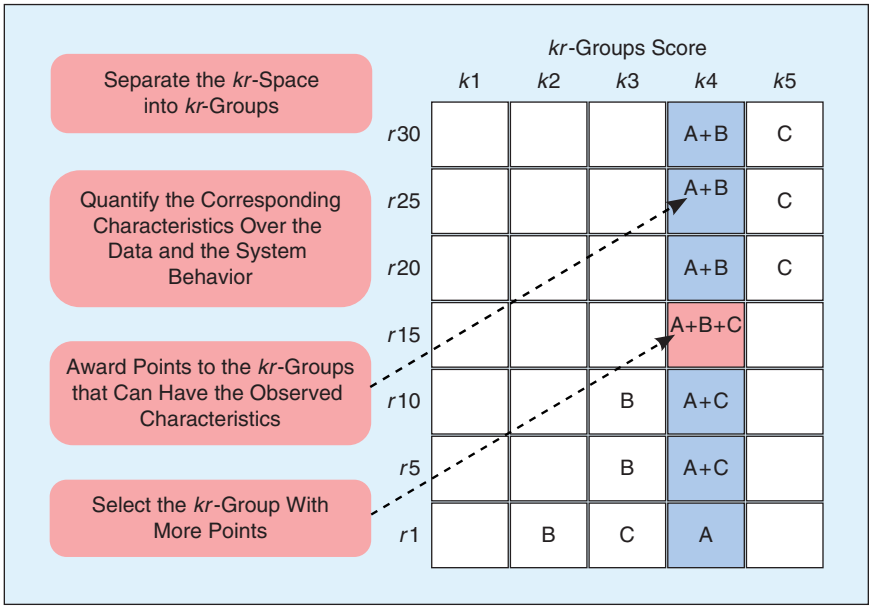


FIGURE 3 High-level representation of the problem structure identification heuristic. A, B and C refer to the criteria that will be used to assign points to the *kr–groups.*

Particularly for BioHEL, three characteristics were considered to classify the problems:

- The number of negative examples in the problem (defined in section 5.1.1).
- The number of *good individuals* in a random sample after evaluating them against the given problem. These are individuals that do not make classification mistakes or that have an accuracy higher than a certain threshold (defined in section 5.1.2).
- The number of attributes expressed in the good individuals (defined in section 5.1.3).

The following sections will explain in greater detail each one of the criteria used to classify the problems (section 5.1). Afterwards, we will show in more detail how the *kr-space* is partitioned in *kr-groups* and what makes a problem belong to a specific group (section 5.2). Finally, we explain the algorithm step-by-step (section 5.3).

5.1. Classification Criteria

This section explains each one of the criteria used within BioHEL to classify the problems: a) the number of negative examples in the problem, b) the number of *good individuals* in a random sample after evaluating them against the given problem and c) the number of attributes expressed in the good individuals. The first two characteristics used are completely theory-driven, which means they use theoretical models to determine the kind of problem we are handling. The last characteristic, even though it does not come from a model, reinforces the two previous criteria in finding the correct *kr-group*. Since the good individuals are already calculated for the second criterion, using the number of attributes expressed in these individuals does not involve an extra computational cost.

5.1.1. Number of Negative Examples in the Problem

Depending on the number of terms r and number of attributes expressed in each term k, the k-DNF problem will present a different percentage of negative examples.

For a randomly generated binary problem defined as the disjunction of r terms, where each term is the conjunction of k randomly picked attributes, the probability of having a negative example in the training set $P(neg)_r^k$ is equal to Equation (4) shown in Section 4.

By counting the number of negative examples in the training set, it is possible to use this formula inversely to determine possible combinations of k and r that are feasible for the given problem. For example a problem with $k = 2$ and $r = 1$ has 75% of negative examples. But also a problem with $k = 6$ and $r = 18$ has on average the same percentage of negative examples. If we observe a particular problem with 75% of negative examples both of these *kr-groups* would receive scores according to this criterion.

5.1.2. Number of Good Individuals in a Randomly Initialized Sample

A good individual or a *representative*, as it was defined by [22], is a rule that specifies (has represented) correctly at least all the attributes in one of the terms of the optimal solution to the problem. For example, if one of the terms of a problem with $d = 5$ and $k = 2$ is $x_1 = 0 \wedge x_4 = 1$ (0**1*) possible representatives would be 0##11, 0##1# and 01110, where # means that the attribute can take any value. Therefore, this rule does not make mistakes, but it can be more specific than the optimal rule where only k attributes are specified. The probabilities of finding a representative were first proposed by [22] for the ternary representation $\{0, 1, \#\}$. However, these models were not entirely suitable for BioHEL, as this system uses a different encoding. Later on, suitable models for the binary domain using the ALKR+GABIL representation were proposed by [31].

Assuming the usage of the default rule and covering mechanisms, the probability of finding a representative for a binary problem depends on k and r, as shown in Equation (5). This function states that the probability of having a good classifier $P(rep)$ is equal to the probability of having at least one of the terms in the k-DNF problem represented, and to have a term represented the rule should express the k relevant attributes. These models are able to hold with a certain amount of rule overlap if the rule coverage is uniform. For more details about this model please see full description presented by [31].

$$P(rep) = 1 - \left(1 - \left(\frac{2^{-k}(l_d(1-p))^k}{1-(1-2^{-k})^r}\right)\right)^r. \tag{5}$$

In this formula p corresponds to the probability of setting to 1 the values in a GABIL attribute (see Section 3), and l_d is the probability that an attribute appears in the ALKR attribute list. This value at the same time depends on the user-defined hyper-parameter *ExpAtts* (expected number of attributes) as follows:

$$l_d = \begin{cases} 1 & d <= ExpAtts \\ \dfrac{ExpAtts}{d} & d > ExpAtts \end{cases}.$$

Figure 4 shows an example of the landscape of this model using different values of p. By counting how many representatives are found in a randomly initialized sample of individuals it is possible to use the formula inversely to determine feasible pairs of k and r for the given problem.

The individuals for the sample are not generated one by one, but by chunks of N individuals (for all experiments in this paper $N = 500$). After evaluating N rules, it might be possible that we do not find any representatives. This could happen due to several reasons. Either the sample is too small and/or the probability of a representative for a particular point is too small as well. To solve these problems the system increases iteratively the total sample size (generates N additional samples) until R representatives (hyper-parameter set by the user) are found. This guarantees that the number of representatives found is not zero while checking the lowest number of individuals as possible. A high value of R will involve checking a bigger sample size, while a small value would have the opposite effect.

Moreover, we try to generate R representatives using the largest value of p possible, because that would create more general rules, as shown in Figure 4. However, more general random rules are more likely to make mistakes. Therefore, when the problem has a larger k, smaller values of p are needed to generate rules

that do not make mistakes. The procedure that the system follows to adjust p, while finding the representatives, is shown in Algorithm 5.1. Within this algorithm, *genSample* initializes N rules following the procedure explained in section 3.1. Moreover, *getAccuracy* computes the accuracy of a rule across the training set. Finally, getMostFrequentK simply identifies the k value most frequently used within the set R of representatives generated by the heuristic.

The system first tries to obtain representatives generating populations of size N created using the largest value of p: *pmax*. Then all these individuals are evaluated against the training set. Afterwards, all the rules with accuracy higher or equal than *minAc* are considered representatives. When R or more representatives are found the system returns the representatives found. If the system has already checked 6 samples and has not found any representatives, the value of p is lowered globally across the system and the search continues. If the value of p has reached its minimum value and the system has not found representatives yet, the search aborts.

The calculation of representatives from a given sample is interesting because it gives room for our third criterion, which is the number of attributes observed in them. However, the identification of the genuine representatives is far from trivial, and some post-processing is needed as it will be explained in the next section.

5.1.3. Number of Attributes in the Representatives

According to the definition of a representative the number of relevant attributes in a candidate representative cannot be less than k, otherwise this rule would make mistakes. This gives us at least an upper bound of the problem's k. However, in order to use the characteristics of the representatives we need to make sure that these "good rules" are actually genuine representatives and that they have the minimum possible amount of attributes without making mistakes.

Two problems might arise with the good rules. First, it can happen that the good rules have more attributes specified in the actual terms of the problem, which is misleading. Second, when the problem consists in more than one rule, the system might find classifiers that do not make any mistakes but they do not represent the terms of the problem. These classifiers instead represent the union or intersection between two or more terms. They might have more or less attributes expressed than k, and they are not really representatives of the problem. Therefore, in order to find genuine representatives it is necessary to post-process the rules in the sample.

To tackle the first problem we need to eliminate the attributes that do not affect the accuracy. As shown in Algorithm 5.1 this is done right before adding the rule to the representative set. To prune unnecessary attributes we perform an iterative search process as shown in Algorithm 5.2. Each one of the attributes in the rule is eliminated, one by one. If the accuracy decreases, the classifier is restored, if not the search continues over the resulting classifier. This local search operator was already proposed by [65] as a post-processing operator to refine the generality of the rules.

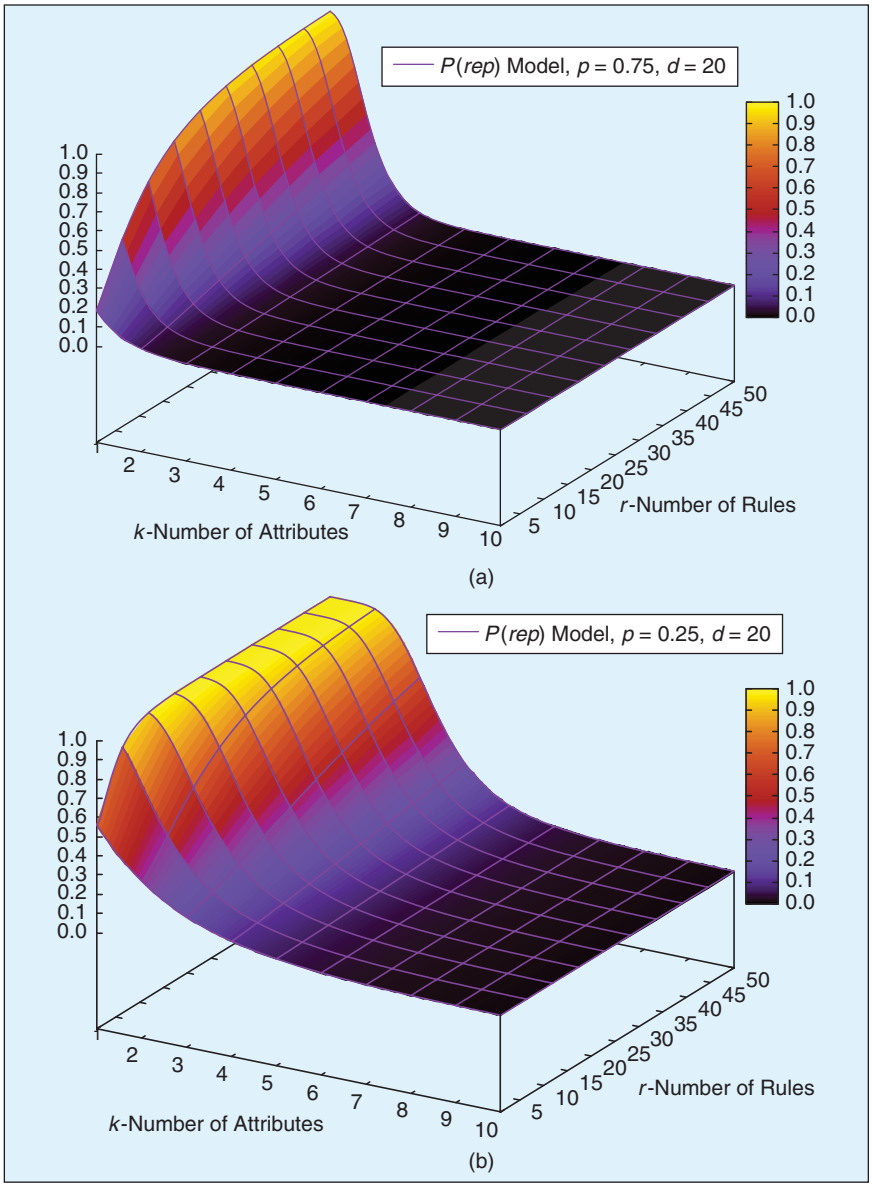


FIGURE 4 Probability of generating a representative with different values of p in a problem with $d = 20$ and *ExpAtts* = 15. (a) $p = 0.75$; (b) $p = 0.25$.

Algorithm 5.1 SEARCHREPS(*N*).

```
p ← pmax
while p ≥ pmin
    do { rep ← ∅
         while i < 6 ∨ rep ≠ ∅
             do { sample ← genSample(N, p)
                  for c ∈ sample
                      do { if getAccuracy(c) ≥ minAcc
                           then { c ← pruning(c)
                                  rep ← rep ∪ c }
                  if |rep| ≥ R
                      then { k* = getMostFrequentK(rep)
                             for c ∈ rep
                                 do { if |c.atts| ≠ k*
                                      then rep = rep − c }
                             return (rep) }
                  i = i + 1 }
         p = p − pstep }
return (null)
```

Algorithm 5.2 PRUNING(*Classifier* c1).

```
prevacc ← getAccuracy (c1)
for each att ∈ getAttributes (c1)
    do { removeAttribute (att, c1)
         if getAccuracy (c1) >= prevacc
           then prevacc ← getAccuracy(c1)
           else restore (att, c1) }
return (c1)
```

To tackle the second problem we need to eliminate the deceptive representatives. That is, the classifiers that do not make mistakes but do not really correspond to the terms of the problem we want to learn. Since these rules usually have a number of attributes either larger or smaller than k (but not exactly k), we keep only the ones that correspond to the most frequent number of attributes observed k^* in the set of R representatives. In the end the rules left in the set are considered genuine representatives and they are the input for the second criterion (Section 5.1.2). Moreover, as we have already calculated the most frequent number of attributes observed among the representatives, we can also use this information k^* as a third metric to award points to the *kr-groups* with $k = k^*$.

5.2. Classifying the Problems

As we explained before, to use the model it is necessary to calculate the standard deviation boundaries for each one of the possible combinations of k and r. For this, we sample uniformly the *kr–space* and we calculate the expected value, the lower bound and the upper bound for each point. To sample the space we calculate these values for $k = \{1..d\}$ and $r = \{1..100\}$ using a step size of 5 for the rules.

Moreover, to calculate the lower and upper bounds for each point we need to analyze further the probabilistic models used. For the probability of a representative we can consider that the probability of having a specific number of representatives in a sample of N classifiers follows a binomial distribution with probability $P(rep)$. This assumption comes from the fact that the generated rules have the same probability of becoming representatives and they are independent from each other. Therefore, the probability of having x representatives can be written as follows:

$$P(\#rep = x) = \binom{N}{x}(1 - P(rep))^{N-x}(P(rep))^{x}.$$

In this case we know that for each point the mean percentage of representatives is $P(rep)$ and the variance is $var = (P(rep)) * (1 - P(rep))$. So for a problem to belong to a specific *kr–group*, we check if the empirical percentage of representatives observed $P'(rep)$ is between the boundaries as follows:

$$P(rep)_r^k - var \leq P'(rep) \leq P(rep)_r^k + var.$$

In the case of the class imbalance we actually do not know the probability distribution, but we know the mean value given k and r. We cannot assume that it is a binomial distribution because the examples in a training set are not independent observations (usually they are not repeated). In this case fixed intervals are used to determine if a problem belongs to a certain group. To do this we calculate the empirical value $P'(neg) \pm 0.1$ to classify the problems as follows:

$$P(neg)_r^k - 0.1 \leq P'(neg) \leq P(neg)_r^k + 0.1.$$

5.3. Hyper-Parameter Setting Procedure Step-by-step

To recapitulate and explain better how the concepts and methods presented are merged together to produce our approach, this section will explain step-by-step the algorithm used within BioHEL to determine the k and r of a problem and its corresponding coverage breakpoint. The algorithm consists in the following steps also shown graphically in Figure 5.

1) Determining the number of attributes in the problem d and the value l_d.
2) Searching R representatives in a randomly initialized sample. Finding representatives involves the following sub-steps:
   a) **Representative generation** Searching iteratively in population of N classifiers until a total of R representatives is found, by evaluating them across the whole training set.
   b) **Representative pruning.** When a good rule is found, the system removes all the attributes that can be eliminated without degrading the accuracy.
   c) **Adjustment of initialization hyper-parameters.** If the system has checked already 6 populations of size N and have not yet found any representatives, the system re-adjusts the value p (See Section 5.1.2).
3) Calculating the most frequent number of attributes activated in the candidate representatives and erasing the misleading ones, keeping only the ones that have a k equal to the most frequent value observed $k^\star$.
4) Determining the number of examples in the training set belonging to the default class ($P'(neg)$).

5) Calculating the observed value $P'(rep)$ as the number of representatives observed divided by the total number of rules observed (total sample size).
6) Calculating the score of a *kr-group* ($Score_r^k$) with equation (6) where $k == k^*$, Neg_r^k and Rep_r^k are boolean variables that take the value of 1 if the empirical observation matches the criteria of the *kr-group* and A, B and C are weights associated to each of the three criteria of the heuristic.

$$\begin{aligned} Score_r^k &= A \cdot (k == k^*) + B \cdot Neg_r^k + C \cdot Rep_r^k \\ Neg_r^k &= P(neg)_r^k - 0.1 \leq P'(neg) \leq P(neg)_r^k + 0.1 \\ Rep_r^k &= P(rep)_r^k - var \leq P'(rep) \leq P(rep)_r^k + var. \end{aligned} \quad (6)$$

7) To finalize, calculating which is the smallest group (smallest k and r) that obtained the highest coincidences between the 3 metrics (highest score). This k is transformed in the coverage breakpoint as $CB = 2^{-k}$.

6. Experimental Design and Results

In this section we present the experimental framework used to test our approach and the corresponding results. First we analyze the hyper-parameter setting approach over a wide variety of synthetic k-DNF problems, with and without noise, created using the generator we provide in http://ico2s.org/datasets/kdnf.html. Then we present an additional test of our approach over a binary protein structure prediction problem already used by [52] which constitutes an interesting challenge for our approach due to the high levels of noise found in the problem. The code and all the datasets used for these experiments can be found in http://ico2s.org/data/instances/cov-break-heu/.

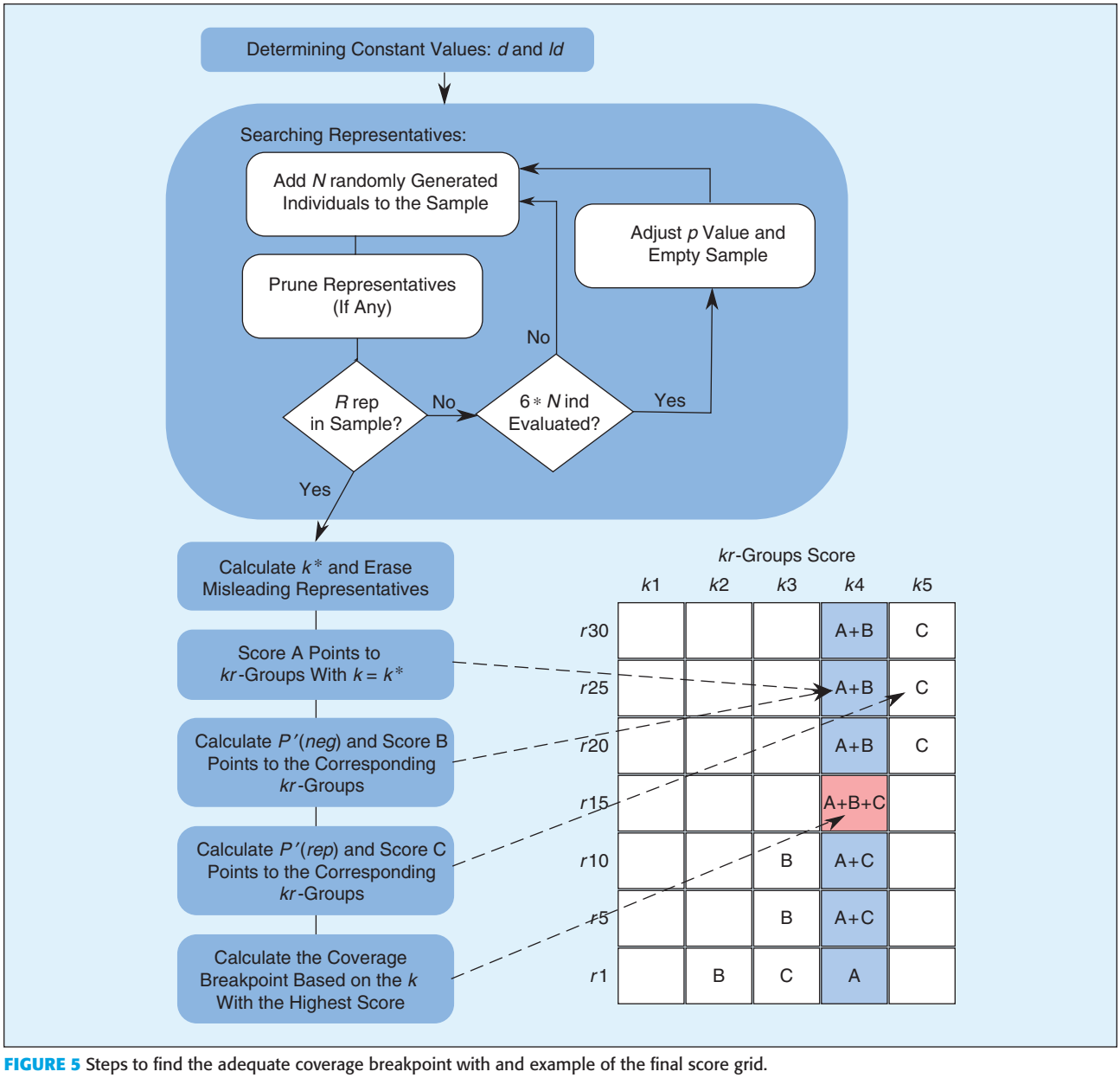


FIGURE 5 Steps to find the adequate coverage breakpoint with and example of the final score grid.

If the number of attributes in the terms is k the adequate coverage breakpoint to solve the problem is 2^k. This translates the problem of finding the adequate coverage breakpoint to finding k.

6.1. Analysis of the Hyper-Parameter Setting Approach Over Binary Problems

In this section we analyze the performance of our approach over a wide variety of k-DNF problems, in terms of probability of success (finding the adequate hyper-parameter value). At the end, we also comment briefly on the additional effort incurred by the heuristic in terms of additional evaluation operations.

The k-DNF problems used in this section have the following characteristics: $d = 20$, $k = \{2-9\}$, and $r = \{5, 10, 20, 40\}$. Moreover, we introduced output noise of 0%, 1%, 5% and 10% over the problems to determine how robust was the classification process towards noise. We generated 5 different problems of each k-DNF scenario, and each problem was run with 5 different seeds. Also, all these runs were performed using fixed default class 0. Since in the k-DNF problems all the generated terms map to class 1, this setting prevents the system form learning the inverse problem, over which calculating the success would not be straightforward.

The learning process was not performed during this stage of experiments, but only the hyper-parameter setting stage. In these experiments, we want to quantify how many times the heuristic finds the optimal k for the problem (or at least a larger one) which would ensure the learning.

We also experiment changing the hyper-parameter *minAcc* (the minimum accuracy demanded in a rule to become a representative) to determine how this hyper-parameter affects the search, and show how it can help tackling problems with noise more efficiently. In these experiments we tested hyper-parameter values $minAcc = \{1.0, 0.95, 0.9\}$. To determine significant differences among using different *minAcc* values we used a Friedman test with its post-hoc Holm test, as shown by [66].

The rest of the hyper-parameters in our approach are shown in Table 1 for clarity and replication purposes. However, according to our preliminary experiments, the hyper-parameters shown in this table can be considered constants and they can remain fixed. Only the minimum accuracy *minAcc* and the number of representatives R have an important impact on the results, because they are directly related to the problem noise and the additional search effort, respectively. Analyzing R in depth would require a very long and complex experimentation. For simplicity in this paper we have set the hyper-parameter to a value (10) that in preliminary work showed to be suitable for all tested scenarios, although a smaller R would also work in some of the easier datasets. Moreover, it should be noted that the reason why the $P(rep)$ function has a score lower than the two other metrics is because, in preliminary experimentation, this metric was not as reliable as the other ones.

TABLE 1 Hyper-parameters for the heuristic used to characterize and find the coverage breakpoint for k-DNF problems.

HYPER-PARAMETER	VALUE
NUMBER OF REPRESENTATIVES NEEDED – R	10
EVALUATED POPS TO CHANGE p	6
MOST FREQUENT k IN REPRESENTATIVES – SCORE A	2
IMBALANCE FUNCTION – SCORE B	2
PREP FUNCTION – SCORE C	1
SAMPLE SIZE – N	500

6.1.1. Results

Table 2 presents the results for the different k-DNF configurations and different values of *minAcc* in terms of percentage of success (finding the k of the problem or at least a larger one). Results for $k = 2$ and $r = 40$ are omitted because in these setting all instances belong to class 1. The cells emphasized represent the configurations where the success rate is less than 100%. For $minAcc < 1$, the cells marked with red show the cases where the success rate is lower than the base case ($minAcc = 1.0$), and the cells marked with green show the cases where the success rate increased. In this table we can observe that using $minAcc = 1$ the heuristic is able to find the appropriate coverage breakpoint for most of the configurations with no noise. Moreover, it is noticeable that the output noise affects the performance of the heuristic.

On the other hand, we can observe that the heuristic fails in the cases where there is very high rule overlapping. These problems are very difficult to solve by the system because of the class imbalance [30], so it is not surprising that they are difficult for the heuristic as well. Such large overlapping makes the heuristic think that the k is smaller than the real one. In the case of a synthetic problem like the k-DNF, where we know the correct answer, this is incorrect. However, what the system is trying to do is not completely wrong, because it is trying to solve the problem with less complex rules compromising the accuracy slightly, which in real-life domains can be advantageous. To understand better these domains further research focusing specifically on the rule overlapping scenario is required.

Figure 6 shows an example of the final score grid of the heuristic using $minAcc = 1$ for a problem with $k = 5$ and $r = 20$ with the 4 levels of noise. In each of the four plots (for a different level of noise) the vertical stripe corresponds to the 2 points that are awarded to the most frequent number of attributes observed in the representatives. The curved stripes correspond to the scores awarded by the other two criteria of the heuristic. In this figure we can see that while the problem increases the representatives present a higher number of attributes. When this happens this area does not intersect the two other areas, thus the heuristic fails to find the appropriate k value. This is because the constraint of $minAcc = 1.0$ is too strict in these

cases where the problem has noise. In the cases with noise we should take in consideration that good rules will have a good accuracy, but not equal to 1. Relaxing this hyper-parameter, as we can see in Table 2, helps finding adequate representatives for problems with noise. We can observe here that the success rate increases in most cases, except when the problem has no noise. As we expected to observe, when the problem has 5% noise, the best results are obtained using *minAcc* = 0.95, and the same occurs when the problem has 10% noise and we use *minAcc* = 0.9.

Moreover, Table 3 contains the statistical analysis to determine which value of *minAcc* is best for the different sets of problems, distinguishing them by the amount of noise. In this table we can observe that the best method changes as expected, and for the problems with high amount of noise (5% and 10%) the respective best method (*minAcc* = 0.95, *minAcc* = 0.90) performs better than the rest of the configurations.

Based on these results we can state that for problems with noise we should use a minimum accuracy equal to the percentage of noise observed in the problem. Even though this

TABLE 2 Probability of success in *k*-DNF problems with different amounts of output noise and minimum accuracy thresholds (*minAcc*). Cells in bold emphasize the cases where the heuristic was not 100% successful. Cells in green and red emphasize the cases where using a lower *minAcc* value obtained better and worst results, respectively.

		***minAcc* = 1.0**				***minAcc* = 0.95**				***minAcc* = 0.9**			
PROB *d* = 20		**NOISE LEVEL**				**NOISE LEVEL**				**NOISE LEVEL**			
k	*r*	0%	1%	5%	10%	0%	1%	5%	10%	0%	1%	5%	10%
2	5	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	**72.00**	100.00	100.00	100.00
2	10	100.00	**0.00**	**0.00**	100.00	**32.00**	**64.00**	100.00	100.00	**0.00**	**4.00**	**20.00**	100.00
2	20	**80.00**	**4.00**	**0.00**	**0.00**	**32.00**	**0.00**	100.00	**0.00**	**32.00**	**0.00**	**24.00**	100.00
2	40		**0.00**	**0.00**	**0.00**		**0.00**	**80.00**	**0.00**	**0.00**	**4.00**	**0.00**	**96.00**
3	5	100.00	100.00	**24.00**	**0.00**	100.00	100.00	100.00	**4.00**	**96.00**	100.00	**88.00**	100.00
3	10	100.00	**84.00**	**44.00**	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	20	100.00	**0.00**	**0.00**	**0.00**	**16.00**	**80.00**	100.00	**0.00**	**0.00**	**0.00**	**12.00**	100.00
3	40	100.00	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	100.00	**0.00**	**0.00**	**0.00**	**0.00**	100.00
4	5	100.00	100.00	**68.00**	**4.00**	100.00	100.00	100.00	**28.00**	100.00	100.00	100.00	100.00
4	10	100.00	**96.00**	**16.00**	**0.00**	100.00	100.00	100.00	**0.00**	100.00	100.00	100.00	100.00
4	20	100.00	**4.00**	**0.00**	**60.00**	100.00	100.00	100.00	**60.00**	**92.00**	100.00	100.00	100.00
4	40	100.00	**0.00**	**0.00**	**0.00**	**4.00**	**24.00**	100.00	**0.00**	**0.00**	**0.00**	**4.00**	100.00
5	5	100.00	100.00	100.00	**44.00**	100.00	100.00	100.00	**44.00**	100.00	100.00	100.00	100.00
5	10	100.00	100.00	**64.00**	**0.00**	100.00	100.00	100.00	**8.00**	100.00	100.00	100.00	100.00
5	20	100.00	**60.00**	**8.00**	**0.00**	100.00	100.00	100.00	**0.00**	100.00	100.00	100.00	100.00
5	40	100.00	**4.00**	**0.00**	**0.00**	100.00	100.00	100.00	**0.00**	**92.00**	**80.00**	100.00	100.00
6	5	100.00	100.00	100.00	**60.00**	100.00	100.00	100.00	**88.00**	100.00	100.00	100.00	100.00
6	10	100.00	100.00	100.00	**28.00**	100.00	100.00	100.00	**8.00**	100.00	100.00	100.00	100.00
6	20	100.00	100.00	**48.00**	**0.00**	100.00	100.00	100.00	**0.00**	100.00	100.00	100.00	100.00
6	40	100.00	**40.00**	**0.00**	**0.00**	100.00	100.00	**80.00**	**0.00**	100.00	100.00	100.00	**84.00**
7	5	100.00	100.00	100.00	**88.00**	100.00	100.00	100.00	**92.00**	100.00	100.00	100.00	100.00
7	10	100.00	100.00	100.00	**28.00**	100.00	100.00	100.00	**48.00**	100.00	100.00	100.00	100.00
7	20	100.00	**88.00**	100.00	**4.00**	100.00	100.00	100.00	**16.00**	100.00	100.00	100.00	**92.00**
7	40	100.00	**76.00**	**4.00**	**0.00**	100.00	100.00	**92.00**	**0.00**	100.00	100.00	100.00	**76.00**
8	5	100.00	100.00	100.00	**80.00**	100.00	100.00	100.00	**76.00**	100.00	100.00	100.00	**96.00**
8	10	100.00	100.00	100.00	**68.00**	100.00	100.00	100.00	**72.00**	100.00	100.00	100.00	100.00
8	20	100.00	100.00	100.00	**4.00**	100.00	100.00	100.00	**12.00**	100.00	100.00	100.00	**80.00**
8	40	100.00	**84.00**	**92.00**	**0.00**	100.00	**84.00**	100.00	**0.00**	100.00	**84.00**	100.00	**44.00**
9	5	100.00	100.00	**84.00**	**48.00**	100.00	100.00	**92.00**	**24.00**	100.00	100.00	**88.00**	**68.00**
9	10	100.00	100.00	**96.00**	**68.00**	100.00	100.00	100.00	**72.00**	100.00	100.00	100.00	**84.00**
9	20	100.00	100.00	100.00	**36.00**	100.00	100.00	100.00	**32.00**	100.00	100.00	100.00	**92.00**
9	40	100.00	100.00	**80.00**	**4.00**	100.00	100.00	**80.00**	**8.00**	100.00	100.00	**72.00**	**44.00**

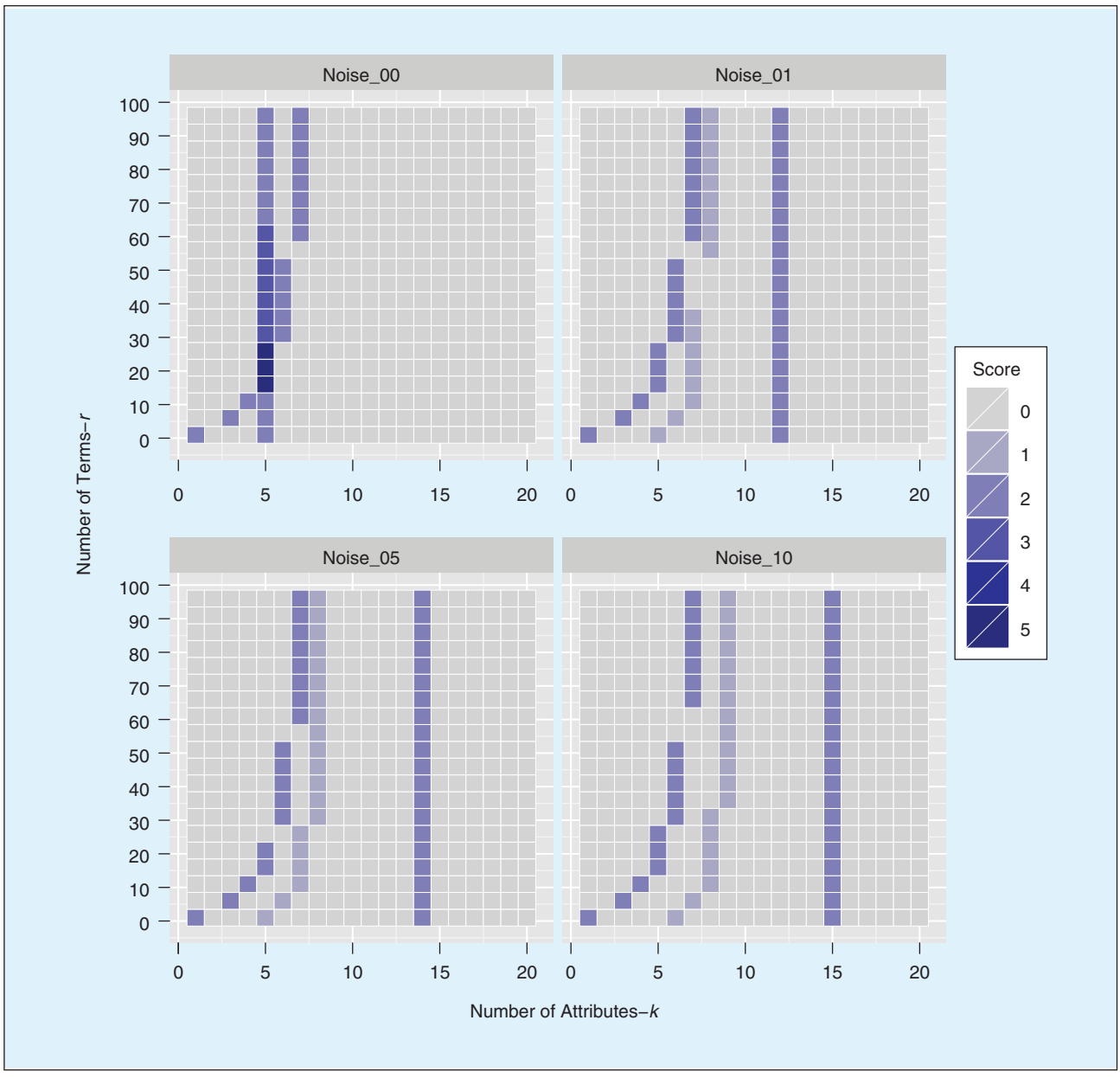


FIGURE 6 Final score grids of the heuristic using *minAcc* = 1 in a problem with $d = 20$, $k = 5$ and $r = 20$ for the 4 levels of noise.

TABLE 3 Results of the Friedman test performed to determine the best *minAcc* value depending on the noise. *Control* shows which was the algorithm that obtained the best ranking. *Dominance* shows the configurations that are significantly worst than the control configuration according to the Holm post-hoc test with confidence threshold 0.05.

			CONTROL			
d	**NOISE**	***p*-VALUE**	**1.0**	**0.95**	**0.9**	**DOMINANCE**
20	0%	0.00050	*			0.9
20	1%	0.00100		*		1.0
20	5%	1.895E-07		*		1.0, 0.9
20	10%	1.119E-11			*	1.0, 0.95

introduces a new hyper-parameter *minAcc*, the percentage of permitted noise is a much more intuitive hyper-parameter to set up than the coverage breakpoint, since it is an structural hyper-parameter of the problem, instead of being a hyper-parameter of the system.

One interesting aspect to notice in Table 2 is that for problems with larger k, the heuristic seems to fail when the problem has either too many or too few terms. When the problem has too many terms, it is possible to find representatives but these representatives are likely to have a large k and this value might not intersect with the other two areas, as it was exemplified in Figure 6. Moreover, if the problem has few terms finding a representative becomes very difficult and it is possible that

the system does not find any representatives during the search process. In this case the mechanism will only rely on the imbalance metric to make a decision.

Our hypothesis is that these difficulties can be tackled by changing the selection policy of the k when there is not a single cell where the three metrics have intersected. Moreover, adjusting the minimum accuracy along the search process or a more granular step size in the adaptation of p could help obtain better results by finding representatives when this task is very difficult. More experimentation is needed to validate these hypotheses.

We present an additional test of our approach over a binary protein structure prediction problem already used by [52] which constitutes an interesting challenge for our approach due to the high levels of noise found in the problem.

Based on these results, we can conclude that our hyper-parameter setting mechanism is able to find the appropriate k value for a wide variety of binary problems, including problems with noise. We explore next the computational overhead of our approach.

6.1.2. Computational Effort of the Heuristic

As we already explained in previous sections, our hyper-parameter control method involves an additional computational effort before the learning process. This extra effort of the heuristic includes the evaluation of the randomly initialized individuals used to find the adequate representatives plus the number of evaluations needed to prune the representatives.

Figure 7 shows the additional effort incurred by our approach for problems with no noise using a $minAcc = 1.0$. The effort is shown in terms of the number of rule evaluations (matching the rule against the complete training set and computing its accuracy). Execution time is not shown as it is proportional to the number of evaluations. In this figure, we can observe that while the k increases it becomes more expensive to run the hyper-parameter setting approach and more iterations are needed to find representatives.

Moreover, it is also noticeable a spiking behavior in the additional effort. This behavior is clarified by Figure 7(b), which shows the frequency in which each value of p is selected. As it was explained before, our approach also adapts the p value to increase the probability of finding representatives when this task becomes very difficult. As we can see, the different stages in the behaviour of the effort observed in Figure 7 correspond to the transition stages between different values of p. When the system uses a smaller p the additional effort to find representatives becomes smaller. A further evaluation of the computational effort of the heuristic is available at [67].

6.2. Evaluation on a Real-world Problem

In order to test the performance of our method over real-world domains, we selected a binary protein structure prediction problem already used in [52]. Specifically, the problem being addressed is called *contact number prediction*. In this problem, for each amino-acid of a protein's sequence, the goal is to predict the number of other amino-acids that in the folded protein state are located at a distance less than a threshold d. In this case the contact number is binarised to (high/low) states to convert the problem into a binary classification dataset. The contact number state of an amino-acid is predicted from information about itself and its immediate ± 4 neighbors in the protein sequence. The information about each amino-acid (which roughly captures the hydrophobicity physico-chemical property of the amino-acid) is represented by a binary variable generated with the method presented in [52]. Hence, each instance in the dataset is represented by 9 binary attributes. This dataset has a total of 257,560 instances.

It is worth mentioning that this problem has a very high noise ratio, and there are no possible rules that have 100% accuracy. Therefore, in order to test the heuristic we have to relax the minimum accuracy required for the classifiers to be representatives to 0.7. This is the maximum value for $minAcc$ for which our heuristic could actually identify representatives. The rest of the heuristic is applied to this dataset exactly as in the k-DNF datasets.

In this section we are going to analyze the results obtained with our approach by comparing them with an exhaustive experimentation to determine the adequate coverage breakpoint. In the exhaustive search we used coverage breakpoint values ranging from 2^{-2} to 2^{-9} (as this is the maximum possible since the problem has 9 attributes), and different values of p (0.75, 0.5 and 0.25). Since we are dealing with a real problem without a known structure, we don't know ahead of time what is the appropriate k for the dataset. Hence, the verification is going to be experimental by running the whole BioHEL algorithm, and determine the success of the heuristic based on the obtained test accuracy. The hyper-parameters for the BioHEL system are the ones uses in [27] except only for three hyper-parameters shown in Table 4. The "Initial MDL TL ratio" is used in the heuristic proposed in [59] to automatically tune the W hyper-parameter of BioHEL's fitness formula (equation 1). This hyper-parameter defines the expected contribution that the TL part of the fitness formula should have in good rules. The "number of windows in ILAS" is used within the Incremental Learning with Alternative Strata (ILAS) scheme employed by BioHEL to speed up its fitness evaluations. In ILAS the training set is divided into a certain number of strata, called *windows*. Each GA iteration uses a different window for its fitness computations using a round-robin policy.

For the analysis of the results, we will first apply a Wilcoxon pairwise test [68] to determine if there are significant differences between using different coverage breakpoints in this problem and which is the most adequate hyper-parameter value to solve it. Having found the adequate coverage breakpoint, we will then compare it with the results obtained by our approach. The results are going to be analyzed in terms of accuracy, execution time and convergence time of both methodologies.

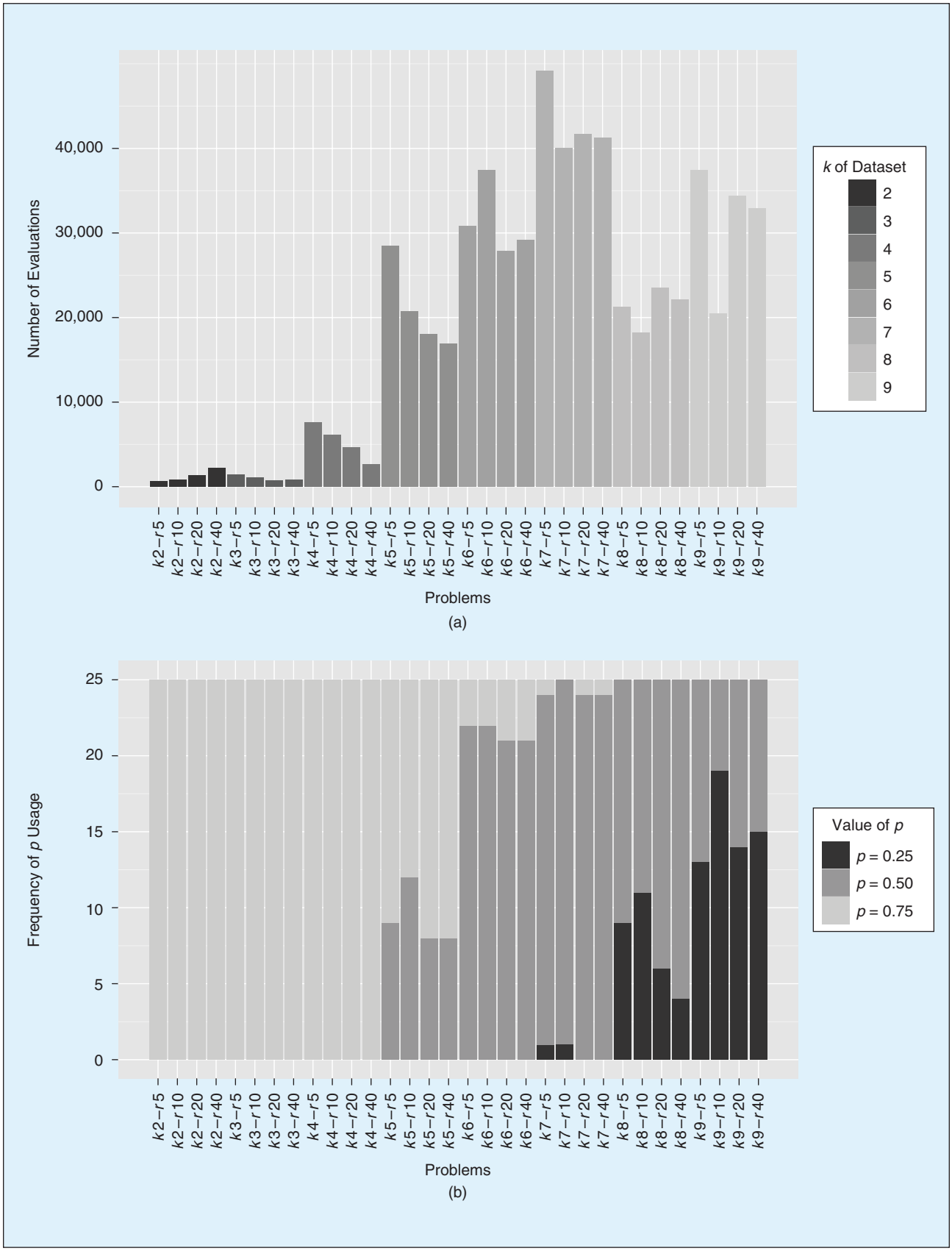


FIGURE 7 Number of rule evaluations and frequency of selection of a specific p value depending on problem characteristics (k, r) in our k-DNF experiments. (a) Number of evaluations; (b) Frequency of p usage on the 25 experiments run for each k-DNF variant.

TABLE 4 Hyper-parameters for the BioHEL system and the heuristic to determine the problem structure.

BIOHEL HYPER-PARAMETER	VALUE
GA ITERATIONS	50
INITIAL MDL TL RATIO	0.025
NUMBER OF WINDOWS IN ILAS	20
HEURISTIC HYPER-PARAMETERS	**VALUE**
SAMPLE SIZE – N	1000
NUMBER OF REPRESENTATIVES NEEDED – R	20

Regarding the dataset, this problem was separated in 10 training and test sets for ten-fold cross-validation. Each one of these problem instances was run with 5 different seeds, so the results are the average of 50 runs.

Table 5 shows the results in terms of accuracy, average time per run and the total time of the experiments when applying different coverage breakpoints of the type $cov = 2^{-k}$ where $k = \{2..9\}$. In this table, we can see that the coverage breakpoint that obtains the best results in terms of accuracy is 2^{-9}. Also, for this problem there are no differences between using different values of p in terms of accuracy or execution time. Moreover, in Table 6 we can see that the coverage breakpoint 2^{-9} is not significantly different than applying other values such as 2^{-7} or 2^{-8}. However, the maximum average accuracy is obtained with the smaller coverage breakpoint.

In Table 7 we can observe the results obtained with our approach. Using a minimum accuracy of 0.7 the system determined that the k of the problem is equal 9 in the majority of the cases and applying the corresponding coverage breakpoint we obtain an accuracy equal to our best results in preliminary experimentation. We also tested a more relaxed *minAcc* value without obtaining good results. Based on this results we can conclude that the heuristic, when using the appropriate noise ratio, is able to categorize this real problem correctly and determine the appropriate coverage breakpoint automatically.

Regarding the computational time, the total amount of CPU time invested in performing the preliminary experiments with the different coverage breakpoints and different values of p is equal to ≈285 hours. On the other hand, the amount of time invested in running the experiments automatically setting the coverage breakpoint, using different *minAcc* values and including the whole learning process was ≈ 81 hours. This constitutes 28% of the time invested in the preliminary experiments, which means a reduction of 71% in the total experimental time.

Moreover, our method also adapts the value p, the probability of setting the bits to 1 in the GABIL representation. Figure 8 reports the average accuracy of the

TABLE 5 Results over the binary PSP problem using fixed coverage breakpoints (where $cov(k) = 2^{-k}$) and different p values.

cov (*k*)		TEST ACCURACY (%)	AVERAGE TIME (s)	TOTAL TIME (s)
$p = 0.75$	2	70.15±0.96	244.67±120.89	12233.71
	3	70.27±0.84	427.00±207.95	21350.19
	4	71.46±0.56	378.64±207.36	18932.08
	5	71.79±0.47	386.40±147.90	19319.86
	6	72.12±0.45	640.26±238.88	32013.22
	7	72.32±0.52	874.15±322.80	43707.67
	8	72.40±0.47	1469.14±496.96	73457.19
	9	72.45±0.47	2526.63±723.67	126331.39
TOTAL TIME			(≈96.48 H)	347345.31
$P = 0.5$	2	70.15±0.96	247.72±101.81	12385.91
	3	70.27±0.84	342.26±180.62	17112.77
	4	71.47±0.56	390.89±208.98	19544.62
	5	71.79±0.47	398.20±180.34	19909.83
	6	72.12±0.45	639.17±240.29	31958.73
	7	72.31±0.53	896.59±287.78	44829.29
	8	72.40±0.48	1601.17±445.91	80058.71
	9	72.45±0.47	2244.94±650.95	112246.99
TOTAL TIME			(≈93.90 H)	338046.85
$P = 0.25$	2	70.46±0.84	239.53±91.12	11976.29
	3	70.27±0.84	362.16±191.81	18107.84
	4	71.46±0.56	332.23±183.90	16611.51
	5	71.80±0.48	409.98±167.93	20498.96
	6	72.12±0.45	622.01±230.56	31100.48
	7	72.31±0.53	846.66±272.08	42333.25
	8	72.40±0.48	1636.27±482.24	81813.37
	9	72.45±0.47	2338.56±586.86	116927.77
TOTAL TIME			(≈94.26 H)	339369.47

TABLE 6 P-values of the Wilcoxon pairwise test to determine significant differences between the usage of different coverage breakpoints of the type 2^{-k} in the binary PSP problem. The cells in bold indicate the cases where there are significant differences.

P-VALUES OF THE WILCOXON PAIRWISE TEST							
k	**2**	**3**	**4**	**5**	**6**	**7**	**8**
3	0.52092	–	–	–	–	–	–
4	**2.9E-07**	**1.2E-08**	–	–	–	–	–
5	**1.5E-11**	**2.9E-11**	**0.01510**	–	–	–	–
6	**4.9E-15**	**4.9E-15**	**2.5E-06**	**0.00265**	–	–	–
7	**<2E-16**	**<2E-16**	**1.0E-08**	**0.00016**	0.21071	–	–
8	**<2E-16**	**<2E-16**	**6.1E-10**	**6.1E-06**	0.05857	0.57281	–
9	**<2E-16**	**<2E-16**	**7.5E-11**	**2.8E-06**	**0.01079**	0.21071	0.57281

The heuristic reduces the computational time needed to set up the algorithm properly.

best rule along the 50 iterations of the GA with different p values. So far the use of adapting this hyper-parameter was just to find representatives during the hyper-parameter control stage, but as this figure shows, using $p = 0.25$ makes the system find good classifiers quicker (although all three values of p eventually manage to learn the correct rules). This is because in this particular problem using a small value of p increases the odds of finding representatives. The spiking behavior in the figure is a normal phenomena due to the usage of ILAS windowing scheme [59], and it is not related to the approach presented in this paper.

In this sense, we can say that the heuristic reduces the computational time needed to set up the algorithm properly, which for large problems can constitute a considerable amount of CPU time. Moreover, it adapts other hyper-parameters of the system, such as the p value, which helps finding good rules quicker within the genetic algorithm.

TABLE 7 Performance of the heuristic over the binary PSP problem.

MIN ACC	*k*	*p*	TEST ACC	AVE. EXEC TIME	TOTAL TIME
0.7	8.98±0.14	0.28±0.08	72.45±0.47	2913.93±1356.41	145696.66
0.6	5.62±1.12	0.73±0.06	71.94±0.74	902.29±382.38	45114.63
TOTAL TIME					(≈80.94 h) 291393.32

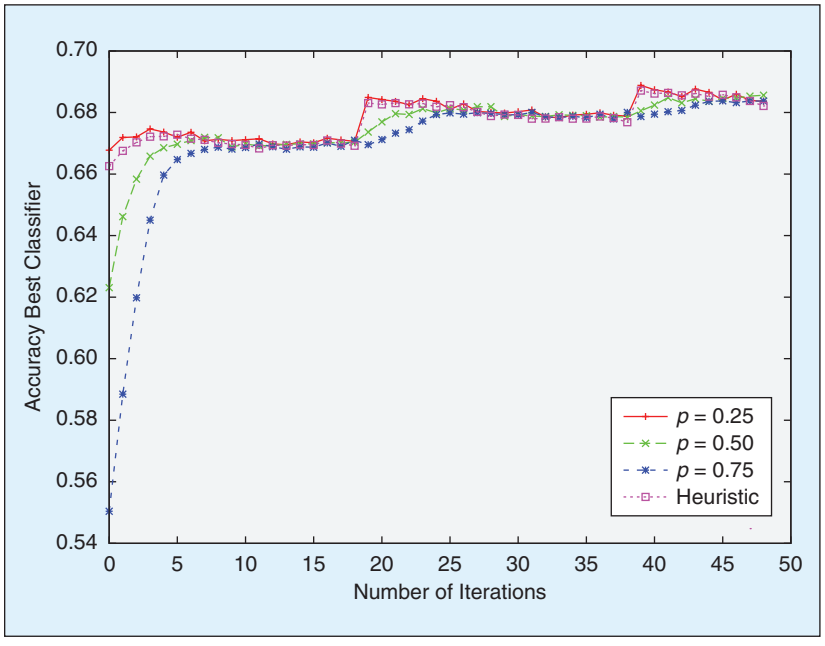


FIGURE 8 Average accuracy of the best classifier during the 50 iterations of the GA using different values of p.

7. Discussion and further work

The initial objective of this paper was to design an automatic procedure to learn the structure of the problem and set the coverage breakpoint hyper-parameter in the BioHEL learning system for (a broad class of) binary classification domains. Our proposed approach does this by using simple models that correlate the behaviour of the system and characteristics of the data to the problem structure. Our thorough experiments show that our procedure is able to successfully estimate the problem's structure (k and r) in binary problems with noise (using many variants of the $k - KDF$ family of boolean functions) and in a real-world problem of protein structure prediction.

Using this method we are able to set up the coverage breakpoint hyper-parameter in BioHEL, facilitating in this way the learning process, reducing the computational time of preliminary experiments and making the system easier to use to an end-user. The final validation stage using a challenging protein structure prediction problem showed that the heuristics works for large real-world problems as well, managing to reduce the total experimental time by 71%. Moreover, our approach also adapts other hyper-parameters like p, the probability of settings bits to 1 in the GABIL representation, which can help finding good rules faster within the GA. And while the objective of this heuristic is to automatically tune crucial hyper-parameters of BioHEL, we are aware that we are introducing several new hyper-parameters. In our experimentation we show that most of these can remain at fixed values and still obtain good performance. The only crucial hyper-parameter to set up is *minAcc*, that specifies that minimum accuracy that the sample of initial rules should have to become representatives. As we argue in the paper, our view is that this hyper-parameter is much more intuitive to set up as it relates to the uncertainty of the dataset rather than being specific to the machine learning algorithm.

While the heuristic is designed with a specific system in mind, BioHEL, we believe that these design principles can be adapted to other machine learning algorithms to develop equivalent heuristics. First, the methodologies presented in this paper to find the structure of the problem can be extended to other systems by using models developed particularly for these systems based on the characteristics of the problem. Knowing in advance the characteristics of the problem at the beginning of the learning process can be advantageous

to guide the search and also set hyper-parameters within the system. For instance, in the GAssist system [59] an adapted heuristic could be used to set the minimal rule set size penalty and the minimal number of rules for the rule deletion operator. For XCS, estimations of k could be used to automatically set up some of its hyper-parameters (e.g. mutation rate, population size), using the models proposed in [22]. Moreover, an adaptation of this heuristic could be used to seed the initial population, as suggested in [64].

The approach we decided to take in this paper is quite different from the *AutoML* approaches which broadly speaking (as explained in the related work section), apply a wrappers on top of the machine learning algorithm. These approaches have shown to be effective across many different domains but all they give you is the set of optimal hyper-parameters (or, in cases like TPOT, also the features selected) to maximize predictive performance. Our approach, while requiring a deep knowledge of the knowledge representation used within BioHEL, is able to estimate knowledge about the structure of the problem being studied, and at the same time as making the process of algorithm tuning easier for the users, is able to provide them with insight about the data.

Moreover, there are several potential lines to extend this work. Firstly, we would like to extend this approach to problems with χ-ary discrete[2] and continuous attributes. In the case of χ-ary attributes, the coverage would not only depend on k but would also depend in the number of values activated in an attribute (number of 1s in the GABIL representation), while for continuous domains coverage depends also on the size of the intervals, and would require a substantial (and challenging) rewrite of all the probabilistic models. If we can apply this heuristic to a more broad class of datasets then we can revisit our observations that most of the heuristic's parameters can remain fixed. Moreover, it would be interesting to investigate policies of selecting the k for the cases in which there is no single cell where the three metrics intersect. In these cases probably it is better to select a k based on the average of the cells that obtained the highest score. In relation to the tuning of the *minAcc* hyper-parameter of our heuristic, there are recent approaches to estimate the uncertainty of dataset labels [69], while it would also be interesting to revisit the use of complexity measures for classification datasets [49] to reliably estimate appropriate values for such hyper-parameter. As our experiments show, our heuristic struggles in scenarios with high rule overlap because k is underestimated as we are not able to identify good representatives. To this aim it would be interesting to explore how we could use *absumption* mechanisms [70] and other types of local search operators [65] to generate better representatives while keeping the computational effort of the heuristic under control.

Finally, a more extensive analysis of the additional effort required by this heuristic should be carried out, focusing on how the hyper-parameter R (target number of representatives to be generated) and the space sampling can reduce or regulate the extra computational effort, and estimating what is the worse-case effort that the heuristic would require. In relation to the adaptation of these approaches to other machine learning algorithms, we have pointed above several potential scenarios where this adaptation could take place. In such case this heuristic probably should be rewritten to become much more modular, and be able to separate the method/representation-specific components from those that could be reused across ML algorithms.

[2] Discrete domains where each attribute has more than two possible values.

Acknowledgments

We are very grateful for the detailed feedback provided by the anonymous reviewers which has made this paper better. We acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) under grants EP/H016597/1, EP/M020576/1 and EP/N031962/1.

References

[1] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore, "Automating biomedical data science through tree-based pipeline optimization," in *Applications of Evolutionary Computation*, G. Squillero and P. Burelli, Eds. Cham: Springer-Verlag, 2016, pp. 123–137.

[2] F. Mohr, M. Wever, and E. Hüllermeier, "Ml-plan: Automated machine learning via hierarchical planning," *Mach. Learn.*, vol. 107, no. 8–10, pp. 1495–1515, 2018. doi: 10.1007/s10994-018-5735-z.

[3] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, "The irace package: Iterated racing for automatic algorithm configuration," *Oper. Res. Perspect.*, vol. 3, pp. 43–58, 2016. doi: 10.1016/j.orp.2016.09.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214716015300270

[4] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook: Curran Associates, 2015, pp. 2962–2970. [Online]. Available: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

[5] A. Fernández, S. García, J. Luengo, E. Bernadó-Mansilla, and F. Herrera, "Genetics-based machine learning for rule induction: State of the art, taxonomy, and comparative study," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 913–941, Dec. 2010. doi: 10.1109/TEVC.2009.2039140.

[6] A. Orriols-Puig, J. Casillas, and E. Bernadó-Mansilla, "Genetic-based machine learning systems are competitive for pattern recognition," *Evol. Intell.*, vol. 1, no. 3, pp. 209–232, 2008. doi: 10.1007/s12065-008-0013-9.

[7] A. L. Swan et al., "Analysis of mass spectrometry data from the secretome of an explant model of articular cartilage exposed to pro-inflammatory and anti-inflammatory stimuli using machine learning," *BMC Musculoskelet. Disord.*, vol. 14, no. 1, pp. 349, 2013. doi: 10.1186/1471-2474-14-349.

[8] A. Swan et al., "A machine learning heuristic to identify biologically relevant and minimal biomarker panels from omics data," *BMC Genom.*, vol. 16, no. Suppl 1, p. S2, Jan. 2015. doi: 10.1186/1471-2164-16-S1-S2.

[9] M. Martínez-Ballesteros, J. Bacardit, A. Troncoso, and J. C. Riquelme, "Enhancing the scalability of a genetic algorithm to discover quantitative association rules in large-scale datasets," *Integr. Comput.-Aided Eng.*, vol. 22, no. 1, pp. 21–39, Jan. 2015. doi: 10.3233/ICA-140479.

[10] S. Baron, N. Lazzarini, and J. Bacardit, "Characterising the influence of rule-based knowledge representations in biological knowledge extraction from transcriptomics data," in *Applications of Evolutionary Computation*, G. Squillero and K. Sim, Eds. Cham: Springer-Verlag, 2017, pp. 125–141.

[11] Y. Bi, B. Xue, and M. Zhang, "An automated ensemble learning framework using genetic programming for image classification," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2019)*. New York: Association for Computing Machinery, 2019, pp. 365–373. doi: 10.1145/3321707.3321750.

[12] J. Liang, E. Meyerson, and R. Miikkulainen, "Evolutionary architecture search for deep multitask networks," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2018)*. New York: Association for Computing Machinery, 2018, pp. 466–473. doi: 10.1145/3205455.3205489.

[13] A. Sohn, R. S. Olson, and J. H. Moore, "Toward the automated analysis of complex diseases in genome-wide association studies using genetic programming," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2017)*. New York: Association for Computing Machinery, 2017, pp. 489–496. doi: 10.1145/3071178.3071212.

[14] A. E. Eiben, Z. Michalewicz, M. Schoenauer, and J. E. Smith, *Parameter Control in Evolutionary Algorithms (Studies in Computational Intelligence), vol. 54*. Cham: Springer-Verlag, 2007, ch. 2, pp. 19–46. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-69432-8_2

[15] A. E. Eiben, M. Horvath, W. Kowalczyk, and M. C. Schut, *Reinforcement Learning for Online Control of Evolutionary Algorithms (Lecture Notes in Computer Science), vol. 4335*. Cham: Springer-Verlag, 2007, ch. 10, pp. 151–160. [Online]. Available: http://www.springerlink.com/index/10.1007/978-3-540-69868-5_10

[16] J. Smith and T. C. Fogarty, *Adaptively Parameterized Evolutionary Systems: Self Adaptive Recombination and Mutation in a Genetic Algorithm (Lecture Notes in Computer Science), vol. 1141*. Cham: Springer-Verlag, 1996, ch. 45, pp. 441–450. [Online]. Available: http://www.springerlink.com/index/10.1007/3-540-61723-X_1008
[17] J. Smith and T. Fogarty, "Self adaptation of mutation rates in a steady state genetic algorithm," in *Proc. IEEE Int. Conf. Evolutionary Computation 1996*, May 1996, pp. 318–323. doi: 10.1109/ICEC.1996.542382.
[18] N. Krasnogor and J. Smith, "A memetic algorithm with self-adaptive local search: TSP as a case study," in *Proc. 2nd Annu. Conf. Genetic and Evolutionary Computation (GECCO)*, L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, and H.-G. Beyer, Eds. San Mateo, CA: Morgan Kaufmann, 2000, pp. 987–994.
[19] N. Krasnogor and J. Smith, "Emergence of profitable search strategies based on a simple inheritance mechanism," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO2001)*. San Francisco, CA: Morgan Kaufmann, 2001, pp. 432–439. [Online]. Available: http://www.cs.nott.ac.uk/~nxk/PAPERS/ga140.pdf
[20] N. Krasnogor and S. Gustafson, "A study on the use of "self-generation" in memetic algorithms," *Nat. Comput.*, vol. 3, no. 1, pp. 53–76, Mar. 2004. doi: 10.1023/B:NACO.0000023419.83147.67.
[21] N. Krasnogor, "Self generating metaheuristics in bioinformatics: The proteins structure comparison case," *Genet. Program. Evol. Mach.*, vol. 5, no. 2, pp. 181–201, June 2004. doi: 10.1023/B:GENP.0000023687.41210.d7.
[22] M. V. Butz, *Rule-Based Evolutionary Online Learning Systems: A Principled Approach to LCS Analysis and Design (Studies in Fuzziness and Soft Computing), vol. 109*. Cham: Springer-Verlag, 2006.
[23] A. Orriols-Puig, X. Llorà, and D. E. Goldberg, "How XCS deals with rarities in domains with continuous attributes," in *Proc. 12th Annu. Conf. Genetic and Evolutionary Computation (GECCO'10)*, 2010, pp. 1023–1030. doi: 10.1145/1830483.1830670.
[24] P. O. Stalph, X. Llorà, D. E. Goldberg, and M. V. Butz, "Resource management and scalability of the XCSF learning classifier system," *Theor. Comput. Sci.*, vol. 425, pp. 126–141, Mar. 2012. doi: 10.1016/j.tcs.2010.07.007.
[25] M. Nakata, W. Browne, T. Hamagami, and K. Takadama, "Theoretical XCS parameter settings of learning accurate classifiers," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2017)*. New York: Association for Computing Machinery, 2017, pp. 473–480. doi: 10.1145/3071178.3071200.
[26] M. Nakata, W. Browne, and T. Hamagami, "Theoretical adaptation of multiple rule-generation in XCS," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2018)*. New York: Association for Computing Machinery, 2018, pp. 482–489. doi: 10.1145/3205455.3205465.
[27] J. Bacardit, E. K. Burke, and N. Krasnogor, "Improving the scalability of rule-based evolutionary learning," *Memet. Comput.*, vol. 1, no. 1, pp. 55–67, Mar. 2009. doi: 10.1007/s12293-008-0005-4.
[28] J. Bacardit, P. Widera, A. Márquez-Chamorro, F. Divina, J. S. Aguilar-Ruiz, and N. Krasnogor, "Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features," *Bioinformatics*, vol. 28, no. 19, pp. 2441–2448, Oct. 2012. doi: *10.1093/bioinformatics/bts472*. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/28/19/2441.abstract
[29] E. Glaab, J. Bacardit, J. M. Garibaldi, and N. Krasnogor, "Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data," *PLoS One*, vol. 7, no. 7, p. e39932, July 2012. doi: 10.1371/journal.pone.0039932.
[30] M. A. Franco, N. Krasnogor, and J. Bacardit, "Analysing BioHEL using challenging Boolean functions," *Evol. Intell.*, vol. 5, no. 2, pp. 87–102, June 2012. doi: 10.1007/s12065-012-0080-9.
[31] M. A. Franco, N. Krasnogor, and J. Bacardit, "Modelling the initialisation stage of the ALKR representation for discrete domains and GABIL encoding," in *Proc. 13th Annu. Conf. Genetic and Evolutionary Computation (GECCO '11)*. New York: ACM Press, 2011, pp. 1291–1298. doi: 10.1145/2001576.2001750.
[32] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*. Stuttgart, Germany: Frommann-Holzboog, 1973.
[33] H. Schwefel, "Evolutionsstrategie und numerische optimierung," Dr.-Ing. Thesis, Dept. of Process Eng., Technical Univ. of Berlin, Germany, 1975.
[34] W. M. Spears, "Adapting crossover in evolutionary algorithms," in *Proc. 4th Annu. Conf. Evolutionary Programming*, 1995, pp. 367–384. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.9322
[35] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann, 1989, pp. 61–69. doi: 10.5555/93126.93146. [Online]. Available: http://portal.acm.org/citation.cfm?id=93146
[36] A. G. Carvalho and A. F. Araujo, "Improving NSGA-II with an adaptive mutation operator," in *Proc. 11th Annu. Conf. Companion Genetic and Evolutionary Computation Conf.: Late Breaking Papers*, 2009, pp. 2697–2700. doi: 10.1145/1570256.1570387. [Online]. Available: http://portal.acm.org/citation.cfm?id=1570387
[37] L. DaCosta, Á. Fialho, M. Schoenauer, and M. Sebag, "Adaptive operator selection with dynamic multi-armed bandits," in *Proc. 10th Annu. Conf. Genetic and Evolutionary Computation*, 2008, pp. 913–920. doi: 10.1145/1389095.1389272. [Online]. Available: http://portal.acm.org/citation.cfm?id=1389272
[38] S. Müller, N. N. Schraudolph, and P. D. Koumoutsakos, "Step size adaptation in evolution strategies using reinforcement learning," in *Proc. 2002 IEEE Congress Evolutionary Computation*, 2002, pp. 151–156. doi: 10.1109/CEC.2002.1006225.
[39] Y. Sakurai, K. Takada, T. Kawabe, and S. Tsuruta, "A method to control parameters of evolutionary algorithms by using reinforcement learning," in *Proc. 6th Int. Conf. Signal-Image Technology and Internet Based Systems*, Dec. 2010, pp. 74–79. doi: 10.1109/SITIS.2010.22. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5714532
[40] S. W. Wilson, "ZCS: a zeroth level classifier system," *Evol. Comput.*, vol. 2, no. 1, pp. 1–18, Mar. 1994. doi: 10.1162/evco.1994.2.1.1.
[41] L. Bull and J. Hurst, *Self-Adaptive Mutation in ZCS Controllers (Lecture Notes in Computer Science), vol. 1803*. Cham: Springer-Verlag, 2000, ch. 33, pp. 342–349. [Online]. Available: http://www.springerlink.com/index/10.1007/3-540-45561-2_33
[42] J. Hurst and L. Bull, *A Self-Adaptive Classifier System (Lecture Notes in Computer Science), vol. 1996*. Cham: Springer-Verlag, 2001, ch. 6, pp. 70–79. [Online]. Available: http://www.springerlink.com/index/10.1007/3-540-44640-0_6
[43] S. W. Wilson, "Classifier fitness based on accuracy," *Evol. Comput.*, vol. 3, no. 2, pp. 149–175, June 1995. doi: 10.1162/evco.1995.3.2.149.
[44] J. Hurst and L. Bull, "A self-adaptive XCS," in *Advances in Learning Classifier Systems (Lecture Notes in Computer Science), vol. 2321*, P. Lanzi, W. Stolzmann, and S. Wilson, Eds. Cham: Springer-Verlag, 2002, pp. 333–360. doi: 10.1007/3-540-48104-4_5.
[45] S. W. Wilson, "Classifiers that approximate functions," *Nat. Comput.*, vol. 1, no. 2/3, pp. 211–234, 2002. doi: 10.1023/A:1016535925043. [Online]. Available: http://portal.acm.org/citation.cfm?id=599708
[46] M. V. Butz, P. O. Stalph, and P. L. Lanzi, "Self-adaptive mutation in XCSF," in *Proc. 10th Annu. Conf. Genetic and Evolutionary Computation*, 2008, pp. 1365–1372. doi: 10.1145/1389095.1389361. [Online]. Available: http://portal.acm.org/citation.cfm?id=1389095.1389361
[47] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*. New York: Elsevier, 2004.
[48] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," *Artif. Intell. Rev.*, vol. 11, no. 1/5, pp. 193–225, 1997. doi: 10.1023/A:1006556606079.
[49] T. K. Ho and M. Basu, "Complexity measures of supervised classification problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 289–300, Mar. 2002. doi: 10.1109/34.990132.
[50] E. Bernado-Mansilla and T. K. Ho, "Domain of competence of XCS classifier system in complexity measurement space," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 82–104, Feb. 2005. doi: 10.1109/TEVC.2004.840153.
[51] J. Luengo, A. Fernández, S. García, and F. Herrera, "Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling," *Soft Comput.*, vol. 15, no. 10, pp. 1909–1936, 2011. doi: 10.1007/s00500-010-0625-8.
[52] J. Bacardit, M. Stout, J. D. Hirst, A. Valencia, R. Smith, and N. Krasnogor, "Automated alphabet reduction for protein datasets," *BMC Bioinform.*, vol. 10, no. 1, p. 6, 2009. doi: 10.1186/1471-2105-10-6. [Online]. Available: http://www.biomedcentral.com/1471-2105/10/6
[53] G. W. Bassel, E. Glaab, J. Marquez, M. J. Holdsworth, and J. Bacardit, "Functional network construction in arabidopsis using rule-based machine learning on large-scale data sets," *Plant Cell Online*, vol. 23, no. 9, pp. 3101–3116, Sept. 2011. doi: 10.1105/tpc.111.088153.
[54] M. Stout, J. Bacardit, J. D. Hirst, and N. Krasnogor, "Prediction of recursive convex hull class assignments for protein residues," *Bioinformatics*, vol. 24, no. 7, pp. 916–923, Apr. 2008. doi: 10.1093/bioinformatics/btn050. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/24/7/916
[55] G. Venturini, "SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts," in *Proc. Eur. Conf. Machine Learning (ECML-93)*, P. B. Brazdil, Ed. New York: Springer-Verlag, 1993, pp. 280–296. doi: 10.1007/3-540-56602-3_142.
[56] J. Bacardit and N. Krasnogor, "A mixed discrete-continuous attribute list representation for large scale classification domains," in *Proc. 11th Annu. Conf. Genetic and Evolutionary Computation (GECCO '09)*. New York: ACM Press, 2009, pp. 1155–1162. doi: 10.1145/1569901.1570057.
[57] S. W. Wilson, "Mining oblique data with XCS," in *Advances in Learning Classifier Systems (Lecture Notes in Computer Science), vol. 1996*, P. Luca Lanzi, W. Stolzmann, and S. Wilson, Eds. New York: Springer-Verlag, 2001, pp. 283–290. doi: 10.1007/3-540-44640-0_11.
[58] K. D. Jong and W. M. Spears, "Learning concept classification rules using genetic algorithms," in *Proc. 12th Int. Joint Conf. Artificial Intelligence*, 1991, vol. 2, pp. 651–656. doi: 10.5555/1631552.1631559. [Online]. Available: http://portal.acm.org/citation.cfm?id=1631559
[59] J. Bacardit, "Pittsburgh Genetics-Based machine learning in the data mining era: Representations, generalization, and run-time," Ph.D. thesis, Ramon Llull Univ., Barcelona, Spain, 2004.
[60] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978. doi: 10.1016/0005-1098(78)90005-5.
[61] M. J. Kearns, *The Computational Complexity of Machine Learning*. Cambridge, MA: MIT Press, 1990.
[62] M. V. Butz and M. Pelikan, "Studying XCS/BOA learning in Boolean functions: structure encoding and random Boolean functions," in *Proc. 8th Annu. Conf. Genetic and Evolutionary Computation (GECCO '06)*, 2006, pp. 1449–1456. doi: 10.1145/1143997.1144236.
[63] A. Hernández-Aguirre, B. P. Buckles, and C. A. C. Coello, "On learning $kDNF_n^s$s Boolean formulas," in *Proc. NASA/DoD Conf. Evolvable Hardware*, 2001, pp. 240–246. doi: 10.1109/EH.2001.937967.
[64] C. Ioannides, G. Barrett, and K. Eder, "XCS cannot learn all Boolean functions," in *Proc. 13th Annu. Conf. Genetic and Evolutionary Computation (GECCO '11)*, 2011, pp. 1283–1290. doi: 10.1145/2001576.2001749.
[65] M. A. Franco, N. Krasnogor, and J. Bacardit, "Post-processing operators for decision lists," in *Proc. 14th Int. Conf. Genetic and Evolutionary Computation Conf. (GECCO '12)*, 2012, pp. 847–854. doi: 10.1145/2330163.2330281.
[66] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006. doi: 10.5555/1248547.1248548. [Online]. Available: http://portal.acm.org/citation.cfm?id=1248548
[67] M. A. Franco, "Principled design of evolutionary learning systems for large scale data mining," Ph.D. thesis, Univ. of Nottingham, 2013.
[68] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945. doi: 10.2307/3001968. [Online]. Available: http://www.jstor.org/stable/3001968
[69] C. G. Northcutt, L. Jiang, and I. L. Chuang, Confident learning: Estimating uncertainty in dataset labels. 2019. [Online]. Available: arXiv:1911.00068
[70] Y. Liu, W. N. Browne, and B. Xue, "Absumption to complement subsumption in learning classifier systems," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO 19)*, 2019, pp. 410–418. doi: 10.1145/3321707.3321719.

Improving Depression Level Estimation by Concurrently Learning Emotion Intensity

©ISTOCKPHOTO.COM/ANTONIOKHR

Syed Arbaaz Qureshi and Sriparna Saha
Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India

Gaël Dias
Department of Computer Science, University of Caen Normandie, Caen, France

Mohammed Hasanuzzaman
Department of Computer Science, Cork Institute of Technology, Cork, Ireland

***Abstract*—**Depression is considered a serious medical condition and a large number of people around the world are suffering from it. Within this context, a lot of studies have been proposed to estimate the degree of depression based on different features and modalities, specific to depression. Supported by medical studies that show how depression is a disorder of impaired emotion regulation, we propose a different approach, which relies on the rationale that the estimation of depression level can benefit from the concurrent learning of emotion intensity. To test this hypothesis, we design different attention-based multi-task architectures that concurrently regress/classify both depression level and emotion intensity using text data. Experiments based on two benchmark datasets, namely, the Distress Analysis Interview Corpus - a Wizard of Oz (DAIC-WOZ), and the CMU Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) show that substantial performance improvements can be achieved when compared to emotion-unaware single-task and multi-task approaches.

Digital Object Identifier 10.1109/MCI.2020.2998234
Date of current version: 15 July 2020

1. Introduction

Depression is a common mental disorder that causes people to experience depressed mood, loss of interest or pleasure, feelings of guilt or low self-worth, disturbed sleep or appetite, low energy, and poor concentration [1]. It is the predominant mental health problem worldwide, followed by anxiety, schizophrenia and bipolar disorder [2]. In 2013, depression was the second leading cause of years lived with a disability worldwide, and in 26 countries, depression was the primary driver of disability [2]. More than 300 million people are now living with depression, an increase of more than 18% between 2005 and 2015.[1]

Depression lasts between 4 and 8 months on average and can actually change one's ability to think, impair attention and memory, as well as debilitate information processing and decision-making skills. It can also lower one's cognitive flexibility and executive functioning. As a consequence, in extreme cases, depression may be characterized by thoughts of death and suicide. Approximately 800,000 people suffering from depression die due to suicide yearly and the annual number of death cases due to depression is on the rise.[2]

Corresponding Author: Gaël Dias (Email: gael.dias@unicaen.fr).

[1]A statistic reported by the World Health Organization, available at https://bit.ly/2rsqQoP.
[2]A study by Hannah Ritchie and Max Roser in 2018, available at https://bit.ly/2mnyVZ6.

There are many possible causes of depression, including faulty mood regulation (for example, inability to deal with failure and rejection), genetic vulnerability, stressful life events (for example, divorce, death of a family member, childhood trauma), and medical problems. It is believed that several of these forces interact to bring on depression [3].

A depression diagnosis is often difficult to make because clinical depression can manifest in many different ways. Observable or behavioral symptoms of clinical depression also may sometimes be minimal despite a person experiencing profound inner turmoil. Diagnosis of depression has traditionally been made based on clinical criteria, including patient current symptoms and history. This process is widely used but relies on subjective interpretation. To standardize both the data obtained and data interpretation, various interview-based instruments and non-interview methods exist for screening and testing for depression in various clinical settings [4]. In particular, interview-based screening tools include the Hamilton Depression Rating Scale (HDRS), the Beck Depression Inventory (BDI), the Center for Epidemiologic Studies Depression Scale (CES-D), the Hospital Anxiety and Depression Scale (HADS), and the Montgomery and Asberg Depression Rating Scale (MADRS).[3]

The Patient Health Questionnaire (PHQ) [5] has been established as a valid diagnostic and severity measure for depressive disorders [6]. In particular, PHQ-8 contains eight questions, whose answers range from 0 (not at all) to 3 (nearly every day), to provide an overall mark between 0 and 24, that estimates the level of depression. Different versions of the PHQ exist, such as PHQ-8, PHQ-9 and PHQ-15, containing 8, 9 and 15 questions respectively. The PHQ-9 is the most widely used questionnaire [6], but researchers generally use PHQ-8, which consists of all the PHQ-9 questions except for the last one (a question on suicidal thoughts). The absence of the ninth question has little effect on scoring between the PHQ-8 and PHQ-9. Studies found that scores between the two tests are highly correlated [7].

However, filling these forms is a tedious task that can be perceived as insuperable by many patients, thus leading to a great deal of medically unfollowed patients. Moreover, due to the increasing number of patients suffering from mental health diseases, the average time for a medical consultation has drastically decreased over the last decade, leading to both patients' and therapists' frustration and limiting the number of interview-based screening acts.

Effective treatments for depression are available, however, only fewer than half of those affected in the world undergo with such treatments. In some countries, this number can go down to less than 10%. Possible reasons for this may be lack of resources, lack of trained health-care providers, social stigma associated with mental disorders and also an inaccurate assessment. Simultaneously, people who are depressed may not be correctly diagnosed, and others who do not have the disorder are too often misdiagnosed and prescribed antidepressants. The above facts prove that there is a steadily increasing global burden of depression and mental illness. Thus development of more advanced, personalized and automatic technologies for the detection and estimation of depression is highly essential.

In order to help therapists in their diagnosis, a great deal of studies have been proposed for the automatic estimation of depression level based on different features over various modalities, such as text, vision and acoustics [8]–[10]. All these methodologies focus on the improvement of single-task learning models, trying to increase the performance by better characterizing depression itself. However, some studies in mental health have shown that depression is a disorder of impaired emotion regulation [11], [12]. In particular, patients with major depression are often unable to control their emotional responses to negative situations, and overuse emotional expressions of sadness, disgust or fear. As a consequence, we hypothesize that the estimation of depression level can benefit from the concurrent learning of emotion intensity, which can be evaluated on a [0,3] scale for the six emotions of Ekman [13] - happiness, sadness, anger, fear, disgust and surprise. So, we propose to use the text data provided in the interviews of the different datasets (depression and emotion) to concurrently estimate depression level and emotion intensity, expecting that both tasks have common backgrounds and can boost performance over single-task processing. For illustration, we show below sentences that are indicative of depression.

Interviewer:" *How easy was it for you to get used to living in Los Angeles?"*
Participant:" *It was not easy for me. It took about three years."*
Interviewer:" *Can you tell me about that?"*
Participant:" *Umm… just the move. I moved away from my family so I was uncomfortable. I didn't know anyone here and even though I did make friends I just felt out of place."*

To test our hypothesis, we particularly explore three different multi-task architectures that concurrently regress/classify both depression level and emotion intensity using textual modality exclusively. Thus, (1) the fully-shared, (2) the shared-private and (3) the adversarial shared-private models are designed, following the ideas of [14]. However, we include an attention layer in the last two models, to let the network decide by itself the weights of the private and shared representations in the decision process. We extend the multi-task architectures to three tasks, which include depression level regression, depression level classification[4] and emotion intensity regression, thus extending the ideas of [15], who have shown that depression level classification and depression level regression can be complementary in the decision process.

An exhaustive series of experiments using these models are carried out using two benchmark datasets: the Distress Analysis Interview Corpus - a Wizard of Oz (DAIC-WOZ) [16], and the

[3]Recommendation of the French Haute Autorité de la Santé For more information, go to https://bit.ly/2EaOs92.

[4]For that purpose, discretization follows medical scales.

Carnegie Mellon University—Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) [17]. Although both datasets contain multimodal (text, vision, acoustics) information, we exclusively focus on the text modality, as [9] showed that lexical models perform reasonably well to monitor depression level. Overall results for both depression level classification and regression show that notable performance gains can be obtained by emotion-aware models, when compared to emotion-unaware single-task (ST) and multi-task (MT) baseline approaches.

Since depression is a disorder of impaired emotion regulation, we hypothesize that its automatic diagnosis can benefit from the concurrent learning of markers of depression and emotions.

With such studies, we expect that in a near future systems can be built that automatically detect depression, thus playing a great role in supporting the therapist's diagnosis. Such applications may also help in early detection of clinical depression by suggesting the sufferer to consult a psychiatrist. We anticipate that our work may reduce cases of late treatment for clinical depression.

2. Related Works

Due to the impulse for the development of automatic technologies that can aid the detection of mental health disorders, a great deal of research studies in Computer Science have been emerging over the past few years [18]. Within this particular context, the automatic detection of depression level has received major focus.

Initial initiatives targeted the understanding of relevant non-verbal descriptors that could be used in machine learning frameworks such as gaze, smile, self-touches and heart-rate descriptors [19]. Other non-verbal descriptors include acoustics. Within this context, [8] focused their research on finding how common paralinguistic speech characteristics are affected by depression, namely prosodic, source, formant and spectral features. With respect to verbal descriptors, [20] hypothesized that researchers should look beyond the acoustic properties of speech and build features that capture syntactic structures and semantic contents. Following these ideas, [9] showed that classification performance suggests that lexical models are reasonably robust to play an important role in the diagnosis and monitoring of depression. But, the analysis also suggests that users may be able to fool algorithms by avoiding direct discussion of depression. Some other interesting work directions using text features include the study of social media [21], eventually using specific corpora tuned for such tasks [22].

More recently, new solutions proposed to combine verbal and non-verbal descriptors (or modalities) within a single learning model [10], [23]. Although the idea is seducing as it can be viewed as a way of avoiding fooling behaviors, the first results were mitigated [24]. But, recent studies [25] evidence successful results. It is also interesting to notice that non-Deep Learning approaches have been proposed but with less successful results [26]. This may suggest that Deep Learning techniques are able to capture high-level features and long-term dependencies at levels not seen before.

All previous related works focus on finding better descriptions of depression characteristics based on verbal and/or non-verbal indicators. In this paper, we aim to investigate the effect of simultaneously learning related tasks such as depression level and emotion intensity estimation. As stated in the study of [27], simultaneous learning of every task combination is not beneficial, but, tasks having cognitive similarities often get benefited from concurrent learning. In recent years, multi-task learning frameworks have become powerful in solving different NLP tasks [27], [28]. The possible reasons for this success (i.e. learning the decision boundaries of related tasks) are: (1) knowledge transfer across tasks in the form of generating more robust representations and (2) the use of more training data. In [29], it has also been discussed that multi-task learning can act as a regularization process which avoids overfitting by maintaining competitive performance across different tasks. In particular, a multi-task framework has recently been proposed by [15] who explore in concurrently learning depression level classification and regression. Inspired by the success of such models, we propose to compare three multi-task learning models (fully-shared, shared-private and adversarial shared-private) that combine three concurrent tasks: depression level classification, depression level regression and emotion intensity regression. Indeed, as depression can be viewed as the impaired regulation of emotion intensity, it is likely that better models can be built based on the concurrent learning of depression level and emotion intensity estimations.

3. Methodology

In order to estimate the level of depression and the intensity of emotions concurrently, we propose three different multi-task architectures that take as input the transcript files from the DAIC-WOZ [16] and the CMU-MOSEI datasets [17]. These datasets are described in detail in section 4. In the following subsections, we describe the tasks to be handled, the preprocessing steps and the multi-task architectures.

3.1. Learning Tasks

In this section, we define the three tasks used in our experiments: depression level regression, depression level classification and emotion intensity regression.

Depression Level Regression (DLR). Given the interview transcript associated with a patient, we predict its PHQ-8 score. This can be modeled as a simple regression task, where a score in the range of [0-24] must be predicted.

Depression Level Classification (DLC). In this task, we discretize the PHQ-8 score, which ranges from 0 to 24, into five classes of equal length: [0-4], [5-9], [10-14], [15-19], and [20-24].[5] We

[5] More details about this process are given in section 4.

now treat this problem as multi-class classification, where a class is predicted given the interview transcripts. Note that this task is highly correlated to DLR.

Emotion Intensity Regression (EIR). In the CMU-MOSEI dataset, the emotion intensity is labeled at sentence-level, in contrast to transcript-level for depression estimation. Each sentence has a 6-D vector label, that contains scores in the range [0-3], for the six Ekman's emotions.[6] Note that in this dataset, many of the transcripts do not have labels for all its sentences. For such transcripts, we append a 0 on top of the labels of all the labeled sentences (representing that some emotion is exhibited in the utterance), and we manually label all the unlabeled sentences with [1,0,0,0,0,0,0], where 1 denotes that no emotion is exhibited in the utterance. As such, each sentence is labeled by a 7-D vector. If there are T sentences in a transcript, its label matrix is of 7×T dimension. So, given the monologue transcript, we must predict this 7×T matrix.

3.2. Sentence Preprocessing and Encoding

The initial step of our methodology aims to preprocess and encode each sentence of the respective transcripts.

Sentence Preprocessing. In DAIC-WOZ, many participants speak colloquially. So, we formalize all utterances by replacing contractions with corresponding full words. The sentences may also contain filler words such as "umm" or "hmm". We let them remain unchanged, as they may be important features to estimate depression. No preprocessing was required for the sentences in CMU-MOSEI, as they are already clear and formal.

Sentence Encoding Network. Inspired by the success of the universal sentence encoder [30] in finding semantic similarity between two sentences, we use its transformer variant to encode the sentences of the transcripts. It encodes a sentence using the encoding sub-graph of the transformer architecture. This sub-graph uses attention to compute context-aware representations of words in a sentence that take into account both the ordering and the identity of all the other words. The context-aware word representations are converted to a fixed-length sentence encoding vector by computing the element-wise sum of the representations at each word position. The encoder takes as input a lower-cased Penn Treebank tokenized string and outputs a 512 dimensional vector as the sentence embedding. As there are different numbers of sentences in different transcripts, we left-pad all the transcripts with 512-D zero-vectors, to a common length.

3.3. Learning Architectures

We describe three different multi-task models with respect to three tasks DLR, DLC and EIR. The three multi-task models are the Fully-Shared (FS MT.), the Shared-Private (SP MT.), and the Adversarial Shared-Private (ASP MT.). Each of the multi-task architectures has been designed for a combination of DLR, DLC and EIR, which are DLR-DLC, DLC-EIR, DLR-EIR and DLR-DLC-EIR. Note that the inputs of each model are the encoded sentences. We also implement a series of single-task models (ST.) for each of DLR, DLC and EIR.

Single-Task (ST.). The single-task model consists of a one-to-one Long Short Term Memory (LSTM) network [31], that encodes the transcript. The LSTM unit was chosen as the recurrent unit because it is very efficient in modeling long dependencies in time series data. In particular, LSTM networks are a special kind of recurrent neural network capable of learning long-term dependencies. As stated by [32], LSTM networks are the state-of-the-art structures for NLP tasks, as they have the ability to retain data through many time steps, a feature which no other deep neural networks have.

The output from the LSTM network (which may be the individual outputs of all sentences[7] or the sum of the outputs from all the LSTM units[8]) is fed to a set of fully connected and dropout layers. The output representation is then passed on to one or more (depending upon the task) linear regression units, in case the task is regression, or to a softmax classifier, in case the task is classification.

Fully-Shared Multi-Task (FS MT.). The fully-shared multi-task model consists of one LSTM network, that acts as the shared space for all the tasks. The outputs (or their summation, for DLC and DLR) from this LSTM network are fed to a task-specific network of fully connected and dropout layers. The representation obtained from this network is passed on to the output layer, which is a single linear regression unit for DLR, a 5-class softmax layer for DLC, or a layer of 7 regression units, in case the task is EIR.

This architecture forces the LSTM network to learn both the shared and task-specific features, as shown on the right side of Figure 1. Indeed, it does not have any facility to separate both shared and private spaces. The main drawback of this architecture is that it is bound to fail for increasingly less-correlated pairs of tasks, as the LSTM network is likely to fail to capture the task-specific features of all the tasks, if they are not enough correlated. However, in case tasks are heavily correlated, this network is expected to perform well.

Shared-Private Multi-Task (SP MT.). The shared-private multi-task model consists of three LSTM networks - two task-specific and one shared. All of the networks have the same number of units. In particular, the input of a task is fed to the task-specific and the shared LSTM network. The outputs from the task-specific and the shared LSTM layers are fused using an attention fusion mechanism [33], to obtain a fusion vector. The attention fusion network is explained further in this section. This fusion vector is then fed to a network of fully-connected

[6] Further details of the dataset are given in section 4.

[7] This is the case for EIR as labels are given at sentence level.

[8] This is the case for DLR and DLC as labels are given at text level.

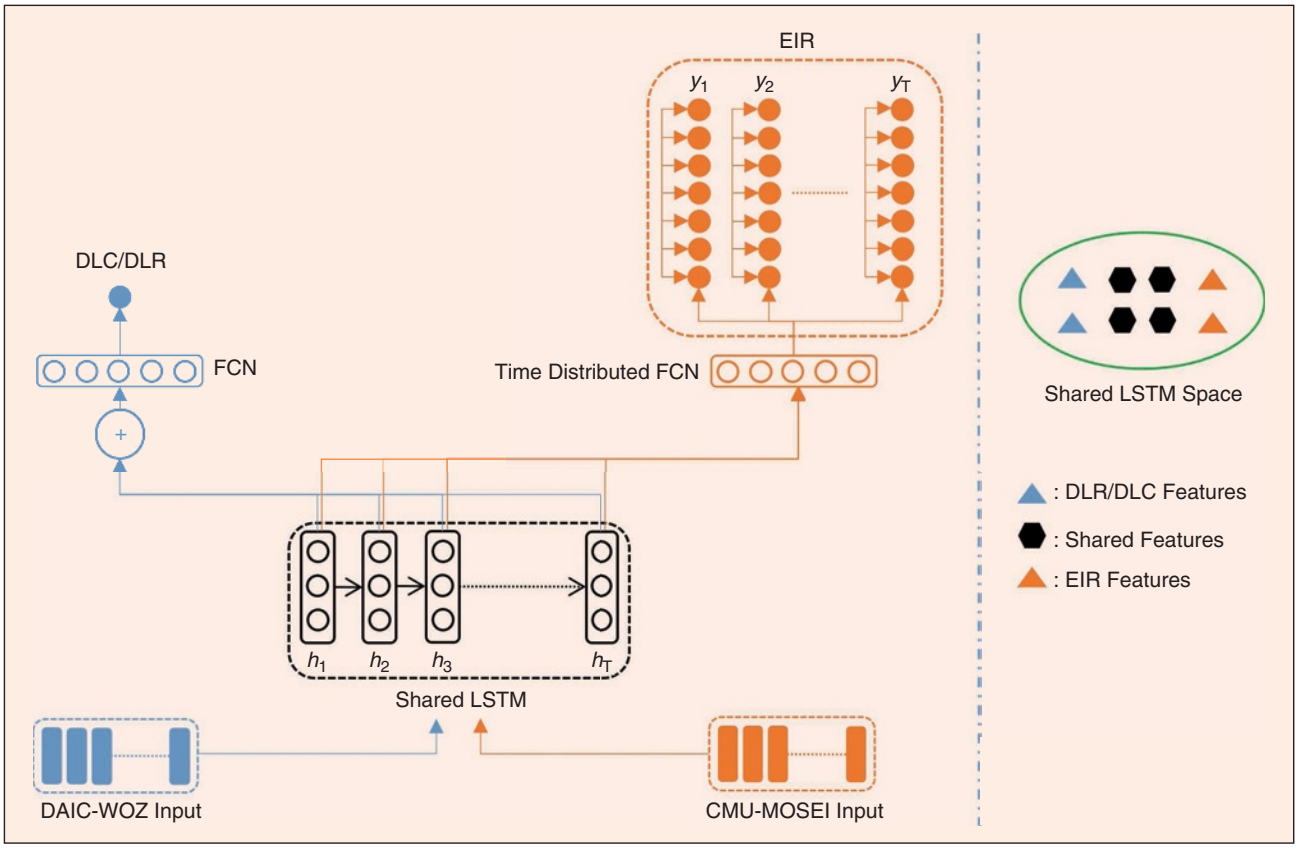


FIGURE 1 Fully-shared multi-task model (FS MT.). FCN stands for fully connected network, EIR for emotion intensity regression and DLC (resp. DLR) for depression level classification (resp. depression level regression).

and dropout layers, whose output is fed to the task-specific output layer.

This architecture, presented in Figure 2, improves over the fully-shared multi-task architecture by providing an infrastructure that has separate spaces for task-specific and shared features. But this too may have drawbacks. The shared feature space could contain some unnecessary task-specific features, while some shared features could also be mixed with the private space, thus suffering from feature redundancy as shown on the right side of Figure 2.

Adversarial Shared-Private Multi-Task (ASP MT.). Inspired by the results obtained by [14], [28], we design a similar architecture with two modifications. The adversarial shared-private multi-task architecture consists of three LSTM networks, that is, two task-specific and one shared, all of which have the same number of units. The input of a task is fed to the task-specific and the shared LSTM networks. The outputs from the task-specific and the shared LSTM layers are then fused using the attention fusion mechanism, oppositely to [14], [28], who use concatenation.

The output from the shared LSTM layer is also fed to a network N_D of fully-connected dropout and softmax layers. This network outputs the task label (for example, if there are two tasks T_1 and T_2, the task label for T_1 is [1, 0], the task label for T_2 is [0, 1]). The shared LSTM layers and N_D act as an adversarial network, the shared LSTM layer acting as the generator and N_D acting as the discriminator.

Finally, a L_{diff} loss function acts as an orthogonality constraint between private and shared layers and differs from the one used in [14], [28]. It is defined in Equation 1, where $\|\cdot\|_1$ is the L_1 norm, H and S are two matrices, whose rows are each unit output of the task-specific LSTM network and the shared LSTM network, respectively, and m and n are the first and second dimensions of $H^\top S$ respectively. This definition of L_{diff} was empirically settled after testing other definitions. The architecture is shown in Figure 3.

$$L_{\text{diff}} = \frac{\|H^\top S\|_1}{m \times n}. \tag{1}$$

This architecture ensures that the task-specific and shared spaces are as separate as possible, as shown on the right side of Figure 3. The introduction of the adversarial network (shared LSTM - N_D pair) removes the possibility of task-specific features creeping into the shared-LSTM space. The orthogonality constraint ensures that the task-specific and shared spaces are as orthogonal as possible, which means the task-specific LSTM space should not contain any of the shared features as its space should be orthogonal to the shared LSTM space. Note that when the tasks are highly correlated, this architecture tends to perform poorly, as it would be very tough for the shared LSTM (generator) to create such a representation that can fool N_D (discriminator).

Attention Fusion Network (AFN.). In attention fusion, we first concatenate the outputs from the task-specific and the shared layers, pass them to a network of fully-connected and dropout layers, the output of which is passed to a softmax layer. This softmax layer outputs two values: α_{task} and α_{shared}, which weight the task-specific LSTM network and the shared LSTM network, respectively, in calculating the final output. So, α_{task} is multiplied with the output of the task-specific LSTM network, α_{shared} is multiplied with the output of the shared-LSTM network, and the corresponding products are summed. This summation represents the fusion vector. The attention fusion network is shown in 4.

We particularly included the attention mechanism to better understand the behavior of each of the task-specific and shared features in the decision process. For estimating depression, if task-specific embeddings are more important than the shared embeddings, then α_{task} would have a value greater than 0.5, and α_{shared} would be less than 0.5. This would allow the network to learn the importance of the shared and task-specific embeddings by itself, in order to estimate depression/emotion levels. Moreover, networks with an attention mechanism usually perform better than their counterpart without attention [34].

Three-task Architectures. The definition of multi-task architectures that contain more than two tasks (here DLC+DLR+EIR) may not be straightforward in all cases. In the case of the fully-shared model, the definition is simple. Each task is solved using the single shared LSTM layer. With respect to the shared-private and the adversarial shared-private models, different strategies are possible. In our case, we take advantage of previous findings, namely that highly related tasks should perform better when fully-shared architectures are used. So, as DLC and DLR are highly correlated, we choose to combine them using a fully-shared architecture and combine the pair of tasks with EIR using the other two possible architectures (SP MT. and ASP MT.). The architecture for ASP MT. on three tasks is shown in Figure 5, and the SP MT. architecture can easily be inferred from the same illustration, by removing the discriminator and the orthogonality constraints.

4. Datasets and Learning Setups

In this section, we present two benchmark datasets, namely DAIC-WOZ for depression estimation and CMU-MOSEI for emotion intensity detection, as well as we define the learning setups of our different architectures.

4.1. DAIC-WOZ Dataset

The DAIC-WOZ depression dataset[9], that is used in the current study, is a subset of the DAIC corpus [16] containing clinical

[9]http://dcapswoz.ict.usc.edu/.

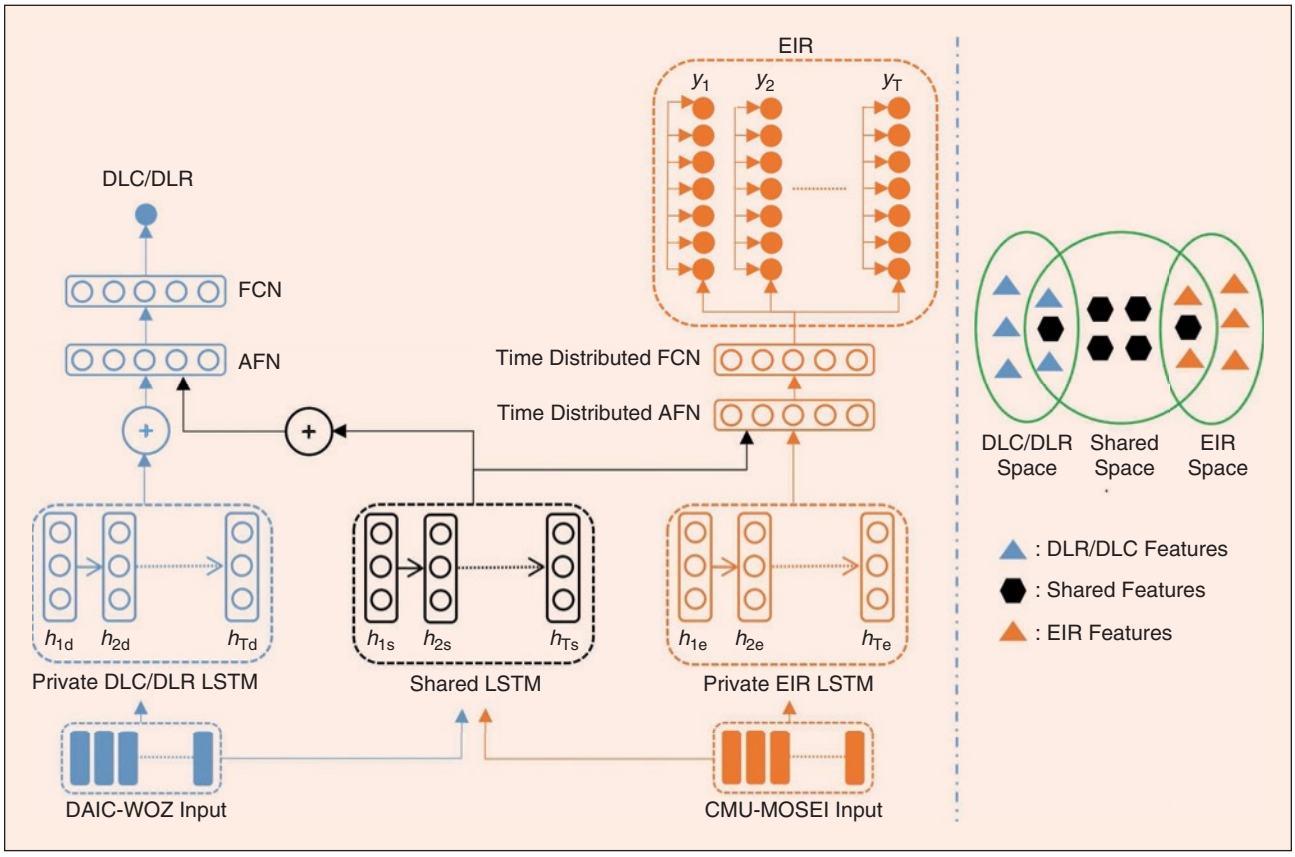


FIGURE 2 Shared-private multi-task model (SP MT.). AFN refers to attention fusion network, FCN to fully connected network, EIR to emotion intensity regression and DLC (resp. DLR) to depression level classification (resp. depression level regression).

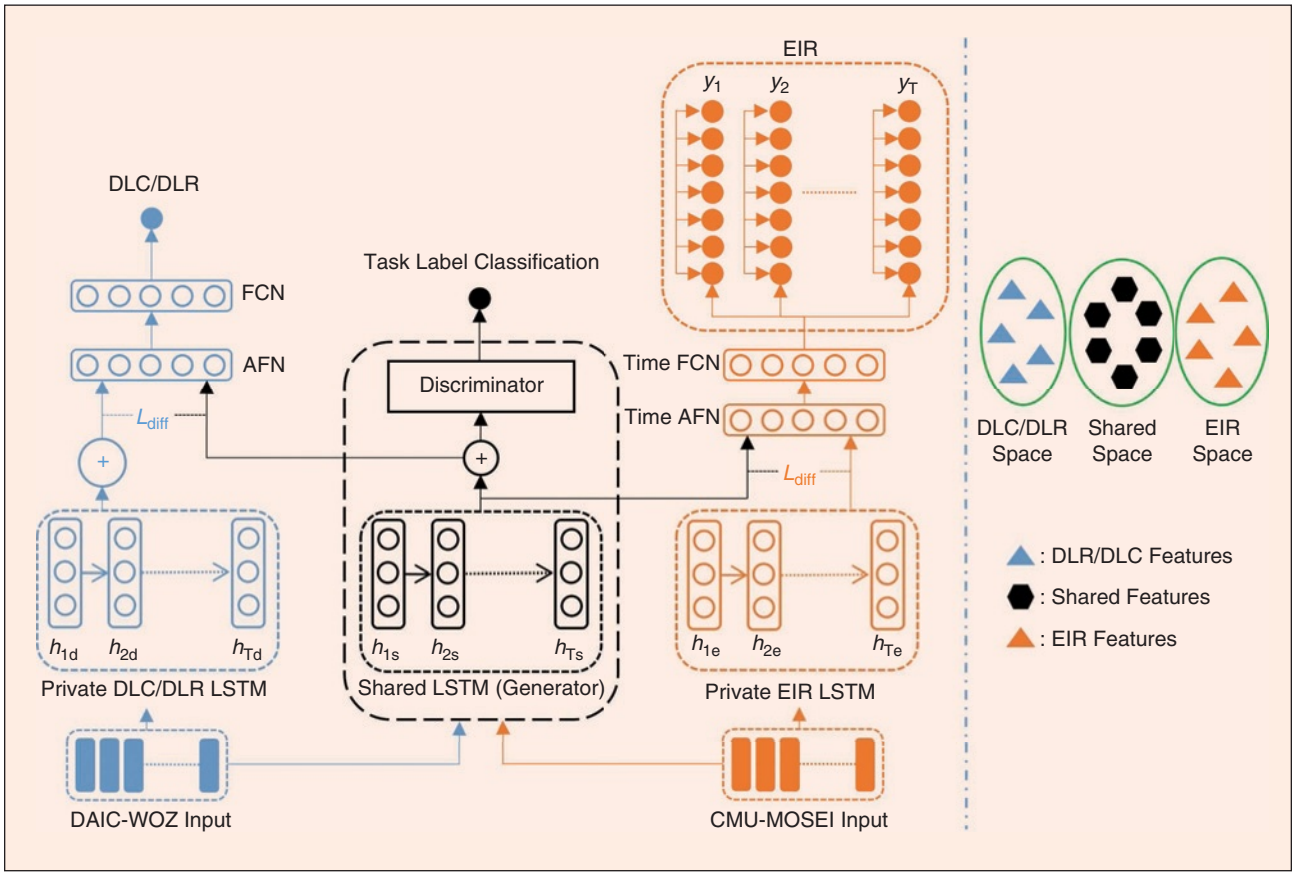


FIGURE 3 Adversarial shared-private multi-task model (ASP MT.). AFN refers to attention fusion network, FCN to fully connected network, EIR to emotion intensity regression and DLC (resp. DLR) to depression level classification (resp. depression level regression).

interviews of situations of psychological distress, which was generated by scientists from University of Southern California. These interviews were taken by a computer agent controlled by a human (wizard-of-oz virtual interviewer) who interacted with common people asking about their mental states and identified different verbal and non-verbal indicators for the same. The audio and video recordings and extensive questionnaire responses from the interviews are a part of the dataset. The data is annotated with a variety of verbal and non-verbal features.

189 sessions of dialogues are in the dataset, out of which, 45 are affiliated with the official test split, whose labels are not given. Out of the remaining 144, 6 of them were rejected as they had partial recording and interruptions, prompting to a final number of 138 samples. The accompanying features are (1) a raw audio document of the dialogue session combined with its transcript, (2) files gathering coordinates of 68 facial indicators, the histogram of oriented gradients (HoG) characteristics of the face, head pose and gaze directionality characteristics (extracted with OpenFace [35]), (3) a document containing the continuous facial activity units extracted with CERT [36], and (4) files with the COVAREP and formant voice characteristics computed with the COVAREP software [37].

TABLE 1 Distribution of the DAIC-WOZ dataset by depression class.

CLASS	TRAIN + DEV.	TEST
NONE-MINIMAL - [0-4] PHQ-8 SCORE	47	16
MILD - [5–9] PHQ-8 SCORE	28	5
MODERATE - [10-14] PHQ-8 SCORE	19	5
MODERATELY SEVERE - [15-20] PHQ-8 SCORE	7	6
SEVERE - [20-24] PHQ-8 SCORE	4	1

As we are only focusing on the text modality, we only retain the transcript files that contain the sentences spoken by the virtual interviewer and the participant. The class-wise distribution of our training, development and test splits is summarized in Table 1. Note that medical studies [38] state that a PHQ-9 score in the interval [0-4] stands for None-minimal depression, in [5-9] for Mild, in [10-14] for Moderate, in [15-19] for Moderately severe, and in [20-27] for Severe depression. In the particular case of the PHQ-8 score, one question about suicidal condition is missing. As a consequence, the exact same discretization can be used, where severe depression is in the range of [20-24].

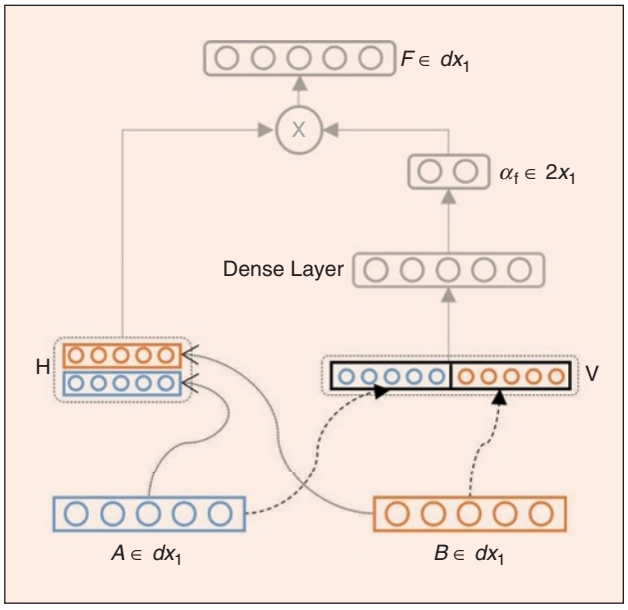


FIGURE 4 Attention fusion network (AFN).

The DAIC-WOZ dataset, however, has some limitations. The number of samples in the entire dataset is small and not evenly distributed, with just one sample of the "severely depressed" category in the test set. It is clear that further efforts are needed to increase such a dataset, although this remains out of the scope of this paper. In all cases, all obtained results of our study will have to be put in perspective relatively to this small amount of learning instances.

4.2. CMU-MOSEI Dataset

The CMU Multimodal Opinion Sentiment and Emotion Intensity (CMU-MOSEI) dataset[10] comprises 3,228 videos from 1,000 different speakers over 250 topics [17]. The videos were gathered from an online video platform, where users emit their opinions in the form of monologues. Each video contains a unique person, who discusses in front of the camera about a given topic. Each video can be transformed into three information sources: language (spoken utterances), visual (gesture

[10]https://github.com/A2Zadeh/CMU-MultimodalSDK.

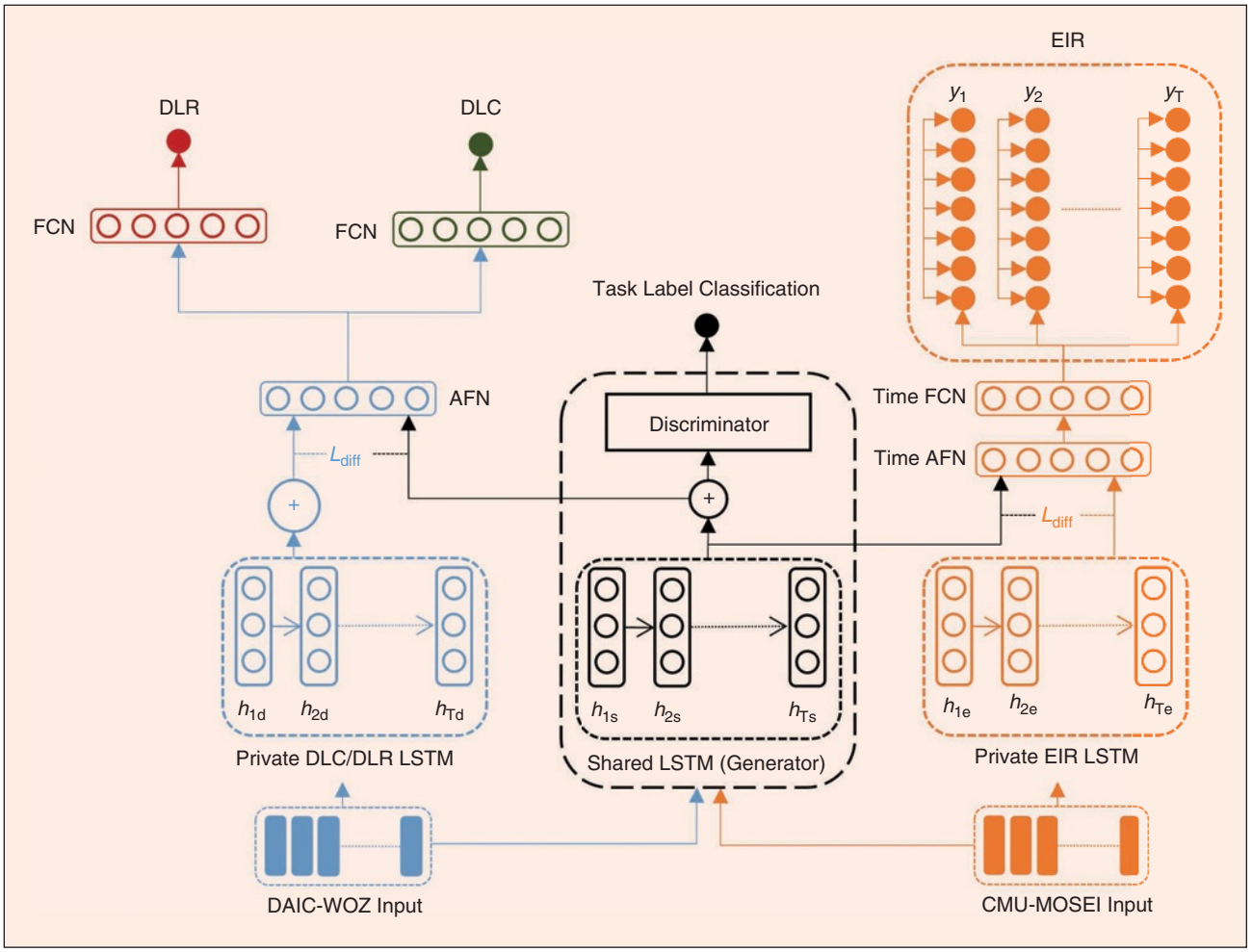


FIGURE 5 Adversarial shared-private multi-task model (ASP MT.) for three tasks. AFN refers to attention fusion network, FCN to fully connected network, EIR to emotion intensity regression and DLC (resp. DLR) to depression level classification (resp. depression level regression).

analysis), and acoustics (intonations and prosody). During data acquisition, videos were analyzed by automatic face detection to verify whether a unique speaker is present. Moreover, only the videos where the speaker's attention is exclusively towards the camera were kept. The number of videos acquired from each channel was restricted to 10 to avoid bias and all videos must have correct transcriptions provided by the speaker. The quality inspection has been made by 14 expert judges, and 3,228 videos were selected from the 5,000 initially gathered.

The 3,228 videos were then segmented into 23,453 annotated pieces, where each segment contains a manual transcription aligned with audio to phoneme level. The annotation of CMU-MOSEI closely follows the annotation rules of the CMU-MOSI [39] dataset. In particular, sentences were annotated for Ekman's six emotions, that is happiness, sadness, anger, fear, disgust and surprise, on a [0,3] Likert scale for the presence of emotion. As such, 0 stands for no evidence of *x*, 1 for weakly *x*, 2 for *x*, and 3 for highly *x*. With respect to sentiment evaluation, a [-3,3] Likert scale was used such that: -3 is highly negative, -2 is negative, -1 is weakly negative, 0 is neutral, 1 is weakly positive, 2 is positive, and 3 is highly positive. Note that in this paper, we do not use the annotation for sentiment evidence. As stated in [17], the annotation was carried out by 3 crowd-sourced judges from Amazon Mechanical Turk platform, where judges were provided with a 5 minutes training video on how to use the annotation system in order to avoid extreme annotation, and all judges were master workers with an approval rate higher than 98%.

Note that as in CMU-MOSEI each of the 3,228 video transcripts contains an average of 7.3 utterances, and in DAIC-WOZ, the 138 interview transcripts contain an average of 90 utterances, we randomly selected 517 transcripts from CMU-MOSEI to reduce imbalance between datasets.

4.3. Learning Setups

With respect to multi-task learning, the task-specific LSTM layers are trained alternatively using the entire training split. As an example, consider the training of the shared-private multi-task network for depression level regression and emotion intensity regression: SP MT. DLR+EIR. The DLR-specific LSTM layer, the shared-LSTM layer, and the corresponding attention fusion network and fully-connected network are trained for N_{DLR} epochs without updating the weights of the EIR-specific layers. For the next N_{EIR} epochs, the EIR-specific LSTM layer, the shared-LSTM layer, and the corresponding attention fusion network and fully-connected network are trained without updating the weights of the DLR-specific layers. Here, N_{DLR} and N_{EIR} are treated as hyperparameters. We go on training the network in this alternating fashion till a maximum number of iterations N_{total} (the total number of times the shared-LSTM layer is trained) is reached. The model that shows best performance on the development split over all iterations is chosen for testing. The pseudo-code for our training procedure is shown in algorithm 1.

Algorithm 1 FS/SP/ASP MT. T_1+T_2 training.

```
n_total ← 1
while n_total < N_total do
    for n_T1 ← 1 to N_T1 do
        Update T1-specific and Shared weights
        n_total ← n_total + 1
    for n_T2 ← 1 to N_T2 do
        Update T2-specific and Shared weights
        n_total ← n_total + 1
```

The architectures have been implemented with Keras[11] and hyperparameters have been optimized through grid search. Note that all learning models are trained on the basis of stratified 5-cross validation, thus keeping the data distribution between training, development and test datasets. In particular, the best of the 5 models over the development set is applied to classify/regress the examples in the test set.

5. Results

In order to test our hypothesis, we perform a series of experiments for three different tasks: (1) Depression Level Regression (DLR), which aims to assign a value between 0 to 24 (that is, the PHQ-8 score) to a given patient interview transcript, (2) Depression Level Classification (DLC), whose objective is to identify the correct discrete class of depression level (None-minimal, Mild, Moderate, Moderately severe, Severe), and (3) Emotion Intensity Regression (EIR), which regresses a [0-3] value for each of the six Ekman emotions (happiness, sadness, anger, fear, disgust and surprise) for a given user transcript.

Five different models serve as baselines. That is, each task is first modeled as a single-task problem, and two unaware-emotion multi-task (fully-shared and shared-private) architectures are implemented that combine both DLR and DLC.[12] Three different combinations of emotion-aware multi-task frameworks are tested, for each one of the three theoretical models (fully-shared, shared-private and adversarial shared-private): (1) DLC combined with EIR, (2) DLR combined with EIR, and (3) DLC combined with both DLR and EIR.[13]

To evaluate regression/classification results, we use well-known evaluation metrics that are standard for depression level estimation [40]: (1) Accuracy, F1 score and Matthews Correlation Coefficient (MCC) for classification; (2) Root Mean Square Error (RMSE), Mean Average Error (MAE), Coefficient of Determination (R^2) and Symmetric Mean Absolute Percentage Error (SMAPE) for regression. In particular, we include two other metrics (Over and Under), that complement Accuracy and evaluate how much a learning model over-evaluates (Over) or under-evaluates (Under) the correct result. Such metrics are important to understand the behavior of learning models. But, as far as we know, they are not presented in related works. For classification, Accuracy, Over and Under sum to 100% and are defined in equations 2 and 3. For

[11] https://keras.io.
[12] These baselines correspond to the 5 first rows of Table 2.
[13] These models correspond to the 9 last rows of Table 2.

TABLE 2 Overall classification results including single-task (ST.) models as well as fully-shared multi-task (FS MT.), shared-private multi-task (SP MT.) and adversarial shared-private multi-task (ASP MT.) models. Acc., Ov. and Un. metrics are given in % and respectively correspond to accuracy, over and under. F1 stands for F1 score, MCC for Matthews correlation coefficient. RMSE refers to root mean squared error, MAE to mean average error, R^2 to coefficient of determination, SM. to the symmetric mean absolute percentage error and $\overline{Ov.}$ and $\overline{Un.}$ to over-evaluation and under-evaluation metrics for regression. $\overline{MSE}$ stands for average mean squared error.

MODELS	EVALUATION METRICS													
	DLC							DLR						EIR
	ACC.	F1	MCC	RMSE	MAE	OV.	UN.	RMSE	MAE	R^2	SM.	$\overline{Ov.}$	$\overline{Un.}$	$\overline{MSE}$
BASELINES WITHOUT EMOTION INTENSITY REGRESSION														
ST. DLC	60.61	0.54	0.38	1.31	0.75	3.03	36.36	–	–	–	–	–	–	–
ST. DLR	–	–	–	–	–	–	–	4.90	3.99	0.46	0.97	3.21	5.18	–
ST. EIR	–	–	–	–	–	–	–	–	–	–	–	–	–	7.15
FS MT. DLC+DLR	**66.66**	**0.62**	**0.49**	1.23	0.66	3.03	30.31	4.96	3.89	0.44	0.98	2.81	5.19	–
SP MT. DLC+DLR	60.61	0.51	0.39	1.26	0.72	0.00	39.39	4.70	3.81	0.50	0.99	3.39	4.32	–
MULTI-TASK RESULTS WITH EMOTION INTENSITY REGRESSION														
FS MT. DLC+EIR	60.61	0.51	0.42	1.58	0.90	0.00	39.39	–	–	–	–	–	–	6.98
SP MT. DLC+EIR	57.57	0.50	0.35	1.27	0.76	6.07	36.36	–	–	–	–	–	–	7.05
ASP MT. DLC+EIR	60.61	0.54	0.38	1.26	0.73	9.09	30.30	–	–	–	–	–	–	7.19
FS MT. DLR+EIR	–	–	–	–	–	–	–	4.60	3.74	0.52	0.99	3.16	4.63	6.88
SP MT. DLR+EIR	–	–	–	–	–	–	–	**4.51**	3.89	**0.54**	**0.94**	3.91	3.85	**6.82**
ASP MT. DLR+EIR	–	–	–	–	–	–	–	4.72	3.96	0.50	**0.94**	3.80	4.15	7.08
FS MT. DLC+DLR+EIR	57.57	0.46	0.38	1.36	0.82	3.04	39.39	4.83	4.03	0.47	0.97	3.13	5.11	6.96
SP MT. DLC+DLR+EIR	**63.64**	0.58	**0.48**	**0.94**	**0.51**	24.24	12.12	4.56	3.79	0.53	0.97	3.20	4.59	7.02
ASP MT. DLC+DLR+EIR	60.61	**0.60**	0.42	1.14	0.64	12.12	27.27	4.61	**3.69**	0.52	0.95	2.87	4.81	7.11

regression, $\overline{Over}$ and $\overline{Under}$ metrics quantify average continuous over-evaluation and under-evaluation and are defined in equations 4 and 5.

$$Over = \frac{\sum_{y_i < \widehat{y_i}} 1}{\sum_{y_i} 1} \tag{2}$$

$$Under = \frac{\sum_{y_i > \widehat{y_i}} 1}{\sum_{y_i} 1} \tag{3}$$

$$\overline{Over} = \frac{\sum_{y_i < \widehat{y_i}} \widehat{y_i} - y_i}{\sum_{y_i < \widehat{y_i}} 1} \tag{4}$$

$$\overline{Under} = \frac{\sum_{y_i > \widehat{y_i}} y_i - \widehat{y_i}}{\sum_{y_i > \widehat{y_i}} 1}. \tag{5}$$

Finally, for emotion intensity regression, we present a global metric $\overline{MSE}$ that averages the squared errors over the indicator of the presence of emotion, and all six emotions. It is defined in Equation 6. As our main focus is on depression, we do not compute emotion-wise metrics, and $\overline{MSE}$ acts as a global indicator.

$$\overline{MSE} = \frac{1}{|\gamma|} \sum_{\gamma} \sum_{i=1}^{i=7} (y_i - \hat{y}_i)^2. \tag{6}$$

Overall evaluation results are given in Table 2. Note that we provide all confusion matrices as supplementary online material[14] to show the overall sketch for DLC.

5.1. Results by Task

DLC can be seen as a coarse-grain task compared to DLR. In this paper, we study both tasks contrarily to previous related works, which only focus on the fine-grained task.

With respect to **DLC**, the best results in terms of Accuracy are obtained by the emotion-unaware multi-task baseline that combines both DLC and DLR, outdoing the best emotion-aware model by 3.03%. However, best results in terms of RMSE and MAE are evidenced by the emotion-aware shared-private multi-task model that concurrently learns all tasks DLC, DLR and EIR. In this case, improvements respectively reach 23.57% for RMSE and 22.7% for MAE. So, although the baseline tends to produce more accurate results, incorrect guesses largely deviate from the correct answer. Moreover, baseline decisions tend to under-evaluate the degree of depression. Indeed, for the best baseline model, 90.9% (Under=30.31%) of the incorrect guesses are under-evaluated, compared to only 9.1% (Over=3.03%), which are over-evaluated. In comparison, the emotion-aware model tends to over-evaluate depression levels in 66.6% (Over=24.24%) of the incorrect cases, and

[14] http://dias.users.greyc.fr/cm.pdf.

under-evaluates them in 33.3% (Under=12.12%), showing a more balanced behavior. In terms of medical decisions, this phenomenon can be an important issue, as under-evaluating the degree of depression of a given patient may have worst consequences than over-evaluating it, although none of these cases should be encountered.[15]

With respect to **DLR**, the best results overall are obtained for the emotion-aware models. In this case, a minimum RMSE=4.51 is obtained by the shared-private model that combines DLR and EIR, and a minimum MAE=3.69 is achieved by the adversarial shared-private model that combines DLR, DLC and EIR. Note that the best evidenced model for DLC (that is, shared-private multi-task model combining DLC, DLR and EIR) shows very similar results with RMSE=4.56 and MAE=3.79 for DLR. As a consequence, an improvement of 4.04% in terms of RMSE and 3.14% in terms of MAE can be achieved over the best baseline, embodied by the shared-private multi-task model that combines DLC and DLR. Interestingly, the emotion-aware models tend to show that in case of over-evaluation, the exceeding values are smaller for the baselines, a situation that also occurs for under-evaluation, although values of under-evaluation are larger than figures evidenced by over-evaluation. As a consequence, there is a tendency of under-evaluation of all models, which may be a drawback in terms of medical issue as mentioned above.

With respect to **EIR**, best results are unexpectedly obtained for depression-aware models, suggesting that emotion intensity regression may also benefit from depression level regression/classification. In particular, the best improvement is evidenced by the shared-private two-task model, which learns DLR and EIR concurrently, with $\overline{MSE}$=6.82, closely followed by the fully-shared model that combines DLR and EIR with $\overline{MSE}$=6.88, evidencing the second best result. As such, an improvement of 4.6% can be obtained compared to the baseline.[16]

The first results show that emotion-aware models can improve the performance of the depression level estimation. In particular, the shared-private multi-task model combining DLC, DLR and EIR seems the more regular architecture to improve over all three tasks on average, as it is highly ranked for all tasks individually across all evaluation metrics. Nevertheless, in order to better understand these results, we propose a class-wise analysis.

5.2. Results by Class

The overall idea of the class-wise analysis is to verify whether some classes of depression are better handled by the classifiers than others. Note that as far as we know, previous related works do not incorporate such an analysis and rely exclusively on overall results, thus failing to take into account important medical issues. The overall results by class are given in Table 3. Note that we do not show all evaluation metrics as it has been evidenced in Table 2 that they are all highly correlated.

For that purpose, we present the exact same results of Table 2 down-described by the 5 classes of depression, which are, none-minimal, mild, moderate, moderately severe and severe. Although this information is interesting, it must be carefully interpreted as the number of test examples is small and not equally distributed. For example, there is only one test example for severe depression, and the DAIC-WOZ dataset contains only four such cases. Overall results are presented in Table 3.

Within this context, overall results show high inequalities between class. The **none-minimal class** seems to be well-handled with high accuracy values and respectively low RMSE and MAE on average for both DLC and DLR for all models, including baselines. Note that the best three-task multi-task model evidences the lowest RMSE and MAE values for this particular class, although it fails to correctly classify all examples. Moreover, there is a clear tendency for over-evaluation, which is understandable as many examples have a PHQ-8 score equals to 0. These observations are clearly positive indicators that strong classification/regression results can be obtained for the class with more patients involved both at training and test splits.

On the other side, the **severe class** shows worst class-wise results as none of the models is capable of correctly classifying the single example present in the test set. Moreover, almost all models fail to correctly estimate this example by a large margin: two classes difference for DLC, and large RMSE and MAE values for DLR, although less expressive values are obtained for emotion-aware architectures. Of course, these concluding remarks can not be generalized due to the lack of statistical evidence over more examples.

As for the **moderately severe class**, all models perform like-wise in terms of DLC accuracy. However, the emotion-aware models evidence lower RMSE and MAE values than baseline models, thus showing more accurate classification estimations. However, in terms of DLR, huge average under-evaluation values are shown by all models, thus showing the difficulty to handle this class in terms of regression. Note that this class is the one that evidences worst results overall in terms of RMSE and MAE for DLR over all models. In fact, some patients within this class can easily be classified, but others are rather difficult to estimate in terms of depression level, and odd low values are usually given by the learning model to these cases.

The **mild class** receives best accuracy levels for the baseline model, and the best three-task model clearly fails within this class, showing worst results overall in terms of DLC. In this case, emotion-aware models do not benefit from the introduction of the concurrent learning of emotion intensity. Moreover, almost no improvement is obtained in terms of DLR by emotion-aware models, to the exception of the shared-private multi-task models combining DLR and EIR, with minor improved results. This class is certainly the one where our initial hypothesis does not clearly stand.

Finally, the **moderate class** receives highest classification results with the three-task model by a large margin. In this case, it clearly outperforms all emotion-aware and emotion-unaware models, for accuracy, RMSE and MAE. With respect to DLR, the best performing model is still an emotion-aware model, but the two-task model. In this case, it clearly outperforms all other tested models.

[15] We will see in this section that most DLR models under-evaluate estimations.
[16] Stronger analysis is out of the scope of this paper.

TABLE 3 Detailed classification/regression results by depression level class: None-minimal (0-4 PHQ-8 score), mild (5-9 PHQ-8 score), moderate (10-14 PHQ-8 score), moderately severe (15-19 PHQ-8 score), severe (20-24 PHQ-8 score). Results for the best performing architecture only are given. Acc., Ov. and Un. metrics are given in % and respectively correspond to accuracy, over and under. RMSE refers to root mean squared error, MAE to mean average error, and $\overline{Ov.}$ and $\overline{Un.}$ to over-evaluation and under-evaluation for regression.

MODELS	EVALUATION METRICS								
	DLC					DLR			
	ACC.	RMSE	MAE	OV.	UN.	RMSE	MAE	$\overline{Ov.}$	$\overline{Un.}$
	BEST FOR DLC WITHOUT EIR: FS MT. DLC+DLR					BEST FOR DLR WITHOUT EIR: SP MT. DLC+DLR			
NONE-MINIMAL	100	0.00	0.00	0	–	3.97	3.22	3.51	1.14
MILD	40	1.10	0.80	20.00	40	3.80	3.11	3.82	2.05
MODERATE	40	1.34	1.00	0.00	60	4.04	3.50	0.00	3.50
MODERATELY SEVERE	33.33	2.27	1.83	0.00	66.67	6.78	5.75	0.47	6.81
SEVERE	0	2.00	2.00	–	100	6.81	6.81	0.00	6.81
	BEST FOR DLC+EIR: ASP MT. DLC+EIR					BEST FOR DLR+EIR: SP MT. DLR+EIR			
NONE-MINIMAL	100	0.00	0.00	0	–	4.28	3.85	4.05	0.74
MILD	20	1.18	1.00	40	40	3.51	3.07	3.56	2.32
MODERATE	20	1.61	1.40	20	60	2.94	2.60	0.00	2.60
MODERATELY SEVERE	33.33	2.16	1.67	0	66.67	6.70	6.05	2.77	6.71
SEVERE	0	2.00	2.00	–	100	2.03	2.03	0.00	2.03
	BEST FOR DLC+DLR+EIR: SP MT. DLC+DLR+EIR								
NONE-MINIMAL	93.75	0.50	0.13	6.25	–	3.42	2.89	2.97	1.79
MILD	0	1.00	1.00	60	40	3.78	3.49	3.89	2.88
MODERATE	80	0.89	0.40	0	20	3.84	3.37	0.00	3.37
MODERATELY SEVERE	33.33	1.41	1.00	0	66.67	7.54	6.78	4.67	7.21
SEVERE	0	2.00	2.00	–	100	3.85	3.85	0.00	3.85

Note that in all cases, there is a clear tendency for under-evaluation as no estimator over-evaluates any patient's level of depression.

Although, as explained before, no strict concluding remarks can be drawn from this analysis due to the small number of test examples, this class-wise analysis should systematically be included in related works of depression level estimation. Indeed, it seems that some classes are more difficult to handle than others, and also models do not perform equally over all classes, although there is a tendency, which confirms the initial hypothesis that emotion intensity estimation can be beneficial to depression level classification/regression.

5.3. Results by Learning Models

Finally, we analyze the behavior of each multi-task model in terms of the fully-shared, shared-private and adversarial shared-private architectures. In particular, improved results were expected by the adversarial shared-private models following initial results reported in [14], [28]. However, this architecture never reaches the highest results, with the exception of the two-task model that includes DLC and EIR, even though it is with a tiny margin over the shared-private architecture. In fact, the adversarial shared-private framework relies on a generator, which learns a shared representation that is capable of fooling the discriminator in terms of task label. This architecture can indeed be beneficial when the concurrent tasks are closely related and in particular when they share some ambiguous features. However, this is not really the case in our experiments as the length of the transcripts is unequal for depression and emotion levels, as well as the vocabulary may not highly overlap. As a consequence, finding a shared representation that can discriminate between both tasks is not a difficult problem, and the learned representation may not be informative enough to handle the concurrent tasks individually. So, the shared-private models regularly evidence stronger results both for DLC and DLR, to the exception of the baseline model, which combines both DLC and DLR. In this case, the fully-shared model shows the best results. This can easily be understood, as (1) the same training dataset is used twice in the training step enforcing the generalization process in terms of shared representation and (2) DLC can be seen as a subtask of DLR, thus including a strong regularization process within the model. Note that this finding is at the origin of the proposed shared-private three-task architecture, that includes a fully-shared layer between DLC and DLR, and globally evidences more stable results overall.

6. Conclusion

In this paper, we tested the hypothesis that depression level classification/regression can leverage from the concurrent learning of emotion intensity. For that purpose, we implemented a series of

emotion-aware and emotion-unaware multi-task architectures over combinations of three tasks: depression level classification, depression level regression and emotion intensity regression. Strong evaluation including new metrics and class-wise results shows that emotion-aware models outperform emotion-unaware baselines in a vast majority of tested situations over the standard benchmarks DAIC-WOZ and CMU-MOSEI. We anticipate that our work will help to reduce the number of cases of late treatment of depression, as one can always get an estimate of his/her PHQ-8 score, without needing to consult a psychiatrist, especially considering the stigma surrounding this illness. However, current results are not accurate enough to help the therapist in his diagnosis as model performance is still too low. This should be a great motivation for future work in depression level estimation. Such research directions include (1) the combination of text, visual and acoustic modalities following the ideas of [10], [17], [33], (2) the study of different concurrent tasks for depression estimation and (3) the creation of larger datasets to better evaluate depression models in terms of class-wise results, that may also include new biomarkers or descriptors.

Acknowledgments

Dr. Sriparna Saha gratefully acknowledges the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research.

References

[1] T. Vos, R. M. Barber, B. Bell, and A. Bertozzi-Villa, "Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: A systematic analysis for the Global Burden of Disease Study 2013," *Lancet*, vol. 386, no. 9995, pp. 743–800, Aug. 2015. doi: 10.1016/S0140-6736(15)60692-4.

[2] A. Ferrari et al., "Burden of depressive disorders by country, sex, age, and year: Findings from the Global Burden of Disease Study 2010," *PLoS Med.*, vol. 10, no. 11, p. e1001547, Nov. 2013. doi: 10.1371/journal.pmed.1001547.

[3] A. Beck and B. Alford, *Depression: Causes and Treatment*. Philadelphia: Univ. of Pennsylvania Press, 2009.

[4] K. Smith, P. Renshaw, and J. Bilelloa, "The diagnosis of depression: Current and emerging methods," *Compr. Psychiatry*, vol. 54, no. 1, pp. 1–6, 2013. doi: 10.1016/j.comppsych.2012.06.006.

[5] K. Kroenke, T. Strine, R. Spitzer, J. Williams, J. T. Berry, and A. Mokdad, "The PHQ-8 as a measure of current depression in the general population," *J. Affect. Disord.*, vol. 114, no. 1–3, pp. 163–173, 2008. doi: 10.1016/j.jad.2008.06.026.

[6] S. El-Den, T. Chen, Y.-L. Gan, E. Wong, and C. O'Reilly, "The psychometric properties of depression screening tools in primary healthcare settings: A systematic review," *J. Affect. Disord.*, vol. 225, pp. 503–522, Jan. 2018. doi: 10.1016/j.jad.2017.08.060.

[7] C. Shin, S.-H. Lee, K.-M. Han, H.-K. Yoon, and C. Han, "Comparison of the usefulness of the PHQ-8 and PHQ-9 for screening for major depressive disorder: Analysis of psychiatric outpatient data," *Psychiatry Investig.*, vol. 16, no. 4, pp. 300–305, 2019. doi: 10.30773/pi.2019.02.01.

[8] N. Cummins, S. Scherer, J. Krajewski, S. Schnieder, J. Epps, and T. F. Quatieri, "A review of depression and suicide risk assessment using speech analysis," *Speech Commun.*, vol. 71, pp. 10–49, July 2015. doi: 10.1016/j.specom.2015.03.004.

[9] J. T. Wolohan, M. Hiraga, A. Mukherjee, Z. A. Sayyed, and M. Millard, "Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with NLP," in *Proc. Int. Workshop Language Cognition and Computational Models*, Santa Fe, Aug. 2018, pp. 11–21.

[10] M. R. Morales, "Multimodal Depression Detection: An Investigation of Features and Fusion Techniques for Automated Systems," Ph.D. thesis, City Univ. of New York, 2018.

[11] J. Joormann and I. Gotlib, "Emotion regulation in depression: Relation to cognitive inhibition," *Cogn. Emot.*, vol. 24, no. 2, pp. 281–298, 2010. doi: 10.1080/02699930903407948.

[12] R. Thompson, M. Boden, and I. Gotlib, "Emotional variability and clarity in depression and social anxiety," *Cogn. Emot.*, vol. 31, no. 1, pp. 98–108, 2017. doi: 10.1080/02699931.2015.1084908.

[13] P. Ekman and R. Davidson, *The Nature of Emotion: Fundamental Questions*. London, U.K.: Oxford Univ. Press, 1994.

[14] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," in *Proc. Annu. Meeting Association for Computational Linguistics*, Vancouver, B.C., Canada, July 30–Aug. 4, 2017, pp. 1–10. doi: 10.18653/v1/P17-1001.

[15] S.-A. Qureshi, S. Saha, M. Hasanuzzaman, and G. Dias, "Multitask representation learning for multimodal estimation of depression level," *IEEE Intell. Syst.*, vol. 34, no. 5, pp. 45–52, 2019. doi: 10.1109/MIS.2019.2925204.

[16] J. Gratch et al., "The distress analysis interview corpus of human and computer interviews," in *Proc. Int. Conf. Language Resources and Evaluation*, Reykjavik, May 26–31, 2014, pp. 3123–3128.

[17] A. Zadeh et al., "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proc. Annu. Meeting Association for Computational Linguistics*, Melbourne, July 15–20, 2018, doi: 10.18653/v1/P18-1208.

[18] N. Dewan, J. Luo, and N. Lorenzi, *Mental Health Practice in a Digital World: A Clinicians Guide*. New York: Springer-Verlag, 2015.

[19] M. Chatterjee, G. Stratou, S. Scherer, and L.-P. Morency, "Context-based signal descriptors of heart-rate variability for anxiety assessment," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Florence, Italy, May 4–9, 2014, doi: 10.1109/ICASSP.2014.6854278.

[20] M. Morales and R. Levitan, "Speech vs. text: A comparative analysis of features for depression detection systems," in *Proc. IEEE Spoken Language Technology Workshop*, San Diego, CA, May 13–16, 2016, doi: 10.1109/SLT.2016.7846256.

[21] D. Hovy, M. Mitchell, and A. Benton, "Multitask learning for mental health conditions with limited social media data," in *Proc. Conf. European Chapter Association for Computational Linguistics*, Valencia, Spain, Apr. 3–7, 2017, pp. 152–162. doi: *10.18653/v1/E17-1015*.

[22] D. Losada and F. Crestani, "A test collection for research on depression and language use," in *Proc. Int. Conf. Cross-Language Evaluation Forum for European Languages*, Évora, Portugal, Sept. 5–8, 2016, pp. 28–39. doi: 10.1007/978-3-319-44564-9_3.

[23] H. Dibeklioğlu, Z. Hammal, Y. Yang, and J. Cohn, "Multimodal detection of depression in clinical interviews," in *Proc. ACM Int. Conf. Multimodal Interaction*, Seattle, WA, Nov. 9–13, 2015, pp. 307–310. doi: *10.1145/2818346.2820776*.

[24] M. Morales, S. Scherer, and R. Levitan, "A linguistically-informed fusion approach for multimodal depression detection," in *Proc. Workshop Computational Linguistics and Clinical Psychology: From Keyboard Clinic*, New Orleans, LA, June 2018, pp. 13–24. doi: 10.18653/v1/W18-0602.

[25] S.-A. Qureshi, M. Hasansuzzaman, S. Saha, and G. Dias, The verbal and non verbal signals of depression: Combining acoustics, text and visuals for estimating depression level. Apr. 2019. [Online]. Available: arXiv:1904.07656

[26] B. Sun et al., "A random forest regression method with selected-text feature for depression assessment," in *Proc. Annu. Workshop Audio/Visual Emotion Challenge*, Mountain View, CA, Oct. 23–27, 2017, pp. 61–68. doi: 10.1145/3133944.3133951.

[27] J. Bingel and A. Søgaard, "Identifying beneficial task relations for multi-task learning in deep neural networks," in *Proc. Conf. European Chapter Association for Computational Linguistics*, Valencia, Spain, Apr. 3–7, 2017, pp. 164–169. doi: 10.18653/v1/E17-2026.

[28] S. Yadav, A. Ekbal, S. Saha, P. Bhattacharyya, and A. Sheth, "Multi-task learning framework for mining crowd intelligence towards clinical treatment," in *Proc. Conf. North American Chapter Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, June 1–6, 2018, pp. 271–277. doi: 10.18653/v1/N18-2044.

[29] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1998. doi: 10.1023/A:1007379606734.

[30] D. Cer et al., Universal sentence encoder. Mar. 2018. [Online]. Available: arXiv:1803.11175

[31] S. Hochreiter and J. Schmidhuber, "Long short-term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[32] T. Young, D. Hazarika, S. Poria, and E. Cambria, Recent trends in deep learning based natural language processing. Aug. 2017. [Online]. Available: arXiv:1708.02709

[33] S. Poria, E. Cambria, D. Hazarika, N. Mazumder, A. Zadeh, and L.-P. Morency, "Multi-level multiple attentions for contextual multimodal sentiment analysis," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, Nov. 18–21, 2017, pp. 1033–1038. doi: 10.1109/ICDM.2017.134.

[34] T. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Sept. 17–21, 2015, pp. 1412–1421. doi: 10.18653/v1/D15-1166.

[35] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: An open source facial behavior analysis toolkit," in *Proc. IEEE Winter Conf. Applications Computer Vision*, Lake Placid, NY, Mar. 7–9, 2016, pp. 1–10. doi: 10.1109/WACV.2016.7477553.

[36] G. Littlewort et al., "The computer expression recognition toolbox (CERT)," in *Proc. IEEE Int. Conf. and Workshops Automatic Face and Gesture Recognition*, Santa Barbara, CA, Mar. 21–25, 2011, pp. 298–305. doi: 10.1109/FG.2011.5771414.

[37] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, "COVAREP: A collaborative voice analysis repository for speech technologies," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Florence, Italy, May 4–9, 2014, pp. 960–964. doi: 10.1109/ICASSP.2014.6853739.

[38] K. Kroenke, R. Spitzer, and J. Williams, "The PHQ-9: Validity of a brief depression severity measure," *J. Gen. Intern. Med.*, vol. 16, no. 9, pp. 606–613, 2001. doi: 10.1046/j.1525-1497.2001.016009606.x.

[39] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, MOSI: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. June 2016. [Online]. Available: arXiv:1606.06259

[40] F. Ringeval et al., "AVEC 2017: Real-life depression, and affect recognition workshop and challenge," in *Proc. Annu. Workshop Audio/Visual Emotion Challenge*. Mountain View, CA, Oct. 23–27, 2017, pp. 3–9. doi: 10.1145/3133944.3133953.

An-Jen Liu*, Ti-Rong Wu*, I-Chen Wu*, Hung Guei, and Ting-Han Wei
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

Strength Adjustment and Assessment for MCTS-Based Programs

Abstract

This paper proposes an approach to strength adjustment and assessment for Monte-Carlo tree search based game-playing programs. We modify an existing softmax policy with a strength index to choose moves. The most important modification is a mechanism which filters low-quality moves by excluding those that have a lower simulation count than a pre-defined threshold ratio of the maximum simulation count. Through theoretical analysis, we show that the adjusted policy is guaranteed to choose moves exceeding a lower bound in strength by using a threshold ratio. Experimental results show that the strength index is highly correlated to the empirical strength. With an index value between ±2, we can cover a strength range of about 800 Elo ratings. The strength adjustment and assessment methods were also tested in real-world scenarios with human players, ranging from professionals (strongest) to kyu rank amateurs (weakest). For amateur levels, we tested our mechanism on two popular Go online platforms - Fox Weiqi and Tygem. The result shows that our method can adjust program strength to different ranks stably. In terms of strength assessment, we proposed a new dynamic strength adjustment method, then used it to evaluate human professionals, predicting reliably their playing strengths within 15 games. Lastly, we collected survey responses asking players about strength perception, entertainment, and general comments for different aspects of analysis. To our best knowledge, this result is state-of-the-art in terms of the range of strengths in Elo rating while maintaining a controllable relationship between the strength and a strength index.

©ISTOCKPHOTO.COM/DMYTO

I. Motivation

Super-human level game playing programs capture the imagination and fascination of society at large; for human players, these programs pose an interesting challenge and offer opportunities for learning. Since the super-human programs AlphaGo [1] and AlphaGo Zero [2], many other programs such as FineArt [3], Leela Zero [4], and ELF OpenGo [5] have successfully reproduced the AlphaGo Zero algorithm. In addition to Go, AlphaGo Zero's method was also applied to other games such as chess and shogi, reaching strength levels much higher than human champions [6] and state-of-the-art game-playing programs [7], [8].

However, it is also important to fit program difficulty to appropriate levels for purposes of entertainment and education. On the low end of the difficulty scale, human players tend to lose interest when playing against an AI opponent far weaker than themselves; on the other hand, excessive difficulty tends to lead to frustration [9]. In the context of learning to play with programs, it is difficult to offer feedback for human players if they consistently win or lose. Thus, in order to

Digital Object Identifier 10.1109/MCI.2020.2998315
Date of current version: 15 July 2020

Corresponding Author: I-Chen Wu (E-mail: icwu@cs.nctu.edu.tw). *: Equal contribution.

1556-603X/20©2020IEEE

achieve an overall better game experience, and to improve the learning process for players, it is imperative to balance program difficulty accordingly.

In addition to matching the player with an appropriate opponent, strength adjustment can also be used as a player assessment tool. More specifically, given a player of unknown strength, an equally matched opponent (i.e., playing at about 50% win rate) can be used as a metric of measuring the player's performance [10], [11]. Ideally, the goal is to design a mechanism that is able to provide a wide-ranging and evenly distributed set of playing strengths, such that it would be possible for most human players to choose the most appropriate strength to play against. Similarly, any player that falls within the range of provided strengths can also be measured by this mechanism.

Coming up with a stable and wide-ranging strength adjustment mechanism is a non-trivial problem. Using the AlphaGo Zero algorithm as an example target, there are two straightforward tweaks that can be applied to the two major components of the algorithm: Monte-Carlo tree search (MCTS) and deep neural networks (DNNs).

First, we could reduce the superhuman program strength to lower levels by reducing the total thinking time, or the total simulation count in MCTS. However, with this method, the search tree's relatively smaller size leaves the program vulnerable to tactical traps. For example, the ladder problem in Go is one of the most elementary shapes taught to human players [5]; however, for MCTS-based programs, a significant number of simulations are often required before the program can handle ladders properly. It has been shown that when adjusting program strength through reduction, simulation count and playing strength do not form a linear relationship [12]. In fact, once the number of simulations falls below a certain threshold, the program playing strength drops catastrophically. In other words, by reducing the simulation count, we would not be able to offer an evenly distributed skill level, especially towards the lower end of the playing strength scale.

On the low end of the difficulty scale, human players tend to lose interest when playing against an AI opponent far weaker than themselves; on the other hand, excessive difficulty tends to lead to frustration [9].

The second straightforward approach is to offer different programs for different strength levels. For example, we could train separate DNNs for each difficulty. A good example of this type of strength adjustment is Paulsen and Fürnkranz's work[1] on training chess evaluation functions for different strengths [13]. However, this approach usually requires large amounts of time and effort to tune and test the programs. This is especially difficult for games like Go, where strength levels span a wide range from 30 kyu to professional 9 dan [14], corresponding to about a 3,000 Elo rating [15] difference.

The main contributions of this paper are as follows:

- We improve upon an existing strength adjustment approach with a threshold mechanism to filter out low-quality moves.
- We present an in-depth strength analysis to justify that there is a lower bound on the move quality.
- We introduce three dynamic strength adjustment methods to predict opponents' expected strengths automatically.
- We successfully applied our methods in real-world cases ranging from professionals to kyu rank amateurs in terms of strength adjustment and strength assessment.

The structure of this paper is as follows. First, we review a strength adjustment (SA) approach based on the softmax policy and propose our modification in Section II. We then apply the method to two programs: 1) the open-source Go program ELF OpenGo (abbr. ELF for the remainder of this paper); 2) the Go program CGI [16]. From these two implementations, we demonstrate that the modified strength adjustment method can be easily used to adjust the strength of a program, covering a range of over 800 Elo ratings. In Section III, this paper presents a hypothesis and performs theoretical analyses to justify the empirical strengths shown in Section II. Having demonstrated that the program strength can be adjusted with relative ease, we introduce methods to adjust the strength dynamically in Section IV. In Section V, we test our method on real-world scenarios, playing against human players. Finally, we summarize our contributions in Section VI.

This paper is an extension of our work, which was published in the AAAI 2019 conference [17]. The mixed dynamic strength adjustment (MDSA) mechanism (Subsection IV-C) and real-world experiments (Section V) contain new work and data for this version of the paper.

II. Static Strength Adjustment (SSA)

In this section, we first review past work on strength adjustment, then present our modifications to the method. We apply the modified approach to the Go program ELF and provide empirical data.

A. Past Work

For strength adjustment, Sephton et al. [12] investigated some selection mechanisms for MCTS-based game-playing programs. One of these mechanisms uses a simple softmax policy as follows. Given strength index z, choose moves i with probability $N_i^z / \sum_j N_j^z$, where N_i is the number of simulations on move i in MCTS. For simplicity of discussion in the rest of this paper, let $N_i \geq N_j$ if $i < j$, i.e., N_1 is the maximum.

Conceptually, z is the inverse of the softmax temperature. When z is higher,

[1]Paulsen and Fürnkranz's method does not use DNNs, but is similar to our example here in that they create multiple programs, each offering one static strength.

the policy tends to choose moves with higher simulation counts, which tends to be a higher-quality move as is the case with MCTS. When z approaches infinity, the move with the highest simulation count is guaranteed to be chosen, and thus the policy exhibits the same behavior as the original MCTS. When $z = 0$, all moves are chosen with equal probability, i.e., moves are chosen randomly. When z approaches negative infinity, the move with the lowest simulation count is chosen, i.e., the policy tends to choose the lowest quality moves.

Thus, z can serve as an index of strength. Sephton et al. [12] showed through experiments that z is correlated to the empirical strength. In their experiments, the strength indices only covered six trials $z \in \{1, 2, 3, 4, 5, 6\}$ for the game Lords of War, where the differences of win rates for these values of z range from 5% to 24%, equivalent to a range of about 100 Elo ratings.

As stated above, when z is low, the policy tends to choose low-quality moves. However, in MCTS, many moves are not visited during simulation, or in some cases, visited very few times only because of the exploration bias [18]. For this reason, it is not a good idea to allow the policy to choose the lowest-quality moves, which would result in a much weaker program or unpredictable behavior.

TABLE I The win rates (against ELF with $z = 0$) and Elo ratings (relative to the original ELF) with respect to z when $R_{th} = 0.1$.

z	WIN RATE (± ERRORS)	ELO RATING (± ERRORS)
∞	97.6% (±1.9%)	0 (−106, +289)
2.0	94.4% (±2.9%)	−153 (−78, +133)
1.5	92.4% (±3.4%)	−210 (−70, +107)
1.0	91.2% (±3.6%)	−237 (−66, +98)
0.5	71.6% (±5.7%)	−483 (−46, +52)
0.0	50.0%	−644
−0.5	35.6% (±6.1%)	−747 (−48, +44)
−1.0	21.6% (±5.2%)	−868 (−59, +49)
−1.5	13.2% (±4.3%)	−971 (−76, +58)
−2.0	12.4% (±4.2%)	−983 (−79, +59)
−∞	7.2% (±3.3%)	−1,088 (−111, +71)

In order to avoid choosing the very lowest-quality moves, Sephton et al. suggested choosing the first n best moves as candidates, where n is a given fixed value. However, it is still possible to choose a very low-quality move, e.g., in the case that only one move is viable while the others are extremely bad, the policy is still likely to choose bad moves.

B. Our Approach

In our approach, we follow the softmax policy to choose moves via the strength index z. With a simple softmax policy, there is the possibility of selecting poor moves. To perform move screening and in turn ensure move quality, we use a threshold ratio R_{th} to avoid moves with lower simulation counts. Namely, given a threshold ratio R_{th}, we only consider the moves i with $N_i \geq N_1 \times R_{\text{th}}$ as candidates. Assuming that the move quality is correlated to the simulation count (we discuss this in greater detail in Section III), this approach ensures that the qualities of the chosen moves are higher than the screened moves, which do not reach the threshold. At the very least, the modified policy is less likely to choose extremely bad moves, as Sephton et al.'s method [12] is prone to do.

For a high threshold ratio, more low-quality moves are filtered. When $R_{\text{th}} = 1$, the move with the highest simulation count is always chosen, behaving the same way as the original MCTS. In contrast, for a low threshold ratio, many low-quality moves are not filtered. Thus, it is important to set a reasonable threshold ratio, where the goal is to filter most low-quality moves, while simultaneously allowing reasonable moves to be considered.

In contrast to Sephton et al.'s method [12], our empirical results in the next subsection show that strengths can be adjusted across a wide range over 800 Elo ratings with the threshold ratio set to 0.1 and $z = [-2, 2]$. Thus, our approach is very suitable for games that are considered to have very high depth [19].

C. Empirical Evaluation

We apply the above approach to the Go program ELF [5] with the 20-block model and present the experimental results. All the experiments are performed on machines equipped with one GTX 1080Ti GPU, one Intel Xeon E5-2683 v3 (14 cores in total), 2.6 GHz, 128 GB memory, and with Linux. All games are played with one second per move, using one GPU and six CPU cores. For each benchmark, 250 games are played against a baseline consisting of ELF with $R_{\text{th}} = 0.1$ and $z = 0$. Note that the original ELF is equivalent to the setting with $z = \infty$, which we do not use as the baseline since it is much too strong for some trials, such as when $z = -\infty$.

Table I shows the win rates and the relative Elo ratings of the ELF versions with $R_{\text{th}} = 0.1$ and with different values of z against the baseline. The shown Elo ratings are relative to the original ELF ($z = 0$) for simplicity of analysis. Since ELF follows the process of training AlphaGo Zero with 20 residual blocks, its Elo rating of the ELF version is expected to be between 4,000 and 5,000 based on AlphaGo Zero [2].

Figure 1 shows the correlation between z and the Elo ratings. Interestingly, both are highly correlated with a low linear regression error of 47.95 Elo, in terms of the Elo rating, when z is between −2 to 2. In addition, the range of strength is very wide, covering 1,088 Elo rating for all z and 830 for the interval of z in $[-2, 2]$.

Furthermore, Figure 2 depicts the correlation between z and the Elo rating for different threshold ratios, 0, 0.02, 0.05, 0.1, 0.25, and 0.5. All games are also played against the same baseline as above. From Figure 2, the correlation between Elo ratings and z is also highly correlated to R_{th} in most cases. A higher value of z tends to correspond to higher Elo ratings.

We observe that high values of R_{th} do not exhibit the intended strength adjustment effects. For example, when $R_{\text{th}} = 0.5$, the Elo rating has no significant changes across different values of z. An intuitive explanation for this is that with a high threshold ratio, most candidate moves are filtered, so the value of z does not matter as much. Figure 3 shows that the average number of candidates is only

1.4 for $R_{th} = 0.5$ and 1.9 for $R_{th} = 0.25$. Another effect is that the adjusted strength range is narrower, e.g., smaller than 500 Elo rating for $R_{th} = 0.25$.

On the other hand, for low threshold ratios, the Elo rating drops quickly, and the strength for different values of z show no difference, e.g., when $R_{th} = 0$ and $z \leq 0$, and when $R_{th} = 0.02$ and $z \leq -1$.

Thus, judging from both Figure 2 and Figure 3, the threshold ratios of 0.05 and 0.1 appear to be suitable for our needs. For simplicity of analysis, 0.1 will be used as the threshold ratio, unless otherwise stated.

III. Theoretical Analysis

The above empirical results show that the strengths are highly correlated to z. In fact, between $-2 \leq z \leq 2$ and a threshold ratio of 0.1, z and the strength show a near linear relationship with regression error 47.95 Elo. However, the strength or Elo rating should be fixed when z approaches ∞ or $-\infty$. Thus, intuitively, the curve of the Elo rating strength according to the z value should be shaped similar to a logistic function. Applying logistic regression [20], the curve is close to a logistic function with error 26.00 Elo ($\beta_0 = -0.25$, $\beta_1 = 1.16$).

This section investigates this conjecture of logistic regression from a theoretical perspective. First, we review the generalized Bradley-Terry model. Second, we present a hypothesis on move strength. Then, from theoretical analysis, we show that the derived strengths are close to the empirical strengths. We calculate the regression error between the derived and the empirical strengths to justify the hypothesis.

A. Generalized Bradley-Terry Model

The Bradley-Terry model [15] has been the foundation of various ranking systems, including the Elo rating system. The model is used to estimate the strengths of players and predict the win rates among these players. Namely, each player i is associated with a positive value γ_i representing the strength of i, and the probability that i wins over j is $\gamma_i/(\gamma_i + \gamma_j)$. Obviously, the higher γ_i is,

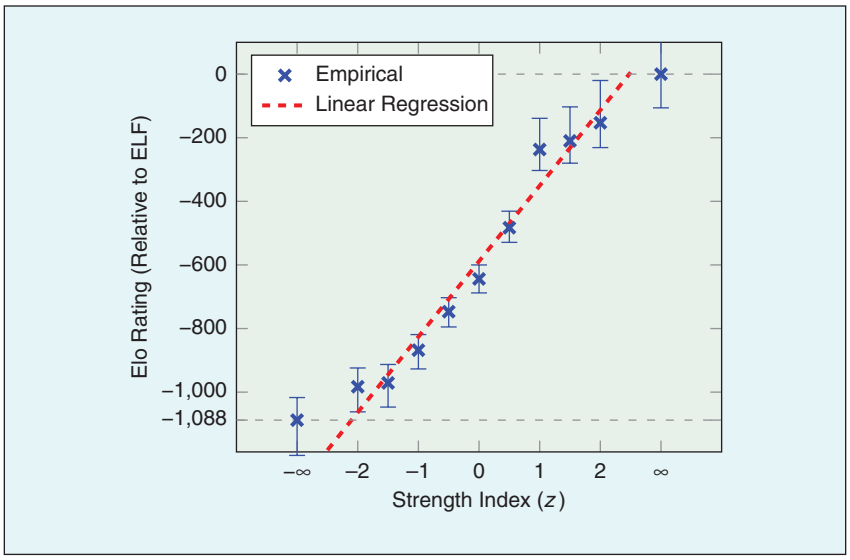


FIGURE 1 The correlation between z and Elo rating with $R_{th} = 0.1$.

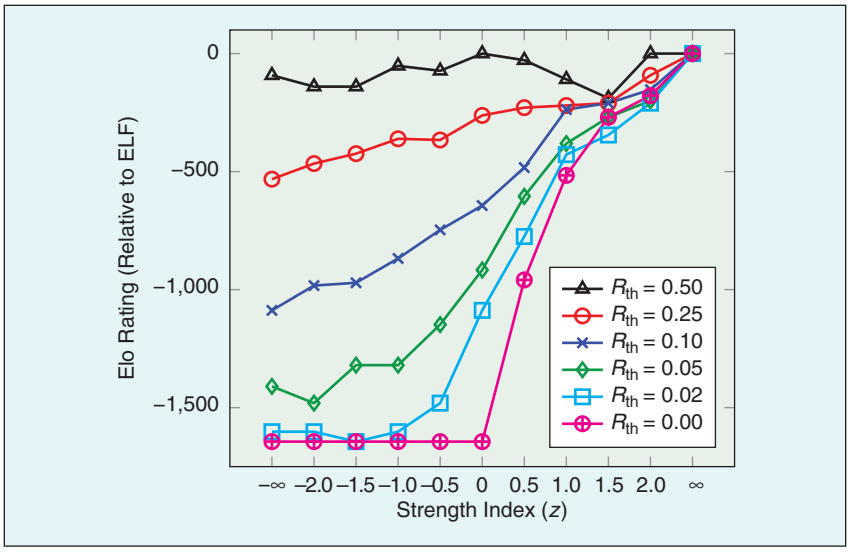


FIGURE 2 Elo rating (relative to ELF) in different threshold ratios and strength indices.

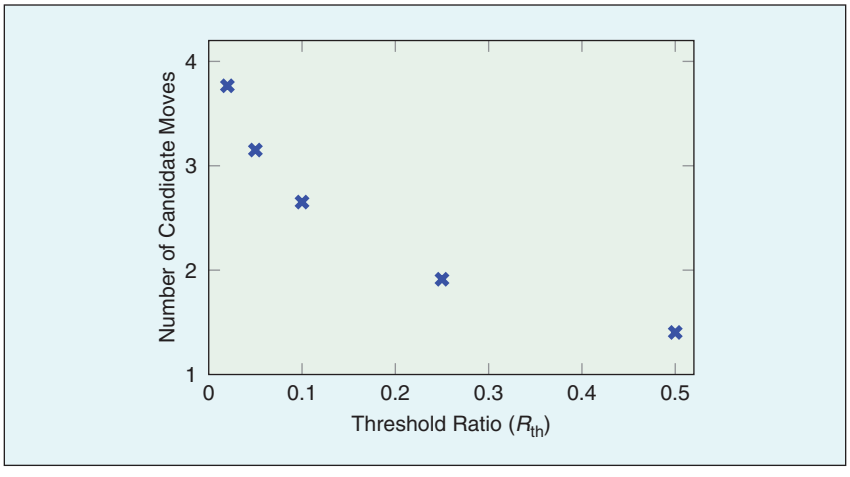


FIGURE 3 The number of candidate moves with respect to R_{th}.

the higher the winning rate (implying a stronger player). The Elo rating of individual i is $Elo_i = 400 \log_{10} \gamma_i$ [21]. For simplicity of discussion in this paper, we also define the rating $E_i = \ln \gamma_i$, whose corresponding Elo rating is $Elo_i = 400 (\log_{10} e) E_i$.

The Bradley-Terry model has also been generalized to handle competitions involving more than two players [22], [23]. Namely, the probability that i wins among n players, 1, …, n, is formulated as $\gamma_i / \sum_i^n \gamma_i$. Another generalization is to allow competitions among teams, instead of players. The strength and its corresponding rating of a team of n players is estimated as

$$\gamma^{\text{all}} = \prod_i^n \gamma_i, \text{ and } E^{\text{all}} = \sum_j^n E_j. \quad (1)$$

In this paper, we also define the average strength and rating of a team of n players to be

$$\gamma^{\text{avg}} = \left(\prod_i^n \gamma_i\right)^{\frac{1}{n}}, \text{ and } E^{\text{avg}} = \left(\frac{1}{n}\right) \sum_j^n E_j. \quad (2)$$

This is useful when n is not fixed. In addition, consider a team that can choose one and only one player to participate and choose player i with probability π_i, where $\sum_j^n \pi_j = 1$. Thus, the strength and rating of the team are

$$\gamma = \prod_i^n \gamma_i^{\pi_i}, \text{ and } E = \sum_j^n \pi_i E_j, \quad (3)$$

respectively, for the reason as illustrated below. For example, let $\pi_i = N_i / \sum_j^n N_j$. We can consider the team composed of $\sum_j^n N_j$ players, among which the number of players i is N_i. Thus, the average strength and rating of the team are the same as γ and E in Equation (3), respectively.

B. Hypothesis

As mentioned above, moves with higher simulation counts N_i in MCTS normally tend to have higher quality. Following this notion, we present a hypothesis for further theoretical analysis. Assume that given a position the strength of move i is proportional to N_i^H. Here, H denotes a conjectured strength index for moves to be selected in MCTS in the previous sections. Namely, let $\gamma_i = c \times N_i^H$. Here, c is a constant coefficient with respect to the same game position, i.e., different positions may have different relative strengths, and therefore will have a different value of c.

If we view the moves as separate players that follow the generalized Bradley-Terry model, the rating of move i is $E_i = \ln \gamma_i = \ln c + H \ln N_i$. For simplicity of analysis, we use E_i in the following analysis without loss of generality. If the Elo rating is preferred, Elo_i can be obtained by a simple conversion, as described above.

Let $\gamma(z)$ and $E(z)$ denote the overall strength and rating following the above method for strength adjustment, which chooses among all moves i using the softmax policy $\pi_i(z) = N_i^z / \sum_i (N_i^z)$. If we view all i moves as a team comprised of individual moves, from the above Bradley-Terry model for team strength (Equation 3), we can derive that

$$\gamma(z) = \prod_i \gamma_i^{\pi_i(z)}, \text{ and} \quad (4)$$

$$\begin{aligned} E(z) = \ln \gamma_i &= \sum_i \pi_i(z) E_i \\ &= \sum_i \pi_i(z) (\ln c + H \ln N_i) \\ &= \ln c + H \sum_i \pi_i(z) \ln N_i. \end{aligned} \quad (5)$$

In the above formula, the first item in Equation (5) is fixed for this position, and therefore it can be removed to obtain relative ratings, say, relative to the rating where $z = \infty$ (which always chooses the move with the maximum simulation count N_1) as follows.

$$\begin{aligned} E_{\text{rel}}(z) &= E(z) - E(\infty) \\ &= H \sum_i \pi_i(z) (\ln N_i - \ln N_1) \\ &= H \sum_i \pi_i(z) (\ln R_i), \end{aligned} \quad (6)$$

where R_i is the ratio N_i / N_1. Since all moves with the ratio less than R_{th} are filtered out, $R_i \geq R_{\text{th}}$. In addition, since N_1 is the maximum among N_i, $R_i \leq 1$ and $\ln R_i$ are therefore all non-positive. Thus, we obtain

$$E_{\text{rel}}(z) \geq H \times (\ln R_{\text{th}}). \quad (7)$$

An important implication in Equation 7 is that the relative ratings of the chosen moves are at worst $H \times (\ln R_{\text{th}})$. Assume $R_{\text{th}} = 0.1$. The relative ratings of all chosen moves are at worst $H \times \ln 0.1 \cong -2.3\, H$, not worse than move 1 by 2.3 H. Since H is a constant under this hypothesis, this implies that the strength of any chosen move is at worst a fixed value. This ensures the quality of all chosen moves.

Now, let us consider following the above SSA method to play a game g, containing a sequence of m_g moves or m_g positions to move. Let $\gamma^{(j)}(z)$ and $E^{(j)}(z)$ denote the strength and rating of the move made at the j_{th} position (i.e., on the j_{th} turn). From Equations 4 and 5, which correspond to the strength and rating for a specific position, we can derive the following two equations:

$$\gamma^{(j)}(z) = \prod_i \gamma_i^{\pi_i(z)}, \text{ and} \quad (8)$$

$$\begin{aligned} E^{(j)}(z) &= \sum_i \pi_i^{(j)}(z) E_i^{(j)} \\ &= \ln c^{(j)} + H \sum_i \pi_i^{(j)}(z) \ln N_i^{(j)}, \end{aligned} \quad (9)$$

where $\gamma_i^{(j)}$, $E_i^{(j)}$, $\pi_i^{(j)}$ and $N_i^{(j)}$ are respectively the strength, rating, policy and simulation count of moves i at the j_{th} position in the game, and $c^{(j)}$ is the coefficient with respect to the position.

Furthermore, let $\gamma^g(z)$ and $E^g(z)$ denote the averaged strength and rating as follows.

$$\gamma^g(z) = \left(\prod_j \gamma^{(j)}(z)\right)^{\frac{1}{m_g}}, \text{ and} \quad (10)$$

$$\begin{aligned} E^g(z) &= \left(\frac{1}{m_g}\right) \sum_j E^{(j)}(z) \\ &= \left(\frac{1}{m_g}\right) \sum_j \left(\ln c^{(j)} + H \sum_i \pi_i^{(j)}(z) \ln N_i^{(j)}\right) \\ &= \left(\frac{1}{m_g}\right) \sum_j \ln c^{(j)} \\ &\quad + \left(\frac{H}{m_g}\right) \sum_j \sum_i \pi_i^{(j)}(z) \ln N_i^{(j)}. \end{aligned} \quad (11)$$

Note that we evaluate the averaged strength and rating as Equation (1), instead of the aggregated values in Equation (2), simply because the number of moves in a game is not fixed.

In the above formula, the first item is fixed, and therefore can be omitted when calculating ratings relative to the one with $z = \infty$, similarly, as follows.

$$E^g_{\text{rel}}(z) = E^g(z) - E^g(\infty) = \left(\frac{H}{m_g}\right)\sum_j \sum_i \pi_i^{(j)}(z)(\ln R^{(j)}), \quad (12)$$

where $R_i^{(j)} = N_i^{(j)}/N_1^{(j)}$. Moreover, let the relative rating be normalized to be independent of the value H as follows.

$$E^g_{\text{norm}}(z) = E^g_{\text{rel}}(z)/H = \left(\frac{1}{m_g}\right)\sum_j \sum_i \pi_i^{(j)}(z)(\ln R^{(j)}) = \mathbb{E}_g\left[\sum_i \pi_i^{(j)}(z)(\ln R_i^{(j)})\right]. \quad (13)$$

For stochastic analysis, we extend by collecting some sets of games, each of which is collected from the games under a designated threshold ratio in the above empirical experiments. We exclude extreme cases to minimize the effect of noise for our analysis. For example, the cases of $z = \infty$ and $z = -\infty$ are not included.

For simplicity of analysis, let us illustrate the case for $D_{0.1}$, denoting the set of games with threshold ratio 0.1, which contains about 2000 games. The expected relative rating under the set $D_{0.1}$ is

$$E^{D_{0.1}}_{\text{norm}}(z) = \mathbb{E}_{g\sim D_{0.1}}[E^g_{\text{norm}}(z)] = \mathbb{E}_{g\sim D_{0.1}}\left[\mathbb{E}_g\left[\sum_i \pi_i^{(j)}(z)(\ln R_i^{(j)})\right]\right] = \mathbb{E}_{g\sim D_{0.1}}\left[\sum_i \pi_i^{(j)}(z)(\ln R_i^{(j)})\right]. \quad (14)$$

Figure 4 depicts the solid curve of $E^{D_{0.1}}_{\text{norm}}(z)$ calculated from the set $D_{0.1}$ according to Equation (14). The left y-axis indicates the value of $E^{D_{0.1}}_{\text{norm}}(z)$. The curve resembles a logistic function. Now, let $E^{D_{0.1}}(z)$ denote the expected rating, and $H^{D_{0.1}}$ be the value H, under the set of games $D_{0.1}$. Thus, we have

$$E^{D_{0.1}}(z) = E^{D_{0.1}}(\infty) + H^{D_{0.1}} E^{D_{0.1}}_{\text{norm}}(z). \quad (15)$$

Then, we can derive that

$$E^{D_{0.1}}(-\infty) = E^{D_{0.1}}(\infty) + H^{D_{0.1}} E^{D_{0.1}}_{\text{norm}}(-\infty), \quad (16)$$

and

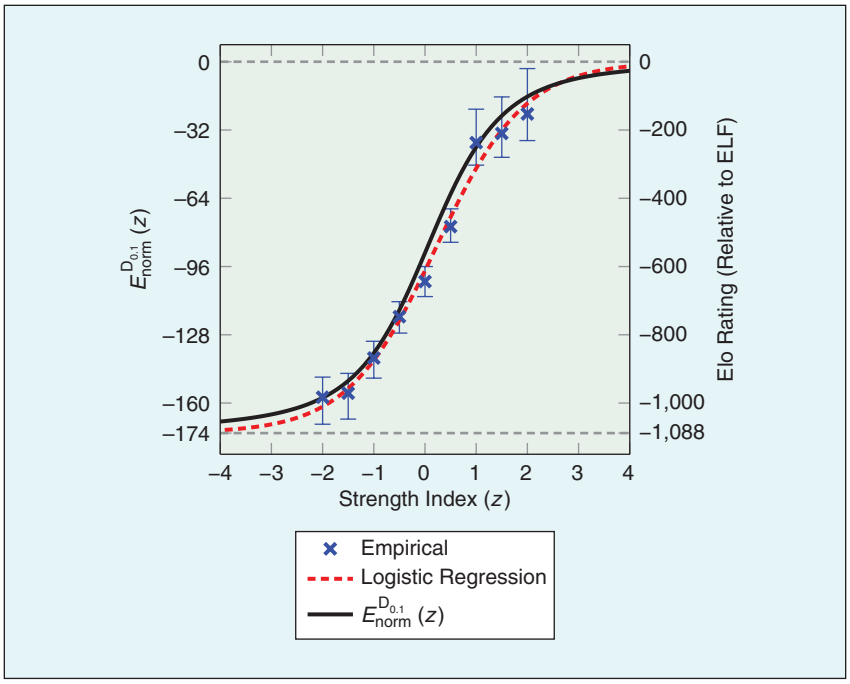


FIGURE 4 The curve of $E^{D_{0.1}}_{\text{norm}}(z)$ and the empirical data.

$$H^{D_{0.1}} = \frac{E^{D_{0.1}}(-\infty) - E^{D_{0.1}}(\infty)}{E^{D_{0.1}}_{\text{norm}}(-\infty)}. \quad (17)$$

Since the values $E^{D_{0.1}}(\infty)$ and $E^{D_{0.1}}(-\infty)$ are supposed to approximate the strength in the empirical experiments, they can be replaced by the empirical strengths at $z = \infty$ and $z = -\infty$, whose relative Elo ratings are 0 and −1088 as shown in Table I. Thus, the value $H^{D_{0.1}}$ is derived to be 6.243 according to the above formula. The right y axis in Figure 4 follows the y axis in Figure 1. The regression error to the empirical strengths for z between −2 and 2 is about 40.45 Elo, and the regression error to a logistic regression curve is about 10.51 Elo ($\beta_0 = -0.05$, $\beta_1 = 1.17$). These low errors justify the hypothesis.

In our experiments, we also derived the value H for other sets of games, as shown in Table II. From the table, $H^{D_{0.25}}$ is almost the same as $H^{D_{0.1}}$, while $H^{D_{0.5}}$, $H^{D_{0.05}}$ and $H^{D_{0.02}}$ are lower. For $H^{D_{0.05}}$ and $H^{D_{0.02}}$, our conjecture is that the noise incurred from having a low threshold ratio is high as the following illustration. In the case of $R_{\text{th}} = 0.02$, since the average number of simulation counts for the best move is about 259.4

TABLE II The conjectured strength indices estimated in different data sets.

DATA SET	$D_{0.02}$	$D_{0.05}$	$D_{0.1}$	$D_{0.25}$	$D_{0.5}$
H	4.385	5.369	6.243	6.244	3.292

with the one second time limit, it is highly likely to include the moves with very low simulation counts (the threshold is about $259.4 \times 0.02 \cong 5.2$). Since many of these simulations may be generated simply because of the exploration bias, these simulations may introduce noise and therefore affect the verification of our hypothesis.

As for $H^{D_{0.5}}$, we observe that the average number of candidates is 1.4 from Figure 3. Since the number is relatively low in many cases, the policy chooses only from a single candidate move. Therefore, the distribution is insufficient to justify our hypothesis. As an example, the most extreme case is where the threshold ratio is 1, and only the moves with the highest simulation counts are chosen, as in the original MCTS. The value of H in this case does not affect the policy at all, since there is only one choice.

IV. Dynamic Strength Adjustment (DSA)

As stated in the previous sections, this paper presents a flexible strength adjustment method simply by altering the value z with an appropriate R_{th}, say 0.1. Moreover, the strength ratings are approximately linear with respect to z in the interval $[-2, 2]$. This allows us to fit the program's strength to its opponents' dynamically, provided the opponents' strengths are within Elo rating differences of $[-983, -153]$ (compared to the original ELF), corresponding to the range of z in $[-2, 2]$ (shown in Figure 2, $R_{\text{th}} = 0.1$). This section introduces two types of dynamic strength adjustment, inter-game and intra-game strength adjustment. For the former, strengths are adjusted between multiple games based on previous game results, while for the latter strengths are adjusted within each game. We present two methods of dynamic strength adjustment (DSA) to showcase how we can predict opponent strengths and adjust accordingly with relative ease. There are many ways to design DSA mechanisms; the presented methods are by no means a comprehensive review of all available methods.

A. Inter-game Strength Adjustment

Inter-game strength adjustment is relatively easy. Namely, the strength index z of a game is adjusted based on the previous game results and the index remains unchanged within the game.

In this section, a simple adjustment method is presented and demonstrated to predict the opponent's strength. The prediction can then be used to set z accordingly. The strength index z is decreased for every win and increased for every loss, both by a small amount Δz. The initial value of z is set to 0. The value Δz is initialized to Δz_{init} and decreased by a discount factor r for each game, capped by a lower bound Δz_{low}.

In our experiments for the method, Δz_{init} is 0.375, approximately equivalent to 100 in Elo rating based on the linear regression in Figure 1, then decreased by a factor of $r = 0.95$ for each game, with $\Delta z_{\text{low}} = 0.03$, approximately equivalent to 8 in Elo rating. In the experiment, 100 games are played against each of the five opponents whose strength indices are $z = 2,\ 1,\ 0,\ -1,\ -2$ for a total of 500 games. The experiment is repeated five times and the following experimental results are based on the average values of the five times.

In Figure 5, each of the five lines indicates the predicted z for each opponent. The result shows that our method can approximately predict opponents' strengths and clearly distinguish five opponents after 10 games. Table III also shows that the averaged win rate for each opponent is within $43\% \sim 54\%$ and the averaged predicted z is very close to the opponent's.

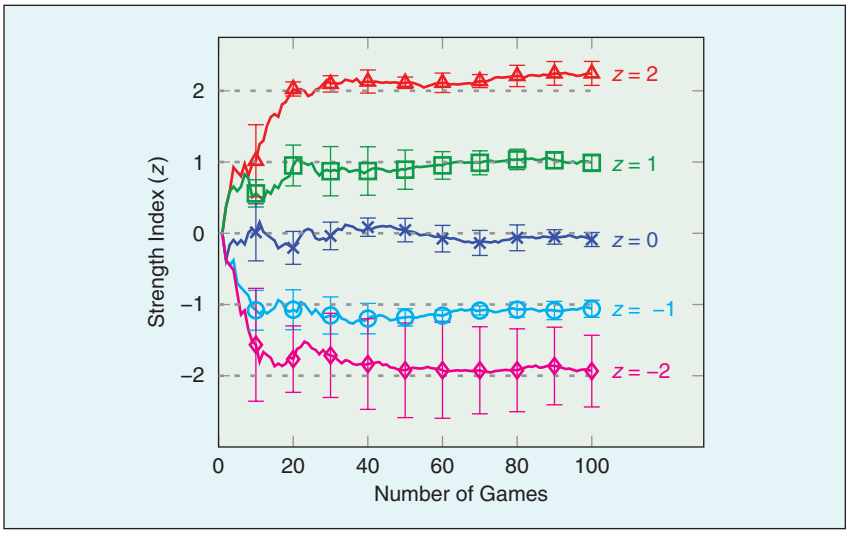


FIGURE 5 Strength index estimation for inter-game SA.

TABLE III Win rate (WR) and average dynamic strength index (Avg. z) against different opponents using inter-game SA.

OPPONENT	$Z = 2$	$Z = 1$	$Z = 0$	$Z = -1$	$Z = -2$
W/O DSA WR	5.6% (±2.9%)	8.8% (±3.6%)	50.0%	78.4% (±5.2%)	87.6% (±4.2%)
INTER-GAME WR	43.4% (±4.4%)	46.6% (±4.5%)	52.0% (±4.5%)	50.0% (±4.5%)	54.8% (±4.5%)
AVG.Z	1.93	0.88	-0.04	−1.06	−1.73

B. Intra-game Strength Adjustment

Intra-game strength adjustment is relatively challenging, given that the algorithm only has one game to predict the opponent's approximate level of play. Players often play inconsistently, mixing objectively good and bad moves within the same game. On the one hand, adjusting by large amounts leads to high variance of program strength. On the other hand, if strengths are adjusted by a small amount, the effects may not be sufficiently obvious.

Our method is as follows. In principle, we still attempt to maintain all moves so that the overall win rate is around 50%. For each move, we first estimate the current win rate W, by using the MCTS win rate of the move with the most simulation counts. The index z is decreased when $W > 50\%$ and increased when $W < 50\%$, both by a value of Δz. The program chooses moves based on the softmax policy proposed earlier.

For stability, Δz is set to be relatively small when W is within a range $(50\% - \epsilon,\ 50\% + \epsilon)$ where ϵ is a user-defined value, say 10%. Namely,

$$\Delta z = \begin{cases} \Delta Z, & \text{if } |W - 50\%| \geq \epsilon \\ \Delta Z \times \dfrac{|W - 50\%|}{\epsilon}, & \text{if } |W - 50\%| < \epsilon. \end{cases} \tag{18}$$

In addition, ΔZ decreases linearly from an initial value, say 0.1, to 0 after a number of moves, say 150 moves. The purpose is to cool down the magnitude of changes as games progress, since the program should have an idea of its opponent's strength by later stages in the game. This cool down mechanism is important because without it, the program will make unreasonable concessions when it is in the lead, or ramp up in strength indefinitely when it is behind.

In the experiment, the above method is used to play against five opponents with strength indices $z = 2,\ 1,\ 0,\ -1,$ and -2. For each opponent, we consider two cases of initial ΔZ, $\Delta Z_{\text{init}} = 0.2$ and 0.1, where for each case, 100 games are played.

Table IV presents the experimental results. The results show that the average z values over 100 games are close to the opponents' strength indices, especially when $\Delta Z = 0.2$. Note that the predicted z for each game is the value z at the end of the game. The standard deviation is high as expected.

The results in Table IV also show that while all the win rates except for $z = 0$ are not around 50%, when compared to the baseline win rates without DSA (as shown in the second row), the overall win rates are closer to 50%. This shows that intra-game DSA can predict opponents' strengths and even out the games. The reason why the win rates are not balanced around 50% despite the predicted z to be more or less accurate, is that the early moves in a game influence the outcome significantly, but the program has yet to observe its opponents' strengths sufficiently at that point.

C. Mixed Dynamic Strength Adjustment (MDSA)

We can expect many different variants for DSA by, say, modifying hyper-parameters for the above methods, or by using hybrids of inter-game and intra-game DSA. For example, we can use intra-game DSA for the first few games in a series to make rapid adjustments to the right strength, then follow up by applying inter-games DSA to arrive at the strength more accurately for the remaining games in the series. A hybrid case is demonstrated in the following experiment: we first apply intra-game DSA to the first two games with $\Delta Z_{\text{init}} = 0.2$, then apply inter-game DSA to the remaining games with the same settings as those in Figure 6. Figure 6 presents the experimental result which shows a fast convergence to opponents' strengths, especially when $z \in \{1, -1, -2\}$. In summary, the above MDSA method uses intra-game SA in the beginning to get a rough strength estimate, then follows up by using inter-game SA to fine tune the player's strength estimation.

V. Real-World Experiments

In this section, we use SSA and DSA to test them against human players of various playing abilities. We applied strength adjustment to three versions of our Go program CGI [24], 3.0, 2.0 and 1.0, which can cover a strength ranging over 2800 Elo ratings, listed as follows from strongest to weakest: professionals, dan rank amateurs, and kyu rank amateurs. For professionals, we use CGI 3.0, consisting of both the policy network and the value network (as is the case with AlphaGo Zero [2]), with settings that follow AlphaGo's [1], where the simulation

TABLE IV Win rate (WR), average z (Avg. z) and standard deviation of z (Std. z) against different opponents using intra-game SA.

OPPONENT		$Z = 2$	$Z = 1$	$Z = 0$	$Z = -1$	$Z = -2$
W/O DSA	WR	5.6%	8.8%	50.0%	78.4%	87.6%
$\Delta z_{\text{init}} = 0.2$	WR	29.0%	38.0%	43.0%	64.0%	71.0%
	AVG.Z	1.85	0.92	−0.10	−1.39	−1.57
	STD.Z	1.82	1.81	1.41	1.73	1.56
$\Delta z_{\text{init}} = 0.1$	WR	15.0%	32.0%	49.0%	73.0%	72.0%
	AVG.Z	1.02	0.75	−0.21	−0.66	−0.96
	STD.Z	0.94	0.90	0.81	0.84	0.85

TABLE V Win Rate (WR) and average dynamic strength index (Avg.z) against different opponents using MDSA.

OPP.	$Z = 2$	$Z = 1$	$Z = 0$	$Z = -1$	$Z = -2$
WR	45.0%	46.0%	46.7%	48.0%	54.3%
AVG.Z	1.69	0.76	−0.10	−1.21	−1.83

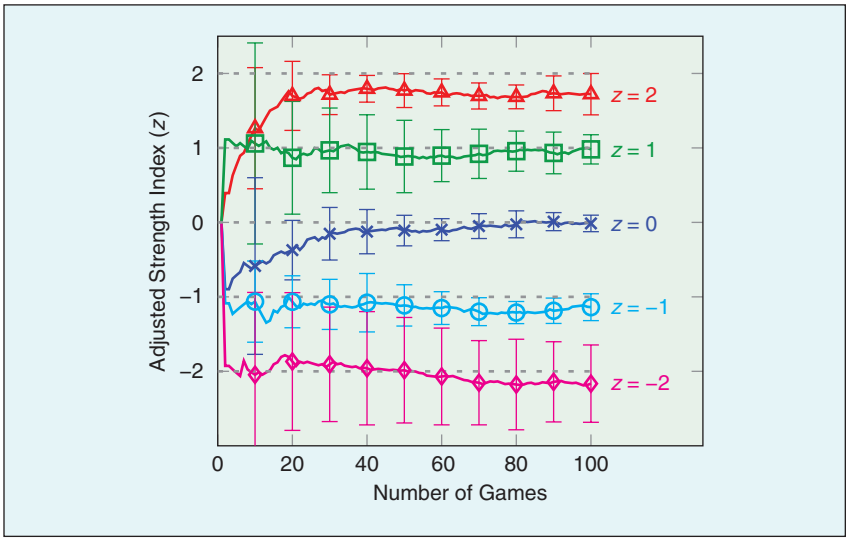


FIGURE 6 Strength index estimation for MDSA.

time was set to 20 seconds. For the amateur level 1 dan to 7 dan, CGI 2.0 uses only the policy network (without the value network), also with settings following AlphaGo's [1], with 4 seconds of simulation time. For amateur 18 kyu to 1 kyu (weaker than the dan ranks), CGI 1.0 uses the traditional minorization-maximization method [22] (without neural networks) with 1 second simulations.

A. Static Strength Adjustment

First, to verify that our strength adjustment is able to provide a steady range of program strengths, we perform the following experiment. Let us assume there is a fair and accurate rating system for a list of human Go players, serving as a baseline system (e.g., the Go rating site [25]), where each player has a rating. As the human players play against a range of strength-adjusted programs, we can obtain a series of game records which form the basis of a separate rating system. If the ratings obtained by this separate rating system is highly correlated with the baseline ratings, we can conclude that the programs have stationary strengths, and that the strengths must range from the weakest human player to the strongest.

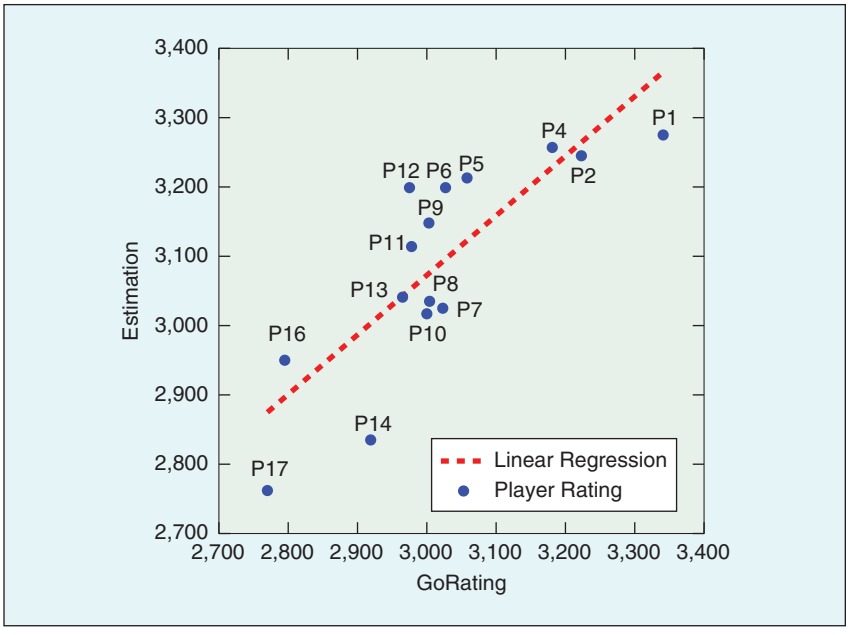


FIGURE 7 SSA HaiFong professional player rating estimation using WHR (see Table VIII for player name references).

TABLE VI Normalized contingency table (row).

ENTERTAINMENT \ PERCEIVED STRENGTH	1	2	3	4	5	6	7	8	9
1	0.10	0.10	0	0	0.20	0	0	0.20	0.40
2	0	0	0.11	0.11	0.11	0.11	0.11	0.22	0.22
3	0	0	0.12	0.12	0.18	0.24	0.24	0.12	0
4	0	0	0.05	0.09	0.41	0.23	0.18	0.05	0
5	0	0	0.05	0.10	0.43	0.33	0.08	0	0.03
6	0	0	0	0.18	0.32	0.24	0.18	0.03	0.06
7	0	0	0.02	0.12	0.24	0.26	0.30	0.04	0.02
8	0	0	0	0	0.13	0.19	0.44	0.19	0.06
9	0	0	0	0.08	0.19	0.22	0.25	0.08	0.17

For this experiment, six different strength indices $z \in \{1.1, 1.4, 1.7, 2.0, 2.5, 3.0\}$ for CGI 3.0 were tested against 15 professional Go players from HaiFong Go Association [26], the biggest Go association in Taiwan. The names of the players are listed in Table VIII. A total of 167 games were played for game record collection, where each player contributed to at least 8 games. Although not enforced, the professional players were instructed to adjust his/her opponent according to their experience with the programs; i.e., pick a stronger one if he/she wins, or vice versa. For the baseline, we used ratings from the most widely used system for professional Go players, the Go Ratings website [25]. The website uses the whole-history rating (WHR) rating system [21] to estimate player strengths. The second set of ratings derived from human-program game records also follow the WHR algorithm.

The results are shown in Figure 7. The correlation between the human-program ratings and the ratings obtained from Go Ratings is high ($R^2 = 0.6661$, note that R^2 is R-squared values). There are a wide variety of reasons that may affect human player performance. For example, for this first experiment, we did not control for the environment in which the professional players played against our program. Their performance may vary depending on if they played against our program in their own home or in at the Go association, which is more professional. Another difference is the timing system used. The players were free to choose their preferred time system, ranging from 1 to 10 minutes of main time; byo-yomi (overtime) is uniformly set to three 30 seconds intervals. Ideally, we should eliminate these factors, but to minimize the disruption to the professional players' training schedule, we had to use the collected records as best as we could for analysis. This may explain outliers such as P14 and P17 in Figure 7.

During these 167 games, we also collected survey responses following the work in [27] on dynamic difficulty adjustment for video games. The survey

asked participants about the perceived strength of the opponent, entertainment level, and any other general comments.

We first annotate general comments before analyzing numerical survey results. In regards to game balancing and perceived strength:

- P4: White (CGI)'s moves at 48 and 50 are interesting, from 58 to 66, white played wonderfully, winning the game at 94.
- P6: This version does not seem to be too strong ($z = 1.1$). This one is stronger ($z = 1.4$).
- P8: It feels like this game ($z = 2.5$) has a sizeable gap in strength (stronger) compared to the previous game ($z = 2$).
- P9: The moves at 20, 24 (CGI) are plays that have not been seen in any human game records.

In terms of quantifiable survey results, we construct a normalized contingency table, as shown in Table VI. The wording on the survey instructs the player to give a perceived strength rating from 1 to 9 (weakest to strongest), relative to her own playing strength. First, we can see that in terms of entertainment value, professional players tend to enjoy opponents that are slightly stronger than themselves (for entertainment scores of 7 to 9). For opponents that are much weaker than the players, the entertainment value is rated worst with no exceptions. In terms of strength, it is interesting to observe that for the strongest relative opponents (perceived strength of 9), there are two extremes. Some find highly challenging opponents to be very entertaining, while others do not enjoy playing against opponents that are far stronger than themselves.

Next, we collect data for amateur players by collaborating with the two largest online Go platforms (in terms of players): Tygem [28] and Fox Weiqi [29] (abbr. FoxWQ for the remainder of this paper). Both websites provide AI developers with toolkits for interfacing upon request. A total of 14 differently strength-adjusted programs (7 each for dan and kyu ranks) were available to play against, for any player who wishes to do so on the two websites. On these platforms, there are options to indicate openly to the community whether an account is AI or human. To minimize irregular play from human opponents, we did not disclose this information. No surveys to human players were conducted.

The results indicate that SSA can maintain a positive strength correlation with the strength index, which enabled professional players to choose an appropriate opponent and provided game balancing, subsequently improving the entertainment value and creating more diverse game play.

Since we do not have a set of reference Go ratings for the online human opponents, we use the dan and kyu ranks on Tygem and FoxWQ as the indicator for actual program strength. The online ranking mechanism (on these platforms) is described as follows. Each new account is given an initial rank; subsequently her rank is adjusted using the most recent n games. For instance, a common setting would be to promote the player's rank by one if she has 12 wins in the most recent 20 games against players with a similar rank, or demote one rank if there were 12 losses. Once a promotion/demotion occurs,

TABLE VII Normalized contingency table (column).

ENTERTAINMENT \ PERCEIVED STRENGTH	1	2	3	4	5	6	7	8	9
1	1.00	1.00	0	0	0.03	0	0	0.13	0.24
2	0	0	0.14	0.04	0.02	0.02	0.02	0.13	0.12
3	0	0	0.29	0.08	0.05	0.07	0.08	0.13	0
4	0	0	0.14	0.08	0.14	0.09	0.08	0.06	0
5	0	0	0.29	0.17	0.27	0.24	0.06	0	0.06
6	0	0	0	0.25	0.17	0.15	0.12	0.06	0.12
7	0	0	0.14	0.25	0.19	0.24	0.31	0.13	0.06
8	0	0	0	0	0.03	0.05	0.14	0.19	0.06
9	0	0	0	0.13	0.11	0.15	0.18	0.19	0.35

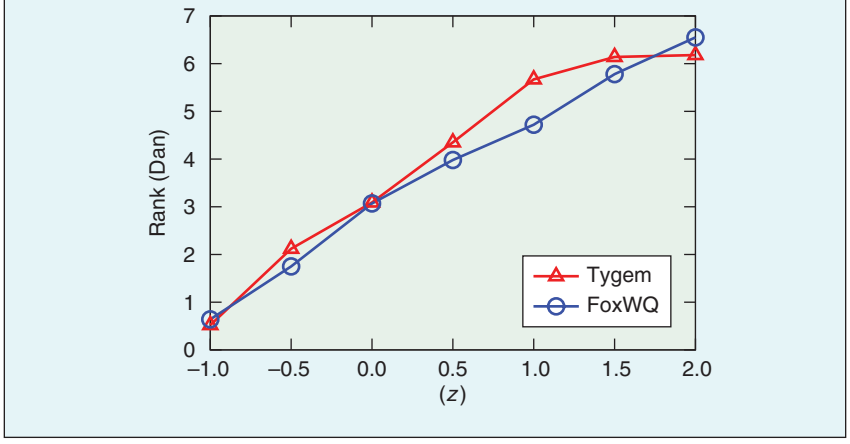


FIGURE 8 Dan rank strength correlation to z for the online Go platforms Tygem and FoxWQ. 0 dan corresponds to 1 kyu.

the "recent game buffer" is cleared. It is worth noting that this ranking mechanism is a dynamic system, in that each player's rank depends on the active player population's overall strength, and that it measures relative instead of absolute strength. Given a large enough number of game samples and a static player population, the rank should converge to a player's ability. While we cannot control for a static player population, each version of our program played at least 400 games, with an overall average of 1,000 games each, of which we use the most recent 100 games for the following analyses.

Figure 8 shows that with different strength indices z, our AI program can obtain a steady distribution of strengths, ranking from 1 dan to 7 dan; z is highly correlated with dan rank ($R^2 = 0.9474$ for Tygem, $R^2 = 0.9931$ for FoxWQ) and a deviation of 0.5012 for Tygem and 0.6083 for FoxWQ. From Figure 9, we can see that the rank of each program stays within a range of ± 1 ranks in the most recent 100 games.

For the lower ranks from 18 kyu to 1 kyu, even with a stable playing ability, it can be difficult to maintain the same rank since beginners introduce more noise in the form of various blunders, and improve strengths more rapidly, etc. As a quick example, from a human perspective, Go can be a strenuous game to play for beginners; it is unreasonable to expect human players to commit the same level of concentration for each game at this level, especially when we did not choose the players that participated in our experiments. For these reasons, the data does not fit as well as the results for professionals or the dan ranks. Figure 10 shows the relationship between z and the programs' kyu ranks on these two websites; the correlation of z to the kyu rank is moderately high ($R^2 = 0.8985$ for Tygem, $R^2 = 0.8675$ for FoxWQ), while the deviation is much higher than the dan rank result (deviation = 1.6671 for Tygem, deviation = 1.6055 for FoxWQ). Despite the large deviation, there is still an incremental relationship between z and the average rank. We can still discriminate between different strength settings to some degree in Figure 11.

In summary, to verify that strength adjustment can be performed to obtain stable, evenly distributed programs, we performed human-program experiments in which humans ranging from professional to amateur kyu rank played against our programs. The results indicate that SSA can maintain a positive strength correlation with the strength index, which enabled professional players to choose an appropriate opponent and provided game balancing, subsequently improving the entertainment value and creating more diverse game play. For the amateur levels, we showed that the strength index is highly correlated with the dan ranks, and moderately correlated with the kyu ranks.

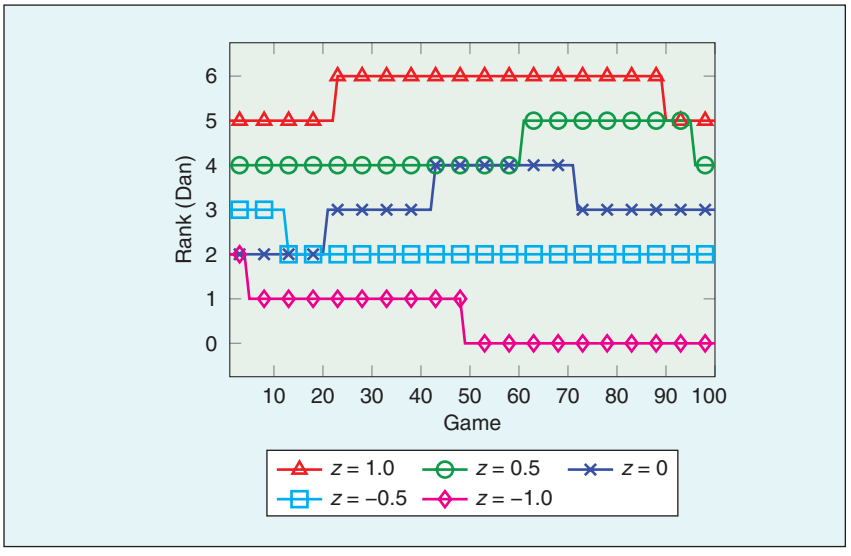


FIGURE 9 Tygem dan rank chart over a period of 100 games.

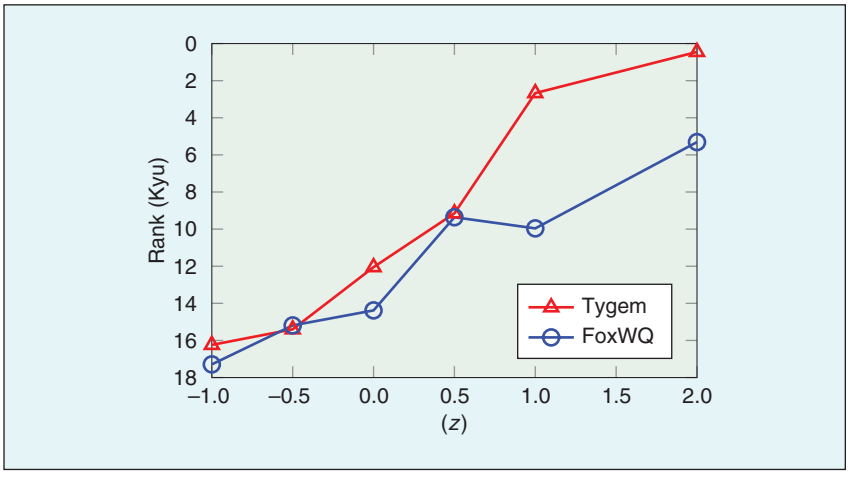


FIGURE 10 Kyu rank strength correlation of z on Tygem and FoxWQ. 0 kyu corresponds to 1 dan.

B. Dynamic Strength Adjustment

We now test how effective our dynamic strength adjustment works to assess the strength of human professional players. For DSA, we directly used MDSA (as in Subsection IV-C) to both professional and amateur players with intra-game SA for the first three games and inter-game SA for the remaining 12 game series in this experiment (15 games in total).

Each professional player was instructed to play 15 games. Different from the experiment in Figure 7, where the participants were asked to manually choose from a list of adjusted

programs, in this experiment, one single version of the program with DSA was used for all participants. The participants were not informed of the adjustment mechanism.

In Figure 12, we show the MDSA estimation process for the participants P1, P8, and P16 (corresponding to the players in Table VIII). The strength relation of the three players is $P1 > P8 > P16$, according to the Go Rating website [25]. The intra-game SA mechanism was used for the first three games, where we can observe a rapid change to the value of z. At the end of the first three games, the z values appear to correspond to each player's strengths ($P1 > P8 > P16$) in order. For the remaining 12 games, inter-game SA was used; in contrast, z adjustment between the games are more stable.

TABLE VIII List of 17 HaiFong Professional Players.

SYMBOL	ENGLISH NAME	GORATING
P1	HSU, HAO-HUNG	3,341
P2	CHEN, CHI-JUI	3,223
P3	CHIEN, CHING-TING	3,202
P4	LIN, LI-HSIANG	3,181
P5	LAI, CHUN-FU	3,058
P6	LI, WEI	3,027
P7	HUANG, SHIH-YUAN	3,023
P8	TSAI, CHENG-WEI	3,004
P9	LU, I-CHUAN	3,003
P10	HSU, CHING-EN	3,000[2]
P11	LIN, YEN-CHENG	2,978
P12	YANG, TZU-HSUAN	2,975
P13	LIN, CHIEH-HAN	2,965
P14	NIU, SHIH-TE	2,919
P15	CHANG, CHIA-HUAN	2,900[2]
P16	PAI, HSIN-HUI	2,795
P17	YU, LI-CHUN	2,770

[2]The player does not have a GoRating on the website. The listed rating is an estimate given by his/her peers.

The Elo rating estimation results of the 12 professional players are shown in Figure 13. Compared with SSA, the DSA mechanism estimated each player's rating with slightly less accuracy ($R^2 = 0.5986$). However, with SSA, the player had to choose the appropriate difficulty level based on their personal experience. Unlike SSA, DSA is able to adjust the program's strength automatically to fit its opponent's strength, resulting in a win rate that is close to 50% (47% ~ 57%).

Table IX lists the z value, intra-z after intra-games and final-z after 15 games for each player. Let the inter-z-diff be intra-z minus final-z. By averaging these absolute values of inter-z-diff, we obtain the value 0.45, which we can think of as the remaining strength adjustment required by inter-game SA for the program to fit to a player's actual strength. Now, let us consider the difference between the final z and the initial z (1.5), indicated in the row of z-diff. If we average these absolute values of z-diff, we obtain the amount of adjustment required only through inter-game SA, which is about 1.08. In other words, by using intra-game SA, we are able to adjust to the player's actual strength with a shorter distance, and follow up by fine-tuning with inter-game SA.

During the inter-game SA process, most players can detect the strength

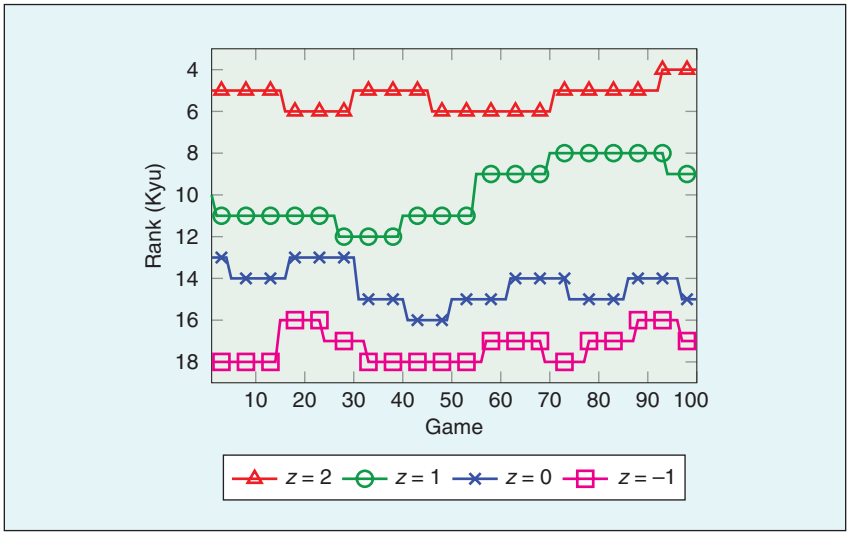


FIGURE 11 FoxWQ kyu rank chart over a period of 100 games.

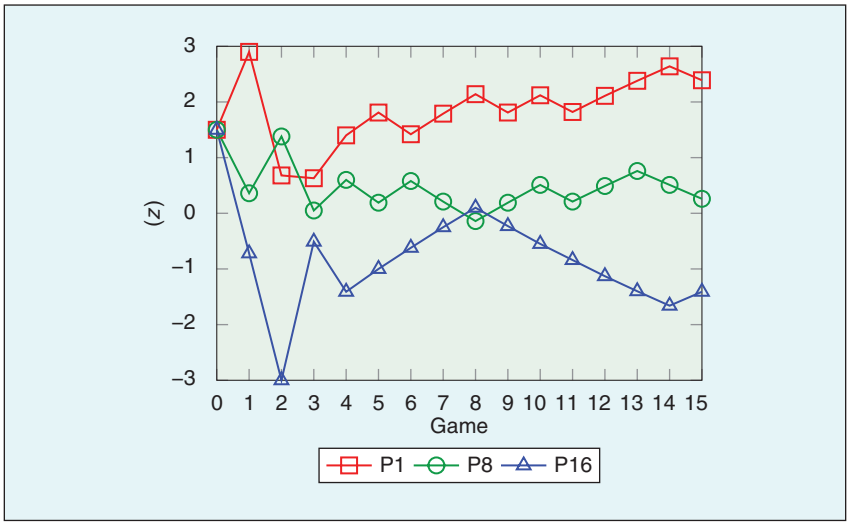


FIGURE 12 MDSA strength estimation of selected players.

In terms of future work, the AlphaZero algorithm has also been successfully applied to other games such as chess and shogi, also achieving super-human level strength [6]. With our approach, we expect to be able to provide a wide range of strength levels for each of these games. We expect our approach to not only impact the Go community, but also the games community at large.

differences game by game. The strongest participant's comments are listed by game in Table X.

Subjectively, players also reported that the programs played to balance each game. One participant reported that "the game felt close all the way to the end, even as I made constant gains". In terms of move diversity, players also reported back the following comments:

- P9: the 20th move made by CGI (W) is refreshing.
- P9: the 11th move by CGI helped broaden my perspective on the game; I think it is not a bad move.
- P12: CGI seems to play more new variations.
- P14: the winning move for CGI occurred at 101; I am rather surprised that a computer was able to come up with something like that.

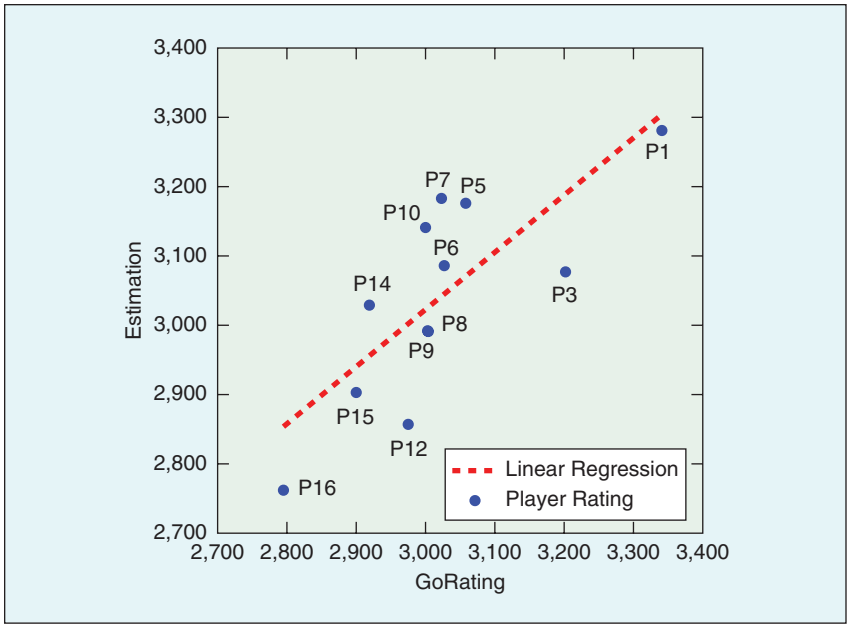


FIGURE 13 MDSA strength estimation of HaiFong professional players.

VI. Conclusion

In this paper, we propose an approach to strength adjustment for MCTS-based game-playing programs. In this approach, we follow a softmax policy [12] with a strength index z to choose moves. Most importantly, this approach uses a threshold ratio R_{th} to filter out low-quality moves i whose simulation counts in MCTS are $N_i \leq N_1 \times R_{\text{th}}$.

In practice, we applied the approach to the Go programs ELF and CGI, demonstrating the ease with which program strengths can be adjusted. The empirical results show the strength covers a range of about 830 Elo ratings with a low linear regression error of 47.95 Elo, with respect to z in the range $[-2, 2]$. To our best knowledge, this result is state-of-the-art in terms of the range of strengths in Elo rating while maintaining a controllable relationship between the strength and a strength index. Another advantage is that the program is still able to play diverse moves despite its adjusted, weaker strength.

Furthermore, we present an in-depth strength analysis for the above empirical results. First, we make the hypothesis that given a position, the strength of move i is proportional to N_i^H. From this hypothesis, the strength ratings of chosen moves are shown to be at worst a fixed value, $H \times \ln R_{\text{th}}$, lower than the best move. This justifies that the move quality is under control, avoiding exceptionally bad moves. In addition, the analysis also shows that the derived strengths are also close to the empirical strengths with regression error 40.45 Elo, and to a logistic function with regression error 10.51 Elo.

With the ease of strength adjustment using z, we introduce three methods to adjust strength dynamically, including

TABLE IX Intra-game SA estimation.

LABEL	P1	P3	P5	P6	P7	P8	P9	P10	P12	P14	P15	P16
INTRA-z	1.40	0.74	1.33	1.40	2.34	0.60	−0.50	1.09	−0.39	−0.02	−0.98	−1.41
FINAL-z	2.39	0.89	1.62	0.96	1.67	0.26	0.27	1.36	−0.72	0.54	−0.38	−1.41
INTER-z-DIFF	−0.99	−0.15	−0.29	0.43	0.67	0.34	−0.77	−0.27	0.33	−0.56	−0.60	0.00
z-DIFF	−0.89	0.61	−0.12	0.54	−0.17	1.24	1.23	0.14	2.22	0.96	1.88	2.91

TABLE X Comments of P1 (HSU, HAO-HUNG).

GAME	RESULT	Z	COMMENT
4	WIN	1.40	I FEEL IT'S (AI) NOT STRONG IN THIS GAME
5	LOSE	1.81	THIS GAME IS OK
6	WIN	1.42	SLIGHTLY WEAK
7	WIN	1.79	MODERATE
8	LOSE	2.14	SLIGHTLY STRONG
9	WIN	1.81	SLIGHTLY WEAK
10	LOSE	2.12	SLIGHTLY STRONG
11	WIN	1.82	WEAK
12	WIN	2.11	SLIGHTLY WEAK
13	WIN	2.38	SLIGHTLY WEAK
14	LOSE	2.64	OK
15	LOSE	2.39	STRONG

inter-game, intra-game and mixed dynamic strength adjustment. The experimental results show that these methods are able to predict the opponents' expected strengths, though the variances can be high.

Finally, we test our methods in real-world cases against professionals, dan rank amateurs and kyu rank amateurs. The players are able to subjectively judge the strength of the programs corresponding to different strength indices. From player surveys, we conclude that professionals tend to feel more entertained when the program is slightly above their strength. When our strength adjusted program is made available online, it can play consistently at evenly-distributed ranks according to each program's preset strength index. As a player performance assessment tool, our program can predict opponent strength accurately in 15 games.

We are currently in the process of applying our strength adjustment mechanism into a so-called lifelong learning system for a variety of games. As a user grows in playing ability, the lifelong learning system is able to keep up with the user and provide appropriate opponents at all levels of play. We have demonstrated in this paper that our method can cover all ranks on Go websites, and even for professional players. The lifelong learning system using our strength adjustment method is therefore suitable for long-term learning, training, and entertainment.

In terms of future work, the AlphaZero algorithm has also been successfully applied to other games such as chess and shogi, also achieving super-human level strength [6]. With our approach, we expect to be able to provide a wide range of strength levels for each of these games. We expect our approach to not only impact the Go community, but also the games community at large.

Acknowledgment

This research is partially supported by the Ministry of Science and Technology (MOST) under Grant Number MOST 107-2634-F-009-011 and MOST 108-2634-F-009-011 through Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan. We would also like to thank the HaiFong Go Association, Tygem, and Fox Weiqi for helping us conduct experiments against human players, and National Center for High-performance Computing (NCHC) for partially supporting computing resource.

References

[1] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. doi: 10.1038/nature16961.
[2] D. Silver et al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, Oct. 2017. doi: 10.1038/nature24270.
[3] Wikipedia, "Fine Art (software)," Dec. 20, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Fine_Art_(software)
[4] G.-C. Pascutto, Leela-Zero GitHub repository, 2018. [Online]. Available: https://github.com/gcp/leela-zero
[5] Y. Tian et al., "ELF OpenGo: An analysis and open reimplementation of AlphaZero," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, 2019, pp. 6244–6253.
[6] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018. doi: 10.1126/science.aar6404.
[7] T. Romstad et al., "Stockfish: A strong open source chess engine," 2019. [Online]. Available: https://stockfishchess.org/
[8] "Results of the 27th world computer shogi championship," Computer Shogi Association, 2019. [Online]. Available: http://www2.computer-shogi.org/wcsc27/index_e.html
[9] R. Hunicke and V. Chapman, "AI for dynamic difficulty adjustment in games," in *Proc. Conf. Association Advancement Artificial Intelligence (AAAI) Workshop Challenges Game AI*, 2004, pp. 91–96.
[10] S. Demediuk, M. Tamassia, W. L. Raffe, F. Zambetta, X. Li, and F. Mueller, "Monte Carlo tree search based algorithms for dynamic difficulty adjustment," in *Proc. IEEE Conf. Comput. Intell. and Games (CIG)*, 2017, pp. 53–59.
[11] S. Demediuk, M. Tamassia, W. L. Raffe, F. Zambetta, F. F. Mueller, and X. Li, "Measuring player skill using dynamic difficulty adjustment," in *Proc. Australasian Computer Science Week Multiconf.*, 2018, p. 41. doi: 10.1145/3167918.3167939.
[12] N. Sephton, P. I. Cowling, and N. H. Slaven, "An experimental study of action selection mechanisms to create an entertaining opponent," in *Proc. IEEE Conf. Comput. Intell. and Games (CIG)*, 2015, pp. 122–129.
[13] P. Paulsen and J. Fürnkranz, "A moderately successful attempt to train chess evaluation functions of different strengths," in *Proc. 27th Int. Conf. Machine Learning (ICML) Workshop Machine Learning and Games*, Haifa, Israel, 2010, p. 114.
[14] H. Arno and P. Morten, "Sensei's library," 2015. [Online]. Available: https://senseis.xmp.net/?RankWorldwideComparison
[15] A. E. Elo, *The Rating of Chess Players, Past and Present*. Arco Pub., 1978.
[16] C.-C. Shih, A.-J. Liu, I.-C. Wu, "2017 CITIC Securities Cup–the 1st world AI Go open," *ICGA J.*, vol. 40, no. 4, pp. 363–368, Jan. 2018. doi: 10.3233/ICG-180076.
[17] I.-C. Wu, T.-R. Wu, A.-J. Liu, H. Guei, and T. Wei, "On strength adjustment for MCTS-based programs," in *Proc. 33th Association Advancement Artificial Intelligence (AAAI) Conf.*, 2019, vol. 33, pp. 1222–1229, doi: 10.1609/aaai.v33i01.33011222.
[18] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Proc. European Conf. Machine Learning*, 2006, pp. 282–293.
[19] M.-L. Cauwet et al., "Depth, balancing, and limits of the Elo model," in *Proc. IEEE Conf. Comput. Intell. and Games (CIG)*, 2015, pp. 376–382.
[20] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, vol. 398. Hoboken, NJ: Wiley, 2013.
[21] R. Coulom, "Whole-history rating: A Bayesian rating system for players of time-varying strength," in *Proc. Int. Conf. Computers and Games*, 2008, pp. 113–124.
[22] R. Coulo, "Computing Elo ratings of move patterns in the game of Go," *ICGA J.*, vol. 30, no. 4, pp. 198–208, June 2007.
[23] D. R. Hunter et al., "MM algorithms for generalized Bradley-Terry models," *Ann. Statist.*, vol. 32, no. 1, pp. 384–406, Feb. 2004. doi: 10.1214/aos/1079120141.
[24] T.-R. Wu et al., "Multilabeled value networks for computer Go," *IEEE Trans. Games*, vol. 10, no. 4, pp. 378–389, July 2018. doi: 10.1109/TG.2018.2852806.
[25] R. Coulom, "The website of Go ratings," 2019. [Online]. Available: https://www.goratings.org/en/
[26] HaiFong Go Association, 2018. [Online]. Available: http://www.haifong.org/
[27] M. P. Silva, V. do Nascimento Silva, and L. Chaimowicz, "Dynamic difficulty adjustment on MOBA games," *Entertain. Comput.*, vol. 18, pp. 103–123, Oct. 2017. doi: 10.1016/j.entcom.2016.10.002.
[28] Tongyang Online Co., Ltd., Tygem, 2019. [Online]. Available: http://www.tygemgo.com/
[29] Tencent, "Fox Weiqi," 2019. [Online]. Available: http://www.foxwq.com/

Conference Calendar

Marley Vellasco
Pontifícia Universidade Católica do Rio de Janeiro, BRAZIL

* Denotes a CIS-Sponsored Conference
Δ Denotes a CIS Technical Co-Sponsored Conference

*** 2020 IEEE Conference on Games (IEEE CoG 2020)**
August 24–27, 2020
Place: Higashiosaka, Japan – virtual
General Co-Chairs: Ruck Thawonmas and Kyung-Joong Kim
Website: http://ieee-cog.org/

Δ 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA CECNSM 2020)
September 25–27, 2020
Place: Corfu, Greece
General Co-Chairs: Michael Dossis and Christos Douligeris
Website: http://hilab.di.ionio.gr/seeda2020/

Δ 2020 International Conference on Process Mining (ICPM 2020)
October 5–8, 2020
Place: Padua, Italy
General Co-Chairs: Massimiliano de Leoni and Alessandro Sperduti
Website: https://icpmconference.org/2020/

*** 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)**
October 6–9, 2020
Place: Sydney, Australia – virtual
General Co-Chairs: Vipin Kumar and Usama Fayyad
Website: http://dsaa2020.dsaa.co

Digital Object Identifier 10.1109/MCI.2020.2998305
Date of current version: 15 July 2020

Δ 2020 Fourth International Conference on Intelligent Computing in Data Sciences (ICDS 2020)
October 21–23, 2020
Place: Fez, Morocco – virtual
General Co-Chairs: Robert Kozma, Chakir Loqman, Mohammed Mestari
Website: http://www.researchnetwork.ma/icds2020/index.html

*** 2020 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)**
October 27–29, 2020
Place: Viña del Mar, Chile – virtual
General Chair: Gonzalo A. Ruz
Website: https://cibcb2020.uai.cl

*** 2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDLEpiRob)**
October 28–30, 2020
Place: Valparaíso, Chile – hybrid
General Co-Chairs: Giulio Sandini and Javier Ruiz-del-Solar
Website: https://cdstc.gitlab.io/icdl-2020/

Δ 7th International Conference on Soft Computing and Machine Intelligence (ISCMI 2020)
November 14–15, 2020
Place: Stockholm, Sweden
General Chair: Suash Deb
Website: http://www.iscmi.us

*** 2020 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2020)**
December 1–4, 2020
Place: Canberra, Australia
General Chair: Hussein Abbass
Website: http://www.ieeessci2020.org/

*** 2021 IEEE Congress on Evolutionary Computation (IEEE CEC 2021)**
June 28–July 1, 2021
Place: Kraków, Poland
General Co-Chairs: Jacek Mańdziuk and Hussein Abbass
Website: https://cec2021.mini.pw.edu.pl

*** 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2021)**
July 11–14, 2021
Place: Luxembourg, Luxembourg
General Co-Chairs: Christian Wagner and Holger Voos
Website: TBA

*** 2021 IEEE International Conference on Development and Learning (ICDL)**
August 23–26, 2021
Place: Beijing, China
General Co-Chairs: TBA
Website: TBA

*** 2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)**
October 13–15, 2021
Place: Melbourne, Australia
General Co-Chairs: TBA
Website: TBA

*** 2021 IEEE Smart World Conference**
October 18–21, 2021
Place: Atlanta, USA
General Co-Chairs: Yi Pan, Rajshekhar Sunderraman, Yanqing Zhang
Website: TBA

*** 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**
November 2–4, 2021
Place: Temuco, Chile
General Co-Chairs: Millaray Curilem and Doris Saez
Website: http://la-cci.org/

*** 2021 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2021)**
December 5–8, 2021
Place: Orlando, FL, USA
General Co-Chairs: Sanaz Mostaghim and Keeley Crockett
Website: TBA

*** 2022 IEEE World Congress on Computational Intelligence (IEEE WCCI 2022)**
July 18–23, 2022
Place: Padua, Italy
General Co-Chairs: Marco Gori and Alessandro Sperduti
Website: TBA

Call for Papers for Journal Special Issues

Special Issue on "Evolutionary Neural Architecture Search and Applications"

Journal: ***IEEE Computational Intelligence Magazine***
Guest Editors: Yanan Sun (ysun@scu.edu.cn), Mengjie Zhang, and Gary G. Yen
Submission Deadline: August 30, 2020
https://yn-sun.github.io/si_enasa.html

Special Issue on "Computational Intelligence for Smart City Services"

Journal: ***IEEE Computational Intelligence Magazine***
Guest Editors: Hao Sheng, Hui Xiong, Zhipeng Cai (zcai@gsu.edu), and Xiuzhen Cheng
Submission Deadline: November 1, 2020
https://research-web.github.io/posts/si/cim2020/

Special Issue on "New Frontiers in Extremely Efficient Reservoir Computing"

Journal: ***IEEE Transactions on Neural Networks and Learning Systems***
Guest Editors: Gouhei Tanaka (gouhei@sat.t.u-tokyo.ac.jp), Claudio Gallicchio, Alessio Micheli, Juan Pablo Ortega, and Akira Hirose
Submission Deadline: September 15, 2020
https://cis.ieee.org/images/files/Documents/call-for-papers/tnnls/CFP_Special_Issue_RC_TNNLS.pdf

Special Issue on "Biologically Learned/Inspired Methods for Sensing, Control and Decision Making"

Journal: ***IEEE Transactions on Neural Networks and Learning Systems***
Guest Editors: Yongduan Song (ydsong@cqu.edu.cn), Jennie Si, Sonya Coleman, and Dermot Kerr
Submission Deadline: October 31, 2020
https://cis.ieee.org/images/files/Documents/call-for-papers/tnnls/CFP_Special_Issue_BLMSCDM_TNNLS.pdf

Special Issue on "Fuzzy Systems Toward Human-Explainable Artificial Intelligence and Their Applications"

Journal: ***IEEE Transactions on Fuzzy Systems***
Guest Editors: Zehong (Jimmy) Cao, Chin-Teng Lin, Yong Deng, and Gerhard-Wilhelm Weber
Submission Deadline: October 31, 2020
https://cis.ieee.org/images/files/Publications/TFS/special-issues/TFS_SI_FSTHEAIA_CFP.pdf

Digital Object Identifier 10.1109/MCI.2020.2998310

IEEE Symposium Series on COMPUTATIONAL INTELLIGENCE

IEEE SSCI 2020 December 1-4, 2020 Canberra Australia

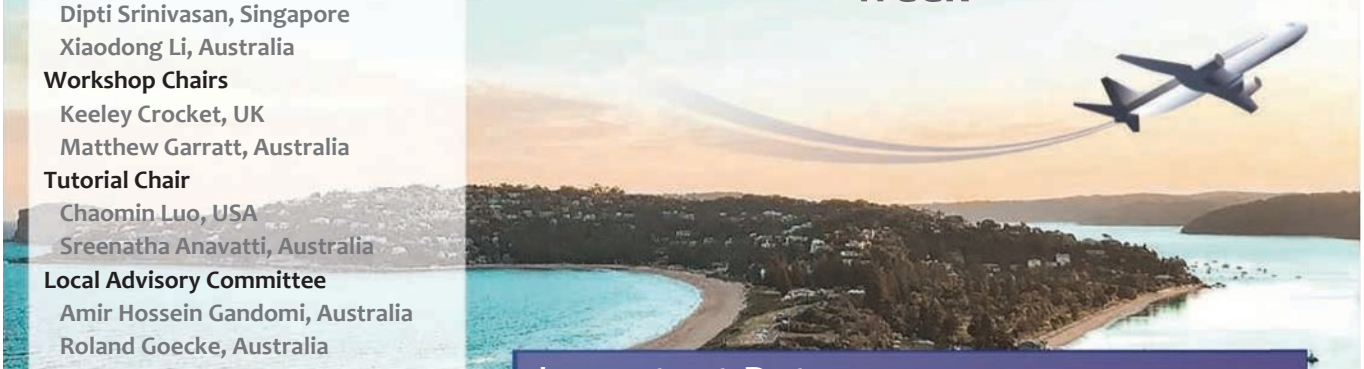
Organising Committee

General Chair
Hussein Abbass, Australia
Program Chair
Carlos A. Coello Coello, Mexico
Conflict of Interest Chair
Hisao Ishibuchi, China
Finance Chair
Kathryn Kasmarik, Australia
Proceedings Chair
Hemant Singh, Australia
Keynote Chair
Kay Chen Tan, Hong Kong
Special Session Chairs
Dipti Srinivasan, Singapore
Xiaodong Li, Australia
Workshop Chairs
Keeley Crocket, UK
Matthew Garratt, Australia
Tutorial Chair
Chaomin Luo, USA
Sreenatha Anavatti, Australia
Local Advisory Committee
Amir Hossein Gandomi, Australia
Roland Goecke, Australia
Dharmendra Sharma, Australia
Local Organizing Chairs
Saber Elsayed, Australia
Markus Wagner, Australia
Cross-event Coordination Chair
George Leu, Australia
Webmaster Chairs
Jiangjun Tang, Australia
Raul Fernandez Rojas, Australia
Whova Chair
Albert Lam, Hong Kong
Publicity Chairs
Harith Al-Sahaf, New Zealand
Min Jiang, China
Rong Qu, UK
Chuan-Kang Ting, Taiwan
Nishchal Verma, India
Min Wang, Australia

45 Technical Symposia

Co-located with the 33rd Australasian Joint Conference on AI

Running during the Canberra Artificial Intelligence (CAI) week



Important Dates

Tutorials, Workshop & Special Session proposals	April 1
Paper submission (no extension)	August 7
Notification to authors	September 4
Camera ready manuscript	September 18

IEEE Computational Intelligence Society

UNSW CANBERRA

NATIONAL CONVENTION CENTRE Canberra

VISIT CANBERRA

CANBERRA CONVENTION BUREAU

BUSINESS EVENTS AUSTRALIA

Digital Object Identifier 10.1109/MCI.2020.2998307