University of Cagliari

Department of Mathematics and Computer Science
Ph.D. Course in Computer Science
Cycle XXXII

Ph.D. Thesis

Machine Learning Models for Educational Platforms

S.S.D. INF/01

Candidate

Mirko Marras
ORCID: 0000-0003-1989-6057

Supervisor

Prof. Gianni Fenu

Ph.D. Coordinator

Prof. Michele Marchesi

Final Examination Academic Year 2018/2019
Thesis Defence: February 2020 Session

Statement of Authorship

I declare that this thesis entitled "Machine Learning Models for Educational Platforms" and the work presented in it are my own. I confirm that:

- this work was done while in candidature for this PhD degree;
- when I consulted the work published by others, this is always clearly attributed;
- when I quoted the work of others, the source is always given;
- I have acknowledged all main sources of help;
- with the exception of the above references, this thesis is entirely my own work;
- appropriate ethics guidelines were followed to conduct this research;
- for work done jointly with others, my contribution is clearly specified.

Abstract

Scaling up education online and onlife is presenting numerous key challenges, such as hardly manageable classes, overwhelming content alternatives, and academic dishonesty while interacting remotely. However, thanks to the wider availability of learning-related data and increasingly higher performance computing, *Artificial Intelligence* has the potential to turn such challenges into an unparalleled opportunity. One of its sub-fields, namely *Machine Learning*, is enabling machines to receive data and learn for themselves, without being programmed with rules. Bringing this intelligent support to education at large scale has a number of advantages, such as avoiding manual error-prone tasks and reducing the chance that learners do any misconduct. Planning, collecting, developing, and predicting become essential steps to make it concrete into real-world education.

This thesis deals with the design, implementation, and evaluation of *Machine Learning* models in the context of online educational platforms deployed at large scale. Constructing and assessing the performance of intelligent models is a crucial step towards increasing reliability and convenience of such an educational medium. The contributions result in large data sets and high-performing models that capitalize on *Natural Language Processing*, *Human Behavior Mining*, and *Machine Perception*. The model decisions aim to support stakeholders over the instructional pipeline, specifically on content categorization, content recommendation, learners' identity verification, and learners' sentiment analysis. Past research in this field often relied on statistical processes hardly applicable at large scale. Through our studies, we explore opportunities and challenges introduced by *Machine Learning* for the above goals, a relevant and timely topic in literature.

Supported by extensive experiments, our work reveals a clear opportunity in combining human and machine sensing for researchers interested in online education. Our findings illustrate the feasibility of designing and assessing *Machine Learning* models for categorization, recommendation, authentication, and sentiment prediction in this research area. Our results provide guidelines on model motivation, data collection, model design, and analysis techniques concerning the above applicative scenarios. Researchers can use our findings to improve data collection on educational platforms, to reduce bias in data and models, to increase model effectiveness, and to increase the reliability of their models, among others. We expect that this thesis can support the adoption of *Machine Learning* models in educational platforms even more, strengthening the role of data as a precious asset. The thesis outputs are publicly available at https://www.mirkomarras.com.

Biography

Mirko Marras was born on *April 19, 1992* in *Iglesias* (Italy). He is a PhD Candidate in Computer Science at the *Department of Mathematics and Computer Science* of the *University of Cagliari* (Italy), advised by prof. *Gianni Fenu*. He received the MSc Degree in Computer Science (cum laude, 18 months) from the same University in 2016. He received the Computer Science Engineering license from the *University of Pisa* (Italy) in 2016.

In 2015, he has been Research Intern at *UnitelCagliari* (Italy) for the "*E-Learning Interactive Opportunities - ELIOS*" project (MIUR, 1.2 ME). In 2017, he spent five months at *EURECAT* (Spain), collaborating with the *Data Science and Big Data Analytics Unit* on the "*DEcentralized Citizen Owned Data Ecosystem DECODE*" project (EU, 5 ME), among others. In 2018, he spent three months within the *Department of Informatics and Systems* at the *University of Las Palmas* (Spain). In 2019, he spent two months at the *Department of Computer Science and Engineering* of the *New York University: Tandon School of Engineering* (U.S.A). Since 2018, he has been contributing to the "*ILEARNTV, Anywhere, Anytime*" project (EU-MIUR, 10 ME). Since 2017, he has been teaching assistant for the "*Computer Networks*" course and thesis assistant for several students in Computer Science.

In 2014, he was recognized as the *Best Third-year Student of the BSc Degree in Computer Science* of the *University of Cagliari* (Italy). In 2016, he was recognized as the *Best MSc Student of the Faculty of Science* and one of the *Top 12 MSc Students* of the same University. In 2018, he got to the podium of the *Call for Visionary Ideas - Education* competition organized by *Nesta Italia*. He has received the *Best Poster Award* at the *European Semantic Web Conference 2017 (ESWC2017)*, the *Demo Honorable Mention* at *The Web Conference 2018 (WWW2018)* and the *Best Paper Award* at *Didamatica 2018*. He has been awarded two *Erasmus+ PlaceDoc* and one *GlobusDoc* grants to spend *11* months abroad, totally.

His research interests focus on machine learning for educational platforms in the context of knowledge-aware systems, recommender systems, biometric systems, and opinion mining systems. He has co-authored papers in top-tier international journals, such as *Pattern Recognition Letters* (Elsevier), *Computers in Human Behavior* (Elsevier), and *IEEE Cloud Computing*. He has given talks and demostrations at several conferences and workshops, such as *TheWebConf 2018*, *ECIR 2019*, *INTERSPEECH 2019*. He has been involving in the program committee of the main Technology-Enhanced Education conferences, such as *AIED*, *EDM*, *ITICSE*, *ICALT*, *UMAP*. He has been also acting as a

reviewer for Q1-level journals, such as *IEEE Transactions on Big Data* and *IEEE Transactions on Image Processing*. He has been part of the local organizing committee of *SITIS 2018*. He has been co-chairing the *Bias 2020 workshop* on algorithmic bias in search and recommendation – with education as a sub-topic - at *ECIR 2020*. He is member of several associations, including *CVPL*, *AIxIA*, *GRIN*, *IEEE*, and *ACM*. For further information, please would you like to visit my personal website http://mirkomarras.com.

Dissemination

The research that contributed to the skills mastered for and the content part of this Ph.D. thesis has resulted from *22* papers fully published in national and international journals and conference proceedings. I would sincerely thank my co-authors for their precious contribution, and such a gratitude would be demonstrated with the adoption of the scientific 'We' throughout the thesis.

I firstly make it clear my contribution. I envisioned the research presented in this thesis and completed the majority of the work. I designed the approaches and chose the research directions. I collected the datasets and was responsible for data analysis. Moreover, I was in charge of the implementation of the related scripts. Finally, I wrote the papers for submission, managed the peer-review pipeline, and subsequently revised them. I collaborated closely with the listed co-authors throughout all stages. Co-authors provided feedback on the approaches, offered technical support, discussed techniques, and contributed to the preparation of the submitted work. Furthermore, I was in charge of the presentation of *9* papers at conferences and workshops (orange-text highlighted). Three papers have received an award or an honorable mention (red-text highlighted).

Exception is made for papers numbered as (i), (iii), (x), (xiv), (xvi), and (xx) as I and Dr. Danilo Dessí equally contributed. Similarly, I and Dr. Silvio Barra put comparable effort for papers numbered as (xi) and (xii).

The detailed references to the produced papers are provided below.

Peer-reviewed Book Chapters

i. Dessí, D., Dragoni, M., Fenu, G., **Marras, M.**, & Reforgiato, D. (2019). *Deep Learning Adaptation with Word Embeddings for Sentiment Analysis on Online Course Reviews.* In: Deep Learning-Based Approaches for Sentiment Analysis, 57-83, Springer. https://doi.org/10.1007/978-981-15-1216-2_3

ii. **Marras, M.**, Marin-Reyes, P., Lorenzo-Navarro, J., Castrillon-Santana, M., & Fenu, G. (2019). *Deep Multi-Biometric Fusion for Audio-Visual User Re-Identification and Verification.* In: Revised Selected Papers of the International Conference on Pattern Recognition Applications and Methods ICPRAM 2019, 11996, 136-157, Springer. https://doi.org/10.1007/978-3-030-40014-9_7

Peer-reviewed Publications in Journals

iii. Dessí, D., Fenu, G., **Marras, M.**, & Reforgiato, D. (2019). *Bridging Learning Analytics and Cognitive Computing for Big Data Classification in Micro-Learning Video Collections.* In: Computers in Human Behavior, 92, 468-477, Elsevier. https://doi.org/10.1016/j.chb.2018.03.004

iv. Fenu, G., & **Marras, M.**. (2018). *Controlling User Access to Cloud-Connected Mobile Applications by Means of Biometrics.* In: IEEE Cloud Computing, 5(4), 47-57, IEEE. https://doi.org/10.1109/MCC.2018.043221014

v. Fenu, G., **Marras, M.**, & Boratto, L. (2018). *A Multi-Biometric System for Continuous Student Authentication in E-Learning Platforms.* In: Pattern Recognition Letters, 113, 83-92, Elsevier. https://doi.org/10.1016/j.patrec.2017.03.027

vi. Fenu, G., **Marras, M.**, & Meles, M. (2017). *Learning Analytics Tool for Usability Assessment in Moodle Environments.* In: Journal of e-Learning and Knowledge Society, 13(3). Italian E-Learning Association. https://www.learntechlib.org/p/180986/

Peer-reviewed Publications in International Conference Proceedings

vii. **Marras, M.**, Korus, P., Memon, N., & Fenu, G. (2019). *Adversarial Optimization for Dictionary Attacks on Speaker Verification.* In: Proceedings of the 20th International Conference of the International Speech Communication Association (INTERSPEECH2019), 2913-2917, ISCA. Talk. http://doi.org/10.21437/Interspeech.2019-2430

viii. **Marras, M.**, Marin-Reyes, P., Lorenzo-Navarro, J., Castrillon-Santana, M., & Fenu, G. (2019). *AveRobot: An Audio-visual Dataset for People Re-identification and Verification in Human-Robot Interaction.* In: Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM2019), 255-265, SciTePress. http://doi.org/10.5220/0007690902550265

ix. Boratto, L., Fenu, G., & **Marras, M.**. (2019). *The Effect of Algorithmic Bias on Recommender Systems for Massive Open Online Courses.* In: Proceedings of the European Conference on Information Retrieval (ECIR2019), 457-472, Springer. Talk. https://doi.org/10.1007/978-3-030-15712-8_30

x. Dessí, D., Dragoni, M., Fenu, G., **Marras, M.**, & Reforgiato, D. (2019). *Evaluating Neural Word Embeddings Created from Online Course Reviews for Sentiment Analysis.* In: Proceedings of the ACM/SIGAPP Symposium On Applied Computing (SAC2019), 2124-2127, Springer. http://doi.org/10.1145/3297280.3297620

xi. Barra, S., **Marras, M.**, & Fenu, G. (2018). *Continuous Authentication on Smartphone by Means of Periocular and Virtual Keystroke.* In: Proceedings of the International Conference on Network and System Security 2018 (NSS2018), 212-220, Springer. https://doi.org/10.1007/978-3-030-02744-5_16

xii. Abate, A. F., Barra, S., Casanova, A., Fenu, G., & **Marras, M.**. (2018). *Iris Quality Assessment: A Statistical Approach for Biometric Security Applications.* In: Proceedings of the 10th International Symposium on Cyberspace Safety and Security 2018 (CSS2018), 270-278. Springer. https://doi.org/10.1007/978-3-030-01689-0_21

xiii. **Marras, M.**, Manca, M., Boratto, L., Fenu, G., & Laniado, D. (2018). *BarcelonaNow: Empowering Citizens with Interactive Dashboards for Urban Data Exploration.* In: Companion of The Web Conference 2018 (WWW2018), 219-222. ACM. Talk. Demo Honorable Mention. https://doi.org/10.1145/3184558.3186983

xiv. Dessí, D., Fenu, G., **Marras, M.**, & Reforgiato, D. (2018). *COCO: Semantic-Enriched Collection of Online Courses at Scale with Experimental Use Cases.* In: Proceedings of the World Conference on Information Systems and Technologies 2018 (WORLDCIST2018), 1386-1396, Springer. Talk. https://doi.org/10.1007/978-3-319-77712-2_133

xv. Fenu, G., & **Marras, M.**. (2017). *Leveraging Continuous Multi-Modal Authentication for Access Control in Mobile Cloud Environments.* In: Proceedings of the International Conference on Image Analysis and Processing 2017 (ICIAP2017), 331-342, Springer. Talk. https://doi.org/10.1007/978-3-319-70742-6_31

xvi. Dessí, D., Fenu, G., **Marras, M.**, & Reforgiato, D. (2017). *Leveraging Cognitive Computing for Multi-Class Classification of E-Learning Videos.* In: Proceedings of the European Semantic Web Conference 2017 (ESWC2017), 21-25, Springer. Best Poster Award. https://doi.org/10.1007/978-3-319-70407-4_5

Peer-reviewed Publications in National Conference Proceedings

xvii. Fenu, G., & **Marras, M.**. (2019). *Servizi Intelligenti per l'Elaborazione di Dati Multi-Biometrici in Piattaforme di Apprendimento.* In: Proceedings of "DIDAttica e informATICA" Conference (DIDAMATICA2019). Talk. https://www.aicanet.it/ documents/10776/2659822/Informatica_per_la_didattica_2019_pa_17.pdf

xviii. Fenu, G., & **Marras, M.**. (2019). *Analisi e Mitigazione del Pregiudizio Algoritmico nei Sistemi di Raccomandazione Didattici.* In: Proceedings of the Italian CINI Conference on Artificial Intelligence (Ital-IA2019). Talk. http://www.ital-ia.it/submission/245/paper

xix. Barra, S., Fenu, G., & **Marras, M.**. (2019). *Sistemi Multi-Biometrici per l'Autenticazione Continua e Trasparente per l'Accesso a Servizi Erogazione Contenuti.* In: Proceedings of the Italian CINI Conference on Artificial Intelligence (Ital-IA2019). Talk. http://www.ital-ia.it/submission/248/paper

xx. Dessí, D., Dragoni, M., Fenu, G., **Marras, M.**, & Reforgiato, D. (2018). *Analisi Emozionale di Recensioni di Corsi Online basata su Deep Learning e Word Embedding.* In: Proceedings of the Italian GARR Conference (GARR2018), 87-91. http://doi.org/10.26314/GARR-Conf18-proceedings-16

xxi. **Marras, M.**, Barra, S., Zucchetti, D., & Chesi, F. (2018). *Abylia: Un Ambiente Digitale per la Scuola in Trasformazione*. In: Proceedings of the Italian GARR Conference (GARR2018), 51-55. http://doi.org/10.26314/GARR-Conf18-proceedings-09

xxii. Fenu, G., **Marras, M.**, Barra, S., Giorgini, F., Zucchetti, D., Chesi, F. (2018). *ILEARNTV: Ecosistema di Conoscenza Condivisa Produzione Collaborativa.* In: Proceedings of "DIDAttica e inforMATICA" Conference (DIDAMATICA2018). Talk. Best Paper Award. http://www.aicanet.it/documents/10776/2101882/didamatica2018_paper_49.pdf/77f26668-d7e8-4355-b747-451035fdec06

Acknowledgements

This thesis has been enriched by the professional and affective support of a range of people who exchanged their precious competence.

First of all, I would express my gratefulness to my advisor, prof. *Gianni Fenu*, for believing in me starting from the time I asked him to assign me a bachelor thesis and for being my role model and mentor. I am thankful to prof. *Diego Reforgiato Recupero*, and I will treasure his approach and practicality. For their intense collaboration and friendship, thank you to dr. *Silvio Barra* and dr. *Danilo Dessì*, with whom I spent great time during our joint research. Heartfelt thanks to dr. *Ludovico Boratto* together with dr. *David Laniado*, dr. *Matteo Manca*, and dr. *Pablo Aragon* from EURECAT. They supported me, guided me, and shared with me exciting moments of research life. Thank you to prof. *Modesto Castrillón-Santana* and prof. *Javier Lorenzo-Navarro* for hosting me into the Department of Informatics and Systems at the University of Las Palmas. A particular acknowledgment is required for dr. *Pedro Marín-Reyes*, as we shared knowledge, skills, and fun in Las Palmas. I am extremely thankful to prof. *Nasir Memon* and dr. *Pawel Korus* from New York University for leading me with expertise during such an amazing research period. Thank you to my parents, *Ferdinando* and *Lorella*, and my girlfriend *Mariarosa*. Thank you to all people who took part of my Ph.D. experience I am not explicitly mentioning here, including relatives, friends, and academics.

Furthermore, I gratefully acknowledge the financial support I have received. Firstly, I recognize the *Sardinia Regional Government* for the financial support of my PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014-2020 - Axis III Education and Training, Thematic Goal 10, Priority of Investment 10ii, Specific Goal 10.5). My research has been additionally and substantially supported by the *Italian Ministry of Education, University and Research* (MIUR) under the "*iLearnTV, Anytime, Anywhere*" Project (DD n.1937 5.6.2014, CUP F74G14000200008 F19G14000910008). I am also thankful to the *University of Cagliari* and the *ECIR Community* for providing me travel grants to conduct research abroad and attend international conferences, respectively. They made it possible to take part in exciting missions, that greatly contributed to enrich me both professionally and humanly.

Thank you all. Grazie a tutti.

Nomenclature

Abbreviations

AI	Artificial Intelligence
AL	Attention Layer
API	Application Programming Interface
CNN	Convolutional Neural Network
CSV	Comma Separated Value
CV	Computer Vision
DL	Deep Learning
DT	Decision Trees
EER	Equal Error Rate
FAR	False Acceptance Rate
FRR	False Rejection Rate
FNN	Feed-forward Neural Network
GPU	Graphic Processing Unit
HRI	Human-Robot Interaction
LMS	Learning Management System
LSTM	Long Short-Term Memory
MOOC	Massive Open Online Course
MAE	Mean Absolute Error
MIUR	Italian Ministry of Education, University and Research
ML	Machine Learning
MP	Machine Perception
MSE	Mean Squared Error
MV	Master Voice
NB	Naive Bayes
nDCG	normalized Discounted Cumulative Gain
NLP	Natural Language Processing
PoI	Person of Interest
RF	Random Forest
RNN	Recurrent Neural Network

ROC	Receiver Operating Characteristic
SA	Sentiment Analysis
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TF-IDF	Term-Frequency-Inverse Document Frequency
t-SNE	t-distributed Stochastic Neighbor Embedding

Latin Expressions

i.e.	id est, ‘that is’
e.g.	exempli gratia, ’for the sake of example’
ergo	’therefore’

Numerical Expressions

{n}K	{n} thousands
{n}M	{n} millions

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In a prosperous digital economy, the development of a successful career lies with the individual's ability to continuously acquire knowledge, gain competencies, and get qualifications. The demand for skilled professionals is fostering job creation and competition amongst companies interested in securing the best candidates [1]. In such a scenario, *Online Education* plays a crucial role as an ecosystem wherein *people* (e.g., learners, teachers, tutors), *content* (e.g., videos, slides), *technology* (e.g., platforms, devices, tools), *culture* (e.g., community sharing), and *strategy* (e.g., business models, learning goals) interact for instilling knowledge to life-long learners, without time or place constraints [2].

Educational and training providers are being encouraged to host online learning by its technical, economic, and operational feasibility. Similarly, end users are taking advantage of the flexibility, accessibility, and costs of learning and teaching online. This win-win situation has led to a proliferation of online initiatives [3]. For instance, in 2019, more than 40 million students and educators have relied on *Google Classroom* [4], *Coursera* [5] has hosted more than 33 million users over 3,600 courses, and *Udemy* [6] has supported 30 million students and 43 million educators within 100,000 courses. Scaling up online education towards these numbers is posing key challenges, such as hardly-manageable classes, overwhelming content alternatives, and academic dishonesty, that often distract students and educators from their main goals: learning and teaching [7].

Capitalizing on large amounts of data and high performance computing, the rapid evolution of *Artificial Intelligence* (AI) and its *Machine Learning* (ML) sub-field is turning the above challenges into an unparalleled opportunity for automated intelligent support. For instance, computerized content tagging could allow teachers to get supported on this error-prone task. AI can provide personalized content by analyzing those available online. Moreover, biometrics could represent a reliable way to ensure academic integrity. Such circumstances have generated interest among researchers, educators, policy makers, and businesses [8]. Evidence of this attention has originated from the investments

made by public and private entities [9]. Moreover, forecasts have announced a growth in the AI-based education market from € 2.6 billion in 2018 to € 7.1 billion by 2023 [10]. Planning, collecting, developing, and predicting are being increasingly recognized as essential steps to make AI real into real-world education, both online and onlife [11].

1.2 Challenges

Recently, there has been an explosion in research laying out new machine learning models and calling attention to applicative issues [12]. This research has greatly expanded our understanding on such a technology, but there has been less work on how it applies on online education at scale. While the state of the art is posing several critical concerns [13], this thesis will focus on research around the following key challenges:

- The learning-related data made publicly available to the research community is currently insufficient to reasonably build and validate machine and deep learning models, which is needed in the design of meaningful educational interventions.
- The huge number of learning resources makes impracticable their manual categorization, while current automated systems based on term frequencies fail to catch semantics. This results in low support to learners and teachers while managing material.
- While current practices in providing automated personalized online education have been proved to be accurate, they are not designed with bias robustness in mind; so, they can introduce bias that might affect millions of users in large-scale education.
- Existing automated systems for checking learners' identities tend to be considered intrusive due to explicit actions asked to users, susceptible to impersonations as mostly based on a single trait, and/or expensive as they often require additional hardware.
- The huge amount of feedback generated by learners after attending courses is hardly manageable by humans and automated models based on plain statistical approaches; consequently, large part of the insights behind this feedback remains uncovered.

As AI-based education will play a relevant role in our everyday life, it becomes imperative to face such critical challenges and provide appropriate support for solving them.

1.3 Contributions

In this thesis, the design, development, and evaluation of machine learning models for education-related services are deeply investigated in the view of empowering learning environments with intelligent algorithms. The focus is on how machines can understand human language and behaviour in learning-related data through Natural Language

Processing (NLP), Computer Vision (CV), and Machine Perception (MP). The proposed contributions consist of targeted data sets and models that support learners and educators over the instructional pipeline, specifically on categorization, recommendation, learners' authentication, and learners' sentiment analysis (*Fig. 1.1*). The design and validation of such new models have been guided by thoughtful continuous feedback from teachers and learners under the "*iLearnTV, Anywhere, Anytime*" research project [14].

Going more into detail, we provide four data sets that expand existing educational data sets in terms of scale, completeness, and comprehensiveness. To the best of our knowledge, these public collections are among the first ones that allow to validate machine and deep learning techniques for educational platforms. On top of them, this thesis introduces novel approaches to assess the contributions of machine learning technology to educational environments. First, differently from existing state-of-the-art baselines, the proposed categorization models mapping micro-learning videos to pre-existing categories exploit semantics rather than term frequency, making more accurate predictions. Second, instead of being optimized only over accuracy and suffering from biased decisions, the proposed recommendation models mitigate the effects of biases in data while still being effective. Third, in contrast to other competitive approaches, the

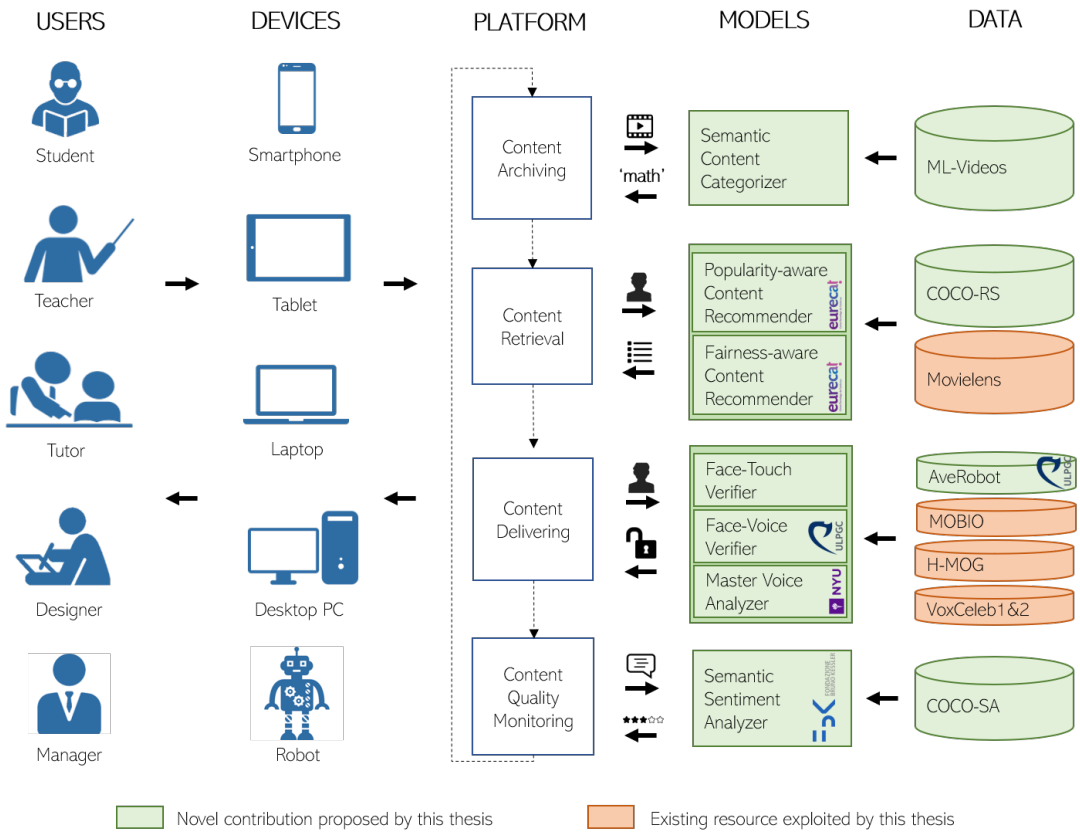


Fig. 1.1: Educational context with a platform empowered with the thesis contributions.

proposed multi-biometric learners' verification models target ubiquitous scenarios and cost-effective transparent recognition. Finally, the proposed sentiment prediction models differ from other opinion mining models as the former fed education-specific text representations into a more effective deep neural network tailored to such an input.

Overall, this thesis provides guidelines, insights, limitations, and future directions to researchers studying machine learning for education. They can apply our findings to improve reliability, to increase effectiveness, and to foster user acceptance. This closes the circle on the thesis goal: improving the understanding on machine learning models for large-scale education, strengthening the role of data as a precious scientific asset.

1.4 Outline

The remainder of this thesis is organised as follows: *Chapter 2* provides a brief introduction to the most representative machine-learning concepts underlying this thesis.

Chapter 3 illustrates our categorization models that map educational videos on pre-existing categories. They integrate speech-to-text methods to get video transcripts, natural language processing techniques to extract semantic concepts and keywords from video transcripts, and big data technologies for scalability. This work has been partially studied jointly with dr. *Danilo Dessí* and prof. *Diego Reforgiato* from *University of Cagliari* (Italy), and published on the "*European Semantic Web Conference*" (ESWC) conference proceedings [15] and the "*Computers in Human Behavior*" (CiHB) journal [16].

Chapter 4 proposes educational recommendation models that mitigate biases existing in data while still being effective. This work has been partially studied jointly with dr. *Ludovico Boratto* from *EURECAT* (Spain), and published on the "*European Conference on Information Retrieval*" (ECIR) [17] and the World Conference on Information Systems and Technologies (WorldCist) conference proceedings [18]. Two extended papers regarding this topic have been also submitted for publication on top-tier journals.

Chapter 5 analyzes existing identity verification approaches in education, and depicts a new multi-biometric framework. Several authentication subsystems including behavioral and physical biometrics have been proposed on top of that. Moreover, to raise awareness on multi-biometrics, we used voice verification as use case to prove a new attack that puts at risk otherwise developed uni-biometrics. The overall multi-biometric framework has been described in a paper published on the "*Pattern Recognition Letters*" journal [19]. The work on touch and face biometrics has been partially studied jointly with dr. *Silvio Barra* from the *University of Cagliari* (Italy), and primarily published on the "*International Conference on Image Analysis and Processing*" (ICIAP) [20] and the "*International Conference on Network and System Security*" (NSS) [21] conference proceedings. Extended works have been published on the "*IEEE Cloud Computing*" (ICC) journal [22]. The work on face and voice biometrics has been partially studied jointly with dr. *Pedro*

Marin-Reyes, prof. *Modesto Castrillon-Santana*, and prof. *Javier Lorenzo-Navarro* from *University of Las Palmas* (Spain), and published on the "*International Conference on Pattern Recognition Applications and Methods*" (ICPRAM) conference proceedings [23]. The extended version has been published on the revised book prepared on top of the same conference [24]. Finally, the work on biometric attacks has been partially studied jointly with dr. *Pawel Korus* and prof. *Nasir Memon* from *New York University* (U.S.A.), and published on the "*The International Speech Communication Association*" (INTERSPEECH) conference proceedings [25].

Chapter 6 proposes our sentiment prediction models that mine learners' opinions, starting from sparse context-specific text representations. This work has been partially studied jointly with dr. *Danilo Dessí* and prof. *Diego Reforgiato Recupero* from *University of Cagliari* (Italy), and dr. *Mauro Dragoni* from *Fondazione "Bruno Kessler"* (Italy), and published on the "*ACM/SIGAPP Symposium on Applied Computing*" (SAC) conference proceedings [26] and the "*Deep-learning Approaches for Sentiment Analysis*" book [27].

Finally, *Chapter 7* offers concluding remarks on the implications of our research and provides several opportunities for future work in this field.

For the sake of reproducibility, code, data, models, demos, and posters accompanying this thesis are made publicly available at https://www.mirkomarras.com.

Chapter 2

Machine Learning Fundamentals

This chapter provides essential context around artificial intelligence, machine learning, and deep learning concepts leveraged by this thesis.

2.1 Historical Background

For a long time, researchers have investigated how to instruct computers in order to reduce the effort for intellectual tasks performed by humans [28]. This field, namely *Artificial Intelligence* (AI), includes *Machine Learning* (ML) and *Deep Learning* (DL), in addition to more traditional methods that often relied on hard-coded rules (Fig. 2.1).

Originally, people working in AI believed that automating human tasks could be achieved by providing to computers a range of rules that describe conditions and commands of action, namely *symbolic AI* [29]. Humans used to input rules and data to be processed according to these rules, and get answers. However, symbolic AI came up to be inflexible to face more intelligent tasks, such as object recognition, content categorization, and machine translation. This opened up to more sophisticated approaches.

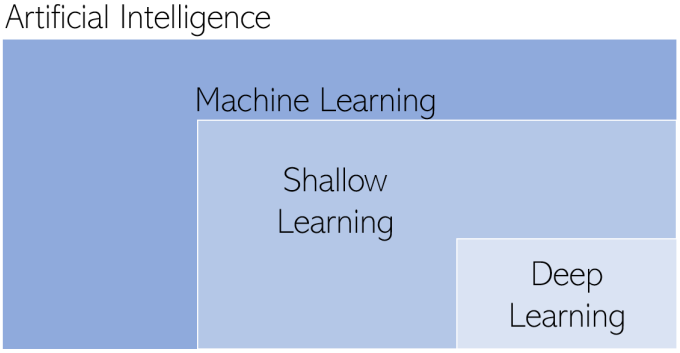


Fig. 2.1: Hierarchy in artificial intelligence, machine learning, deep learning fields.

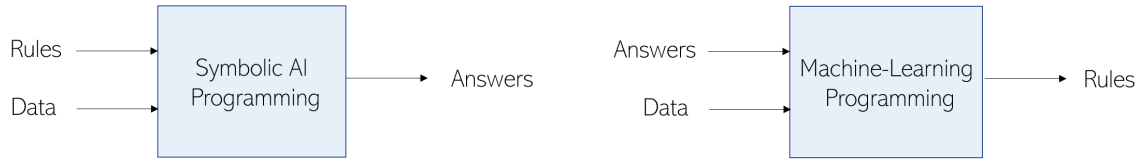


Fig. 2.2: The differences between symbolic AI and machine-learning programming.

In response to this limitation, it was investigated how machine can learn patterns on its own, starting from data relevant for the targeted task [30]. Such a new paradigm, namely *machine learning*, implies that humans input data and answers expected from the machine with this data, and the machine learns patterns that can be applied to data unseen so far (Fig. 2.2). Examples relevant to the task are fed into the machine-learning system, that learns patterns from such a data, making it possible to get complex patterns for solving the task. For instance, a machine-learning system for object classification is fed with human-labelled images from which a set of rules for associating pictures to object labels are learned.

Deep learning is a ML sub-field wherein patterns are learned from data through consecutive manipulation by means of a sequence of stacked layers [31]. It differs from traditional machine learning, namely *shallow learning*, that learns only one or two layers of data representations. The transformation implemented by a deep neural layer is parameterized by its *weights*. Hence, learning means optimizing the weights of all layers, such that the network correctly maps inputs to expected targets. Given the predicted and true targets, the system computes a score through a *loss function* that captures how well the network maps the current samples. The score is then used by the *optimizer* that, through a *Back-propagation* algorithm, arranges the weight values, so that the loss score will be lower in the next iteration. Repeating the loop a sufficient number of times makes it possible to learn weight values that minimize the loss, obtaining a *trained model*.

2.2 Problem Design Branches

Several approaches can be suitable for defining problems faced by a machine-learning system (Fig. 2.3). They can be divided into four broad categories [32], namely:

- *Human-Supervised learning* is based on learning how to map data to known annotations (i.e., labels). Most applications, such as object recognition, speaker verification, sentiment prediction, and language understanding, fall into this category.
- *Self-Supervised learning* implies supervised learning with labels generated from the input data, typically using a heuristic algorithm. For instance, auto-encoders, where the inputs are also the generated targets, make full advantage of self-supervised learning.

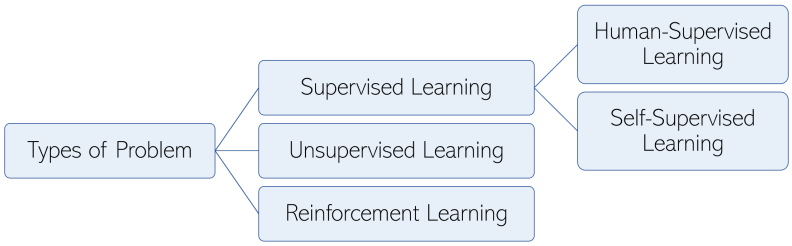


Fig. 2.3: Common classes of problem encountered in machine learning.

- *Unsupervised learning* aims to find interesting transformations of the input data without knowing any or a subset of targets. Sample applications are survival analysis, data denoising, dimensionality reduction and clustering.
- *Reinforcement learning* is based on an agent that receives information about its environment and learns to choose actions that will maximize some reward. For instance, a neural network that outputs game actions to maximize its score can leverage it.

Over this thesis, we mostly focus on supervised learning problems. Therefore, the subsequent sections provide information tailored to this type of problem.

2.3 Experimental Workflow

Common pipelines solving a machine-learning problem include problem definition, data pre-processing, model development and model evaluation (Fig. 2.4).

Problem Definition

This step serves to define the type of problem (e.g., binary classification, multi-class classification, multi-label classification, scalar regression). Identifying the problem type guides the choices made at next steps, such as the model architecture, the loss function, and so on. In addition, inputs and outputs need to be defined and, based on that design choice, proper training data should be retrieved. For instance, learning to classify the sentiment of reviews implies having both reviews and sentiment annotations.

The hypothesis is usually that the outputs can be predicted given the inputs; hence, the retrieved data should be sufficiently informative to learn the relationship between inputs and outputs. Therefore, ML can be used only to get patterns present in training data and recognize what it has been seen there.

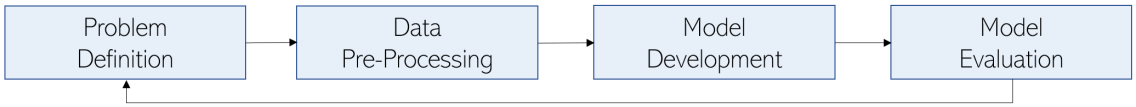


Fig. 2.4: The common pipeline to face a machine-learning problem.

Data Pre-Processing

This step aims to make data more manageable by algorithms through vectorization, normalization, missing values handling, and/or feature extraction:

- *Vectorization.* Inputs and outputs should be numerical vectors, irrespective of the data (e.g., images, text). Turning data into vectors is called *vectorization.* For instance, text can be represented as a list of integers standing for sequences of words. On the other hand, this step is not needed when data is already in numerical form.
- *Normalization.* It should be noted that feeding into a ML model data that takes large values or is heterogeneous, as it can prevent the model from converging. To make learning easier, data should have values in [0,1] range. For instance, image data encoded as integers in range [0,255] is usually cast to float and divided by 255, so that they become float values in [0,1] range. Similarly, when predicting user's identities, each feature could be normalized to have a standard deviation of 1 and a mean of 0.
- *Missing Values Handling.* If there could missing values when predicting through a ML model, it is generally a good practice to simulate such a situation also during model training. To this end, while training, missing values as 0 could be introduced by copying some training samples and dropping some features that may become missing while predicting. In this way, the model can learn that the value 0 means missing data and starts ignoring the value.
- *Feature Extraction.* Using human knowledge about the data and about the ML algorithm can make the algorithm work better. Feature extraction is usually adopted by shallow algorithms not having hypothesis spaces rich enough to learn useful features by themselves. For instance, in speaker verification, inputs for neural networks are typically based on pre-processed data, such as spectrograms and filterbanks extracted from the raw audio. Modern DL is making it possible to fed raw data and let neural networks extract useful patterns from it.

Model Development

The goal at this step is to develop a ML model able to solve the original task [33]. Three key choices to build the model should be considered: (i) model architecture that should learn meaningful data representations, (ii) differentiable optimization function

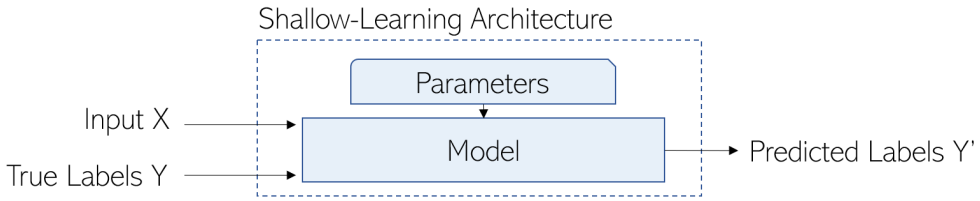


Fig. 2.5: The common components of a shallow-learning model.

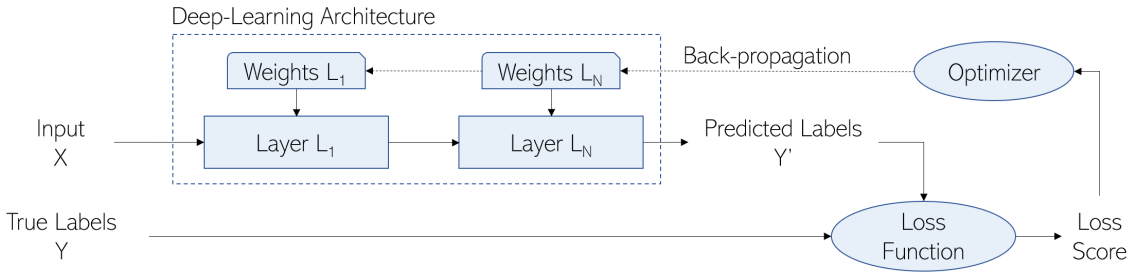


Fig. 2.6: The common components of a deep-learning model.

that should match the type of problem, (iii) optimization configuration that should support the model in minimizing the objective function.

Some shallow-learning models structured as depicted in Fig. 2.5 are described below.

- *Decision Trees* (DTs) [34] predict the value of a target variable by learning decision rules from input features. The model has a root node containing all data features of the training set. Then, the root node is split into several children according to a given criteria. This process recursively continues on children until no nodes to be split exist.
- *Support Vector Machines* (SVMs) [35] map each training data sample to a point in a N-dimensional space, where N is the number of features and the value of each feature is the value of a particular coordinate. Then, it finds the set of hyper-planes that better differentiate the points based on the targets. A linear combination of vectors determines the location of the decision boundaries producing the best separation.
- *Random Forest* (RF) [36] is a meta estimator that (i) fits a number of decision tree classifiers on various random data sub-samples and (ii) uses averaging to improve the predictive accuracy and to control over-fitting. Each decision tree is a weak classifier, while all the decision trees combined together aim to be a stronger classifier.

Some deep-learning models structured as depicted in Fig. 2.6 are described below [37].

- *Feed-forward Neural Networks* (FNN) were one of the first components applied to learn from data using DL [38, 39]. One or more levels of nodes, namely *perceptrons*, are randomly joined by weighted connections in a many-to-many fashion. These networks were historically thought in order to simulate a biological model where nodes are neu-

rons and links between them represent synapses. On the basis of the input values fed into the network, nodes of a certain level can be activated and their signal is broadcasted to the subsequent level. In order to activate nodes of a subsequent level, the signal generated at a level is weighted and must be greater than a given threshold.

- *Recurrent Neural Networks* (RNN) are tailored for processing data as a sequence [40, 41]. In contrast to FNNs, RNNs have cyclic connections among nodes of distinct levels. Recurrent connections connect past data with the one that is currently being processed, simulating a state memory. The forward pass is similar to FNN forward pass. The difference is that the activation of a node depends on both the current input and the previous status of the hidden layers. This workflow is useful when data presents patterns from the past to the future. As an extension, Bidirectional RNNs (BiRNNs) present the training data forwards and backwards to two hidden RNNs combined into a common output layer, making it possible to find patterns from both past and future data [42].

- *Long Short-Term Memory* (LSTM) extends RNNs by employing recurrent connections and adding memory blocks in their recurrent hidden layers [43, 44]. These memory blocks save the current temporal state and make it possible to learn temporal observations hidden in the data. Using memory blocks allows to relate the current data being processed with the data processed long before, solving the problem experienced by common RNNs. For this reason, LSTMs have been proved to have a positive impact on sequence prediction tasks. Bidirectional layers using two hidden LSTMs can be leveraged to process data both forward and backward.

- *Convolutional Neural Networks* (CNNs) perform filtering operations on the nodes of a layer, abstracting and selecting only meaningful nodes. Such networks have been historically applied in Computer Vision [45, 46]. Hence, they are not directly applicable on texts, as the text should be vectorized before applying convolutional filters to them. Each filter is composed by a kernel that slides on the vector representation and repeats the same function on each element until all vectors are covered.

- *Attention Layers* (ALs) serve to orient perception as well as memory access [47, 48]. They filter the perceptions that can be stored in memory, and filter them again on a second pass when retrieved from memory. This assumption makes it possible to solve several limits, especially in NLP. For instance, traditional word vectors presume that a word's meaning is relatively stable across sentences, but this is not always the case (e.g., lit as an adjective that describes something burning or as an abbreviation for literature). Moreover, ALs learn how to relate an input sequence to the output of the model in order to pay selective attention on more relevant input features. For example, in order to reflect the relation between inputs and outputs, an AL may compute an arithmetic mean of results of various layers according to a certain relevance.

- Other layers can be integrated to fine-tune the performance of a model. For instance, *Embedding Layers* turn positive integers into dense vectors of fixed size chosen from a pre-initialized matrix [49, 50]. *Noise Layers* usually avoid model over-fitting by modi-

fying a fraction of input or adding/subtracting values following a predefined distribution (e.g., Gaussian) [51]. *Dropout Layers* may be seen as a particular type of a noise layer that assign the value 0 to a randomly chosen fraction of its input data [52]. *Dense Layers* are densely-connected layers used to map large unit inputs in a few unit results. For example, it may be used to map nodes in a few classes outputted by the model.

As part of the optimization for DL, the error for the current state of the model must be estimated repeatedly. This requires the choice of an error function, namely a *loss function*. Such a function is used to estimate the loss of the model, so that the weights can be updated to reduce the loss on the next evaluation. Based on the type of ML problem, there are several loss functions to choose from. For instance, for regression problems, common loss functions are *Mean Squared Error*, *Mean Squared Logarithmic Error*, and *Mean Absolute Error*; for binary supervised classification, *Binary Cross-Entropy*, *Hinge Loss*, and *Squared Hinge Loss* are usually adopted; for multi-class classification, common solutions are *Multi-Class Cross-Entropy*, *Sparse Multi-Class Cross-Entropy*, and *Kullback Leibler Divergence*. Please refer to [53] for further details on each function.

Finally, within DL, an *optimizer* is integrated to update the weights and minimize the loss function. The loss function is used by the optimizer to move towards the right direction to reach the global minimum. Common optimizers include *Root Mean Square Propagation (RMSProp)* [54] and *Adaptive Moment Estimation (ADAM)* [55].

Model Evaluation

Evaluating a model always needs to subdivide the available data into (i) a *training set*, (ii) a *validation set*, and a (iii) *test set*. During training, the model is fed with the training set and evaluated on the validation set. The latter data is used because developing a model usually involves tuning its configuration (e.g., choosing the number of layers or the size of the layers). Such a tuning can be achieved by using as a feedback the performance of the model on the validation set. To test performance on unseen data, a completely different data set is used to evaluate the model: the test set. The common procedures for splitting data are provided in what follows (Fig. 2.7) [56]:

- *Held-Out Validation*. A subset of the data is set apart for testing, typically around 10% and 20% of the whole dataset. The model is trained on the rest of the data, and its performance are evaluated on the test set. However, if little data is available, then the validation and test sets may contain too few samples to be statistically representative, preventing the validity of the experimental results.
- *k-Fold Validation*. The data is split into K equal-sized partitions. An independent instance of the model is trained on K−1 partitions, and evaluated on partition i. The process is repeated K times, with a different partition i as a test set. The final metrics are averaged to obtain the final score. This might solve issues related to significant variance on final metrics over different train-test split.

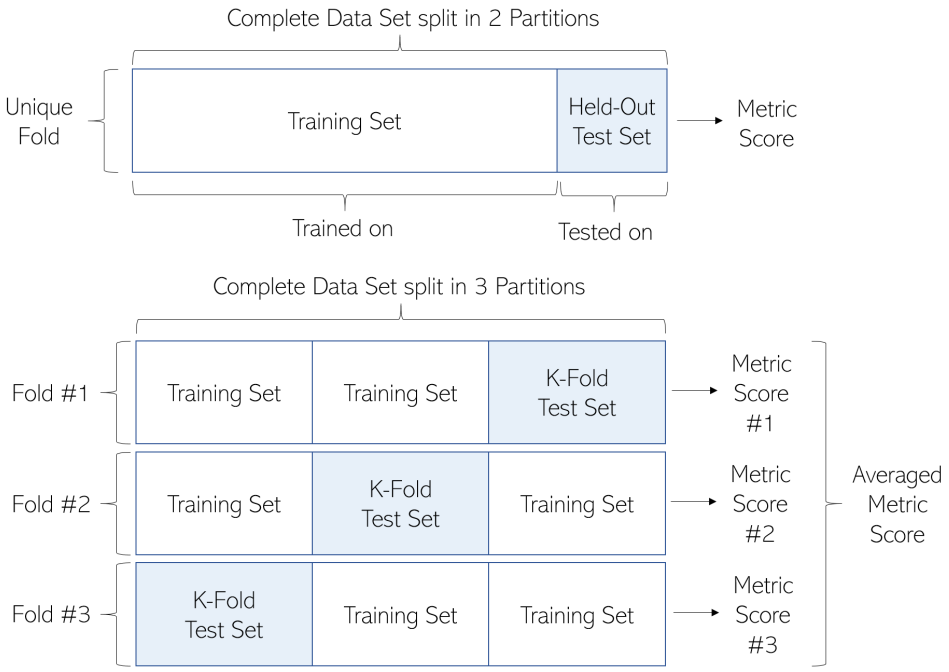


Fig. 2.7: Common ML evaluation protocols: held-out (top) and k-fold (bottom).

After splitting the data, the metrics for measuring success should be defined. The metric should be directly aligned with the high-level goals, such as the success of the business. For balanced-classification problems, where every class is equally likely, *Accuracy* and area under the *Receiver Operating Characteristic* curve (ROC) are common metrics. For class-imbalanced problems, *Precision* and *Recall* are usually used. For ranking problems or multi-label classification, *Mean Average Precision* is generally adopted. In some cases, custom metrics are defined. Such metrics represent the output of the protocol.

As the universal tension in ML is between optimization and generalization, the ideal model is the one that stands right at the border between under-fitting and over-fitting. Hence, all the pipeline steps should be repeated until the model closely reaches this goal.

Chapter 3

Machine Learning Models for Content Categorization

Research Highlights
- *SVM+SGD* trained on *Concepts* is the most efficient model.
- *SVM* trained on *Keywords+Concepts* achieves the highest accuracy.
- *Keywords* are more representative than *Concepts* for prediction.
- Models get comparable performance over *first/second-level* categories.

3.1 Introduction

Micro-learning videos embedded into *online courses* represent one of the most powerful medium to deliver knowledge to learners in small chunks over time. This paradigm promises to have a solid success rate, as proved by recent practical experience [57]. In contrast to traditional educational models, micro learning lasting from 5 up to 15 minutes has a positive impact to knowledge acquisition [58], as it ensures higher *flexibility* and better fits with the *constraints of human brain* with respect to attention span [59, 60].

With the ever-increasing number of micro-learning videos, managing large video collections is turning into a challenge for educators. Human annotation and archiving is time-consuming and non cost-effective. On the other hand, searching videos given specific categories of interest is becoming harder for learners. This has raised the need of tools powered by intelligent methods for effective and efficient video classification. Recent studies tend to model the problem as a text classification task which automatically assigns one category from a set of predefined ones to a video based on its transcript (i.e. a written version of the content presented by the speaker) [61, 62]. These works focused on traditional long face-to-face video lessons which provide a lot of textual information in comparison with the one available in micro-learning videos. Furthermore, these

approaches usually map video transcripts using *Term-Frequency-Inverse Document Frequencies* (TF-IDF). Text documents are modeled as a set of term frequencies, regardless the position of words in the document or semantic links with other words. It follows that a huge amount of the knowledge derived from the text is lost.

Emerging *Semantic Web* resources and techniques have been recently combined with *Data Mining* and *Knowledge Discovery* methods in order to perform analysis and obtain useful insights out of the data [63]. Cutting-edge cognitive computing systems, such as *IBM Watson*[1] and *Microsoft Cognitive Services*[2], can extract concepts, emotions, entities, keywords, and relations from unstructured text, and use advanced machine learning algorithms to derive analytics, generate predictions and hypothesis in a scalable way. Therefore, they can offer *Cognitive Computing potential*, so that the knowledge lost by the existing approaches can be recovered and enriched.

In this chapter, we are interested in supporting the development of tools powered by Cognitive Computing to investigate: (i) how we can extract and merge features from micro-learning videos to improve their representation in the eyes of machine-learning algorithms, and (ii) which machine learning algorithm is best at taking advantage of such features in terms of effectiveness and efficiency for micro-learning video classification. To this end, we analyze micro-learning video collections through a fully-automated pipeline. More precisely, we propose an efficient and effective approach to classify a collection of educational videos on pre-existing categories, capitalizing on (i) a *Speech-to-Text* tool to get video transcripts, (ii) cutting-edge *Natural Language Processing* and Cognitive Computing tools to extract semantic concepts and keywords for their representation, and (iii) *Apache Spark* as Big Data technology for scalability. Several classifiers have been trained on feature vectors extracted by Cognitive Computing tools on a data set we collected from *Coursera*, namely *ML-Videos*. Considering the experimental results, our approach promises to improve micro-learning video classification performance.

The contribution of this chapter is threefold:

- We arranged a *large-scale data set* composed by features extracted from 10,328 videos downloaded from *Coursera*. They are pre-annotated with 7 first-level categories and 34 second-level categories based on the course wherein they are embedded.
- We propose an *automated approach* for classifying micro-learning videos, capitalizing on Cognitive Computing for feature extraction and Big Data technologies for fast computation, going over frequency-based methods, with several practical implications.
- We provide an *extensive evaluation* in terms of effectiveness and efficiency for micro-learning classification, comparing our approach with different combinations of feature type and classification algorithm. This makes it possible to assess which one is the best at taking advantage of the peculiarities of micro-learning videos.

[1] https://www.ibm.com/watson/

[2] https://www.microsoft.com/cognitive-services/en-us

The rest of this chapter is structured as follows: Section 3.2 describes the most representative techniques for content archiving, going deeply on those tested on video lessons. Then, Section 3.3 formalizes the problem we seek to investigate. Section 3.4 introduces the micro-learning data set we collected, while Section 3.5 describes and evaluates the ML models we trained on such a data. Finally, Section 3.6 concludes the chapter.

3.2 Related Work

3.2.1 Content Categorization Techniques

The term *categorization* is defined as the procedure to select the most appropriate category for a resource from a predefined taxonomy. Indexing multimedia content by classifying it is a task required to organize resources so that they can be quickly retrieved.

Several researchers have tried to automatize *multimedia content classification* through accurate, non-time-consuming machine-learning systems [64, 65, 66, 67]. Common solutions adopted pre-existing classification algorithms successfully fed with textual data, such as *Support Vector Machine* (SVM) [35, 49, 68], both in its *C4.5* [69] and *Stochastic Gradient Descent* (SGD) variant [70], *Decision Trees* (DTs) [34], and *Random Forests* (RFs) [36]. However, the usual audio-visual characteristics of a video make video classification harder with respect to plain text classification. Thus, researchers tended to integrate *video metadata* [71] and *video content information*, such as visual frame sequences [72, 73], audio tracks [74], transcripts [75, 76], or multiple combinations of them [77, 78].

In spite of good results obtained using low-level features, emerging *semantic-based alternatives* showed a larger potential. They have been focused on higher level features that model the content of a video. In [79], the authors presented a novel system for content understanding and semantic search on a video collection based on low and high level visual features. In a similar way, [80] investigated the problem of detecting or classifying single video-clips by means of the event occurring in them, which is defined by a complex collection of elements including people, objects, actions, and relations among them. The authors in [81] proposed to generate categories for videos by exploiting *Automatic Speech Recognition* (ASR) and *Optical Character Recognition* (OCR), directly.

However, we point out that the *video type is a relevant characteristic to better understand video content*, and *the information leveraged by a reliable analysis should be selected based on the way knowledge is mainly transmitted.* As micro-learning videos mainly convey knowledge through the instructor's voice, we mostly relate to classification approaches that use *video transcripts* as a source of information. In one of the most representative works, the authors in [75] used the transcripts to improve sentiment classification from general videos. Besides, [76] proposed various video navigation techniques in educational context through *keyword clouds* computed from the transcripts using *TF-IDF*.

3.2.2 Content Categorization in Education

Several classification tasks have been solved by machine-learning models, but there is a limited research on how they perform within the educational domain. For instance, the authors in [62] used *Latent Semantic Allocation* to find and represent topics conveyed by a video lecture. The method demonstrated that the video content allows to achieve better performance than the information extracted from title and tags associated to the video itself. The authors in [82] used knowledge and relationships extracted from *Wikipedia* as a resource to match videos with the most suitable category based on a pre-defined taxonomy. Other classification approaches made use of NLP either directly on the accompanying audio transcript [83] or extracted automatically from the lecture images employing OCR or via speech-to-text [84, 85, 86]. Similarly, the authors in [87] proposed a framework for classifying educational videos from their metadata. The authors in [88] later applied shallow machine learning techniques.

The analysis of the video transcript is an attractive way to be exploited as transcripts represent the richest channel of information for video lessons. However, the existing works that manipulate transcripts focused on long face-to-face video lessons that usually last more than one hour. It follows that the resulting transcripts include a huge amount of words than the ones extracted from micro-learning videos, making classification different at least from the perspective of the available information. Moreover, no existing data set targets the micro-learning scenario, so it exists a research gap that impedes to develop and test solutions for such a learning approach. Furthermore, most of the approaches modeled each transcript as a set of term frequencies, regardless the position of words in the text or semantic links with other words. It follows that a lot of the knowledge derived from the text is lost. Consequently, we first seek to fill the gap in micro-learning video data sets by collecting resources from existing educational platforms. Then, we investigate how higher-level features extracted from transcripts can improve micro-learning classification, enriching *TF-IDF* features with semantics. We finally assess how existing classification algorithms work when fed with these features.

3.3 Problem Formalization

The micro-learning video classification problem can be formalized as follows. We consider N tuples $\mathbb{D} = (x_i, y_i)_{i=1}^{N}$ where each $x_i \subset \mathbb{X}^*$ corresponds to a video transcript of unknown length composed by words from a dictionary $\mathbb{X}$, and each $y_i \subset \mathbb{Y}$ corresponds to the category conveyed by the video from a pre-defined set of categories (e.g. *math*, *physics*). We consider an explicit feature extraction step which produces fixed-length representations in $\mathbb{F} \subset \mathbb{R}^e$ (e.g., TF-IDF). We denote the stage as $\mathscr{F} : \mathbb{X} \rightarrow \mathbb{F}$. Given a feature vector $f \subset \mathbb{F}$, we seek to obtain a classifier $\mathscr{G}_\theta$ that correctly maps f into y. Namely, we want to find a function $\mathbb{G}_\theta : \mathbb{F} \rightarrow \mathbb{Y}$ that outputs the category of the

video transcript x whose feature vector is $f = \mathscr{F}(x)$. Finding such a classifier becomes an optimization problem, which aims to maximize the following objective function:

$$\mathscr{G}_\theta = \operatorname{argmax} \mathop{\mathbb{E}}_{(x,y)\in\mathbb{D}} [\mathscr{G}_\theta(\mathscr{F}(x)) = y] \tag{3.1}$$

In other words, we aim to maximize the cases where the classifier predicts the correct category of a video based on the features extracted from its transcript.

3.4 The Proposed ML-Videos Data Set

Before designing a tailored classification approach, we filled the gap in micro-learning video data sets by collecting resources from existing educational platforms. The collected data set, namely *ML-Videos*, contains over 10,000 micro-learning videos from more than 500 instructors, extracted from courses delivered on *Coursera*. The videos span a wide range of different fields, including maths, physics, economics and so on. Each video included in the data set is shot in different single-speaker controlled environments, typically professional recording studios. To the best of our knowledge, there exists no other benchmark public data set that includes micro-learning videos. To position this data set in the literature, Table 3.1 summarises other representative video data sets whose transcripts have been used for classification. Besides lacking micro-learning conditions, most of them have been labelled with few categories and included longer transcripts.

3.4.1 Collection Methodology

This section describes our *multi-stage approach* for collecting a large micro-learning video data set from existing large-scale learning platforms. The collection pipeline is summarised in Fig. 3.1, and key stages are discussed in the following paragraphs.

Data Set	Videos	Public?	Context	Levels	Categories
20NG [64]	20,000	Yes	News	2	26
NSL-KDD [65]	25,192	No	Medicine	1	23
Youtube [71]	182	No	Miscellaneous	1	3
TRECVID [79]	25,000	Yes	Events	1	20
MCG-WEB [82]	80,031	Yes	Miscellaneous	1	493
KHAN [83]	3,000	Yes	Classroom Education	1	3
GoogleI/O [89]	209	Yes	Classroom Education	1	5
Wikipedia [90]	3,000	No	Classroom Education	1	7
ML-Videos	**10,328**	**Yes**	**Micro Learning**	**2**	**41**

Table 3.1: Representative data sets for video classification based on transcripts.

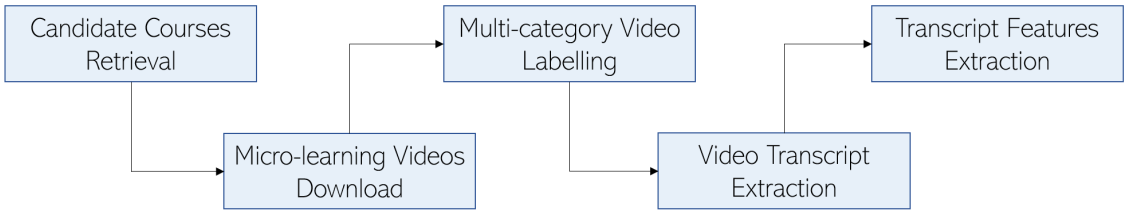


Fig. 3.1: Collection pipeline for ML-Videos data set.

1. **Candidate Courses Retrieval**: the first stage is to obtain a list of courses from *Coursera*. We start from the list of courses that are returned by *Coursera APIs*[3], which is based on an intersection of the most searched courses in the platform, and the courses that are freely available. This list, dumped in *March 2017*, contains 10,352 courses, ranging from economics and computer science to history and art.

2. **Micro-learning Videos Download**: the videos for each of the considered courses are automatically downloaded using *Coursera-dl*[4], a Python script that can be used to download lecture resources (e.g., videos, ppt) from *Coursera* classes. The script is instructed to create a folder for each course and save all the related videos within it. Each video is named with its sequence ID and its title, separated by an underscore.

3. **Multi-category Video Labelling**: first-level and second-level category labels are crawled from *Coursera APIs* for all the courses in the data set, building a two-level taxonomy. Category labels are obtained for all but 24 courses, that were discarded. Categories are assigned to videos based on the course wherein they are integrated.

4. **Video Transcripts Extraction**: each micro-learning video is sent to the *Speech-to-Text service*[5] of the *IBM Watson's suite*, which combines grammar rules with knowledge of audio signals for translating the spoken language of an audio track in its written form. Such a tool is one of the most accurate and easily manageable.

5. **Transcript Features Extraction**: from each video transcript, we extract *TF-IDF* features and *high-level semantic features* computed by the *Natural Language Understanding service*[6] part of the *IBM Watson's suite*. It contains a collection of natural language processing functions aiming at extracting *keywords*, *concepts*, *entities* and so on from texts. For this study, we took only *concepts* and *keywords*.

[3] https://about.coursera.org/affiliates
[4] https://github.com/coursera-dl/coursera-dl
[5] https://www.ibm.com/watson/developercloud/speech-to-text.html
[6] https://www.ibm.com/watson/services/natural-language-understanding/

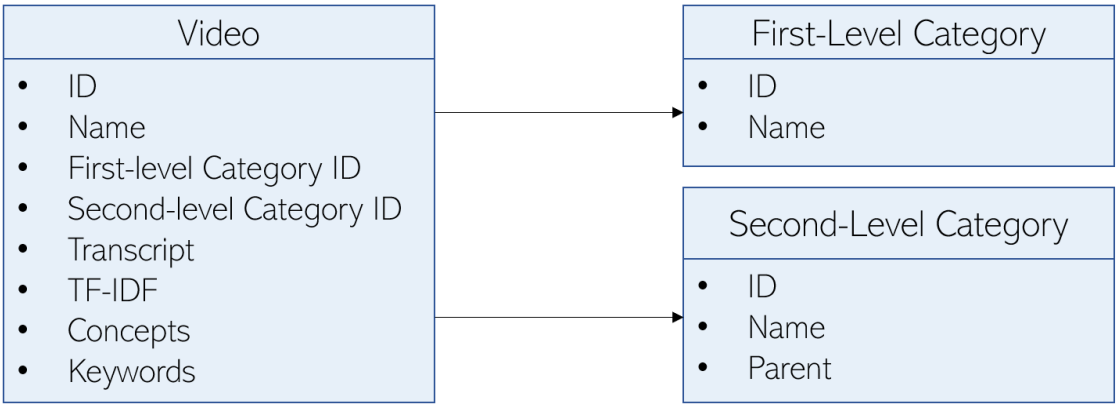


Fig. 3.2: Structure of ML-Videos data set.

3.4.2 Structure and Statistics

ML-Videos is a *CSV*-based collection whose structure in terms of entities and associations is depicted in Fig. 3.2. As expected, *Video* is the most informative entity. Each video is identified by a unique ID and primarily described by a short title. One first-level category ID and one second-level category ID are associated to each video and point to the corresponding entities. The text extracted from each video is saved into the *transcript* attribute, while the subsequent features are reported in the remaining attributes. Both *First-level* and *Second-level Category* entities are identified by an ID and a name. In addition to this, second-level categories include the ID of the parent first-level category.

We collected 10,328 videos from 617 courses taught in *English* language. Coursera pre-assigned each course to one of the 7 first-level categories and one of the 34 second-level categories. Each course contains a set of videos (avg. 18; std. 5). Each downloaded video was assigned to the same categories of the course it belongs, namely one first-level category and one second-level category. Each video lasts from 5 to 18 minutes, and the video transcripts contain from 200 to 10,000 words (avg. 1,525; std. 1,017). The distribution of the number of videos per category is shown in Fig. 3.3 and Fig. 3.4.

The dataset is challenging, since it contains fine-grained categories that require subtle details to differentiate (e.g. *Business Essentials* and *Business Strategy*). Moreover, the video transcripts contain less words than documents typically used in text classification. The language style is also different, since transcripts derive from speaking activities.

3.5 The Proposed Content Categorization Approach

In this section, we describe the proposed approach for micro-learning video classification. Figure 3.5 depicts its components and how they work together.

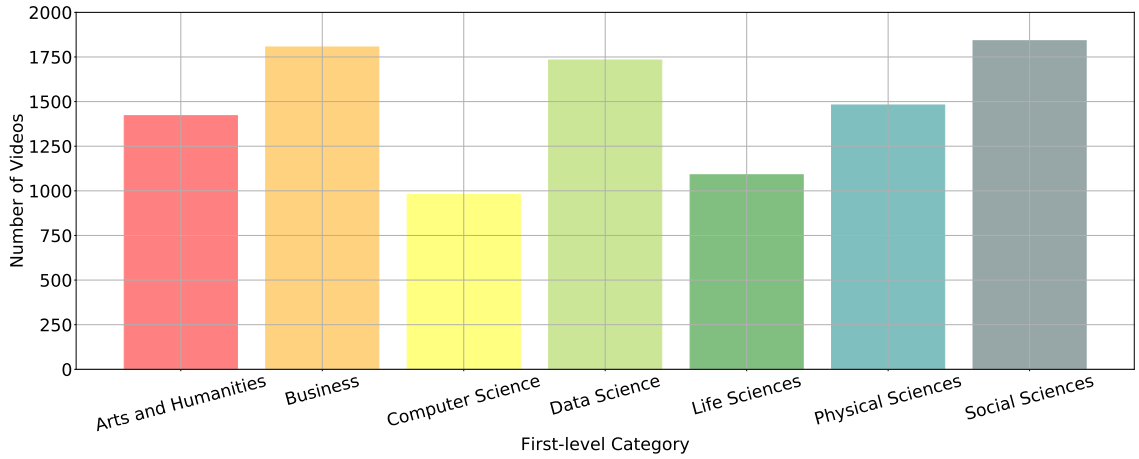


Fig. 3.3: First-level category distribution in ML-Videos.

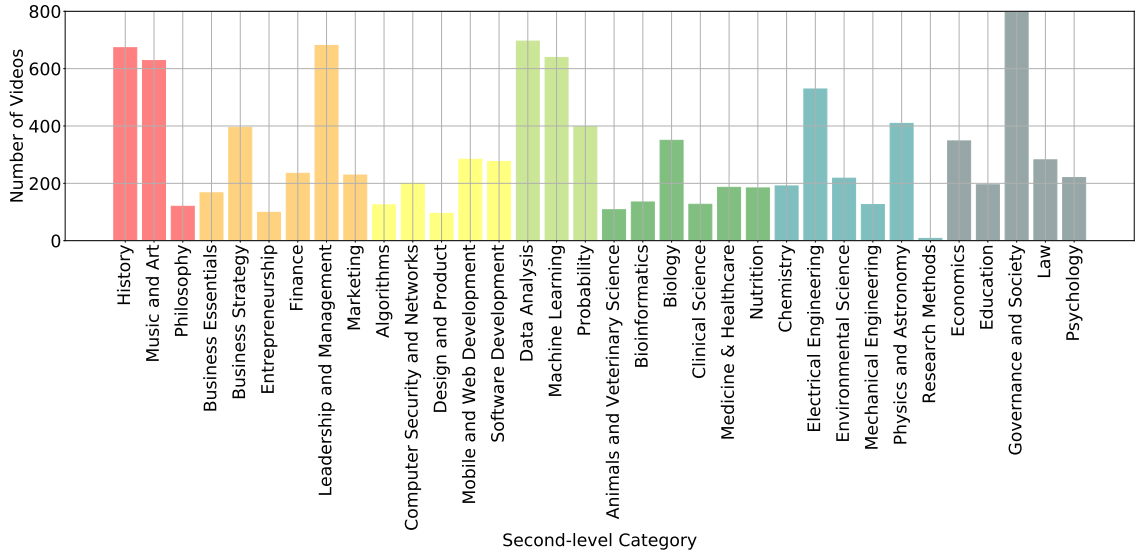


Fig. 3.4: Second-level category distribution in ML-Videos.

3.5.1 Methodology

The *Classification Manager* orchestrates the overall process, from speech-to-text conversion to performance evaluation. First, it calls the *Pre-Processing Module* to extract the transcripts from the videos included into the data set (Steps 1-2). Then, these transcripts are handled by the *Feature Extraction Module* which computes the corresponding features (Steps 3-4). During training, the *Classification Manager* sends both features and category labels to the *Classifer Module* (Steps 5-6). During testing, only the features are sent, and the returned categories are matched with the original ones stored in the data set to evaluate the performance (Step 7). We detail the modules in the following sections.

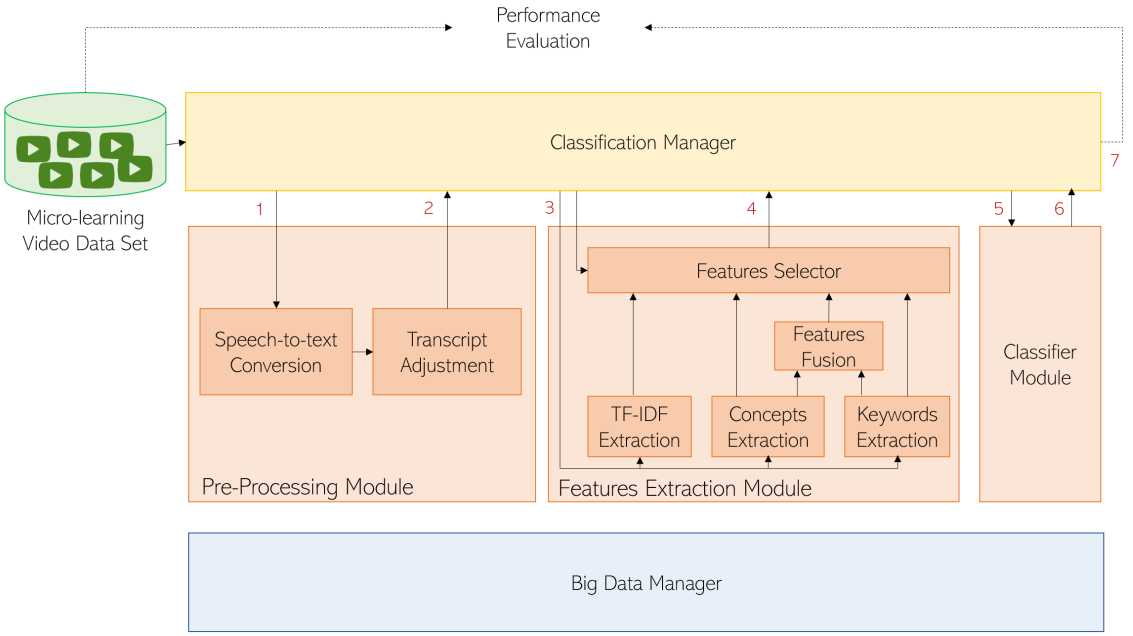


Fig. 3.5: The proposed micro-learning video classification approach.

Big Data Manager Module

This module integrates *Apache Spark* enriched with the *MLlib library*. *MLlib*[7] is the Spark's machine learning library aimed to make practical machine learning scalable and easy. It includes common learning algorithms and utilities, such as classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as lower-level optimization primitives and higher-level pipeline APIs. By integrating it, our approach becomes general and flexible. We can easily use any classifier within the *MLlib* library, whose code is already optimized with the *Map-Reduce paradigm* to run over a cluster. We might also use any other classifier of other libraries not specific for *Apache Spark*. The cluster has been configured to use over 100 machines within our department.

Pre-Processing Module

This module takes a micro-learning video as an input and returns the cleaned version of the associated transcript as an output. Given a micro-learning video v, the module sends it to the *Speech-to-Text service*[8] of the *IBM Watson's suite*, which combines grammar rules with knowledge of audio signals for translating the spoken language of an audio track in its written form. Its choice depends on the fact that it is one of the most accurate and easily manageable speech-to-text tools [91]. The service returns a plain

[7] http://spark.apache.org/mllib/
[8] https://www.ibm.com/watson/developercloud/speech-to-text.html

text *t(v)* corresponding to the transcript of *v*. The module converts all the words in *t(v)* to lowercase and each one of them is compared with the most similar word in WordNet[9] in order to detect spelling errors. Each word *w* in *t(v)* is substituted in *t(v)* with the correct *w'*, obtaining *t'(v)*. Finally, the module removes stop-words from *t'(v)* and returns it as a cleaned transcript. This will be used as a means of analysis for the video.

Features Extraction Module

The *Features Extraction Module* is responsible of turning a micro-learning transcript into a set of features. It takes a cleaned transcript from the *Pre-Processing Module* as an input and returns a set of pairs where the first element is the *identifier string* of the feature and the second element is the *relevance of that feature* for the corresponding transcript. The relevance value spans in the range $[0, 1]$, where a value closer to 0 represents a low relevance and a value closer to 1 a high relevance of that feature.

From the transcript, the module can extract *TF-IDF* features and high-level features computed by the *Natural Language Understanding API* provided by the *IBM Watson's suite*, which contains a collection of natural language processing functions aiming at extracting keywords, concepts, entities and so on from texts. Currently, the module exploits only concepts and keywords, but it can be easily extended to a wider set of features. Keywords include important topics typically used when indexing data. The *IBM* service identifies and ranks keywords directly mentioned in the text. Concepts represent words not necessarily referenced in the text, but with which the transcript text can be associated. These concepts are inferred from the relations between syntactic and semantic elements of the sentences. Both for concepts and keywords, the *IBM* service computes a weight that indicates their relevance.

Given a cleaned transcript *t'(v)* as returned from the *Pre-Processing Module*, the module first computes for each word $w \in t'(v)$ its *TF-IDF* value building the *TF-IDF* vector *tf-idf(t'(v))*. Then, it sends *t'(v)* to the *Natural Language Understanding service* and requests to obtain a concepts vector *c(t'(v))* and a keywords vector *k(t'(v))*. The service returns them as *JSON data*. For each vector, a list of pairs is returned where the first element of each pair is the string identifier and the second element is the relevance associated to it in the range $[0, 1]$. After that, the module builds a unique feature vector *kc(t'(v))* by concatenating *c(t'(v))* and *k(t'(v))*. In the case of a collision between two identifiers from concepts and keywords vectors, the module computes the mean between the associated relevance values and stores a single instance of the identifier, accordingly. As an example, we consider a short segment of a video transcript about computer networks (Fig. 3.6). "*Computer network*" is the only concept extracted even though it is not directly mentioned in the text. A number of four keywords were returned.

[9] https://wordnet.princeton.edu/

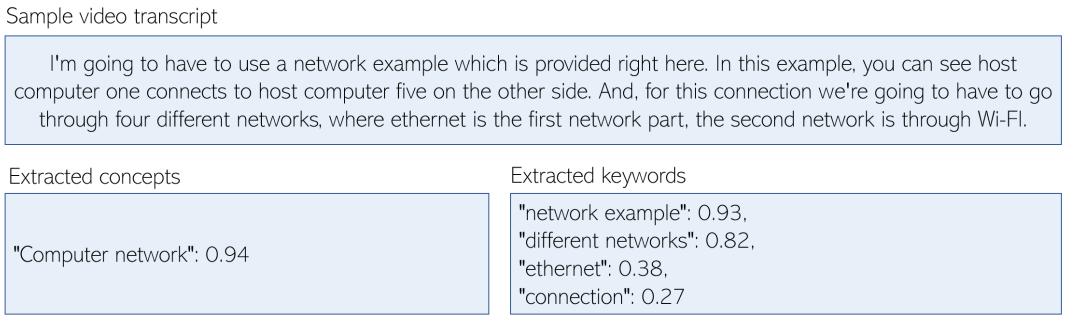


Fig. 3.6: Example of features extracted by IBM Watson.

Classifier Module

The *Classifier Module* aims at finding the most appropriate category for a given video using the underlying classifier trained on a number of labeled samples. The classifier implements a function $\mathtt{f}(\mathtt{t}'(v)) \rightarrow \mathtt{y}$ where $\mathtt{f}(\mathtt{t}'(v))$ is the features vector of $\mathtt{t}'(v)$ and $\mathtt{y}$ is the category returned by the classifier and of the given taxonomy. The module can implement any classification algorithm, independently from the feature type. In particular, our approach integrates *Decision Trees*, *Support Vector Machines*, *Random Forests*, and *Support Vector Machines jointly with Stochastic Gradient Descent*, since they are the most widely used algorithms as emerged from literature review. However, our approach does not depend on this design choice and any classification algorithm can be used.

3.5.2 Evaluation

In this section, we describe the experiments performed to assess both the *effectiveness* and the *efficiency* of our approach on top of the *ML-Videos* data set (see Section 3.4).

Evaluation Metrics and Protocols

We mapped the video classification problem as a text classification problem. For each category y_i of the category space $y_1, \ldots, y_n$, the corresponding precision $P_{\mathtt{y}_i}$, recall $R_{\mathtt{y}_i}$, and F-measure $F_{1\mathtt{y}_i}$ are defined by:

$$P_{\mathtt{y}_i} = \frac{TP_{\mathtt{y}_i}}{TP_{\mathtt{y}_i} + FP_{\mathtt{y}_i}} \quad R_{\mathtt{y}_i} = \frac{TP_{\mathtt{y}_i}}{TP_{\mathtt{y}_i} + FN_{\mathtt{y}_i}} \quad F_{1\mathtt{y}_i} = 2\frac{R_{\mathtt{y}_i} \cdot P_{\mathtt{y}_i}}{R_{\mathtt{y}_i} + P_{\mathtt{y}_i}}$$

where $TP_{\mathtt{y}_i}$ (true positives) is the number of videos correctly assigned to the category $\mathtt{y}_i$, $FP_{\mathtt{y}_i}$ (false positives) is the number of videos erroneously assigned to $\mathtt{y}_i$, and $FN_{\mathtt{y}_i}$ (false negatives) is the number of videos that actually belong to the category $\mathtt{y}_i$, but they are not assigned to this class.

Then, we need to compute the average performance of a binary classifier (i.e. one for each category) over multiple categories. There are three main methodologies for the computation of averaged metrics [92]. They can be summarized as follows:

- *Micro-Averaged (Mic).* The metrics are globally calculated by counting the total number of true positives, false negatives and false positives, with no category differences.
- *Macro-Averaged (Mac).* The metrics are locally calculated for each category and the unweighted mean is computed. This does not consider categories imbalance.
- *Weighted-Averaged (W).* The metrics are locally calculated for each category, then their average is obtained by weighting each category metric with the number of instances of the category in the dataset. Therefore, each category does not contribute equally to the final average and some of them contribute more than the others.

To test the performances of our approach, we considered nine scores obtained from the combination of the metrics (precision, recall, F-measure) with average mode (micro, macro, weighted), for each category space of our data set (first/second-level). The evaluation protocol is based on a *Stratified K-fold Cross-Validation* with *K=10*.

Baselines

In order to evaluate our approach, we tested it with a number of alternative combinations of four features representations and four classification algorithms. The features representations included TF-IDF (baseline), concepts, keywords, and the combination of keywords and concepts. While, the baseline classifiers were the following algorithms:

- *Decision Tree* (DT). This method creates a model based on *C4.5 algorithm*, predicting the value of a target variable by learning decision rules from data features. The model has a root node containing all data features of the training set. Then, the root node is split into several children according to a given criteria.
- *Support Vector Machine* (SVM). This method maps each sample to a point in a n-dimensional space (where n is number of features). Then, it finds the set of hyperplanes that better differentiate the categories. A linear combination of vectors determines the the decision boundaries producing the best separation of categories.
- *Random Forest* (RF). This method is a meta estimator that fits a number of decision tree classifiers on various random sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Each decision tree, individually, is a weak classifier, while all the decision trees taken together would be a strong classifier.
- *Support Vector Machine + Stochastic Gradient Descent* (SVM+SGD). This method extends the standard SVM implementation with the SGD algorithm, that finds the best coefficients describing the decision boundaries by optimizing for a hinge loss function. It allows performing training over large data while reducing the computation time.

Computational Time Evaluation

We evaluated the efficiency of the proposed approach in terms of the size of the features vectors and the time for performing both training and test for a given classifier.

Table 3.2 summarizes the basic statistics with respect to the different feature sets. The first column indicates the name of the feature set, the second column shows its size, while the other two columns report the average and the standard deviation of the number of non-zero elements. The vector sizes for the combination *concepts* and *keywords* is greater than all the others. However, the average number of its non-zero elements is smaller than that of the *TF-IDF*. Hence, high-level features might be more discriminative.

Table 3.3 reports the total time required for a given algorithm with a given features set to perform the training and the test phases. From the results, the *SVM+SGD* algo-

Feature Set	Total Size	Average Size	Standard Deviation
TF-IDF (baseline)	117,073	260	513
Concepts	29,952	21	13
Keywords	332,023	78	26
Keywords + Concepts	361,975	99	31

Table 3.2: Basic statistics about the considered features sets.

Features Set	Algorithm	Execution Time [s]	
		First-level Categories	*Second-level Categories*
TF-IDF (baseline)	DT	5.23	7.92
	RF	12.20	39.81
	SVM	3.70	16.25
	SVM+SGD	**0.25**	**1.12**
Keywords	DT	3.32	7.21
	RF	11.06	43.96
	SVM	3.32	11.21
	SVM+SGD	**0.20**	**1.00**
Concepts	DT	1.13	1.88
	RF	2.38	7.67
	SVM	0.31	2.21
	SVM+SGD	**0.05**	**0.20**
Keywords + Concepts	DT	4.00	8.34
	RF	13.53	50.38
	SVM	3.95	12.84
	SVM+SGD	**0.26**	**1.06**

Table 3.3: Computational time for completing both training and test phases.

rithm achieved the best computational time, especially when fed with concepts. The computational time mostly depends on the particular classifier and the number of features. For example, the computational time for *SVM+SGD* classifier using concepts as features takes 0.05 seconds due to the lower time required by *SVM+SGD* and the lower size of the concepts feature set. DT and plain SVM got comparable computational time.

Effectiveness Evaluation

We evaluated the performance of all the classifiers trained on *TF-IDF*, *keywords*, *concepts*, and *keywords plus concepts* using precision, recall and F-measure. The results in Table 3.4 indicate that using *keywords* outperformed *TF-IDF* only with the *SVM+SGD* approach. Using *concepts* generally gave better results than using *keywords*, except when fed into *SVM*-derived algorithms. *Concepts* outperformed *TF-IDF* in case of *SVM+SGD* algorithm. With concepts, we obtained higher precision and lower recall, particularly for macro and weighted evaluations. Overall, *SVM* fed with *TF-IDF* outperformed all the other settings, but it was usually ten time slower than the *SVM+SGD* counterpart.

Table 3.5 shows the results of the classifiers on second-level categories. In this case, *SVM+SGD* classifier trained using *keywords* still outperformed the same algorithms trained on *TF-IDF*, while its performance decreased if trained on *concepts*. When *key-*

Features	**Algorithm**	**Precision**			**Recall**			**F-Measure**		
		Mic	*Mac*	*W*	*Mic*	*Mac*	*W*	*Mic*	*Mac*	*W*
TF-IDF (baseline)	DT	0.48	0.56	0.55	0.48	0.47	0.48	0.48	0.48	0.48
	RF	0.58	0.61	0.60	0.58	0.57	0.58	0.58	0.57	0.57
	SVM	**0.70**	**0.72**	**0.71**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	**0.69**
	SVM+SGD	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.61	0.61
Keywords	DT	0.35	0.49	0.47	0.35	0.32	0.35	0.35	0.32	0.33
	RF	0.43	0.51	0.49	0.43	0.41	0.43	0.43	0.40	0.41
	SVM	0.65	**0.68**	**0.67**	0.65	0.65	0.65	0.65	**0.65**	**0.65**
	SVM+SGD	**0.66**	0.66	0.66	**0.66**	**0.66**	**0.66**	**0.66**	**0.65**	**0.65**
Concepts	DT	0.44	0.68	0.66	0.44	0.42	0.44	0.44	0.44	0.45
	RF	0.52	0.67	0.66	0.52	0.51	0.52	0.52	0.53	0.53
	SVM	0.62	**0.64**	0.63	0.62	0.62	0.62	0.62	0.62	0.61
	SVM+SGD	**0.63**	**0.64**	**0.64**	**0.63**	**0.63**	**0.63**	**0.63**	**0.63**	**0.62**
Keywords + Concepts	DT	0.46	0.65	0.64	0.46	0.44	0.46	0.46	0.47	0.47
	RF	0.55	0.64	0.63	0.55	0.54	0.55	0.55	0.55	0.55
	SVM	**0.69**	**0.71**	**0.70**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**	**0.69**
	SVM+SGD	0.68	0.69	0.69	0.68	0.68	0.68	0.68	0.67	0.67

Table 3.4: Effectiveness on video classification over first-level categories.

Features	Algorithm	Precision			Recall			F-Measure		
		Mic	*Mac*	*W*	*Mic*	*Mac*	*W*	*Mic*	*Mac*	*W*
TF-IDF (baseline)	DT	0.46	0.53	0.52	0.46	0.42	0.46	0.46	0.44	0.46
	RF	0.58	0.59	0.59	0.58	0.54	0.58	0.58	0.54	0.56
	SVM	**0.71**	**0.73**	**0.73**	**0.71**	**0.67**	**0.71**	**0.71**	**0.67**	**0.70**
	SVM+SGD	0.58	0.64	0.69	0.58	0.57	0.58	0.58	0.57	0.61
Keywords	DT	0.34	0.50	0.48	0.34	0.32	0.34	0.33	0.36	0.34
	RF	0.43	0.49	0.49	0.43	0.41	0.43	0.43	0.42	0.42
	SVM	0.62	**0.70**	0.67	0.62	0.55	0.62	0.62	0.57	0.61
	SVM+SGD	**0.66**	0.66	**0.70**	**0.66**	**0.65**	**0.66**	**0.66**	**0.63**	**0.66**
Concepts	DT	0.39	0.49	0.52	0.39	0.38	0.39	0.39	0.39	0.40
	RF	0.48	0.50	0.53	0.48	0.46	0.48	0.48	0.45	0.47
	SVM	**0.57**	**0.58**	**0.59**	**0.57**	0.53	**0.57**	**0.57**	**0.53**	**0.56**
	SVM+SGD	**0.57**	0.55	0.58	**0.57**	**0.54**	**0.57**	**0.57**	**0.53**	**0.56**
Keywords + Concepts	DT	0.41	0.52	0.52	0.41	0.40	0.41	0.41	0.42	0.41
	RF	0.51	0.54	0.54	0.51	0.50	0.51	0.51	0.49	0.50
	SVM	0.66	**0.70**	**0.69**	**0.66**	0.60	**0.66**	0.66	**0.62**	0.65
	SVM+SGD	**0.67**	0.64	**0.69**	**0.66**	**0.65**	**0.66**	**0.67**	**0.62**	**0.66**

Table 3.5: Effectiveness on video classification over second-level categories.

words and concepts were combined, the overall performance improved. In the case of *SVM+SGD*, it can be considered better than *TF-IDF* with an improvement up to 9%. The worst case with *concepts + keywords* showed a maximum loss of 7% when *SVM* is adopted. As far as the second-level categories classification was concerned, it is worth noticing that such categories are not equally distributed as the first-level categories (see Tables 3.3 and 3.4), and this variance negatively affects classification using high-level features.

Effectiveness-Efficiency Trade-off Evaluation

In most cases, our semantic approach produces good performance, regardless the increasing algorithm complexity in terms of video size or transcript size. The first factor influences only the time needed to extract transcripts. No assumptions can be done on transcript sizes in relation to video sizes since the number of words included into a transcript depends on the amount of orally-provided content. The second factor has an impact on feature extraction. Training and test are not influenced by the transcript size since the size of each feature vector depends on the size of the word dictionary.

Considering the trade-off between effectiveness in terms of precision, recall, F-measure and efficiency in terms of computational time, the combinations achieving best results include *SVM* or *SVM+SGD* as algorithm and *TF-IDF* or *Keywords+Concepts* as fea-

tures, respectively. However, *SVM+SGD* algorithm is generally over ten times faster than *SVM*. With *SVM+SGD*, our approach using *Keywords+Concepts* outperforms that using *TF-IDF* from 7% to 9% in terms of precision, recall and F-measure. The combination using *SVM+SGD* and *Keywords+Concepts* achieves performance comparable with that obtained by *SVM* fed with *TF-IDF* in terms of effectiveness, but the former is strongly better in efficiency. The experimental results demonstrate that *Keywords+Concepts* combined with *SVM+SGD* can scale well, maintaining good performance in all cases.

3.6 Findings and Recommendations

In this chapter, we presented a new data set and a new approach supporting micro-learning video classification. Our approach extracts transcripts from videos using novel *speech-to-text* systems and exploits *Cognitive Computing* to represent the features of each video transcript. Moreover, it leverages *Big Data* technologies for fast computation.

Based on the obtained results, we can conclude that:

- Combining *Concepts* and *Keywords* can help existing machine-learning models to exploit the semantic behind the text that traditional approaches fail to capture.
- The proposed approach that fed *SVM+SGD* with *Concepts+Keywords* achieves good performance in most cases for both computational time and precision-recall analysis.
- *TF-IDF* still works better than high-level features with algorithms that incrementally split data for classifications, such as *Decision Trees* and *Random Forest*.
- Leveraging *Apache Spark* allows the proposed approaches to be scalable enough to process and classify millions of videos, fitting to real-world operational conditions.
- No significant drop exists between models performance over first and second-level categories, highlighting how robust the considered features and algorithms are.

Given the short duration of videos, and therefore the short transcripts, we plan to leverage DL methodologies for learning more reliable and robust features tailored for solving this classification task, going beyond general-purpose features. To this end, particular emphasis will be given to attention-based models able to consider the relevance of each feature on the final classification decision. Furthermore, given the short duration of videos and their possibly specialized content, we will devise multi-level classification approaches moving from high-level categories to low-level categories, and investigate whether this can contribute to get better results or could uselessly increase the computational demand. Moreover, we would provide our approach as a base for usability evaluation in order to quantify its impact on learners' and instructors' experience. Our approach might also be combined with experiential data of students to recommend videos that fit learners' interests.

Chapter 4

Machine Learning Models for Content Recommendation

Research Highlights
- The large-scale dataset we proposed enables ML-based recommendation.
- Educational recommenders are highly susceptible to popularity biases.
- Instructors belonging to a minority group might be disadvantaged.
- Biases can be mitigated by regularizing models during training.

4.1 Introduction

Recommender systems learn behavioural patterns from data to support both *individuals* [93] and *groups* [94] at filtering the *overwhelming alternatives* our daily life offers. However, the *biases in historical data* might propagate in the items suggested to the users and foster potentially undesired outcomes [95]. For instance, some recommenders focus on a tiny catalog part composed by popular items, leading to *popularity bias* [96, 97, 98]. Others generate a *category-wise bias* because the rating distribution greatly varies across categories [99]. Historical patterns can promote *unfair recommendations*, discriminating learners and/or educators based on sensitive attributes [100]. In addition, *evaluating only accuracy might not bring to online success* [101]. Movie recommenders make good rating predictions, but focus on few popular items and lack personalization [102]. In contrast, in tourism, high accuracy corresponds to better perceived recommendations [103].

In the view of these context-dependent results, we thus point out that *online education* is an unexplored field where the impact of biases should be investigated. Existing e-learning platforms have attracted *lots of participants* and their interaction has generated a *vast amount of learning-related data*. Their collection and processing have opened up new opportunities for supporting educational experiences, such as *personalization* and *recommendation* [104, 105]. In such a *multi-stakeholder environment*, where a va-

riety of individuals (e.g, learners, educators) or groups benefits from delivering recommendations, the possibility that a recommender might be biased towards *items of certain categories* or *people characterized by certain sensitive attributes* sheds lights on fairness and transparency perspectives. Entirely removing any bias from algorithms is currently impracticable, but uncovering and mitigating them should be a *core objective*.

In this chapter, we study the susceptibility of educational recommenders to existing biases, and we design countermeasures that strike the trade-off between bias mitigation and recommendation effectiveness. We conducted an offline evaluation of different recommendation strategies that took as input the ratings left by learners after attending online courses. Then, we compared the courses recommended by classic and recent methods against data biases, such as popularity biases and educators' gender biases. These biases might have educational (e.g., course popularity bias might affect knowledge diversification) and societal (e.g., educators from minority groups might be disadvantaged) implications. Our results uncover that existing recommenders suffer from several biases, and provide evidence on the need to go beyond the evaluation only of their accuracy.

The contribution of this chapter is five-fold:

- We arranged a *large-scale data set* including over 30k courses, 6K instructors and 32k learners who provided 600k ratings. This outruns most of the existing educational data sets for recommendation in terms of scale, completeness, and comprehensiveness.
- We provide novel definitions of trade-off between recommendation effectiveness and robustness against several biases (i.e., popularity robustness, educators' gender robustness), that make it possible to construct tailored benchmarks for these problems.
- We conduct explorative analysis on several biases (i.e., popularity bias, educators' gender bias) in traditional recommenders and more recent learning-to-rank recommenders, and uncover internal mechanics and deficiencies behind such biases.
- We define regularization terms within training optimization functions for reducing the influence of popularity and educators' gender on item relevance during model development, leveraging pair-wise comparisons of properly-selected pairs of items.
- We extensively validate the proposed mitigation approaches in the collected data set and other public data sets from different domains. We show that they produce significant improvements in popularity and providers' gender bias mitigation, with a reasonable impact on effectiveness, regardless of the applied domain.

The rest of this chapter is structured as follows: Section 4.2 describes the most representative techniques for recommendation and provides a summary of biases in recommendation. Then, Section 4.3 formalizes the problems we investigate, and Section 4.4 introduces the data set. Section 4.5 and Section 4.6 show approaches for providing less biased and more fair recommendations. Finally, Section 4.7 concludes the chapter.

4.2 Related Work

4.2.1 Recommendation Techniques

The term *recommendation* can be defined as predicting the interest of a user for items based on his/her past interactions, and providing to him/her a ranking of unseen items for which such a predicted interest is the highest [93]. Interactions can be *rating values* (e.g., stars) or *binary values* (e.g., like/dislike or interested/not interested). They can be collected either *explicitly*, in combination with reviews entered by users, or *implicitly*, from system logs. Item recommendation approaches built upon such a data can be *personalized* and *non-personalized*. Here, we deal with personalized recommendations, which can be obtained through *content-based*, *collaborative filtering*, or *hybrid techniques*.

Content-based methods mine descriptions and contents of items that have been marked with high ratings by a user, and then recommend to this user unseen items that have similar characteristics with respect to the former items [106, 107]. Recommending purely based on item content might bring some drawbacks, such as *narrow pattern extraction* and *overspecialization*. The former occurs when the feature extractor or the item content itself is often insufficient to detect meaningful patterns. The latter implies that content-based systems might recommended items that are too similar to the ones experienced any the user in the past.

Collaborative filtering approaches capitalize on the idea that users with similar rating patterns might have similar interests in the future [108, 109, 110, 111]. Such a class of recommenders include *neighborhood* and *model-based* methods. In the former, the user-item ratings are directly used to predict ratings for new items in two ways, known as *user-based* or *item-based* recommendation. They evaluate the interest of a target user for an item using the ratings for this item by other users that have similar rating patterns. The latter predicts the rating of a user for an item based on the ratings of the user for similar items. Two items are similar if users have assigned ratings to these items in a similar way.

On the other hand, *model-based* approaches optimize a predictive model fed with rating data [112]. Relevant patterns of interaction between users and items are captured by model parameters learned from data. *Latent Semantic Analysis* [113], *Latent Dirichlet Allocation* [114], *Support Vector Machines* [115], and *Singular Value Decomposition* [116, 117] are all examples of model-base approaches. Moreover, recent works integrated learning-to-rank approaches categorized in point-wise [118], pair-wise [119, 120] and list-wise [121, 122]. Point-wise approaches take a user-item pair and predict how relevant the item is for that user. Pair-wise approaches digest a triplet of user, observed item, and unobserved item, and minimize cases when the unobserved item is more relevant than the observed item. List-wise approaches look at the entire list to build the optimal ordering for that user.

Finally, content-based and collaborative filtering methods can be combined, leading to *hybrid recommendation approaches*. For instance, individual predictions can be merged into a single, robust prediction [123, 124] or content information can be added into a collaborative filtering model [125, 126].

4.2.2 Recommendation in Education

Research on *personalization of learning* is getting more and more important with the increasing use of digital learning environments, such as learning object repositories, learning management systems, and devices for mobile learning. This has made data collection an inherent process of delivering educational content to the students. That means that the analysis of learning behavior is no longer only related to representative pilot studies, but also refers to the interaction of the entire student population. This trend has even become faster with the appearance of *Massive Open Online Courses* (MOOCs) [127] and the emerging of the *Learning Analytics* field [128]. The former provides massive amounts of student data and fosters new opportunities for recommender systems. The latter focuses on understanding and supporting learners based on the data they produce.

Existing taxonomies generally identify *seven clusters* to group educational recommenders systems [129], namely (i) the ones following collaborative filtering approaches as done in other domains [130, 131]; (ii) the ones that propose improvements to collaborative filtering approaches to take into account the peculiarities of the educational domain [132, 133]; (iii) the ones that explicitly consider educational constraints as a source of information [134, 135]; (iv) the ones that explore alternatives to collaborative filtering approaches [136, 137]; (v) the ones that consider contextual information [138, 139]; (vi) the ones that assess the impact of the recommendations [140, 141]; and (vii) the ones that focus on recommending courses instead of resources within them [142, 143, 144]. The field is moving and new approaches are emerging year by year.

Several trends have emerged from the analysis of educational recommender systems according to the above-mentioned categories. *Finding good items* within courses is the most applied task, but *recommendation of sequence of items* that aims to create an effective and efficient learning path through digital contents is also an important task. Along this mainly content-driven recommendations, the *recommendation of other learners* who follow similar learning goals or share the same interests is a very central task. There are some new tasks appearing in the recent years, which go beyond recommending learning content within courses, such as *suggesting online courses*.

Examples for techniques are *hybrid* and *content-based approaches* that started to be reported in 2008 and are increasingly applied in recent years until today. There is an increasing interest in *graph-based* and *knowledge-based* approaches. They are mainly applied to address sparsity and unstructured data problems. However, *collaborative filtering* and *rule-based* approaches are still the most frequently used techniques [129].

4.2.3 Biases in Recommendation

Popularity Bias

In the existing recommendation strategies, popular items (i.e., those with more ratings) are frequently recommended and less popular ones are suggested rarely, if at all. This has led to a phenomenon called *popularity bias*. The authors in [101] were among the first to point at uneven results with respect to popularity in top-k recommendations. In [145], the authors revealed that social media traffic tends to exhibit high popularity bias. In addition to this, it has been proved that popularity bias affects several other domains [146]. In order to cope with popularity bias, researchers developed methods falling into pre-processing, in-processing, and post-processing.

Pre-processing methods alter training data to reduce the potential impact of bias on it. For instance, the work in [147] split the whole item set into short and long-tail parts and clustered their ratings separately. Long-tail recommendations were based on ratings from the corresponding cluster, while short-tail recommendation used the ratings of individual items. They showed that this reduces the gap for long-tail items, while maintaining reasonable performance. The authors in [146] proposed a non-uniform sampling of training triplets. The idea was to normally sample item pairs where the observed element is less popular then the unobserved one. They proved that learning from those triplets can counteract popularity bias. The authors in [148] proposed to pick up unobserved items based on a probability distribution depending on item popularity.

In-processing methods modify an existing algorithm to simultaneously consider relevance and popularity, performing a joint optimization or using one criterion as a constraint for the other. The authors in [149] recommended new items by considering user's personal popularity tendency. This reasonably penalizes popular items while improving accuracy. The work in [150] mitigated popularity bias by enhancing recommendation neutrality, i.e., the statistical independence between a recommendation and its popularity. The authors in [102] modified the *RankALS* algorithm to push optimization towards recommended lists that balance accuracy and short/long-tail item relevance. In [151], the authors calculated the number of common neighbours between two items with given popularities, then a balanced common-neighbour similarity index is developed by pruning popular neighbours. While all approaches improve short/long-tail item balance, they are not linked to an internal aspect causing the bias. Such a property might be desirable to find better solutions for that particular class of algorithms.

Post-processing methods seek to re-rank a recommended list returned by an existing algorithm according to certain constraints. The authors in [152, 153] presented two approaches for controlling item exposure. The first one adapted the *xQuAD* algorithm in order to balance the trade-off between accuracy and long-tail item coverage. The second one multiplies the relevance score for a given item with a weight inversely proportional to the item popularity. Items were re-ranked according to the weighted scores, gener-

ating popularity-distributed recommendations. Furthermore, the work in [146] determined user-specific weights that help to balance accuracy and popularity. While a post-processing approach can be applied to any algorithm, the computation of the weights highly depends on the scale and distribution of the relevance scores. Therefore, they must be arranged to the considered algorithm and require extra computation.

Consumer-related Bias

Consumer fairness (C-Fairness) deals with ensuring that users who belong to different groups or are similar at the individual level will receive recommendations with the same quality. Existing works tackling consumer fairness either propose pre-processing, in-processing, or post-processing approaches.

Considering the *pre-processing* approaches, the authors in [154] introduced the concept of consumer-fairness into tensor-based recommendation algorithms, and optimize for statistical parity (i.e., having the same recommendation quality for both groups of users). They did so by proposing an approach to identify and remove from the tensor all sensitive information about the users. By running the recommendation algorithm on the tensor without the sensitive attributes, more fair recommendations can be generated. The authors in [155], instead, generated additional artificial data to improve the fairness of the system. In their work, fairness is obtained by minimizing the difference in the mean squared error of the recommender system for two groups of users.

Regarding *in-processing* approaches, the authors in [156] proposed approaches to measure consumer-unfairness that might come from population imbalance (i.e., a class of users characterized by a sensitive attribute being the minority) and from observation bias (i.e., a class of users who produced less ratings than their counterpart). They propose four unfairness metrics, and added them to the *Matrix Factorization* loss function to be minimized. The authors in [157] tackled the problem of ensuring consumer-fairness in recommendation ranking through pair-wise comparisons. They studied the likelihood of a clicked item being ranked above another relevant unclicked item, defining *pairwise fairness*. Then, they built a pairwise regularization, which penalizes the model if the its ability to predict which item was clicked is better for one group than the other.

The mitigation of consumer unfairness through a post-processing approach was proposed by the work in [158], mitigating stereotypes that might arise from the produced recommendations, in what the authors call ϵ-fairness. They defined a recommendation list as ϵ-fair if the predictability of a sensitive feature (e.g., the gender) of the user receiving the recommendations is lower than a value ϵ. In order to achieve that goal, a re-ranking algorithm aims to minimize the loss in accuracy while guaranteeing ϵ-fairness.

Provider-related Bias

Provider fairness (P-Fairness) aims to ensure that the providers of the recommended objects, who belong to different groups or are similar at individual level, will get recommended the same amount of times based on their numerical representation in data.

Provider fairness was mostly tackled through *post-processing* approaches. The authors in [159] achieved provider-fairness in the *Kiva.org* platform through a re-ranking function based on *xQuad*; this algorithm balances recommendation accuracy and fairness by dynamically adding a custom bonus to the items of the non-recommended providers. In the same domain, the authors in [160] tried to define the concept of local fairness and identify protected groups through consideration of local conditions. This was done to avoid discriminating between types of loans, and to equalize access to capital across all businesses. While the results showed that it is not possible introduce local fairness in the recommendation, refined approaches could improve local conditions.

Differently from other fair recommender systems, the authors in [161] did not measure provider-unfairness based on sensitive features of the users, but on the popularity of the providers. They focus on a two-sided marketplace, with the consumers being users who listen to music, and the artists being the providers. If only highly popular artists are recommended to users, there could be a disadvantage for the less popular ones. For this reason, artists are divided in ten bins based on their popularity, and a fairness metric that rewards recommendation lists diverse in terms of popularity bins is defined. Several policies are defined to study the trade-offs between user-relevance and fairness, with the ones that balance the two aspects being those achieving the best trade-off.

4.3 Problem Formalization

In this section, we formalize the recommender system, accuracy metrics, popularity bias and educators' fairness concerns, and new metrics we propose to quantify them.

4.3.1 Recommender System Formalization

Given a set of M users $U = \{u_1, u_2, ..., u_M\}$ and a set of N items $I = \{i_1, i_2, ..., i_N\}$, we assume that users have expressed their interest for a subset of items in I. The collected feedback from observed user-item interactions can be abstracted to a set of (u, i) pairs implicitly obtained from natural user activity or $(u, i, \mathtt{rating})$ triplets explicitly provided by users. We denote the user-item feedback matrix $R \in \mathbb{R}^{M*N}$ as by $R(u, i) = 1$ to indicate user u interacted with item i, and $R(u, i) = 0$ otherwise; it is worth noticing that this brings a slight simplification along the way, with no generality loss.

Given this input, the recommender system's task is to predict unobserved user-item relevance, and thereupon deliver a set of ranked items that satisfy the needs of users. To this end, we assume that a function estimates relevance scores of unobserved entries in R for a given user, and uses them for ranking the items. Formally, it can be abstracted as learning $\widetilde{R}(u, i) = f(u, i|\theta)$, where $\widetilde{R}(u, i)$ denotes the predicted relevance, θ denotes model parameters, and f denotes the function that maps parameters to scores.

We assume that each user/item is internally represented through a D-sized numerical vector. More precisely, the model includes a user-vector matrix W and an item-vector matrix X. We also assume that the function f computes the dot similarity between user and item vectors W_u and X_i. The higher the similarity, the higher the relevance. To rank items, they are sorted by decreasing relevance, and the top-k items are recommended, where k is the cut-off (i.e., the number of items to be shown to that user).

4.3.2 Effectiveness Metrics and the Proposed Bias-related Metrics

As we deal with a multi-criteria setting, we are concerned about any accuracy loss that results from considering additional criteria. Therefore, our goal is to strike a balance between recommendation accuracy and debiasing metrics. For the former property, we rely on a well-known metric, while for the latter property we formalize novel metrics that are proposed in this thesis.

Normalized Discounted Cumulative Gain (nDCG). The Discounted Cumulative Gain (DCG) is a measure of ranking quality [162]. Through a graded relevance scale of items in a ranking, DCG measures the gain of an item based on its position in the ranking. The gain is accumulated from the top to the bottom of the list, with the gain of each result discounted at lower ranks. To compare the metric among recommended lists, all relevant items are sorted by their relative relevance, producing the maximum possible DCG, i.e, Ideal DCG (IDCG). The normalized discounted cumulative gain (nDCG) is then computed as:

$$DCG@k(u|\theta) = \sum_{p=1}^{k} \frac{2^{f(u,i_p|\theta)} - 1}{\log_2(p+1)} \tag{4.1}$$

$$nDCG@k(\theta) = \frac{1}{|U|} \sum_{u \in U} \frac{DCG@k(u|\theta)}{IDCG@k(u|\theta)} \tag{4.2}$$

where $f(u, i_p|\theta)$ is the relevance of the item suggested at position p to u, and $IDCG@k$ is calculated with items sorted by decreasing relevance.

Popularity-aware Recommendation Parity (PSP). The proposed parity aims to force the probability distributions of model outputs for different items to be the same. Considering that only the top-k items are recommended to users, we focus on the probabilities of being ranked within the top-k, which is also aligned with basic recommendation accuracy metrics, such as precision@k and recall@k. Therefore, we propose the popularity-aware statistical parity metric, which encourages $P(R@k|i = i_1, \theta) = P(R@k|i = i_2, \theta) = ... = P(R@k|i = i_N, \theta)$, where R@k represents being ranked within top-k, and $P(R@k|i = j, \theta)$ is the probability of ranking item j within top-k. Formally, we calculate the probability as follows:

$$P(R@k|i = j, \theta) = \frac{\sum_{u \in U} \varphi(u, j)}{\sum_{u \in U} 1 - R(u, j)} \tag{4.3}$$

where $\varphi(u, j)$ returns 1 if item j is being ranked in top-k for user u, 0 otherwise, $\sum_{u \in U} \varphi(u, j)$ calculates how many user are being recommended item j in top-k, and $\sum_{u \in U} 1 - R(u, j)$ calculates how many users have never interacted with item j, and thus might receive j as a recommendation. Last, to keep the same scale for different k, we compute the relative standard deviation over the probabilities to determine PSP@k:

$$PSP@k = \frac{std\left(P(R@k|i = i_1, \theta), ..., P(R@k|i = i_N, \theta)\right)}{mean\left(P(R@k|i = i_1, \theta), ..., P(R@k|i = i_N, \theta)\right)} \tag{4.4}$$

where std(·) calculates standard deviation, and mean(·) calculates mean value.

Popularity-aware Equality of Treatment (PEO). The proposed equality of treatment aims to encourage the True Positive Rates (TPRs) of different items to be the same. We define the TPR as the probability of being ranked within top-k, given the ground-truth that the item is relevant for the user in the test set. This is noted as $P(R@k|i = j, y = 1, \theta)$, where $y = 1$ defines that items are relevant for users. This value can be formally computed as:

$$P(R@k|i = j, y = 1, \theta) = \frac{\sum_{u \in U} \varphi(u, j)\, R(u, j)}{\sum_{u \in U} R(u, j)} \tag{4.5}$$

where $\sum_{u \in U} \varphi(u, j)\, R(u, j)$ calculates how many user who interacted with item j (test set) are being recommended item j in top-k, and $\sum_{u \in U} R(u, j)$ calculates how many users have interacted with item j (test set). Then, we calculate the relative standard deviation to determine PEO@k:

$$PEO@k = \frac{std\left(P(R@k|i = i_1, y = 1, \theta), ..., P(R@k|i = i_N, y = 1, \theta)\right)}{mean\left(P(R@k|i = i_1, y = 1, \theta), ..., P(R@k|i = i_N, y = 1, \theta)\right)} \tag{4.6}$$

The reader notices that TPR is equal to recall in classification tasks, while the proposed probability $P(R@k|i = j, y = 1, \theta)$ is equal to recall@k of item j in recommendation tasks. Therefore, mitigating bias based on equality of treatment enforces recall@k for different items to be similar.

Educators' Gender Fairness. We assume to measure it as the equality of representation between the number of items per educator's gender and the number of recommendations per educator's gender. The score ranges between 0 and 1. Higher values indicate greater educators' fairness with respect to their gender. The $\mathtt{ItemExposureFair@k}$ (IEF@k) for women educators is defined as:

$$IEF@k(\theta) = 1 - abs\left(\frac{|I_{women}|}{|I|} - \frac{\sum_{u \in U, i \in I_{women}} \phi(u, i|\theta)}{\sum_{u \in U, i \in I} \phi(u, i|\theta)}\right) \tag{4.7}$$

where $\phi(u, i|\theta)$ is 1 if item i is recommended to u in top-k list, 0 otherwise. The metric can be mutually computed for men educators as well.

Under our multi-criteria setting, note that the higher the nDCG value, the more accurate recommendations are. For both PSP@k, PEO@k, and IEF@k, lower values indicate that recommendations are less biased. From a practical perspective, we argue that a good nDCG-PSP trade-off is important in scenarios where an equal recommendation distribution among items is required regardless of existing imbalance in the data, such as some circumstances where items are connected with critical or sensitive information[1]. Conversely, a good nDCG-PEO trade-off is preferred in recommenders that want to preserve the existing imbalance in original data, but prevent algorithmic bias. Finally, a good nDCG-IEF trade-off characterizes recommender systems that are both accurate and unbiased against educators' gender.

In what follows, we introduce a new data set that serves to investigate the above problems in the educational domain. Then, we present exploratory analysis and technical approaches that aim to tackle the mentioned trade-offs.

4.4 The Proposed COCO-RS Data Set

COCO-RS contains over 600k ratings from more than 30k learners, extracted from courses delivered on *Udemy*, one of the leading global marketplaces for online learning and teaching at scale. Unlike academic-oriented platforms driven to traditional coursework, Udemy enables experts in various areas to offer courses at no charge or for tuition

[1] We point out that, while this popularity metric and the related trade-off may not tend to fit all recommendations scenarios universally, it would help drive algorithm decision for systems that desire to face such popularity concerns due to specific business objectives.

Data Set	#Users	#Items	#Ratings	Context	Attributes
Book Crossing [163]	105,283	340,556	1,149,780	Books	-
Epinions [164]	120,492	755,760	13,668,320	Social Web	-
Last-FM [165]	1,892	17,632	92,834	Music	-
Movielens 1M [166]	6,040	3,705	998,131	Movies	C/P Gender
Movielens 10M [166]	69,878	10,676	9,973,604	Movies	-
Movielens 20M [166]	138,493	26,744	20,000,263	Movies	-
DAJEEE [167]	-	20,000,000	-	Resources	-
TED [168]	69,000	1,000	100,000	Talks	-
Merlot [169]	-	40,000	-	Resources	-
MACE [170]	-	150,000	-	Resources	-
COCO-RS	**37,704**	**30,399**	**617,588**	**Courses**	**C/P Gender**

Table 4.1: Representative data sets for recommendation.

fees. In comparison with other online course platforms, no third-party control on reliability, validity, accuracy or truthfulness of the course content is performed. All copyright and registered trademarks remain property of their owners. To the best of our knowledge, there exists no other benchmark large-scale data set that includes educational ratings from online courses. Table 4.1 summarises other representative data sets used for recommendation. While several large-scale data sets from other domains have been successfully collected, existing data sets from the educational domain include few users, items, and ratings, and lack sensitive attributes suitable for fairness studies.

4.4.1 Collection Methodology

This section describes our *multi-stage approach* for collecting a large educational ratings data set. The pipeline is summarised in Fig. 4.1, and key stages are discussed below:

1. **Candidate Courses Retrieval**: The first stage is to obtain a list of courses from *Udemy*. We start from the list of courses that are returned by *Udemy APIs*[2], that exposes functionalities to help developers accessing content and building external applications. The script retrieved 43,023 courses, dumped in November 2017.

2. **Course Ratings Download**: To retrieve the ratings released by learners, the script uses the *Udemy APIs* method aimed to return course ratings given the course identifier. The query result gave 6M rows, each with the course id, the timestamp, the rating between 0 and 5 with step 0.5, and the name of the learner who released it.

3. **Data Cleaning and Pruning**: To ensure that the data set can be used for benchmarking recommendation, we took only learners who released at least 10 ratings.

[2] https://www.udemy.com/developers/

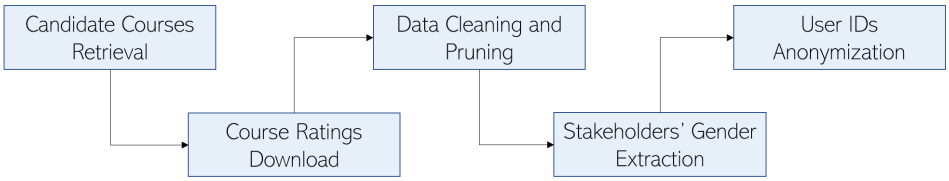


Fig. 4.1: Collection pipeline for COCO-RS data set.

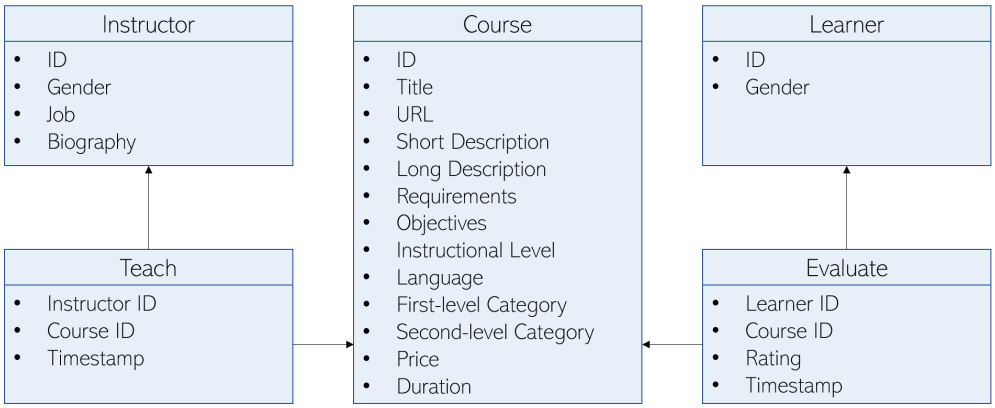


Fig. 4.2: Structure of the COCO-RS data set.

The re-sampled data set includes 37k users, who gave 600k ratings to 30k courses.

4. **Stakeholders' Gender Extraction**: *Udemy APIs* do not give any information regarding sensitive attributes of learners and instructors. By capitalizing on the names of learners retrieved at step 2 and the name of instructors retrieved by an hand-crafted crawler, we used *Gender APIs*[3] to get the gender starting from the name.

5. **User IDs Anonymization**: As we were dealing with sensitive personal attributes, we anonymized both learners IDs and instructors IDs. Each type of ID was re-scaled between 0 and $n\_users - 1$, where n_users corresponds to the number of learners and instructors, respectively. No other identifier is linked to original identities.

4.4.2 Structure and Statistics

COCO-RS is a *CSV*-based collection whose structure in terms of entities and associations is depicted in Fig. 4.2. Text attributes have *Unicode* coding, while languages and timestamps hold *ISO639-1* and *ISO8601* standards, respectively.

Course is the most informative entity. First, *id* and *course URL* provide unique identification attributes. Then, the course is described by *short* and *long descriptions*. *Requirements* and *objectives* list technical and pedagogical needs at the beginning and expected

[3] https://genderapi.io/

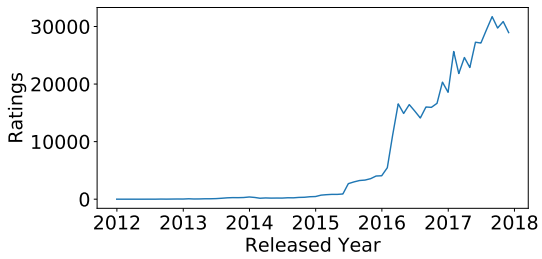


Fig. 4.3: Ratings/year on COCO-RS.

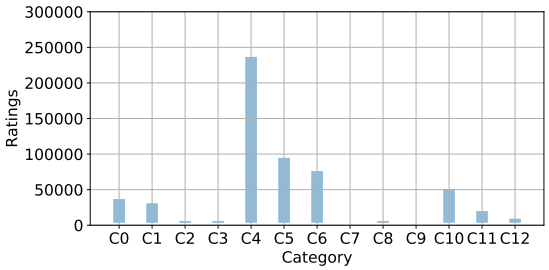


Fig. 4.4: Ratings/category on COCO-RS.

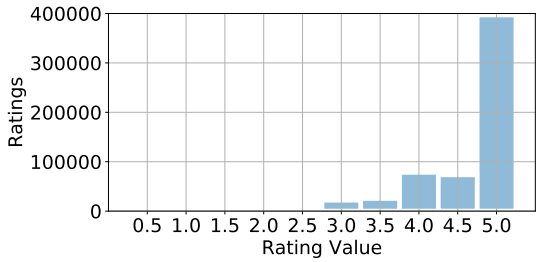


Fig. 4.5: Rating distribution on COCO-RS.

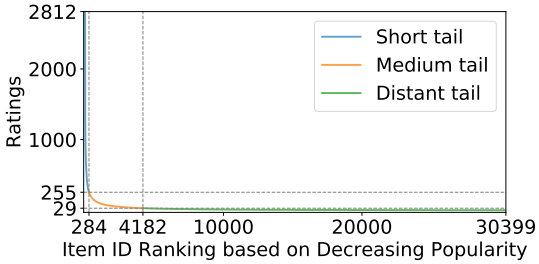


Fig. 4.6: Popularity tail on COCO-RS.

learner skills at the end, respectively. The *language*, the *instructional level* (*beginner*, *intermediate*, *expert*), *first/second-level categories*, and *tags* are listed. Each course has only one first-level category and one second-level category. Other course fields identify the current *price* and the *discount*. Numerical attributes list the *course duration* in hours.

Due to privacy constraints, the *Instructor* and *Learner* entities only include information available on the corresponding public profiles. Each entity instance is uniquely identified by a fake *id*, so that the id stored into the data set does not correspond to the real id of the user. Each instructor is described by the job title and biography. Regarding relationships, in *Teach*, the pairs of *instructor id* and *course id* model the association among instructors and the courses they teach. One instructor can teach more than one course and the same course can have one or more instructors. Each pair of *course id* and *learner id* in *Participate* defines the courses that the learner attended. *Evaluate* contains *learner id*, *course id*, the [0-5] *rating* with step 0.5, and the *timestamp*.

By counting the ratings released over time, it can be observed that there is an increasing trend over years (Fig. 4.3). The data set includes 30,399 courses, distributed over 15 first-level categories (Fig. 4.4), that received 617,588 ratings.

The ratings distribution is unbalanced across categories (avg. 47,506; st.dev. 62,615; min 688; max 240,047). Similarly, the languages distribution along courses follows such a trend. Only 21% of courses do not use English as primary language. Furthermore, the data set includes 37,793 learners. Such ratings are non-uniformly distributed across the range of values they can hold (Fig. 4.5). Most of the ratings have a value of 5, meaning

that learners usually recognize courses as of high-quality. However, only few courses received lots of ratings, while the rest of them has been rarely evaluated. This means that the popularity tail in *COCO-RS* is highly skewed (Fig. 4.6)[4]. The sparsity of the rating matrix is therefore 0.995%. The distribution of users across genders is highly unbalanced, with 84% men and 16% women. The same gender representation is maintained on ratings. Only learners with at least 10 rating are included, while each course can have one or more ratings. The distribution of the number of ratings per courses (avg. 20; st.dev. 72; min. 1; max. 2,812) shows a downward trend, but there is a large number of courses with a lot of ratings. Finally, *COCO-RS* also includes 7,411 instructors, 72% men and 28% women. Few instructors teach several courses (avg. 3; st.dev. 7; min 1; max 453).

4.5 The Proposed Method for Popularity Debiasing

4.5.1 Exploratory Analysis on Traditional Recommenders

To better understand the need of considering recommendation effectiveness and popularity bias metrics, we begin by providing an explorative analysis. We leveraged the Java recommendation framework *LibRec* [171] to evaluate traditional collaborative filtering algorithms on *COCO-RS*, due to their popularity in e-learning contexts [129, 172].

The investigated algorithms are listed below:

- Non-Personalized (NP) baselines:
  - *Random*: randomly recommending items;
  - *MostPop*: recommending the most frequently-consumed items;
  - *ItemAvg*: recommending the items with the highest average rating;
  - *UserAvg*: recommending the items with the highest user average rating.
- Standard Collaborative Filtering (SCF) algorithms:
  - *ItemKNN*: item-based collaborative filter (Cosine, K-NN, $k = 100$);
  - *UserKNN*: user-based collaborative filter (Cosine, K-NN, $k = 100$).
- Matrix Factorization (MF) methods:
  - *SVD++*: gradient descent matrix factorization ($\texttt{LatentFactors} = 40$) [173];
  - *WRMF*: weighted regular matrix factorization ($\texttt{LatentFactors} = 40$)[174].

[4] Existing works often include the first 20% most popular items within the short-tail set and the rest in the long-tail set, but this can lead to unfair intra-set popularity, especially in highly-skewed tails. Therefore, short and long-tail items have been grouped so that the Gini index of the popularity distribution in each set is 33%. Moreover, since distant-tail items receive so few ratings that meaningful cross-user comparison becomes noisy, our works will focus only on short and long tails.

- Learning-To-Rank (LTR) algorithms:
  - *AoBPR*: a variant of BPR manipulating uniform sampling pairs [175];
  - *BPR*: bayesian personalized ranking technique for implicit feedback [119];
  - *Hybrid*: hybrid integrating diversity and accuracy-focused approaches [176];
  - *LDA*: a filtering approach leveraging Latent Dirichlet Allocation [177].

It should be noted that, while we can generate a ranking of the items a user has not evaluated yet by predicting missing ratings, *LTR* methods maximize the ranking quality, generating diverse recommendations against rating-prediction-based algorithms.

Prediction and Ranking Effectiveness

First, we evaluate the recommendation effectiveness, considering metrics that evaluate rating prediction accuracy against those that measure the ranking quality. Like in similar studies [178], we employed a *5-fold cross validation* based on a *user-sampling strategy*. We split the users in five test sets. Each set was the test set of a given fold. In each fold, for each user in the corresponding test set, we selected 5 ratings to be the test ratings, while the rest of their ratings and all the ratings from users not in that test set were the train ratings. Each algorithm was run in both rating prediction and *top-10* item ranking mode. We chose top-10 recommendations since they probably get the most attention and 10 is a widely employed cut-off [93]. *Root Mean Squared Error* (RMSE) evaluated the accuracy of the rating predictions (i.e., the lower the better). *Area Under the Curve* (AUC), *precision*, *recall*, and *Normalized Discounter Cumulative Gain* (NDCG) [162] measured the recommended list accuracy (i.e., the higher the better).

Family	Method	RMSE	AUC	Prec@10	Rec@10	NDCG
MF	SVD++	**0.68**	0.50	0.005	0.001	0.008
NP	UserAvg	0.70	0.50	0.004	0.007	0.005
SCF	UserKNN	0.71	0.68	0.050	0.101	0.095
SCF	ItemKNN	0.76	0.69	0.051	0.102	0.092
NP	ItemAvg	0.78	0.50	0.005	0.008	0.005
NP	MostPop	1.07	0.60	0.023	0.046	0.038
LTR	BPR	2.08	0.69	0.054	0.109	0.094
LTR	AoBPR	2.34	0.69	0.054	0.108	0.094
NP	Random	2.36	0.50	0.004	0.008	0.005
LTR	LDA	4.11	0.66	0.042	0.085	0.074
LTR	Hybrid	4.11	0.55	0.018	0.037	0.029
MF	WRMF	4.12	**0.71**	**0.062**	**0.124**	**0.114**

Table 4.2: The accuracy of the algorithms on rating prediction and top-10 ranking.

Table 4.2 shows the results. The best ones are printed in bold in case they were significantly better with respect to all the others (paired two-tailed Student's t-tests with $p = 0.05$). On RMSE, the *MF* approach, *SVD++*, significantly outperformed all the other schemes. However, the rather simple non-personalized *UserAvg* yielded comparable accuracy to *SVD++*, and was better than other computationally expensive schemes like *ItemKNN* and *BPR*. The latter was significantly better than the other *LTR* approaches. *ItemAvg*, which simply considers an item's average rating, achieved results in line with *ItemKNN*. The *WRMF* method performed, somewhat surprisingly, worse than a lot of the traditional ones. The ranking of the algorithms on *RMSE* is not consistent with respect to other contexts [146]. This confirms that the data set characteristics like size, sparsity, and rating distributions can affect recommendation accuracy [179].

The results on item ranking in terms of nDCG led to a completely different algorithm ranking. *BPR* and *AoBPR* achieved the best performance together with *WRMF* and *UserKNN*. Except *Hybrid*, the *LTR* methods performed consistently better than *MostPop*. In line with the results in [180], *MostPop* performed quite poorly, probably due to the wide range of categories included in the data set. Although *Item-KNN* is rather simple, it performed much better than almost all the *NP* baselines, and reached results comparable to *LTR* schemes. *SVD++* led to mediocre results, while it was the best method in rating prediction. In contrast, *WRMF* achieved the highest accuracy.

While the accuracy of some algorithms is almost equal, the *top-10 lists* greatly varied. In view of these differences, we calculated the average overlap of courses recommended by each pair of algorithms to the same user (Fig. 4.7). Such an overlap is high for pairs like (*WRMF*, *UserKNN*), (*UserKNN*, *LDA*), and (*AoBPR*, *BPR*). *Hybrid* and *SVD++* recommended courses which are not typically proposed by the other algorithms. However, since *MostPop* works well and has some similar recommendations with respect to other algorithms, the latter could tend to recommend popular courses as well.

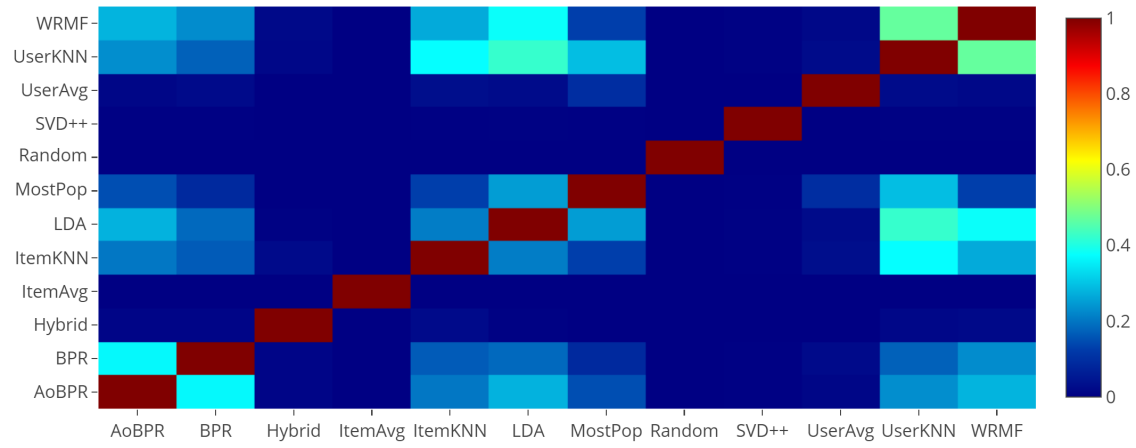


Fig. 4.7: The average overlap per user between the top-10 lists.

Course Popularity Bias

We then explored how the course popularity in data influences the algorithms. To this end, we evaluated how popular are the courses provided by an algorithm, assessing its capability to suggest relevant but not popular ones.

Table 4.3 presents the popularity of the recommended courses as the number of ratings they received. By design, *MostPop* has the highest average popularity, since it recommends best sellers. The recommended courses received about 1,500 ratings on average. *LDA* and *WRMF* also showed a popularity bias, with 586 and 404 ratings per recommended course, respectively. On the other hand, some algorithms are not biased towards course popularity. *SVD++*, *ItemKNN*, *AoBPR*, and *BPR* recommended several courses from the long tail. Interestingly, only *Hybrid* recommended niche and unpopular courses, and its average number of ratings, 11, is lower than the average number of ratings per course in the catalog, 20. Other *NP* baselines achieved a good trade-off.

To obtain a detailed picture, we sorted the courses according to the number of ratings in the dataset and organized them in bins of 1000 courses (Fig. 4.8); the first bin contains the least rated courses, while subsequent ones consider courses of increasing popularity. Then, we counted how many items from each bin an algorithm recommends. Except *Hybrid*, *Random*, and *SVD++*, all the algorithms often recommended courses from the bin of the most popular ones (`bin30`). In *BPR*, course popularity seems to be directly related with the chance of being recommended. *SVD++* and *Hybrid* seem to be good options to recommend niche courses. Interestingly, *Hybrid* tends to recommend more unpopular courses than popular ones.

Receiving a lot of ratings does not imply people liked a course. The correlation between number of ratings and average rating is weak, 0.11. Therefore, we measured the

Family	Algorithm	Avg. / StDev. Rating	Avg. / Std. Dev. Number of Ratings
MF	SVD++	**4.76 / 0.21**	134 / 267
NP	MostPop	4.71 / 0.07	**1545 / 588**
NP	ItemAvg	4.70 / 0.42	15 / 3
MF	WRMF	4.68 / 0.17	404 / 393
LTR	LDA	4.64 / 0.14	586 / 515
SCF	UserKNN	4.63 / 0.21	192 / 296
NP	UserAvg	4.60 / 0.20	341 / 524
LTR	AoBPR	4.58 / 0.25	71 / 152
SCF	ItemKNN	4.55 / 0.23	88 / 168
LTR	BPR	4.55 / 0.27	67 / 144
NP	Random	4.47 / 0.58	20 / 73
LTR	Hybrid	4.44 / 0.72	11 / 57

Table 4.3: The popularity of the recommended items.

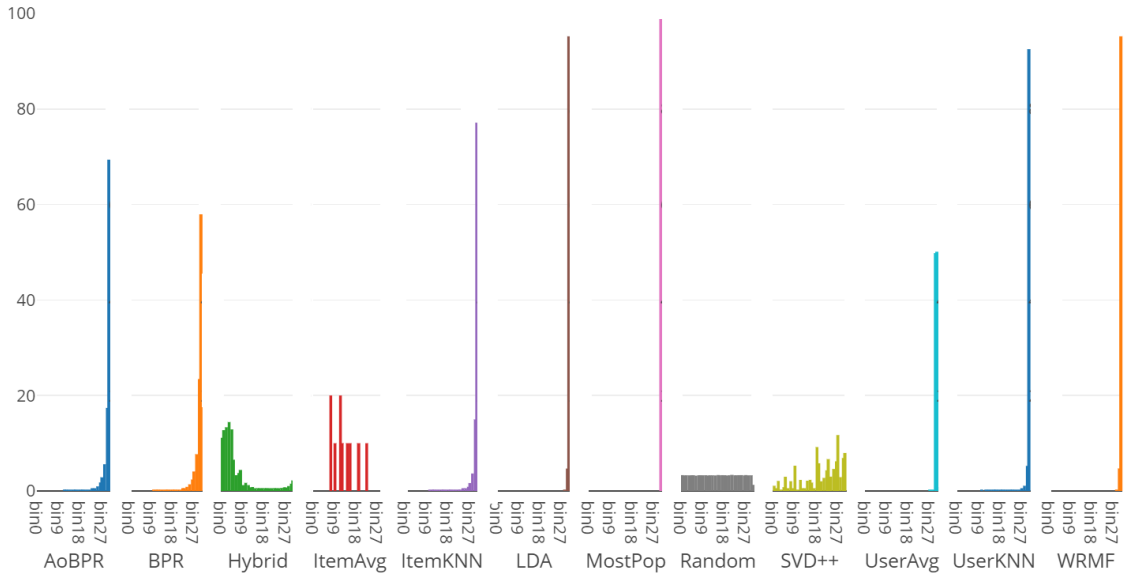


Fig. 4.8: The distribution of the recommended courses over the catalog.

average rating of a course as another popularity indicator. It does not tell if the course is really liked by a large number of people, but it can help to see if some algorithms tend to concentrate on highly-rated and probably less-known courses. Table 4.3 shows that a lot of algorithms recommend courses that were rated, on average, above 4.44 (the global average is 4.47). Furthermore, some algorithms (i.e., *SVD++*, *MostPop*, and *WRMF*) recommended a lot of courses with a high average rating, and low-rated courses are rarely recommended. *LDA* focuses on high-rated courses (4.64) and is significantly different from other *LTR* methods. For algorithms not optimized for rating prediction, the average rating is low and closer to the global average. This means that they do not take the average rating into account and recommended also low-rated courses. The average rating of the *MostPop* recommendations is 4.71, so well-known courses are also top-rated.

Catalog Coverage and Concentration Bias

To check if the recommender system is guiding users to long-tail or niche courses, we should count how many courses in the catalog are recommended. Hence, we looked at the course space coverage and concentration effects of the algorithms.

We counted the number of different courses appearing in the lists (Table 4.4). The results show that the coverage can be quite different across the algorithms. Except *Random*, only *Hybrid* recommend more courses than all other techniques, almost half of the whole catalog. This is in line with the idea behind *Hybrid*: balancing diversity and rating prediction accuracy. However, in our context, we found it achieved good diversity, but low prediction accuracy. Other *LTR* approaches provided a coverage of around 20%, ex-

Family	Algorithm	Coverage	Catalog Percentage	Gini Index
NP	Random	**30399**	**100.00**	**0.16**
LTR	Hybrid	12735	41.90	0.77
LTR	BPR	6514	21.43	0.85
LTR	AoBPR	5857	19.27	0.89
SCF	ItemKNN	4653	15.31	0.89
SCF	UserKNN	1183	3.89	0.89
MF	SVD++	1121	3.68	0.88
MF	WRMF	457	1.50	0.68
LTR	LDA	200	0.65	0.64
NP	MostPop	29	0.09	0.63
NP	UserAvg	14	0.04	0.17
NP	ItemAvg	12	0.04	0.28

Table 4.4: The catalog coverage per algorithm out of 30.399 courses.

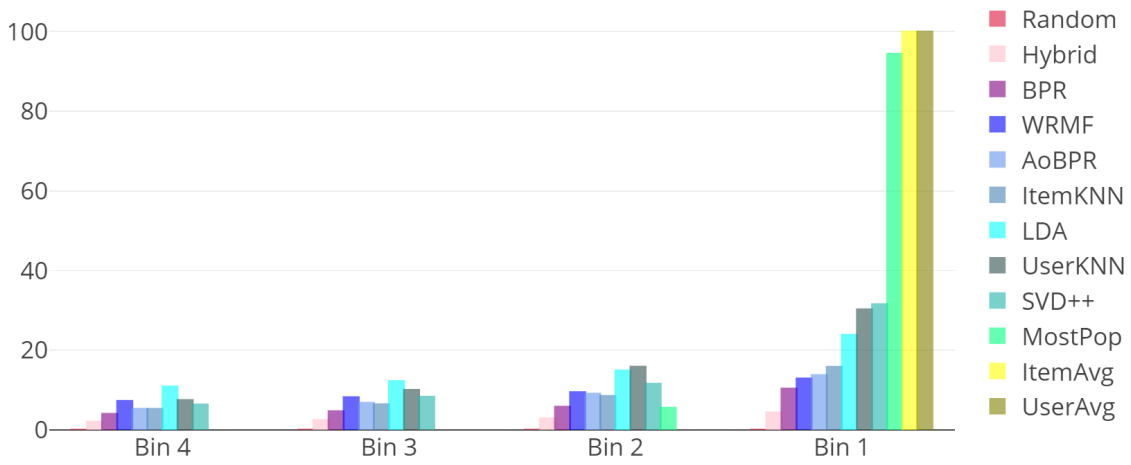


Fig. 4.9: The distribution of the number of recommendations.

cept *LDA* (1%). *KNN* methods showed a limited catalog coverage, confirming the results in [181]. In contrast to [182], the algorithms performing best on prediction accuracy are not the best ones also for the catalog coverage. These differences went unnoticed if only the accuracy was considered.

Catalog coverage does not reveal how often each course was recommended. Thus, we captured inequalities with respect to how frequently the courses appeared. For each course suggested by an algorithm, we counted how often it is contained in the lists of that algorithm. The courses are sorted in descending order, according to the times they appeared in the lists, and grouped in bins of 10 courses. `Bin1` contains the most recommended courses. Fig. 4.9 shows the four bins (out of 3,040) with the 40 most frequently recommended courses. The Y-axis shows the percentage of recommendations the algorithm has given for the courses in the corresponding bin with respect to the total number

of suggestions provided by that algorithm. While *SVD++* and *ItemKNN* recommended a number of different courses, most of them were rarely proposed. *BPR*, *AoBPR*, and *WRMF*, which had a good catalog coverage, provided about 20% of the courses from the 40 most often recommended ones. In Table 4.4, we show the *Gini index* to observe the inequality with respect to how often certain courses are recommended, where 0 means equal distribution and 1 corresponds to maximal inequality [183]. Except for the *NP* baselines, *Hybrid* and *BPR* have the weakest concentration bias. Compared to *BPR*, Hybrid's Gini index is lower, showing a more balanced distribution of recommendations.

Course Category Popularity Bias

E-learning platforms are often equipped with a taxonomy that associates each course with one or more categories. This attribute does not imply the quality of a course, but the distribution of the number of ratings can greatly vary across categories. Nonetheless it is natural, given by the heterogeneity of users and courses, it makes aggregated ratings commonly used by algorithms incomparable across categories and thus prone to bias issues. The course category popularity bias could even influence how learners perceive the recommendations as useful for deepening the knowledge in a preferred category or for fostering a multi-disciplinary knowledge in unexplored categories. Therefore, we focused on the popularity of the category to which courses belong and how the popularity bias affecting course categories in data propagates in the recommended lists.

We counted how many different course categories appeared in the lists. *UserAvg* exhibited only 3 out of 13 different categories, while *MostPop* and *ItemAvg* recommended

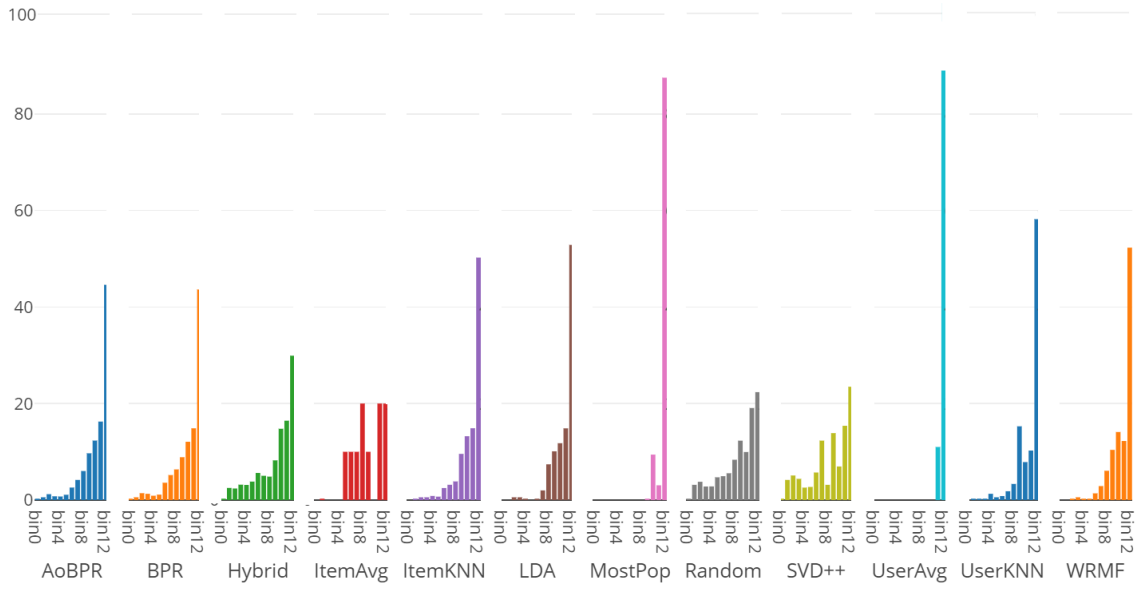


Fig. 4.10: The distribution of the recommended courses over course categories.

5 and 8 categories, respectively. Except for *LDA* (10 categories), all the other algorithms provided a full coverage on categories. To obtain a clear picture, we sorted the 13 categories according to their increasing number of ratings in the dataset. *Bin12* represents the most popular category. For each algorithm, we counted how many recommendations per category were provided in the recommended lists. Fig. 4.10 shows the distribution of the recommendations per category. *BPR*, *ItemKNN*, *LDA*, and *WRMF* showed a bias to the most popular category. More than 50% of their recommendations came from it. *Hybrid* and *SVD++* offered a more uniform distribution across categories.

In this context, it was also important to measure how much each algorithm reinforces or reduces the bias to a given category. Fig. 4.11 shows the bias related to course category popularity. Each rectangle shows the percentage of increment / decrement on the recommended courses per category with respect to the ratings per category in the dataset. Considering that "*development*" is the most popular category, when producing recommendations, MostPop reinforces its popularity by 50%. *Hybrid* and *SVD++* caused a 10% popularity reduction in courses of this category. Hence, their recommendations can meet the needs of those not interested only in "*development*" courses.

To sum up, the differences in accuracy between some algorithms are very small. For instance, the best-performing techniques, SVD++ and UserAvg, have a difference of 0.02 in RMSE. The algorithm ranking based on nDCG was quite different. However, the analysis regarding catalog coverage and concentration showed that our findings on popularity bias can be contrasting with respect to the ones observed for accuracy. **Hence, we point out that, if the goal is to guide learners to different areas of the course catalog, algorithms should not be optimized on accuracy alone**. In fact, Hybrid did not perform well on prediction accuracy, but covered half of the catalog. In contrast, SVD++ had a better prediction accuracy, but recommended only 4% of the courses.

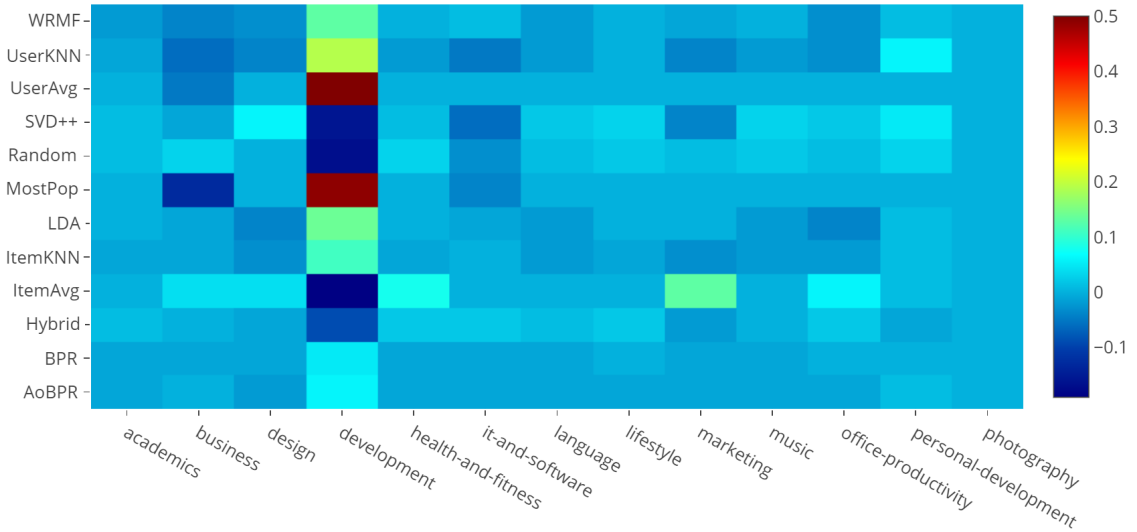


Fig. 4.11: The reinforcement produced by algorithms over course categories.

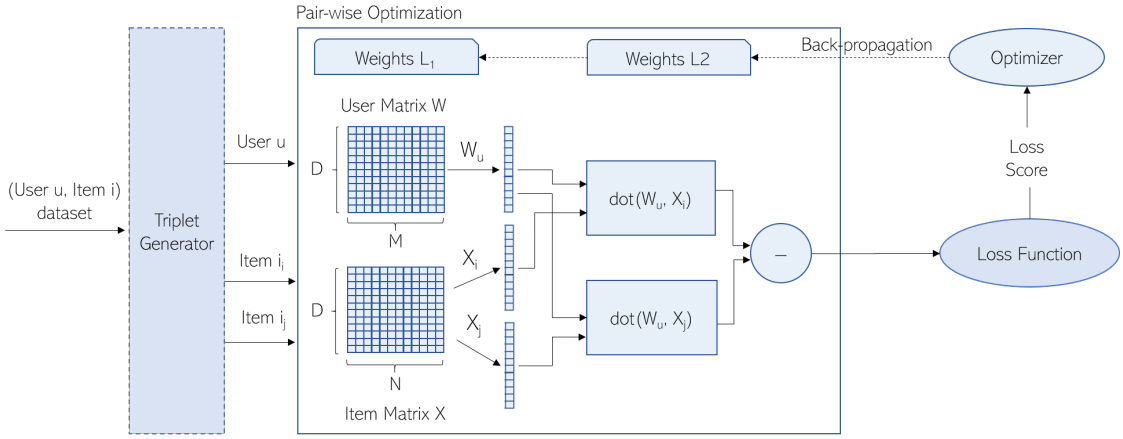


Fig. 4.12: The neural personalized ranking model explored in our analysis.

4.5.2 Exploratory Analysis on Neural Recommenders

In this section, we extend our exploratory analysis to one of the most widely adopted algorithms, Neural Personalized Ranking (NPR) [118]. We show that it is vulnerable to imbalanced data and tends to produce biased recommendation based on PSP and PEO.

Neural Personalized Ranking Description

NPR is one of the most influential methods to solve recommendation problems, which is the foundation of many cutting edge personalized ranking algorithms [184, 185, 186]. This algorithm adopts matrix factorization [187] as the base, and estimates model parameters θ through a pair-wise objective. The latter maximizes the margin between the relevance $f(u, i_i|\theta)$ predicted for an observed item i_i and the relevance $f(u, i_j|\theta)$ predicted for an unobserved item i_j, given interactions in R (Fig. 4.12), defined as:

$$\underset{\theta}{\operatorname{argmax}} \sum_{u \in U, i \in I_u^+, j \in I_u^-} \delta(f(u, i_i|\theta) - f(u, i_j|\theta)) \tag{4.8}$$

where $\delta(\cdot)$ is the sigmoid function, I_u^+ and I_u^- are the sets of items for which user u's feedback is observed or unobserved, respectively.

Before model training, embedding matrices are initialized with values uniformly distributed between 0 and 1, and the loss function is transformed to the equivalent minimization problem. For 20 epochs, the model is served with batches of 1024 triplets. More precisely, for each user u, we create 10 triplets (u, i, j) per observed item i; the unobserved item j is randomly selected. The optimizer used for gradient update is *Adam*[5].

[5] In our work, we are more interested in better understanding algorithm characteristics beyond accuracy, so the further accuracy improvements that can probably be achieved through hyper-parameter tuning would not substantially affect the outcomes of our analyses.

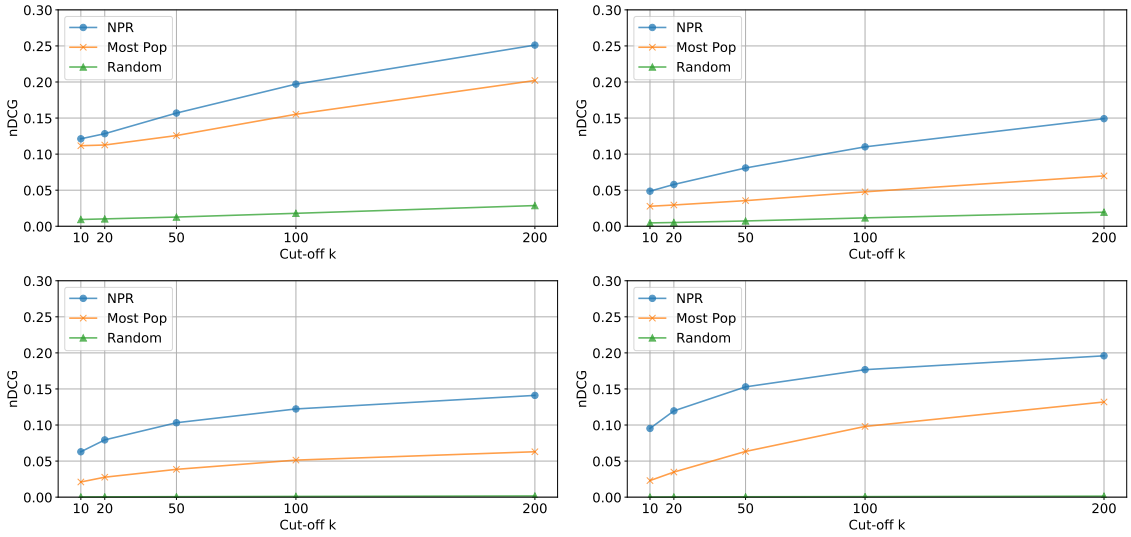


Fig. 4.13: Recommendation accuracy for the explored recommenders.

Recommendation Accuracy Analysis

Throughout our exploratory analysis, *NPR* is compared with two non-personalized baselines, namely popularity-based recommender *Most Pop* and random guess recommender *Random*. Such baselines are opposite with respect to popularity bias: a random recommender is insensitive to popularity and uniformly recommends items across the catalog; a popularity-based recommender ignores tail items, and suggests the same few popular items to everyone[6]. For statistic significance, we use paired two-tailed Students t-tests with a p-value of 0.05.

We adopt two datasets with diverse item distributions: MovieLens 1M (ML1M) [166] contains 1M ratings applied to 3K movies by 6K users of the online service MovieLens. Each user rated at least 20 movies; COCO-RS [18] contains 600k ratings applied to 30K courses by 37K users of the educational platform Udemy. Each user rated at least 10 courses. We treated ratings as positive feedback, i.e., users are interested in rated items.

Figure 4.13 plots the nDCG@k accuracy metric measured for NPR, Most Pop, and Random in Movielens 1M over the ALL test setup (top-left) and the UAR test setup (top-right); in COCO-RS over the ALL (bottom-left) and the UAR test setups (bottom-right). Evaluated on all test ratings (ALL Test Setup - left plots), NPR (blue line) significantly outperformed Most Pop (orange line) and Random (green line) on both datasets. However, the performance achieved by Most Pop seems to be highly competitive. This might reveal that the rating per item distribution on the test set is highly unbalanced towards popular items, and it can bias evaluation metrics in favor of popular items. Furthermore,

[6]Even though comparing an algorithm against Most Pop and Random has been previously well studied [146, 17], there is no evidence on how the new bias metrics model their outcomes.

we observed that the gap in nDCG between NPR and Most Pop increases at higher cut-offs. Most Pop can identifying relevant items, but tends not to put them in top positions.

To measure the robustness of the involved recommenders to rating-per-item unbalances within the test set, we examined an alternative experimental configuration, where the statistical role of popularity gets reduced. More precisely, we considered a test set, which is a subset of the original test set, where all items have the same amount of test ratings (UAR Test Setup) [188]. Results are depicted on the right plots in Figure 4.13. It can be observed that nDCG scores decrease under the latter evaluation setup for both NPR and Most Pop. Both algorithms have some degree of bias towards popular items.

The fact that NPR strongly adhere to Most Pop might suggest that a recommender system optimized for recommendation accuracy would not by default result in recommending sets with low popularity bias estimates. **We conjecture that optimizing for accuracy, without explicitly considering popularity bias, has an adverse impact on the latter**. To demonstrate this, we present a number of results on popularity.

Popularity Bias Analysis

Motivated by the accuracy results, we analyze the ranking probability distributions of NPR, Most Pop, and Random. To this end, Figure 4.14 plots bias scores in Movielens 1M as recommendation parity (top-left) and equality of treatment (top-right); in COCO-RS as recommendation parity (bottom-left) and equality of treatment (bottom-right). From the figure, we can conclude that NPR and Most Pop fail to hold good levels of recommendation parity (left plots). This derived from the fact that their PSP@k is tremendously

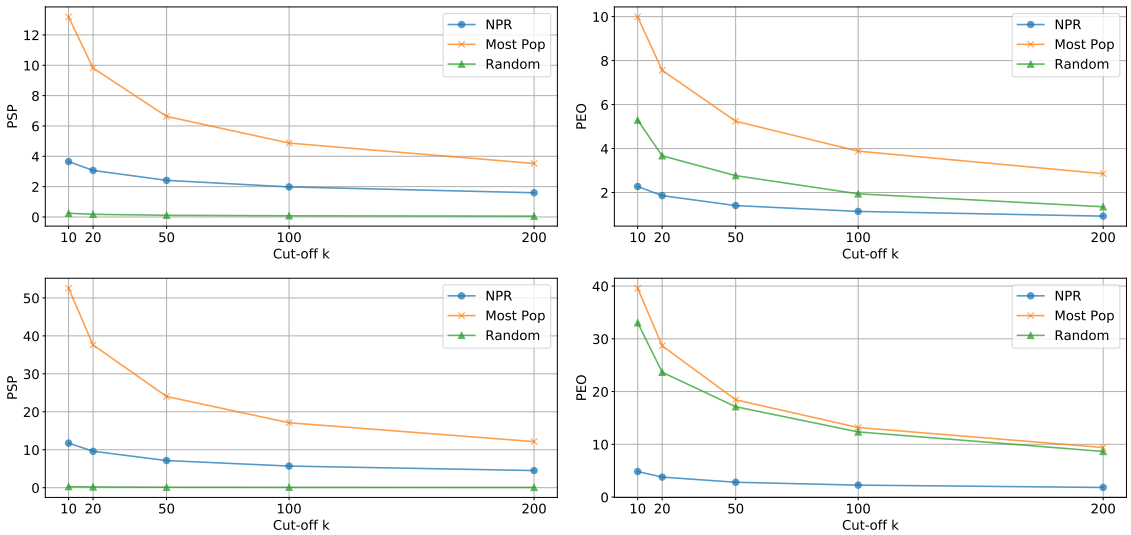


Fig. 4.14: Popularity bias for the explored recommenders.

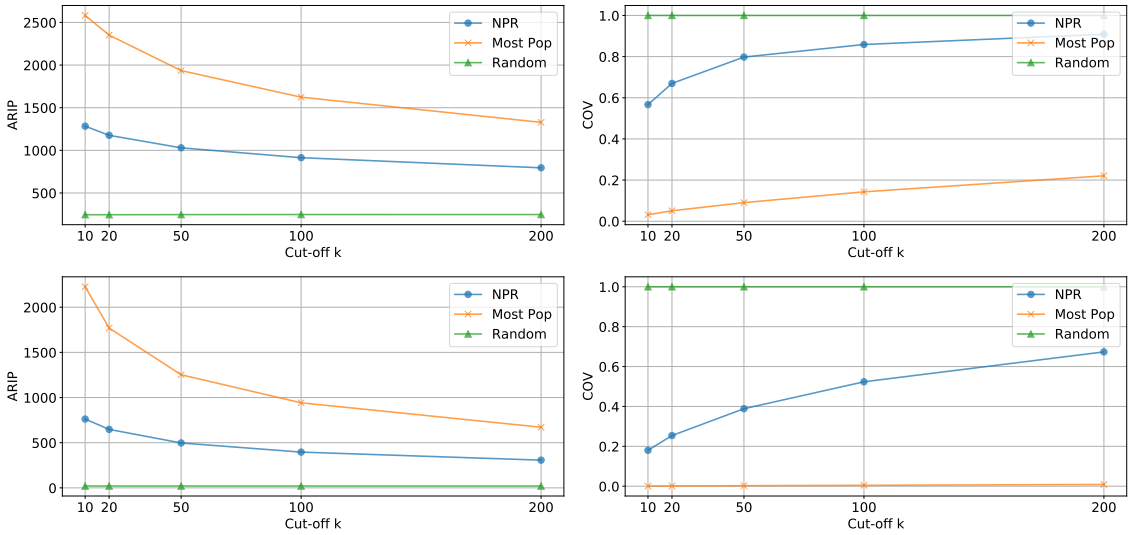


Fig. 4.15: Item catalog metrics for the explored recommenders.

higher than the PSP@k achieved by Random, which has a perfect recommendation parity by default. Furthermore, the results on equality of treatment (right plots) point to a different view. PEO@k can identify situations in which some items, often those which are more popular, might get smaller errors. From the figure, Most Pop and Random bring to very high PEO@k scores. This is due to the fact that they achieve high TPR for few popular items and null TPR for the rest of the catalog.

We also plotted the Average Recommender Item Popularity (ARIP) and the Item Catalog Coverage (COV) for the explored recommenders. The results are plotted in Figure 4.15 - the average recommended item popularity (top-left) and item catalog coverage (top-right) in Movielens 1M; the average recommended item popularity (bottom-left) and item catalog coverage (bottom-right) in COCO-RS. The average popularity of an item recommended by NPR is around 50% lower with respect to the one measured for Most Pop (left plots). It seems to be a significant improvement at first glance but, considering the item popularity distribution, we can observe that such popularity values along cut-offs fall into the head item set. Hence, NPR still tends to recommend very popular items. On item catalog coverage (right plots), NPR achieves full coverage at $k = 200$. This means that, at smaller cut-offs, the recommended items does not greatly differ among users, and a tiny part of the catalog is exposed.

Popularity Bias Diagnosis

Starting from the findings on recommendation accuracy and popularity bias, we seek to uncover some internal mechanics that foster bias propagation in NPR.

First, we applied the t-SNE dimensionality reduction technique to both the embedding

matrices, so that user and item embeddings are mapped onto the same bi-dimensional space. Figure 4.16 plots user and item embeddings computed by NPR on Movielens 1M (left) and COCO-RS (right). It can be observed that user embeddings are mapped closer to short-tail items than long-tail ones. It follows that, when computing user-item relevance, the items closest to a user often come from the head.

To have a detailed view, we also analyzed the distribution of user-item relevance scores for observed head-mid items and observed tail items, separately. Figure 4.17 plots the relevance distributions for head-mid and tail item-user comparisons by NPR in Movielens 1M (left) and COCO-RS (right). Such distributions are surprisingly cleanly Gaussian, and are significantly different. It can be observed a tendency of head-mid observed items of getting higher relevance to users. This should be considered as an undesired behavior of the algorithm that is under-considering observed tail items regardless of the real user interest.

Furthermore, we analyzed the performance in terms of overall, intra- and inter- pair-wise accuracy for head-mid and tail observed items. To this end, we randomly sampled four sets of $100 \times N$ triplets (u, i, j) each, where: the first set includes observed head-mid items as i and unobserved head items as j; the second set includes observed head-mid items as i and unobserved tail items as j; the third set includes observed tail items as i and unobserved head-mid items as j; and the fourth set includes observed tail items as i and unobserved tail items as j. For each set, we computed the recommender pair-wise accuracy on predicting a higher relevance for observed items than unobserved ones.

Figure 4.18 plots the NPR accuracy on getting higher relevance for observed than unobserved items for NPR in Movielens 1M (left) and COCO-RS (right). It can be observed that NPR fails in assigning higher relevance to observed head-mid items, when they were compared to unobserved head-mid items (left-orange bar). It performs better when observed head-mid items are compared against unobserved tail items (left-green bar). Similarly, NPR fails to calculate higher relevance when comparing observed tail items and unobserved head-mid items (right-green bar); it achieves higher accuracy for observed/unobserved tail item comparisons (right-orange bar).

Considering the inter pair-wise accuracy gap between observed head and observed tail items (gap between green bars), we uncover that the recommender fails more frequently on giving higher relevance to observed tail items when compared against unobserved head-mid items (right-green bar); conversely, it performs significantly better in the opposite case, i.e., observed head-mid item against unobserved tail item (left-green bar). Hence, tail items, even when of interest, are significantly under-ranked.

The correlation between (i) the difference in user-item relevance and (ii) the relative difference in popularity for observed and unobserved items is highly positive: SpearmanCorr=0.65 for NPR. This means that there is a direct relation between popularity and relevance, in line with the value obtained by Most Pop: Corr=1.00. On the other hand, it is in contrast with the correlation measured for Random: Corr=0.00.

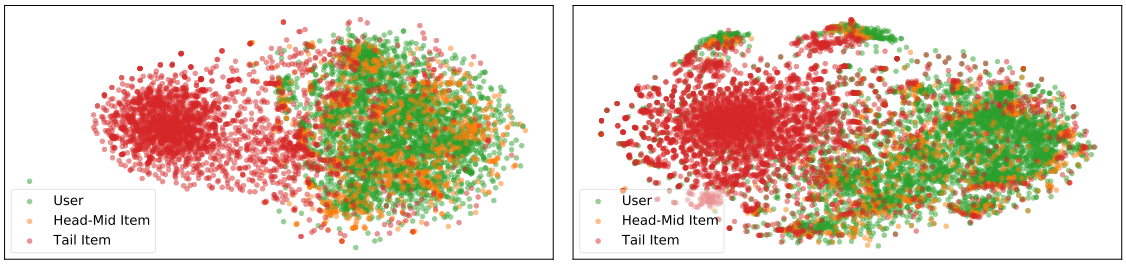


Fig. 4.16: T-SNE embedding representation for NPR.

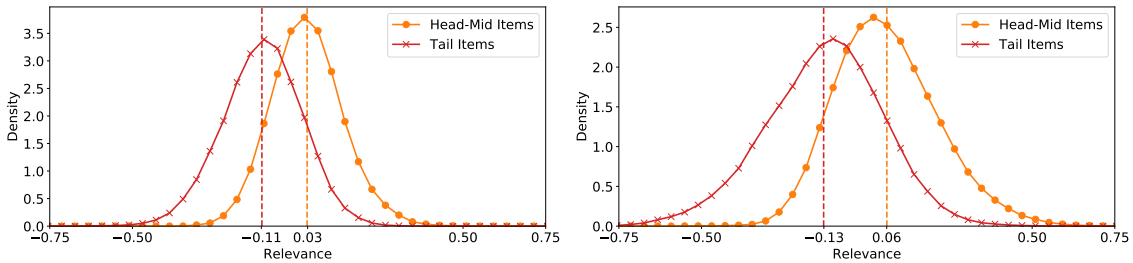


Fig. 4.17: Relevance score distributions for NPR.

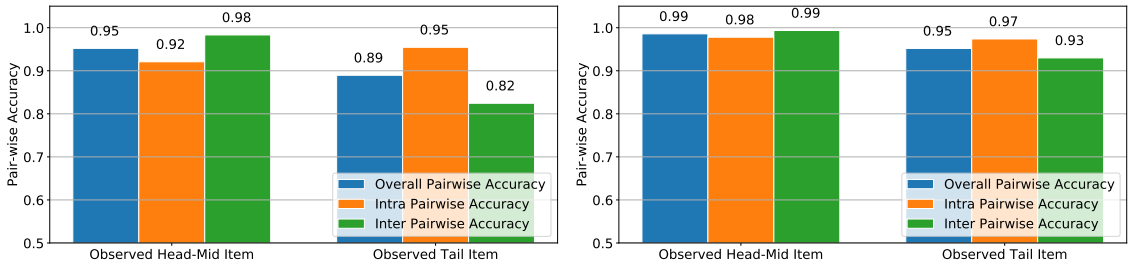


Fig. 4.18: Pair-wise accuracy for NPR.

Hence, **we conjecture that minimizing the absolute value of such a correlation during optimization might have a positive impact on the targeted trade-off** between recommendation accuracy and popularity bias. To demonstrate this, we design a treatment that reduces the strength of this correlation while optimizing for accuracy.

4.5.3 Methodology for Popularity Bias Mitigation

With an understanding of some deficiencies in NPR internal mechanics, we now investigate how we can optimize a recommender system for overcoming them, and seek to generate popularity-unbiased recommendations. To this end, we optimize for equality in pair-wise accuracy between observed head-mid and tail items by minimizing both (i)

the pair-wise error specified in Eq. 4.8, and (ii) the correlation between the difference in relevance and in popularity over observed and unobserved items. The reader notices that our procedure will not rely on any arbitrary split between head-mid and tail items.

The procedure is composed by the following steps:

1. **Triplet Generation (SAM)**. For each user u, we create $t = 10$ triplets (u, i, j) per observed item i; the unobserved item j is selected among the items less popular than i for half of the user triplets, and among the items more popular than i for the other half. We define this set of triplets as T. This step will make sure that there is sufficient data for computing the correlation-based regularization defined at Step 3.

2. **Batch Grouping**. We sample triplets in T in batches of $m = 1024$ samples in order to set up an iterated stochastic gradient descent. Each batch is balanced in terms of triplets where i is more popular than j and j is more popular than i, to ensure the same representation for both sets in a batch.

3. **Regularized Optimization (REG)**. Each training batch we considered, $T_{batch} = \{(u_{t_b}, i_{t_b}, j_{t_b}) \,|\, 0 \leqslant b \leqslant m\}$, is fed into the model that follows a regularized paradigm derived from the standard pair-wise approach (Fig. 4.12). The loss function can be formalized as follows:

$$\underset{\theta}{\operatorname{argmax}} \; (1 - \alpha) \; acc(T_{batch}) + \alpha \; reg(T_{batch}) \tag{4.9}$$

where $\alpha \in [0, 1]$ is the regularizer weight, acc(.) is the standard pair-wise objective, defined as follows:

$$acc(T_{batch}) = \sum_{b=0}^{m} \delta(f(u_{t_b}, i_{t_b}|\theta) - f(u_{t_b}, j_{t_b}|\theta)) \tag{4.10}$$

and reg(.) regularizes the correlation involving (i) the residual between observed and unobserved item relevance and (ii) the relative popularity of the observed item w.r.t. the popularity of the unobserved item, as follows:

$$reg(T_{batch}) = 1 - |Corr(A, B)| \tag{4.11}$$

$$A_b = f(u_{t_b}, i_{t_b}|\theta) - f(u_{t_b}, j_{t_b}|\theta) \;\; 0 \leqslant b \leqslant m \tag{4.12}$$

$$B_b = \begin{cases} 1, & \text{if } \sum_{u \in U} R(u, i_{t_b}) \geqslant \sum_{u \in U} R(u, j_{t_b}) \\ 0, & \text{otherwise} \end{cases} \quad 0 \leqslant b \leqslant m \tag{4.13}$$

where $\sum_{u \in U} R(u, i)$ and $\sum_{u \in U} R(u, j)$ are respectively the popularity level of item i and item j gather in the original dataset.

The model is thus penalized if its ability to predict an higher relevance for an observed item is better when it is more popular than the unobserved item.

Steps 2 and 3 are repeated for all the batches, until convergence.

4.5.4 Evaluation

In this section, we empirically evaluate the proposed model w.r.t. the two proposed bias metrics as well as the recommendation accuracy. We aim to answer four key research questions:

- **RQ1** What are the effects of the proposed data sampling, regularization, and their combination on recommendations?
- **RQ2** How does the regularization weight affect our treatment?
- **RQ3** What is the impact of our treatment on internal mechanics?
- **RQ4** How does our treatment perform compared with other state-of-the-art debiased models in mitigating popularity bias and preserving recommendation quality?

Datasets

We used the same datasets leveraged for the exploratory analysis. The first one, *MovieLens 1M* [166], includes 1M ratings applied to 3K movies by 6K users. The second one, *COCO-RS* [18], includes 37K users, who gave 600K ratings to 30K online courses. This can help us check if the proposal is generalizable across different domains, as they hold different popularity distributions across the item spectrum.

Experimental Setup

In the experiments, we need to consider both recommendation accuracy and recommendation bias. For the recommendation bias perspective, we report PSP@k and PEO@k. As for the recommendation quality we adopt nDCG@k. We report the results with k = 10, 20, 50, 100, 200. We also measured precision and recall in the experiments, which show the same pattern as nDCG, hence we do not report them for conciseness.

We compare the trade-off achieved by the proposed regularized approach, namely *NPR+SAM+REG*, against the one obtained by the following baselines:

- *NPR+$\mathcal{J}$* [146]. It applies a debiased data sampling strategy to the dataset, before training NPR. The main idea is to focus the sampling only on triplets (u, i, j) where i is less popular and j is more popular.
- *NPR+W* [152]. It re-ranks the output of NPR according to a weight-based strategy. The relevance returned by NPR for a given item is multiplied with a weight inversely proportional to the popularity of that item, before re-ranking.
- *NPR+S* [153]. For each user, it iteratively builds the re-ranked list by balancing the contribution of the relevance score returned by NPR and of the diversity level related to two item sets, namely head-mid and tail sets.

We performed a temporal train-test split that includes the last 20% of ratings released by a user on the test set and the remaining 80% oldest ones on the training set. Embedding matrices are initialized with values uniformly distributed in the range [0, 1]. The model is served with batches of 1024 triplets, chosen from the corresponding set. Before each of 20 epochs, we shuffle the training batches. The optimizer used for gradient update is Adam. Models are coded in Python on top of Keras, and trained on GPUs.

As we deal with popularity bias issues, we mostly focus on a testing configuration where the statistical role of popularity in accuracy metrics gets reduced. For the sake of completeness, we also include results on the common configuration that includes a full test set. More precisely, we run the following setups:

- *UAR*. The test set consists of a subset of the original test set where all items have the same amount of ratings, as described in [188].
- *ALL*. The test set includes all the original test ratings as created by the training-test procedure describe above.

RQ1: Effects of Model Components

In this subsection, we run ablation experiments to assess (i) the influence of the new data sampling strategy and the new regularized loss on the model performance, and (ii) whether combining these two treatments together might improve the trade-off between accuracy and popularity bias metrics.

To answer these questions, we compare the base model (NPR) against NPR trained on data created through the proposed sampling strategy only (NPR+SAM), NPR optimized through the proposed regularized loss only (NPR+REG), and NPR combining both our treatments (NPR+SAM+REG). Results on accuracy and popularity are discussed below.

Figure 4.19 plots nDCG scores in Movielens 1M over the ALL test (top-left) and the UAR test (top-right) and in COCO-RS over the ALL test (bottom-left) and the UAR test (bottom-right). We can observe that all the newly introduced configurations (green, orange, and red lines) have a loss in accuracy w.r.t. the base NPR model (blue line), if we

considered all test ratings (left plots), leading to intractable mitigation. However, the gap in accuracy among base and regularized models is positively reduced, when we consider the same number of test ratings for all the items (right plots). We argue that, as large gaps of recommendation accuracy in the ALL setup reflect only a spurious bias in the metric and the underlying test set (see [188] for a demonstration), the real impact of our treatments on accuracy must be considered on the UAR setup. From right plots, we can conclude that there is a negligible gap in accuracy across models. The impact of each treatment on nDCG seems to vary across data sets. On ML1M, interestingly, combining our data sampling and regularized loss slightly improves accuracy, while when treatment

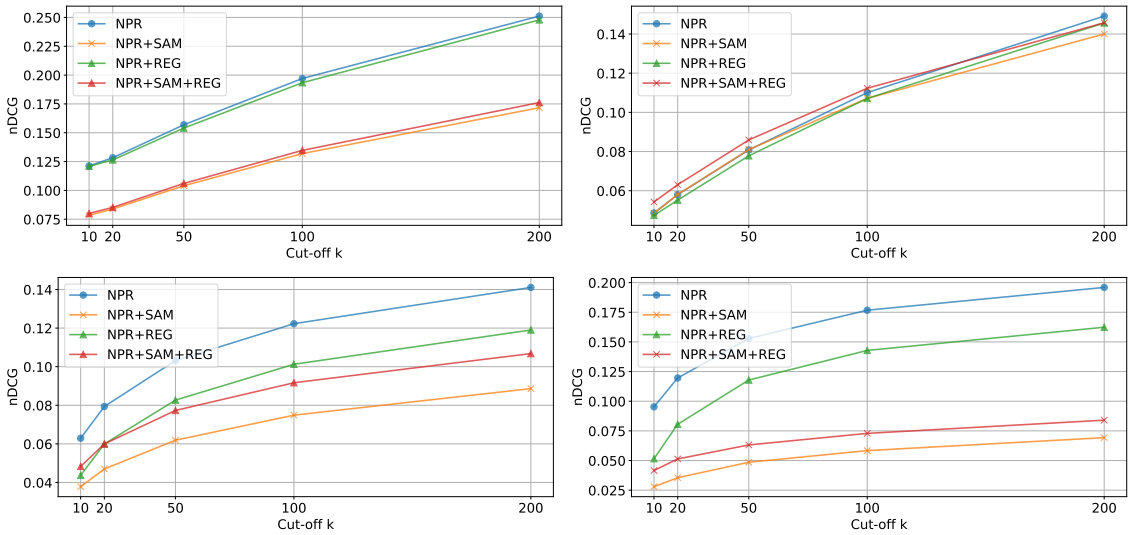


Fig. 4.19: Recommendation accuracy under our different treatment configurations.

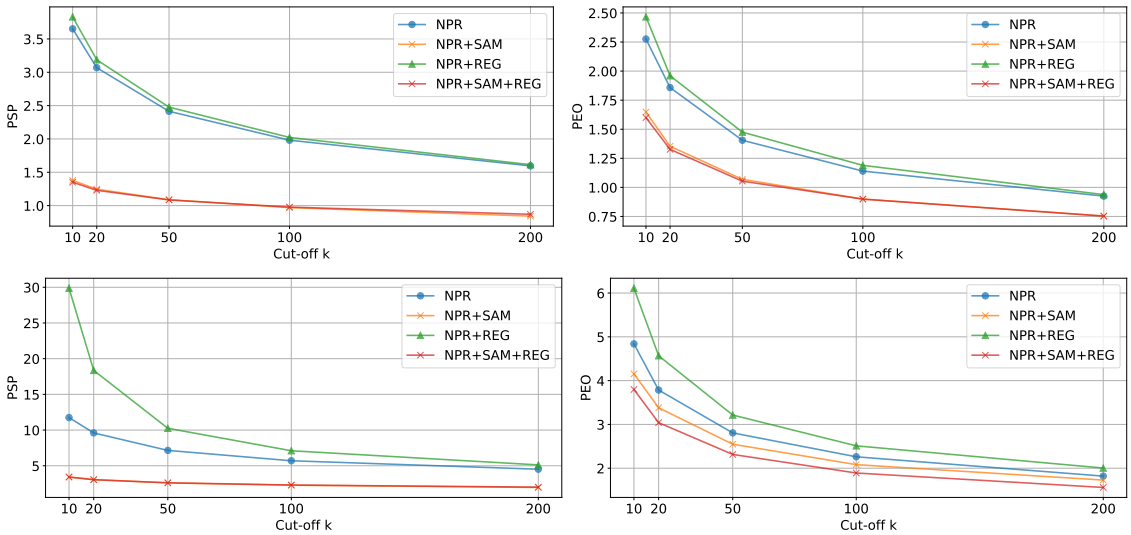


Fig. 4.20: Popularity bias metrics under our different treatment configurations.

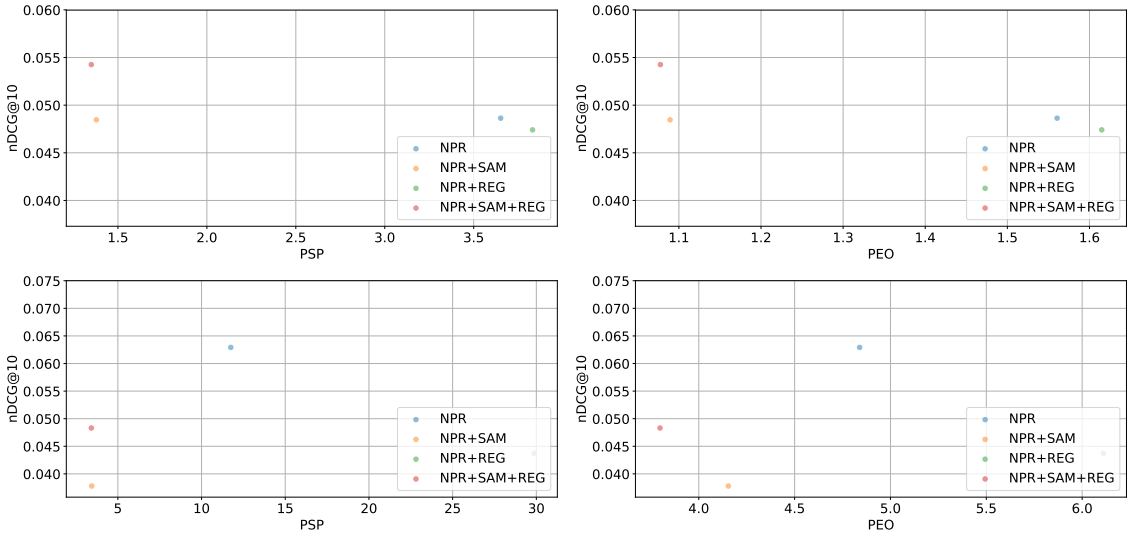


Fig. 4.21: Trade-off under our different treatment configurations.

is individually performed there is no difference w.r.t base NPR.

Figure 4.20 plots bias metrics in Movielens 1M as recommendation parity (top-left) and equality of treatment (top-right) and in COCO-RS as recommendation parity (bottom-left) and equality of treatment (bottom-right). We can observe that our data sampling (orange line) and our combination of data sampling and regularized loss (red line) positively impact PSP and PEO metrics, while our regularized loss alone (green line) still keeps comparable bias w.r.t. plain NPR (blue line). Furthermore, there is no statistical difference on PSP (left plots) between NPR+SAM (orange line) and NPR+SAM+REG (red line). It follows that the regularized loss does not allow to improve PSP directly. On other other hand, NPR+SAM+REG can significantly reduce PEO w.r.t. NPR+SAM. It follows that our regularized loss makes the true positive rates more similar among items.

Overall, our data sampling and regularized loss together achieve a better trade-off between recommendation accuracy and popularity bias. Figure 4.21) plots nDCG@10 scores against recommendation parity (top-left) and equality of treatment (top-right) in Movielens 1M, and nDCG@10 scores against recommendation parity (bottom-left) and equality of treatment (bottom-right) in COCO-RS. Each point represents an algorithm. The closer a point is to top-left corner (high nDCG, low PSP/PEO) the better.

RQ2: Impact of Regularization Weight

We investigate how the model performs when we vary the weight given to the regularization term α within the new proposed loss function. For conciseness, we only report experimental results on ML1M dataset, but please note that the results on COCO-RS show similar patterns.

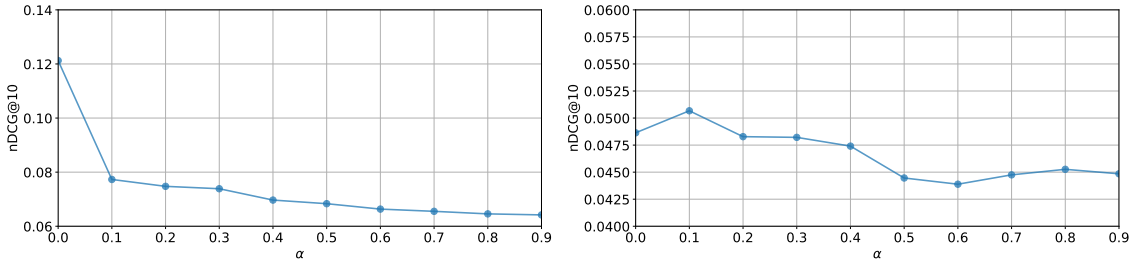


Fig. 4.22: NPR+SAM+REG recommendation accuracy over α.

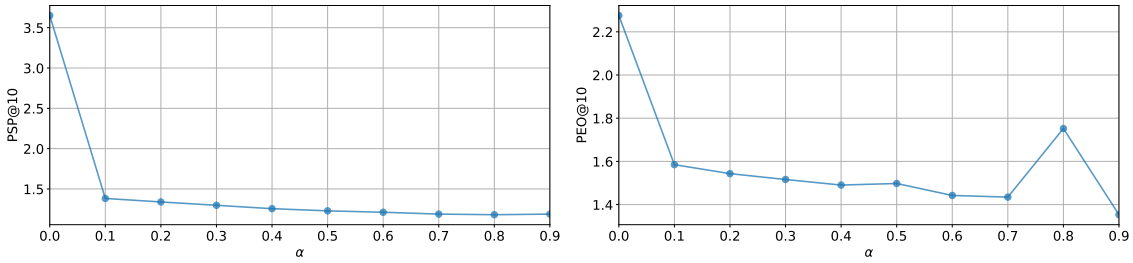


Fig. 4.23: NPR+SAM+REG popularity bias over α.

We vary the trade-off regularizer α and plot the results on accuracy and popularity bias metrics in Figure 4.22 and Figure 4.23, respectively. More precisely, Figure 4.22 plots nDCG@10 score in Movielens 1M over the ALL test (left) and the UAR test (right) by varying α, while Figure 4.23 plots bias metrics at $k = 10$ in Movielens 1M as recommendation parity (left) and equality of treatment (right) obtained by varying α. The x-axis coordinates indicates the value of α, while the y-axis shows the value measured for the corresponding metric at that value of α. Figure 4.22 demonstrates that with larger weight for the regularization term, the recommendation quality decreases more. For the bias reduction performance, as presented in Figure 4.23, both PSP and PEO does not largely vary over different α values, which is most likely due to the nature of the regularization. To balance accuracy and debiasing, setting $\alpha = 0.5$ is a reasonable choice.

RQ3: Impact on Internal Mechanics

In this subsection, we run experiments on ML1M and COCO-RS to assess (i) whether the debiased model can effectively reduce the gap between head-mid and tail relevance distributions, and (ii) whether it can effectively balance inter- and intra- pair-wise accuracy among head-mid and tail items.

To answer the first question, we plot the relevance distributions for head-mid and tail items in Figure 4.24, where the orange lines are calculated on head/mid-item-user pairs, and the red lines are calculated on tail-item-user pairs. More precisely, Figure 4.24 plots relevance score distributions for head-mid and tail item-user comparisons in Movielens

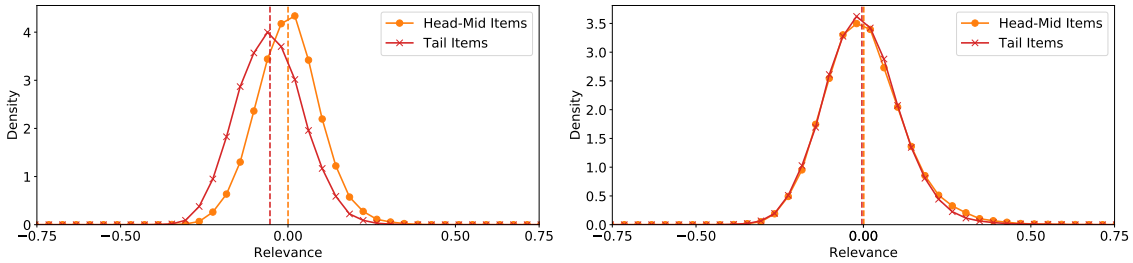


Fig. 4.24: Relevance score distributions for NPR+SAM+REG.

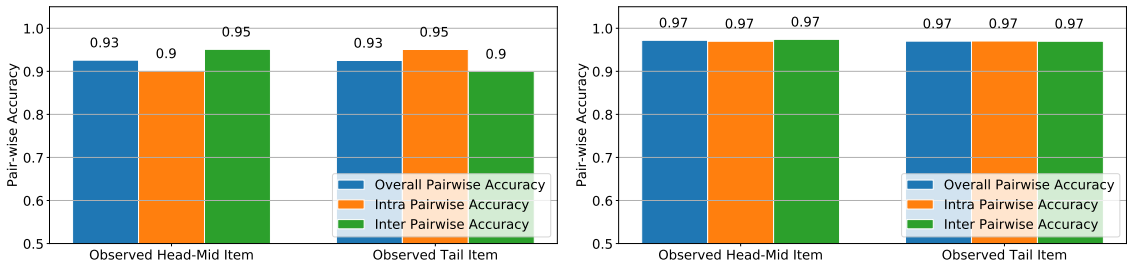


Fig. 4.25: Pair-wise accuracy for NPR+SAM+REG.

1M (left) and COCO-RS (right). From the figure, we can conclude that the proposed model can effectively enhance score distribution similarity. For ML1M, we moved from a Spearman correlation of 0.65 in the NPR model to 0.10 in the regularized model. For COCO-RS, such correlation went from 0.47 in NPR to 0.06 in the regularized model.

For the second question, we show the pair-wise accuracy for observed head-mid and tail items in Figure 4.25. Such a figure plots the accuracy on getting higher relevance for observed than unobserved items in ML1M (left) and COCO-RS (right). From the blue bars, we can see that the overall pair-wise accuracy for head-mid items has been reduced on both data sets of around 2%, while it has been increased of 4% on ML1M and 2% on COCO for tail items. Interestingly, the regularized approach makes it possible to reduce the difference in inter pair-wise accuracy, which was a damaging effect caused by algorithmic bias. Such a gap moved from 16 to 5 percentage-points in M1M. In COCO-RS, the gap reached 0 percentage-points.

RQ4: Comparison with Baselines

We next compare the proposed NPR+SAM+REG model with some state-of-the-art alternatives to assess (i) how the proposed model performs in comparison with baselines for recommendation accuracy and popularity bias individually, and (ii) how the proposed NPR+SAM+REG model manages the trade-off among such two objectives, com-

pared with baselines. To answer these questions, we report accuracy and bias metrics for different models in Figure 4.26 and Figure 4.27. More precisely, Figure 4.26 plots nDCG scores in Movielens 1M over the ALL test (top-left) and the UAR test (top-right) and in COCO-RS over the ALL test (bottom-left) and the UAR test (bottom-right). Figure 4.27 plots bias scores in Movielens 1M as recommendation parity (top-left) and equality of treatment (top-right) and in COCO-RS as recommendation parity (bottom-left) and equality of treatment (bottom-right). Baselines marked as (*). Our proposed model is identified by a blue line.

From Figure 4.26, the proposed NPR (blue line) can achieve an accuracy score comparable to NPR+J (orange line), while it largely outperforms the other baselines (green and red lines), over all datasets, on both test setups. The configuration that shows a statistically significant different between NPR+SAM+REG and NPR+J is related to the UAR test setup on COCO-RS (bottom right plot). This may be caused by the skewed item popularity distribution in COCO-RS, that makes it harder for NPR+SAM+REG to build meaningful triplets where the observed item is less popular than the unobserved item.

From Figure 4.27, it can be observed that our NPR+REG+SAM model (blue line) largely reduces PSP w.r.t. the other baselines (left plots) on both datasets. On ML1M (top left plot), NPR+W exhibits the highest bias on recommendation parity. NPR+J and NPR+S achieve comparable scores between each other, but smaller than NPR+W. On COCO-RS (bottom left plot), NPR+S scores get worse. Smaller improvements of our proposal w.r.t. baselines are achieved on PEO as well (right plots) on both datasets. NPR+S has the lowest PEO among baselines, but still significantly higher than NPR+SAM+REG.

Our proposal beats NPR+J and NPR+W on both recommendation accuracy and popularity bias reduction. Moreover, it achieves comparable accuracy performance w.r.t.

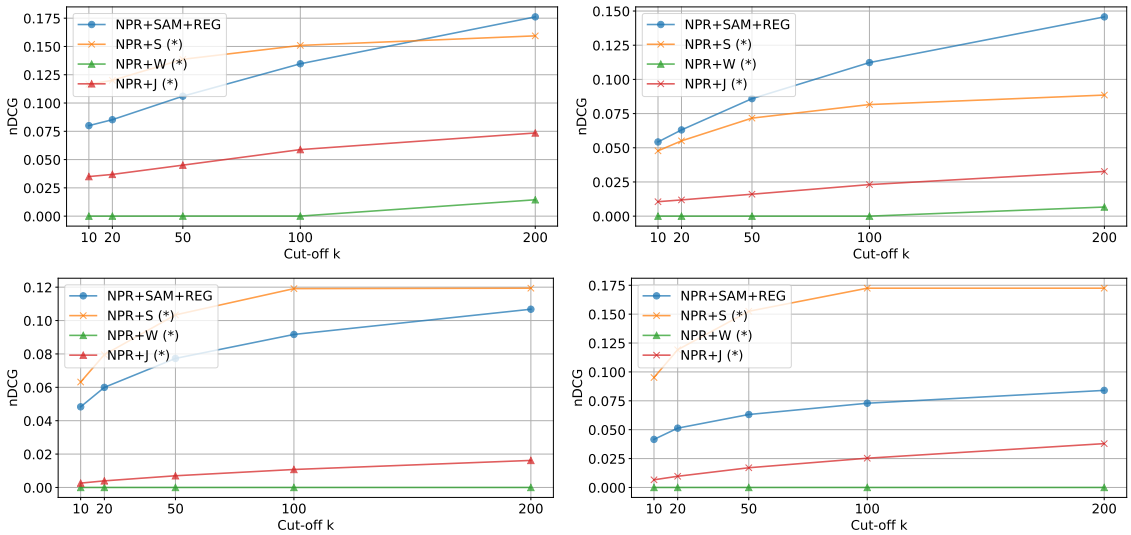


Fig. 4.26: Recommendation accuracy against baselines.

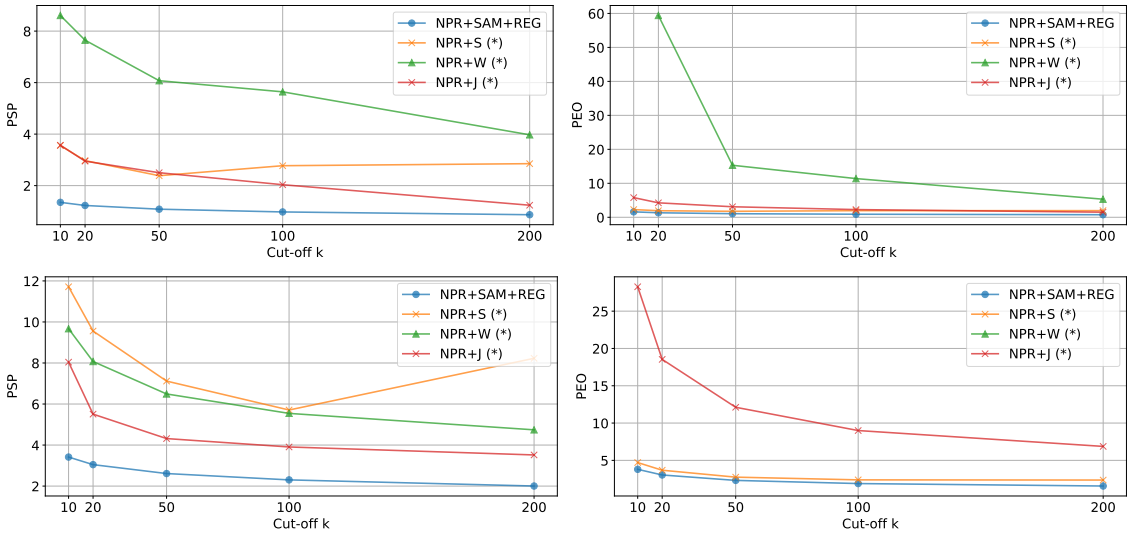


Fig. 4.27: Popularity bias metrics against baselines.

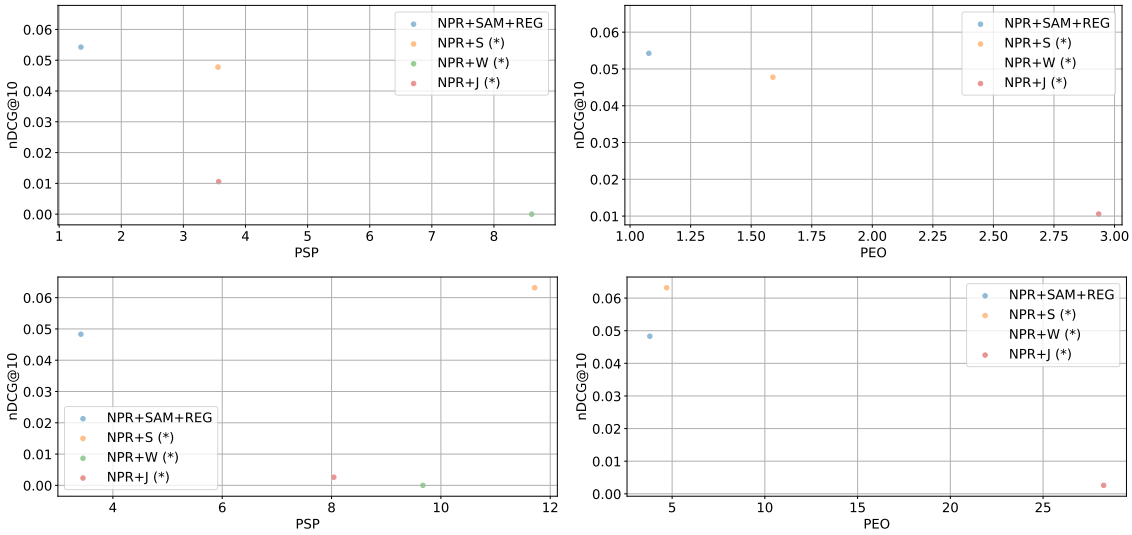


Fig. 4.28: Trade-off against baselines.

NPR+S, but the former outperforms the latter in popularity bias reduction. Therefore, we can conclude that NPR+SAM+REG can better manage the trade-off between such a two objective. To show this finding, Figure 4.28 plots the nDCG@10 score against recommendation parity (top-left) and equality of treatment (top-right) in Movielens 1M, and the nDCG@10 score against recommendation parity (bottom-left) and equality of treatment (bottom-right) in COCO-RS. Each point represents an algorithm. The closer a point is to top-left corner (high nDCG, low PSP/PEO) the better.

4.6 The Proposed Method for Educators' Fairness

4.6.1 Exploratory Analysis on Neural Recommenders

To better understand the need for considering recommendation effectiveness and educators' fairness based on their gender, we begin by providing a descriptive summary of (i) the pair-wise recommendation model we seek to investigate and (ii) how it suffers from educators' unfairness with respect to their gender, when optimized only for accuracy.

The Investigated Synthetic Data Sets

The exploratory analysis treated items from women educators, i.e., I_{women}, as the minority group. To simulate real-world scenarios, we created a set of 10 synthetic data sets, each aimed to simulate different representations of items and ratings related to the minority group. Each data set included $6,000$ users, $2,000$ educators, $6,000$ items, and $1,000,000$ random ratings. Fig. 4.29 (left) summarizes women representation in both items and ratings for each data set. The data set ID is in the form x-y, where x represents the percentage of women educator items and y the percentage of women educator ratings. In Fig. 4.29 (right), we quantified the unbalance associated to women representation across items and ratings by computing the following index:

$$\text{ItemRatingFair@k}(\theta) = 1 - \text{abs}\left(\frac{|I_{women}|}{|I|} - \frac{\sum_{u \in U, i \in I_{women}} y(u,i|\theta)}{\sum_{u \in U, i \in I} y(u,i|\theta)}\right) \quad (4.14)$$

The score ranges between 0 (strongly unfair) and 1 (strongly fair).

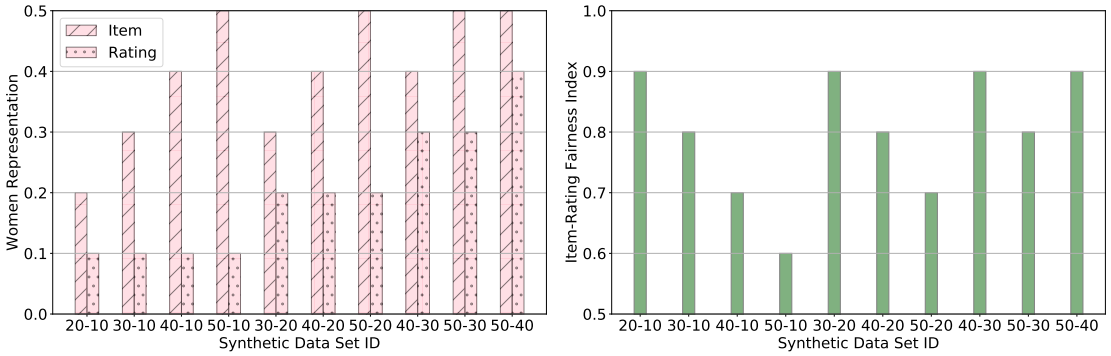


Fig. 4.29: Women representation and fairness in *item/ratings* over synthetic data.

.

Constructing such data sets help us to observe how the investigated recommenders work with different unbalanced representations for the minority group.

The Investigated Pair-wise Learning Approach

To estimate model parameters θ, and consequently compute the user vector matrix W and the item vector matrix X, most of the existing approaches optimize a pair-wise objective function (Figure 4.12). Pair-wise learning maximizes the margin between observed item relevance $y_{u,i}$ and unobserved item relevance $y_{u,j}$ for triplets (u, i, j) in Y. The loss function can be formalized as follows:

$$\underset{\theta}{\operatorname{argmax}} \sum_{u \in U, i \in I_{o_u}, j \in I_{u_u}} \delta(f(u, i|\theta) - f(u, j|\theta)) \tag{4.15}$$

where $\delta(\cdot)$ is the sigmoid function, I_{o_u} and I_{u_u} are the sets of items for which user u's feedback is observed or unobserved, respectively.

Embedding matrices W and X are initialized with values uniformly distributed in the range $[0, 1]$, and the optimization function is transformed to the equivalent minimization dual problem. Given the user-item feedback matrix Y for each synthetic data set, the model is served with batches of $1,024$ triplets. For each user u, we created 10 triplets (u, i, j) per observed item i; the unobserved item j is randomly selected for each triplet. The optimizer used for gradient update is *Adam*. The process is repeated for 20 epochs. The reader notices that the training-testing protocol followed for the previous analysis is adopted for this analysis as well. We do not report them here for conciseness.

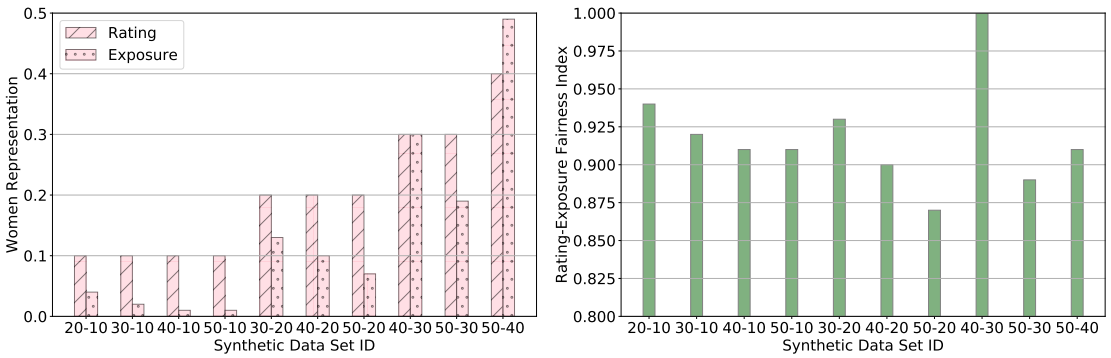


Fig. 4.30: Women representation and fairness in *ratings*/*exposure* over synthetic data.

Educators' Fairness Analysis

Fig. 4.30 (left) plots both the women item representation in ratings and the women item representation in top-10 recommendations (i.e., exposure) for each data set. It can be observed that the pair-wise learning strategy has a tendency to propagate the unfairness against the minority group, so that the women item representation in recommendations is significantly lower than the women item representation in ratings. In Fig. 4.30 (right), such a bias is quantified by the following index:

$$\text{RatingExposureFair@k}(\theta) = 1 - \text{abs}\left(\frac{\sum_{u\in U, i\in I_{women}} y(u,i|\theta)}{\sum_{u\in U, i\in I} y(u,i|\theta)} - \frac{\sum_{u\in U, i\in I_{women}} \phi(u,i|\theta)}{\sum_{u\in U, i\in I} \phi(u,i|\theta)}\right) \tag{4.16}$$

where $\phi(u,i|\theta)$ is 1 if item i is recommended to u in top-k list, 0 otherwise. The score ranges between 0 (strongly unfair) and 1 (strongly fair). For all the data set, except for the $40-30$ data set, such an index is significantly lower than 1. This reveals that the higher the gap between women representations in items and ratings, the higher the unfairness in women educators' exposure. It follows that educators from the minority group tend to be discriminated regardless of the learners' interest in their items.

As our goal is to achieve the same women educator representation in items and recommendations, we computed the related fairness index by following Eq. 4.7. Scores range between 0 (strongly unfair) and 1 (strongly fair). From Fig. 4.31, we can observe that the gap in women representation between the items and recommendations is larger than the one exposed between ratings and recommendations. The pair-wise learning strategy is strongly affected by the number of ratings per educators' gender than the number of items per educators' gender.

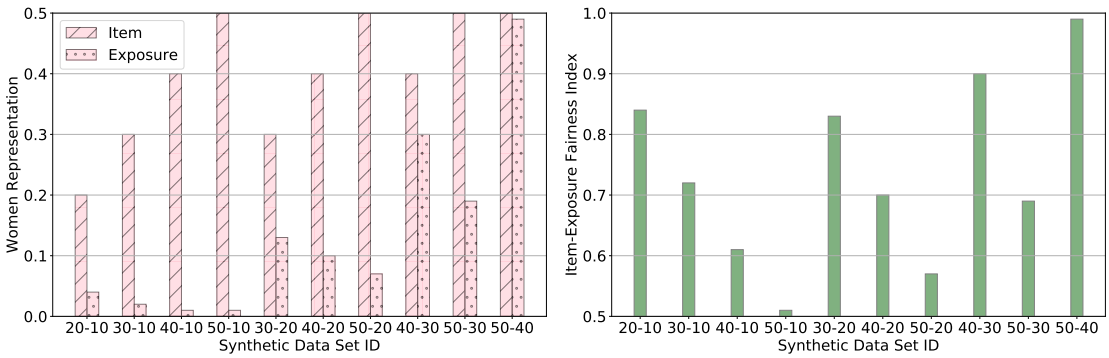


Fig. 4.31: Women representation and fairness in *item*/*exposure* over synthetic data.

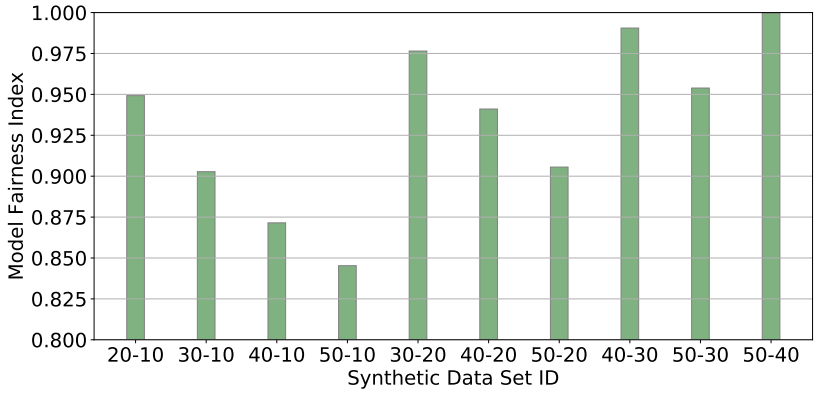


Fig. 4.32: Women *model fairness* index over synthetic data.

Going deeply into the model, we observed also that the pair-wise learning strategy propagates the unfairness in a way that users' embeddings are more similar to embeddings of items provided by men. To measure such an unfairness, we created a set of $1,000,000$ user-item pairs, where half of the pairs includes items from women and the other half includes items from men educator. Then, we measured the average relevance received by items from women and men. Fig. 4.32 shows such a model fairness index as:

$$\text{ModelFair@k}(\theta) = 1 - \text{abs}\left(\frac{1}{|I_{women}|}\sum_{u\in U, i\in I_{women}} \widetilde{y}(u,i|\theta) \; - \; \frac{1}{|I_{men}|}\sum_{u\in U, i\in I_{men}} \widetilde{y}(u,i|\theta)\right) \tag{4.17}$$

The score ranges between 0 (strongly unfair) and 1 (strongly fair). Accordingly to what observed for the above-mentioned indexes, the higher the gap between women representation in items and ratings, the higher the unfairness for women educators' relevance.

Hence, **we conjecture that playing with the women representation in ratings and regularizing relevance scores, so that such scores follow the same distribution for items from both genders, might have a positive impact on the targeted trade-off** between accuracy and fairness. To demonstrate this, we defined a regularization process that reduces the mentioned unfairness while optimizing effectiveness.

4.6.2 Methodology for Educators' Unfairness Mitigation

The goal of the regularization procedure is to generate recommendations that maximize the trade-off between accuracy and fairness by optimizing also for equality of representation between items and recommended lists. To do this, we play with the type of

triplets fed into the model; then, we set up a loss function aimed to minimize both (i) the pair-wise error specified in Eq. 4.15 and (ii) the difference in relevance across items from women and men. We will later show empirically that, although the optimization relies on a given set of interactions, the results are fully generalizable to unseen interactions.

The procedure relies on the following steps:

1. **Ratings Upsampling - Regularization Level I**. We upsampled the ratings related to the minority group, so that the minority group is equally represented in both items and ratings sets. The approach is tested on each of the following techniques:

   (a) *std-rep*: random upsampling on original data, with repetitions.

   (b) *std-syn*: random upsampling on synthetic data, no repetitions.

   (c) *pop-syn*: popularity-based upsampling on synthetic data, no repetitions.

   More precisely, assuming that the minority group is represented by a percent of the items, the upsampling techniques generate new pairs (u, i), where i belongs to the minority group, until the representation of this group in ratings is below a percent.

2. **Triplet Generation**. For each user u, we created 10 triplets (u, i, j) per observed item i associated at Step 1; the unobserved item j is selected among items not yet observed by u. We define the set of triplets as T.

3. **Batch Grouping**. We sampled triplets in T in batches of $m = 1,024$ samples in order to set up an iterated stochastic gradient descent optimization. The batches were created in a way that they hold the same representation of the protected group showed by the entire data set.

4. **Optimization - Regularization Level II**. Each training batch $T_{batch} = \{(u_{t_b}, i_{t_b}, j_{t_b}) \,|\, 0 \leqslant b \leqslant m\}$ is fed into the model that follows a regularized paradigm derived from the standard pair-wise approach. The loss function is formalized as:

$$\underset{\theta}{\operatorname{argmax}} \; acc(T_{batch}) + reg(T_{batch}) \tag{4.18}$$

   where acc(.) is the accuracy objective of a standard pair-wise approach, defined as:

$$\sum_{u \in u_{t_b}, i \in i_{t_b}, j \in j_{t_b}} \delta(f(u, i) - f(u, j)) \tag{4.19}$$

   and reg(.) is a regularization term computed as the difference in relevance across

items from female and male educators, as follows:

$$\begin{aligned} abs[\,(\sum_{u \in u_{t_b}, i \in i_{t_b} \cap I_{women}, j \in j_{t_b}} f(u,i) - f(u,j)) - \\ (\sum_{u \in u_{t_b}, i \in i_{t_b} \cap I_{men}, j \in j_{t_b}} f(u,i) - f(u,j))\,] \end{aligned} \tag{4.20}$$

The model is penalized if its ability to predict an higher relevance for an observed item is better when the item is provided by a men than a women.

Steps 3 and 4 are repeated for all the batches, until convergence.

4.6.3 Evaluation

In this section, we evaluate both levels of the proposed regularization process aimed to strike recommendation effectiveness and educators' fairness based on their gender.

The approach is validated in 12 large-scale data sets, i.e., 10 synthetic data sets previously presented and 2 real-world data sets. To the best of our knowledge, they are among the few data sets that provide sensitive attributes. The first real-world one, *MovieLens-1M (ML-1M)* [166], includes 1M ratings applied to 3K movies by 6K users. On ML-1M, the representation of women providers in items is around 10%, while such a representation is reduced to 7% in ratings. Therefore, all the upsampling techniques create user-item pairs, so that the percentage of ratings related to female providers reaches 10%. The second real-world one, *COCO-RS* [18], includes 37K users, who gave 600K ratings to 30K online courses. On COCO-RS, the representation of women providers in items is around 19%, while it is reduced to 12% in interactions. Therefore, an upsampling technique creates user-item pairs, so that the percentage of women ratings is 19%.

This can help us check if the proposal is generalizable across domains. Moreover, each data set holds different items distributions across educators' genders.

Evaluation Metrics and Protocols

To evaluate the ability of the models in striking recommendation effectiveness and provider fairness, we measured the index formalized in Eq. 4.14.

Moreover, to deeply interpret the results behind such an index, we measured also effectiveness (Precision, Recall, and normalized Discounted Cumulative Gain) and coverage (Overall Coverage, Women Item Coverage and Men Item Coverage) metrics. *Precision* measures how well an algorithm puts relevant items (i.e., items observed by the user and unseen by the algorithm) in top-k recommendations regardless the rank. *Recall*

measures the proportion of relevant items included in the recommendation list with respect to the total number of relevant items. *nDCG* uses graded relevance and positions of the recommended items to compute the ratio between the discounted cumulative gain of the current recommended list and the idealized one. *Item coverage* metrics measure the percentage of the related items in the catalog that are recommended at least once. The overall evaluation protocols follow the steps performed during exploratory analysis, and the evaluation is thus consistent along experiments.

Model Optimization Setups

In what follows, we compare several setups obtained by combining the regularization levels proposed in the previous section. Such setup are identified as follows:

- *no-up*: training without regularizing at any level.
- *std-rep*: regularizing at I level with std-rep upsampling, but not at II level.
- *std-syn*: regularizing at I level with std-syn upsampling, but not at II level..
- *pop-syn*: regularizing at I level with pop-syn upsampling, but not at II level.
- *reg-no-up*: regularizing at II level, but not at I level.
- *reg-std-rep*: regularizing at I level with std-rep upsampling, and at II level.
- *reg-std-syn*: regularizing at I level with std-syn upsampling, and at II level.
- *reg-pop-syn*: regularizing at I level with pop-syn upsampling, and at II level.

Results and Discussion on Synthetic Data Sets

Fig. 4.33 reports the difference in women representation across items and recommendations after applying upsampling on the synthetic data sets. Blue bars refer to models trained without upsampling, while red, green, and yellow bars refer to results achieved when upsampling is performed. The considered upsampling techniques limit providers' unfairness, as the ItemExposureFair@k scores measured on them are closer to 1.

Creating synthetic ratings associated to women items (std-syn and pop-syn) makes it possible to pair representation in all the data sets. In contrast, upsampling by replicating existing interactions and ratings (std-rep) appears working only when the women representation in ratings is around 10%. For a larger representation, the fairness index exhibits significantly lower values. This triggers the fact that replicating user-item pairs might excessively foster the related items, so that the recommender suggests only them.

Fig. 4.34 shows the impact of the regularization at II level. Interestingly, regularizing only at II level mitigates unfairness when women and men representations in ratings

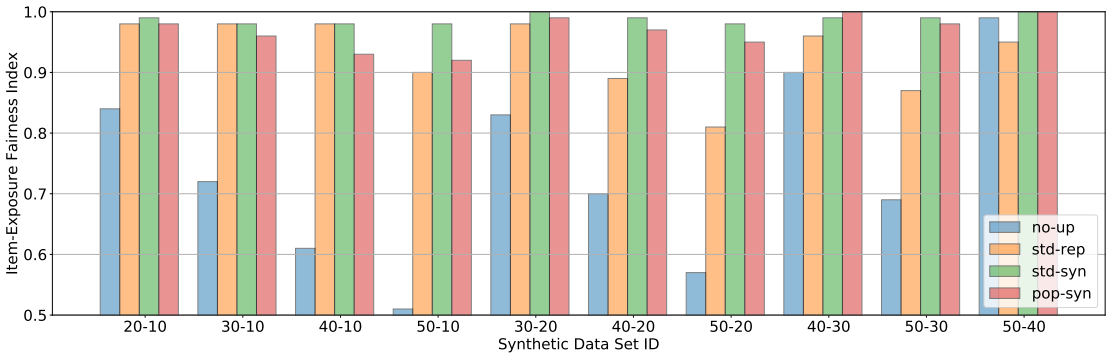


Fig. 4.33: Women *item-exposure fairness* after *mitigation at I level* on synthetic data.

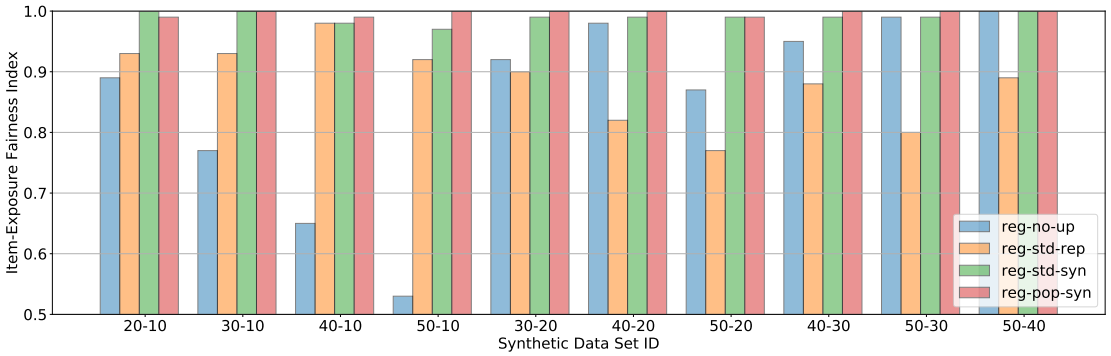


Fig. 4.34: Women *item-exposure fairness* for *mitigation at I+II level* on synthetic data.

are comparable; it fails to strike fairness for larger gaps, even though the difference in exposure across genders is reduced with respect to the counterparts not regularized at all. It appears that recommenders need both levels of regularization to be provider fair. This confirms the validity of the intuition emerged from the exploratory analysis.

Results and Discussion on Movielens-1M

Fig. 4.35 reports the trade-off between effectiveness and provider fairness (i.e., the higher the better). Several regularized setups get a better trade-off with respect to the non-regularized setup (blue bar).

Fig. 4.36 reports the item-recommendation fairness score computed as Eq. 4.7 (i.e., the higher the better). It can be observed that creating synthetic user-item pairs at I level (std-syn, pop-syn) makes it possible to hold higher educators' fairness. However, in contrast to the results observed on synthetic data sets, applying regularization at II level does not help mitigating fairness. In fact, while reg-no-up and reg-rep show unfairness towards women providers, reg-syn and reg-pop are unfair for men providers to the same

extent. Such a behavior could be related to the non-uniform distribution of implicit feedback in Y in a way that replicating few existing user-item pairs is not enough to foster items from women educators, while creating synthetic user-item pairs rapidly biases the recommender towards suggesting items from women educators.

As in any multi-criteria setting, we must be concerned about any accuracy loss that results from taking additional criteria into consideration. Therefore, we also evaluate recommendation effectiveness in Fig. 4.37. It can be observed that *pop-syn* has the highest loss on accuracy metrics, i.e., around 0.70, while it shows the best fairness score.

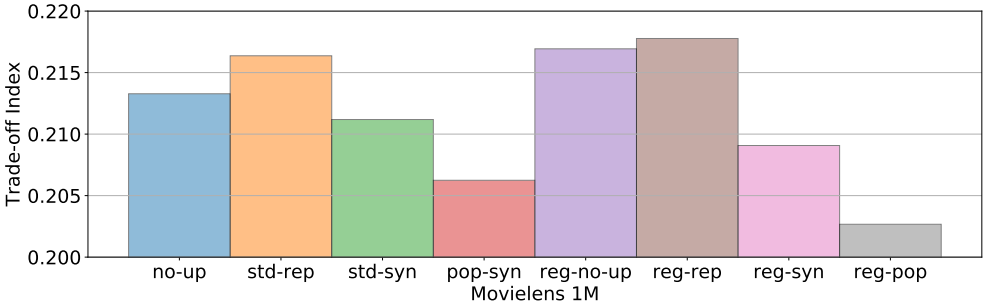


Fig. 4.35: Trade-off between effectiveness and fairness on ML-1M.

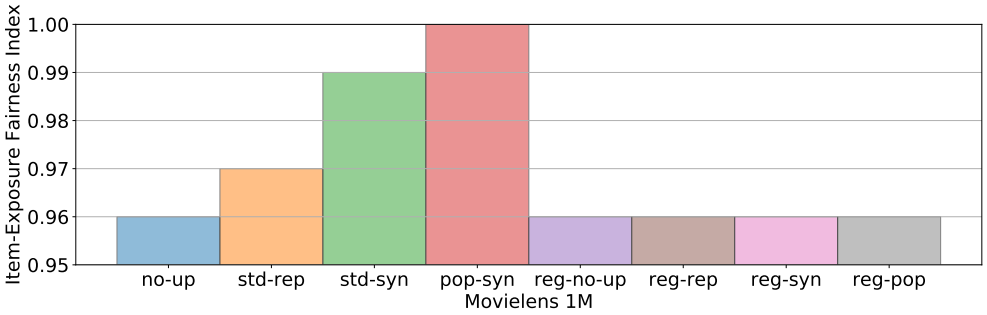


Fig. 4.36: Women *item-exposure fairness* over *ML-1M* after treatment.

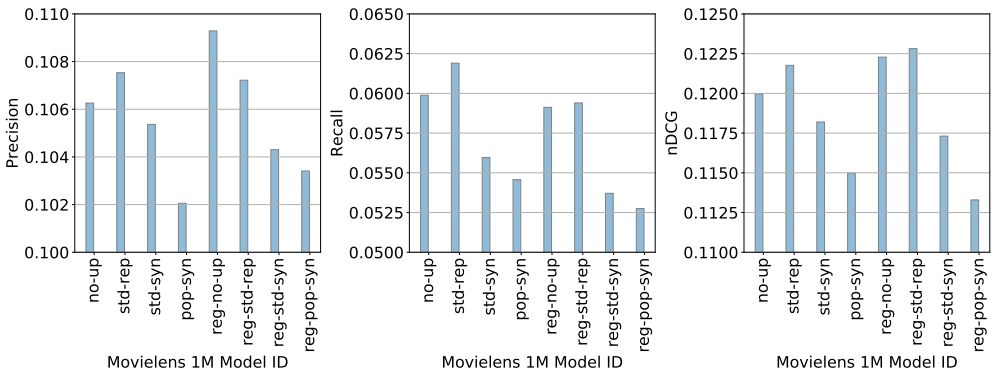


Fig. 4.37: Effectiveness achieved by models over *ML-1M*

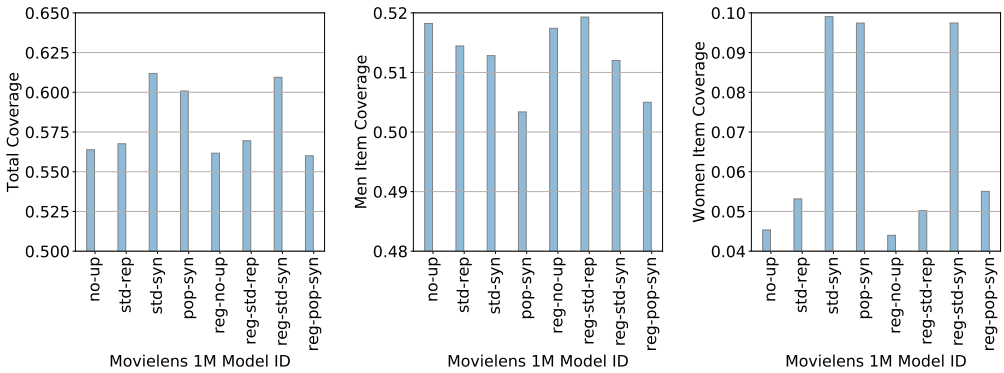


Fig. 4.38: *Coverage* of the models over *ML-1M*

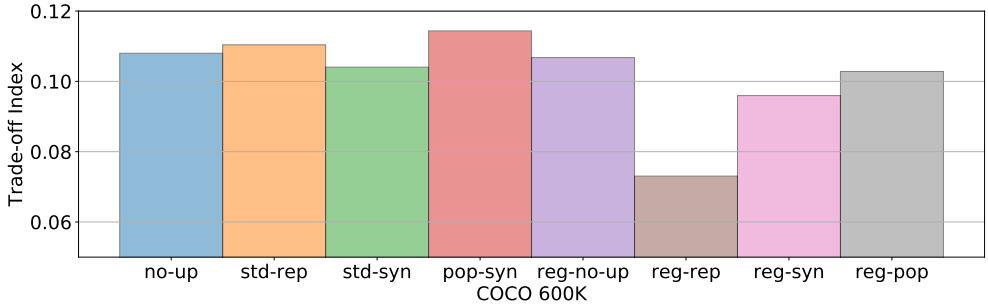


Fig. 4.39: Trade-off between effectiveness and fairness on COCO-RS.

Surprisingly, the setup *reg-std-rep* achieved higher effectiveness and lower unfairness with respect to the *no-up* setup. This means that the proposed multi-level regularization helps the recommender improve accuracy jointly with fairness. Such a behavior brings to larger coverage, as more items from women educators are recommended (Fig. 4.38).

Results and Discussion on COCO-RS

Several regularized setups allow us to hold a significantly better trade-off between effectiveness and fairness even on COCO-RS (Fig. 4.39).

Fig. 4.40 plots the fairness scores obtained by different combinations of regularization. It is confirmed that creating synthetic user-item pairs (std-syn, pop-syn) improves educators' fairness. However, in contrast to the results observed on the other data sets, applying regularization at both levels does not help us in mitigating fairness. Regularizing only at II level (reg-no-up) achieved higher fairness than the standard non-regularized approach (no-up). Accordingly to ML-1M, COCO-RS shows that reg-syn and reg-pop become unfair for men educators. The reason behind this observation could be related to the larger non-uniform distribution of implicit feedback in COCO-RS.

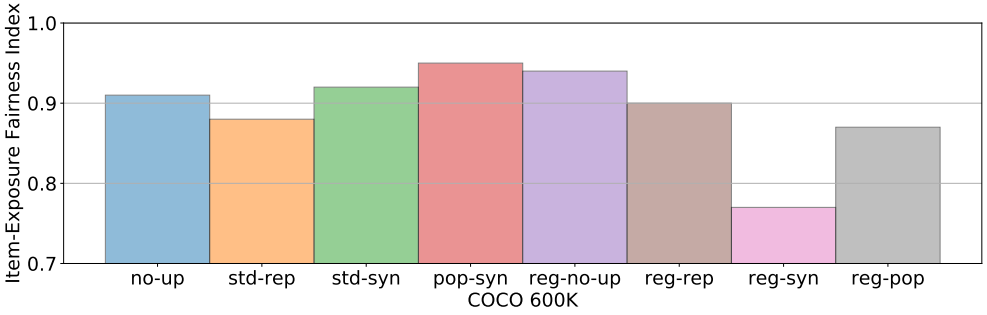


Fig. 4.40: Women *item-exposure fairness* over *COCO-RS* after treatment.

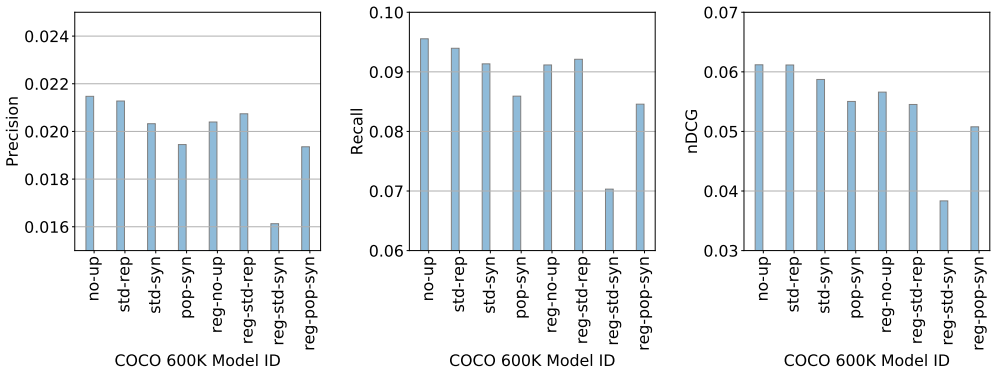


Fig. 4.41: Effectiveness achieved by models over *COCO-RS*

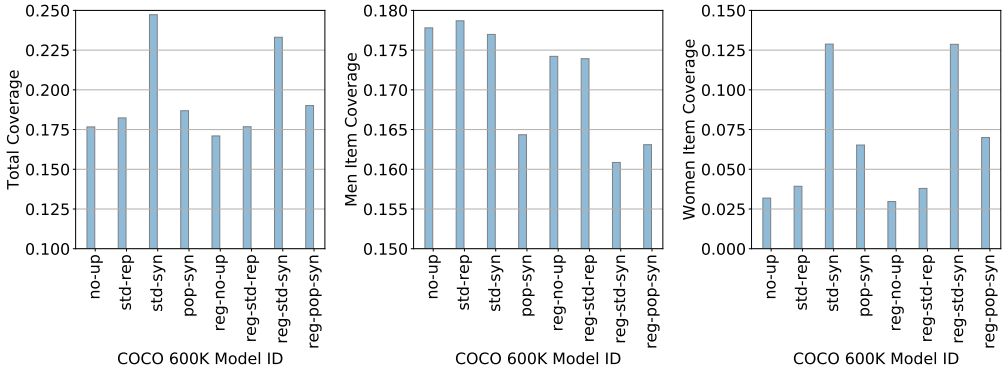


Fig. 4.42: *Coverage* of the models over *COCO-RS*

Fig. 4.41 plots the accuracy metrics for COCO-RS. In contrast to ML-1M, our multi-level regularization brings a loss in accuracy for all the setups. However, such a loss is lower in percentage than the one observed for the other data set. Furthermore, the proposed treatment achieves higher coverage of the item catalog (Fig. 4.42). The coverage of the items from men remains stable, while the coverage for items from women becomes six times bigger than the one obtained by the standard pair-wise approach.

4.7 Findings and Recommendations

In this chapter, we depicted several manageable ways to provide recommendations robust across biases, such as popularity biases and educators' gender biases. This has been possible by leveraging multi-level pair-wise approaches based on data upsampling for minority groups and/or a regularization within the optimization function.

Based on the results, we can conclude that:

1. Neural recommenders are susceptible to popularity bias, and the extent of such a bias depends on both the representation in items and ratings.
2. Randomized user-item relevance distributions for short and long-tail items are significantly different, and the first one exhibits higher relevance values, on average.
3. Inter pair-wise accuracy for observed long-tail items is lower than for observed short-tail items, so the former items are under-ranked regardless of users' interests.
4. While the popularity-aware model shows a loss in accuracy when the entire test set is used, it performs better when tested on long-tail item ratings only.
5. The correlation term makes it possible to reach higher trade-off between recommendation effectiveness and popularity robustness than state-of-the-art baselines.
6. Upsampling can be used to significantly increase educators' gender fairness, especially by replicating existing user-item pairs from the minority group.
7. Minimizing the difference in relevance among items from different educators' genders makes it possible to reduce the unfairness across them.
8. Combining regularization at both levels brings better fairness results based on the extent of the item popularity tail (the less the slope, the higher the fairness).
9. The proposed multi-level regularization can achieve a better trade-off between recommendation effectiveness and educators' gender fairness.

The proposed research opens to several future works. For instance, we can investigate the relation between the recommendations and the tendency of each user to prefer items from minority groups. We can further assess the proposed regularizations on other approaches, i.e., point-wise and list-wise. Moreover, we will consider how models work in presence of constraints from multiple stakeholders (e.g., both learners' and educators' fairness). Other relevant future work will also focus on investigating under which conditions it is possible to distinguish popularity bias by quality-only related popularity. Tackling such a goal is challenging as there is no clue on well-accepted definitions of quality and popularity properties. Moreover, it requires further investigation the fact that current datasets are usually biased towards ratings with high values, making it difficult to consider ratings as a signal of quality.

Chapter 5

Machine Learning Models for Identity Verification

Research Highlights
- Biometrics can enable *transparent* and *continuous* learner verification.
- Combining *face* and *hand-motion* patterns ensures accurate recognition.
- *Intermediate audio-visual fusion* leads to higher verification accuracy.
- Uni-biometric systems are susceptible to *adversarial dictionary attacks*.

5.1 Introduction

The increasing participation in digital education is requiring to educational institutions to ensure the *integrity* of their initiatives against several *threats* [189]. Such a misconduct behavior usually involves an impostor that hijacks the identity of an unsuspected user to conduct malicious activity (*identity theft*), an authorized individual that conducts illegal actions and repudiates such actions (*identity denial*), or an authorized individual that willingly shares their credentials in violation of established policies and regulations (*identity sharing*). The most adopted countermeasure allows students to perform activities while a person monitors them (e.g. *Kriterion*[1] and *ProctorU*[2]), but this is no scalable and highly subjective. Other systems replace humans with algorithms via post or real-time analysis (e.g. *ProctorFree*[3] and *RemoteProctorNow*[4]), but they usually lead to high false-positive rates and limited supported scenarios. Emerging hybrid methods challenge humans only for unclear situations (e.g., *Proctorio*[5]).

[1] https://www.kryteriononline.com/test-taker/online-proctoring-support
[2] https://www.proctoru.com/
[3] http://proctorfree.com/
[4] http://www.softwaresecure.com/products-remote-proctor-now/
[5] https://proctorio.com/

Biometrics are increasingly playing a primary role to verify the *identity of learners* [19], onlife and online. Examples of adopted biometric traits include the *facial* structure [190], the ridges of a *fingerprint* [191], the iris pattern [192], the sound waves of a *voice* [193], and *the way a person interacts* with a digital device [194]. From the system perspective, recognition pipelines require to *detect the modality* of interest in the biometric sample. This is followed by a set of *pre-processing functions*. *Features* are then extracted from pre-processed data, and used by a *classifier* for recognition. From the user perspective, an individual is asked to provide some samples whose feature vectors are stored as a template by the system, i.e., *enrollment*. Then, the recognition process determines if the probe comes from the declared person, i.e., *verification* [195].

Given the nature of the problem, learners should be authenticated *continuously* and *transparently*, without affecting their usual interaction. In fact, a system based solely on something the user has or knows assumes that the student does not share such information. Similarly, an entry-point biometric authentication does not avoid that the legitimated student enters the system and an impostor continues. In response, *physical biometrics* capture random footages or continuous videos to recognize face traits and/or body cues [196, 197]. However, they can be fooled by turning the camera to other video sources. *Behavioral biometrics* enable authentication based on how people interact, but are not completely reliable when provided alone and suffer from limited applicability [198, 199]. *Multiple biometrics* have been combined, but usually require intrusive actions or sensors (e.g. a mouse equipped with a fingerprint reader) [200, 201]. Moreover, existing systems target *PCs* or *laptops*, while learners also use *mobile devices* [202].

In this chapter, we propose a set of *multi-biometric approaches* aimed to continuously and transparently authenticate learners in online and onlife educational scenarios. We contextually combine different physical and behavioral traits based on *the type of interaction* and *the type of device* used at a specific moment. Each approach transparently controls both something the learner is and something the learner does, since physical biometrics are effective but lack synchronization with the actions performed on the platform, while behavioral biometrics are suitable to recognize who does the interactions, but they are not sufficiently accurate alone. This results in a *device/interaction-agnostic* biometric framework that leverages sensors and tracking routines present in almost all the current devices. Please refer to [19] for further information on the whole framework we proposed. Some of its components are described in detail along this chapter with findings and recommendations.

The contribution of this chapter is five-fold:

- We arranged an *audio-visual data set* of 111 participants vocalizing short sentences under robot assistance scenarios. The collection took place into a *three-floor university building* by means of *eight recording devices*, targeting challenging conditions.
- We propose a *transparent continuous authentication* approach that integrates physical (*face*) and behavioral (*touch* and *hand motion*) biometrics to control the learner's iden-

tity during *point-and-click activities* on mobile devices, going beyond one-time login.

- We propose a *transparent continuous authentication* approach that capitalizes on *vocal* and *facial* data to control the learner's identity during *speaking activities*. Each uni-biometric model helps each other to recognize learners when trained jointly.
- We use *voice verification* as use case to study a new *dictionary attack* that puts at risk otherwise developed uni-biometrics systems, showing that adversarial optimization can be used to significantly increase impersonation capabilities of arbitrary inputs.
- We provide an *extensive evaluation* of each approach in terms of effectiveness and efficiency achieved on *public data sets*, comparing different operational setups to assess which one performs better within *education-related scenarios*.

The rest of this chapter is structured as follows: Section 5.2 describes the most representative techniques for biometric verification, going deeply on those tested on learning platforms. Then, Section 5.3 formalizes the problems we seek to investigate. Section 5.4 introduces the audio-visual data set we collected and the other public data sets we manipulated over the presented research. Section 5.5 and Section 5.6 describe and evaluate the verification approaches we designed. Section 5.7 depicts the new attack to biometric systems we crafted. Finally, Section 5.8 concludes the chapter.

5.2 Related Work

5.2.1 Traditional Biometric Verification Techniques

Continuous authentication [195] is a promising area actively studied over the years, but the related methods have been rarely applied in real settings due to their operational complexity. The goal is to regularly check the user's identity throughout the session. Existing applications include monitoring whether an unsuspected impostor has hijacked a genuine user's session on a device [203] or on an online website [204] as well as identifying whether a user has shared their credentials with others in e-learning exams [19].

Physical biometrics, such as face biometrics, have been captured using front-facing cameras, and analyzed to get distinctive user's features [205]. Mouse [206], keyboard [207], touchscreen [208], and other sensors have been used to measure behavioral biometrics (e.g., gait, typing, touching, mouse movements, and hand movements). Such methods continuously check whether the probe comes from the genuine user; if this is true, the user can continue the session; otherwise, the user is locked out. To provide a brief overview, we grouped existing solutions in *uni-modal* and *multi-modal* biometrics.

Uni-modal continuous authentication makes use of only *a single biometric*. *Physical biometrics* are rarely used, as they tend to require considerable *computing power* and

high-quality data not easy to obtain continuously. For instance, *iris recognition* needs the user to face the camera, and usually takes time for authentication [209, 210]. *Fingerprint recognition* requires expensive devices in the view of long computational time [211]. One of the common uni-modal approaches is thus based on *face recognition*. Many algorithms work well on images collected in controlled settings, while their performance degrades significantly on images that present variations in pose, illumination, occlusion, and quality [212]. *Behavioral biometrics*, such as typing, tapping, and speaking, are usually considered a suitable data source for transparent and continuous authentication, while providing *usability*. *Touch sensors data* has been leveraged to get users' distinctive features for authentication purposes [213, 214]. By contrast, other researchers investigated reliability of *motion data* for active authentication. Extensive experiments shown that sensor behavior exhibits sufficient discriminability and stability [215, 216]. Intensive research has been conducted on *keystroke dynamics*, studying its feasibility to verify users' identity on mobile phones and decide whether a text has been written by the legitimated user [217, 218, 219, 220].

Multi-modal biometrics exploit evidence from *multiple biometrics* to mitigate the issues related to uni-modal methods, such as noisy data, intra-class variation, and spoofing attacks [221]. For instance, it exists a framework combining face and touch modalities [222]. Instead of a score-level fusion of the scores from face and touch authentication subsystems, they built a stacked classifiers fed with the scores returned by uni-modal systems. Deploying voice, touch, and hand-motion verification for continuous authentication has been recently studied, integrating these biometrics based on the time they were acquired [223]. This integration would create a trust level on the user based upon the interval from the last successfully captured samples. The literature also proposed a mobile non-intrusive and continuous authentication approach that uses biometric techniques enabled by the device, achieving favorable performance [224]. Their studies investigated authentication systems on laptops. Their experiment adopted an obtrusive login and merely the soft biometrics were verified throughout. Furthermore, some works used a wristband as an initial login fingerprint sensor and, then, as a device that constantly measures the user's skin temperature and heart rate [225]. Lastly, audio-visual recordings have been manipulated to solve visual analysis problems [226]. We point out that, given that learners interact with different devices based on the context, device-agnosticism should be considered.

5.2.2 Deep Biometric Verification Techniques

The recent widespread of deep learning has favoured the use of neural networks as feature extractors combined with common machine-learning classifiers [227]. Backbone architectures rapidly evolved over last years, going from *AlexNet* [228] to *SENet* [229]. Deep learning approaches have been particularly applied to *face* and *voice* biometrics.

From the face perspective, *Deepface* [230] integrates a cross-entropy-based *Softmax*

loss while training the network. However, applying *Softmax* loss is usually not sufficient by itself to learn features separated by a large margin, when samples come from diverse people. Other loss functions have been explored to enhance generalization. For instance, *euclidean-distance-based* losses embed images into an euclidean space, and reduce intra-variance while enlarging inter-variance across samples. *Contrastive* loss [231] and *Triplet* loss [232] are commonly used, but they often exhibit training instability and complex sampling strategies. *Center* loss [233] and *Ring* loss [234] balance the trade-off between accuracy and flexibility. Furthermore, *cosine-margin-based* losses, such as *AM-Softmax* [235], were proposed to learn features separable through angular distance.

From the voice perspective, the most prominent approaches include *d-Vectors* [236], *c-Vectors* [237], *x-Vectors* [238], *VGGVox-Vectors* [239] and *ResNet-Vectors* [240]. Furthermore, deep learning frameworks with end-to-end loss functions to train speaker discriminative embeddings have recently drawn attention [241]. It has been proved that they are more accurate than traditional systems based on hand-crafted features. Such speaker verification strategies relied on *Gaussian Mixture Models* (GMMs) [242] trained on low dimensional feature vectors, *Joint Factor Analysis* (JFA) [243] methods modelling speaker and channel subspaces separately, or *i-Vectors* [244] attempting to embed both subspaces into a single compact, low-dimensional space.

Combining signals from multiple sensors has been traditionally investigated from a data fusion perspective. For instance, such a merge step can happen at *sensor level* or *feature level*, and focuses on how to combine data from multiple sources. It can be performed by either removing correlations between modalities or representing the fused data in a common subspace; the fused data is then fed into a machine-learning algorithm [245]. The literature provides also evidence of fusion techniques at *score level* and *decision level* [246, 247, 248, 249]. There is no a general conclusion on which fusion policy performs better between early and late fusion, but the latter was simpler to be implemented, especially when modalities varied in dimensionality and sampling rates [250].

Good evidence of advantages covered by deep multi-modal fusion comes from the literature. For instance, the authors in [251] aimed to learn features from audio and faces from convolutional neural networks compatible at high-level. The works in [252, 253] depicted time-dependent audio-visual models adapted in an unsupervised fashion by exploiting the complementary of multiple modalities. Their approach allowed to cope with situations when one of the modalities is under-performing. Furthermore, the approach described in [254] used a three-dimensional convolutional neural network to map both modalities into a single representation space. Inspired by findings on high-level correlation of voice and face, the authors in [255] experimented with an attention-based neural network that learns multi-sensory associations for user verification. Differently, the method in [256] extracted static and dynamic face and audio features; then, it concatenated the top discriminative visual-audio features to represent the two modalities, and used a linear classifier for verification. Recent research in [257] depicted an efficient attention-guided audio-face fusion approach to detect speakers.

5.2.3 Biometrics in Education

Collecting and processing biometric data are tasks recently introduced in the context of educational platforms. First, we provide an overview of representative applications. Then, we focus on how they have been used to ensure academic integrity.

Biometrics have been integrated in *onlife school and university scenarios* to simplify administrative processes that, in general, consume a considerable amount of time (e.g., fingerprint-based attendance control [258] or visual-based attendance control [259]). In online environments, biometrics have been adopted to monitor *student participation*, but also to understand their *engagement* [260]. Moreover, the analysis of biometrics enabled *detecting cognitive difficulties* among learners. By observing their behavior, the platform can follow how a student interacts with materials, and detect if they lose interest [261].

Biometric technologies can also help instructors ensure *the integrity of online activities* [189]. These systems verify the identity of the current user and prevent them from accessing unauthorized content. Most representative services include *ProctorU*[6], a student proctoring system based on the human direct control of the examiner. The latter must register to the exam within the *Learning Management System*. Then, the human supervisor looks at the images coming from the webcam to check the learner's ID and the surrounding environment. Once the exam is started, the user is constantly monitored by the supervisor. Similarly, *Pearson VUE*[7] (Pearson Virtual University Enterprises) requires identification through an ID card and a facial image. After a system test, the examiner can start the exam while a human supervisor monitors them throughout the session. Even though both the systems rely on human supervision, they are considered the most popular solutions due to the numerous partnerships with schools and universities.

One of the most representative automated systems is *Proctorio*[8], a remote software that monitors the suspicious behavior of the participants based on pre-configured settings (e.g. if the user will be monitored only via camera, or even via microphone; if the user screen must be recorded, if the web traffic must be checked, if the user must show the room wherein they take the test). The exam can be reported as suspect depending on his behavior, but the teacher will determine if the exam is valid. *ProctorFree*[9] is another exam supervision system that does not require human intervention. It authenticates the student using face recognition and continuously checks their identity by analyzing their face. Furthermore, during the exam, it monitors a variety of typically suspicious events. Once the exam is completed, a report is sent to the instructor. Finally, *TeSLA*[10] (Trust-Based Authentication and Authorship E-Assessment Analysis) allows a user to be controlled during an exam through keystroke, face, and voice dynamics. Moreover, an anti-plagiarism software checks the authenticity of the produced content.

[6] https://www.proctoru.com/
[7] https://home.pearsonvue.com/
[8] https://proctorio.com/
[9] http://proctorfree.com/
[10] https://tesla-project.eu/

5.3 Problem Formalization

The learner's identity verification problem can be formalized as follows. Let A generally denote the domain of input biometric data. We consider a traditional feature extraction step which produces fixed-length representations in $D \subset \mathbb{R}^e$ from each $a \in A$. We denote this stage as $\mathscr{D} : A \to D$. Given a *verification policy* p, a *decision threshold* τ, and N *enrolled biometric samples* per user, a verification system can be defined as:

$$v_{p,\tau} : D \times D_u^N \to \{0, 1\} \tag{5.1}$$

which compares an input feature vector d from an unknown user with a set of enrolled feature vectors $d_u^1, ..., d_u^N$ from user u to confirm (1) or refute (0) the user's identity. We mainly consider a verification policy p that relies on a similarity function $\mathscr{S} : D \times D \to \mathbb{R}$ in order to compare feature vectors:

$$v_{p,\tau} = any\left(\{\mathscr{S}(d, d_u^i) > \tau : i \in 1, \ldots, N\}\right)$$

Let Equal Error Rate (EER) denote the value at which the False Acceptance Rate (FAR) is equal to the False Rejection Rate (FRR), each defined as:

$$\begin{aligned} FAR &= \frac{\text{Number of False Accepts}}{\text{Number of Impostors Comparisons}} \\ FRR &= \frac{\text{Number of False Rejects}}{\text{Number of Genuine Comparisons}} \end{aligned} \tag{5.2}$$

Hence, finding such a verification system becomes an optimization problem, which aims to minimize the following objective:

$$\tau = \underset{\tau}{\operatorname{argmin}} \;\; EER\left[\underset{u \in U, a \in A_{u'}}{\mathbb{E}} v_{p,\tau}\left(\mathscr{D}(\mathscr{F}(a)), D_u^N\right)\right] \tag{5.3}$$

5.4 The Investigated Data Sets

The biometrics research community has provided lots of contributions on data sets settled up onlife or online. As biometrics has been applied on both scenarios within educational environments, we seek to investigate identity verification models that might work onlife and online. To the best of our knowledge, none of the existing data sets nearly reproduces onlife educational scenarios. Therefore, in what follows, we describe a novel data set we collected for this purpose. On the other hand, even though existing

data sets do not target online learning, some of them have been collected under very similar conditions. Therefore, we exploit such data sets to validate our models.

5.4.1 The Proposed AveRobot Data Set for Onlife Scenarios

AveRobot is a multi-biometric dataset of 111 participants vocalizing short sentences under robot assistance scenarios. The collection took place into a three-floor university building by means of eight recording devices. The main goal was to mimic a robot assistance scenario in a semi-constrained indoor environment, as often encountered in public buildings like universities. Considering that the problem was related to the robot sensory part, no real robots were necessary, but they were simulated through the use of various cameras and microphones similar to the ones integrated into robots. The interactions in each floor were recorded with different devices, simulating a total number of eight robot acquisition systems: two in the first floor, three in the second one, and three in the third one. Furthermore, the recordings were made at three locations for each floor: near the stairs, along the corridor, and outside the lift. Fig. 5.1 provides sample faces de-

Fig. 5.1: Samples from the AveRobot dataset.

Dataset	Users	Is Public?	Devices/User	Visual Specs	Audio Specs
[262]	16-171	Yes	2	RGB + RGB-D	-
[263]	50	Yes	1	RGB + RGB-D	-
[264]	22	No	1	RGB + RGB-D	16bit 48kHz
[265]	90	Yes	1	RGB + RGB-D	-
[227]	26	No	1	Grayscale	-
[266]	14	No	1	RGB	-
AveRobot	**111**	**Yes**	**8**	**RGB**	**16bit 16kHz**

Table 5.1: Representative data sets for verification in Human-Robot Interaction.

tected in *AveRobot* videos. As the reader may expect, using diverse acquisition devices poses changes in image illumination, quality, blur, geometry and resolution, and sound quality (e.g., background chatter and room acoustics make our data set challenging).

Differently from AveRobot, other existing data sets collected in Human-Robot Interaction scenarios usually include no audio cue and tend to suffer from one or more limitations: they are obtained under controlled conditions, composed by a small number of users or samples per user, collected from the same device, or not freely available. Refer to Table 5.1 for a comparison among AveRobot and other representative data sets.

Collection Methodology

To collect *AveRobot*, we followed a pipeline whose steps are described below:

1. **Device Selection**: Eight recording devices were selected to make up the dataset, each simulating a different robot acquisition system. Table 5.2 details their characteristics. It should be noted that the devices expose different peculiarities, and they are similar to the sensors embedded in robots. Camera 1 and 7 tended to generate more blurred recordings. On the other hand, Camera 3 and 6 recorded videos using interlaced scan, differently from the progressive scan performed by the others.
2. **Environmental Setup**: We grouped the devices per floor by considering their different type and various operational heights. Each floor hosted a smartphone camera, a compact camera and a video camera, except Floor 0. To assure that the recordings were done in similar conditions, tripods were used for compact and video cameras, while smartphone cameras were held by a human operator at the same height of the other devices. In most cases, we selected a recording height closer to the height of a robot (e.g., Pepper reaches 120cm). The devices were configured with the highest possible resolution at a constant frame rate.

Model	Type	Resolution	Fps	Height	Floor
(1) Casio Exilim EXFH20	Compact Cam	1280 × 720	30	130 cm	0
(2) Huawei P10 Lite	Smartphone Cam	1920 × 1080	30		
(3) Sony HDR-XR520VE	Video Cam	1920 × 1080	30	120 cm	1
(4) Samsung NX1000	Compact Cam	1920 × 1080	30		
(5) iPhone 6S	Smartphone Cam	1920 × 1080	30		
(5) Sony DCR-SR90	Video Cam	720 × 576	25	150 cm	2
(7) Olympus VR310	Compact Cam	1280 × 720	30		
(8) Samsung Galaxy A5	Smartphone Cam	1280 × 720	30		

Table 5.2: The specifications of the recording devices in AveRobot.

3. **User Recording**: The identical recording procedure was repeated for each user. Firstly, for each location, the user selected and memorized the sentence to be articulated, taken from a list of pre-defined sentences. Meanwhile, the devices were arranged in a position near the target location (i.e. stairs, corridor and lift). Then, the human operators switched on the corresponding devices at the same time, while the user approached the camera and reproduced the sentence in front of the capturing devices. In this way, at each location, the same speech was simultaneously recorded with two/three devices. The same process was repeated on each floor and location by selecting a different sentence. The process took between 6 and 10 minutes per user.

4. **Data Protection**: After finishing the session, the user read and signed an agreement in order to respect the European data protection regulation. The information provided by the participant included but it was not limited to: her/his full name, the identification number, whether s/he authorizes to show their data as samples on research articles, and the signature. Gender, height and age were registered.

5. **Video Labelling**: The videos were manually labelled to keep track of the identity, floor and location, the pronounced sentence and the recording device. To this end, each video was properly renamed by using the following convention: UserId-FloorId-LocationId-SentenceId-DeviceId. Moreover, a metadata file was created to save the personal information regarding each user: the assigned id, name, surname, gender, height, native language, age and approval status (i.e. if they authorized to publish their data in research articles). The anonymized version is made publicly available.

6. **Visual Post-Processing**: First, all the videos were converted from the original video format to the target MP4 format. Then, the faces were detected and aligned with MTCNN [267], resized to 224×224 pixels and stored as PNG images. Those frames with detected faces were extracted and saved, with original resolution, as PNG images. Each image was manually checked to remove false positives.

7. **Audio Post-Processing**: Once that the audio was extracted from each video and stored as a WAV file, the silence part at the beginning and ending of the audio was removed through a semi-automated process. It involved the Auditok[11] segmentation tool. Therefore, the resulting audios included only the part where the participant talks. Each audio was converted to single-channel, 16-bit streams at a 16kHz sampling rate. The related spectrograms were generated in a sliding window fashion using a Hamming window of width 25ms and step 10ms for each second of speech.

Structure and Statistics

The proposed dataset contains 2,664 videos from 111 participants (65% male and 35% female) who vocalize different short sentences. The sentences were selected by the par-

[11] https://github.com/amsehili/auditok

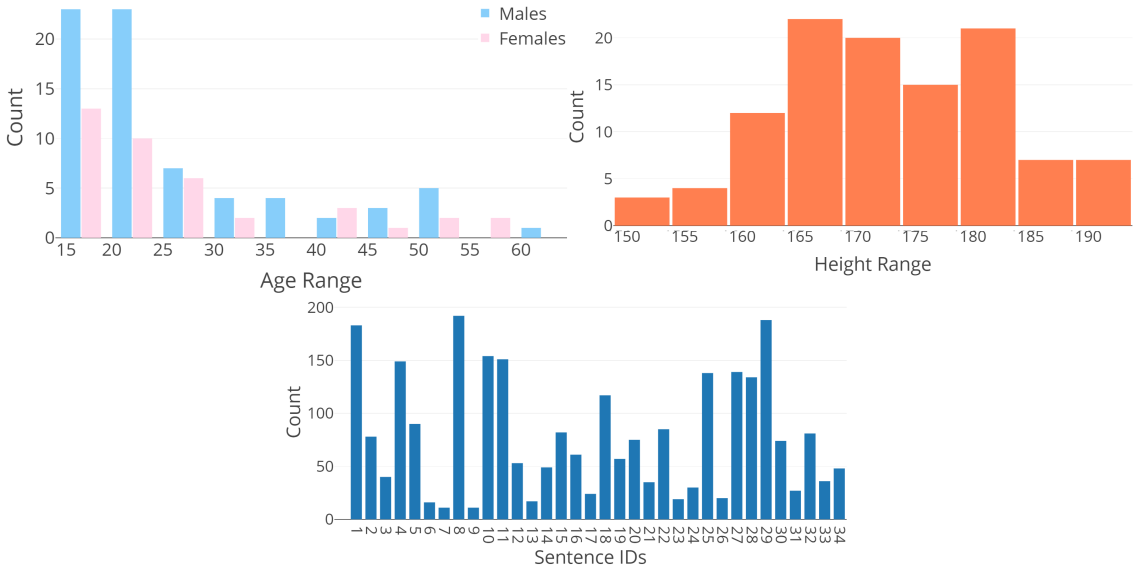


Fig. 5.2: Sample statistics from the AveRobot dataset.

ticipant from a pre-defined set of 34 sentences tailored for a robot assistance scenario. The collected people span different ethnicities (e.g., Chinese and Indian), ages (avg. 27; std. 11; min. 18; max. 60), and heights (avg. 1.74m; std. 0.10m; min. 1.50m; max. 1.92m). Figure 5.2 depicts relevant distributions along the dataset. The gender, height, and age for each participant are also provided together with the videos. Each person was recorded in 3 locations (i.e. stairs, floor and lift) for each one of the 3 floors of the building. As mentioned above, 8 diverse recording devices were leveraged during the collection to simulate the robot acquisition systems. The recording devices assigned to the same floor worked simultaneously. Thus, the dataset comprises 24 videos per user:

- 1st Floor: 2 (devices) $\times$ 3 (locations) = 6 videos.
- 2nd Floor: 3 (devices) $\times$ 3 (locations) = 9 videos.
- 3rd Floor: 3 (devices) $\times$ 3 (locations) = 9 videos.

The total length of the video resources provided by the proposed dataset is 5h 17min, occupying 21.8GB. Each participant is represented by more than 3min of videos, each lasting around 7s. It should be noted that each video includes three phases: (i) when the person is approaching to the devices, (ii) when s/he speaks in front of them, and (iii) when s/he leaves the scene. Hence, looking only at the face content, each video contains around 127 frames with a detected face, and each user is represented by over 3,000 detected faces. The total number of detected faces is 338,578, occupying 18.0GB. On the voice content, each video contains around 3s of speech, and each user is represented by over 1m of content. The voice data lasts 1h 40min, occupying 283MB.

5.4.2 Representative Existing Data Sets for Online Scenarios

To validate our verification approaches for online educational scenarios, we leveraged the following existing data sets collected under very similar conditions:

- The *H-MOG* data set [215] includes 100 users. When a volunteer logs into the data collection tool, s/he is randomly assigned to a reading, writing, or map navigation session. For each session, the volunteer either sits or walks to finish the tasks. One session lasts about 5 to 15 minutes, and each volunteer is expected to perform 24 sessions: 8 reading sessions, 8 writing sessions, and 8 map navigation sessions. Each volunteer in the experiments contributed with about 2 to 6 hours of data. The following categories of data are recorded: accelerometer, gyroscope, magnetometer, raw touch event, tap gesture, scroll gesture, key press on virtual keyboard.

- The *UMDAA-02* data set [268] consists of smartphone sensor signals collected from 48 volunteers over two months. The data collection sensors include the front-facing camera, touchscreen, gyroscope, accelerometer, magnetometer, and pressure sensor. Excluding the data of 5 users, a set of 33,209 images was selected from all sessions of the remaining 43 users at an interval of 7 seconds. The database contains many partial faces. Moreover, single finger touch sequences on the screen are included. The length of gestures vary between 1 to 3,637 touch data points. The data set contains a large number of touch and swipe data per user, and can serve for practical experiments.

- The *MOBIO* database [269] consists of audio and video data taken from 152 people. The database has a female-male ratio or nearly 1:2 (100 males and 52 females), and was collected in six different sites from five different countries. This led to both native and non-native English speakers. In total, 12 sessions were captured for each client, i.e., 6 sessions for phase I and 6 sessions for phase II. Each phase consists of 21 questions, with the question types belonging to short response questions, short response, and free speech. The database was recorded using a mobile phone and a laptop computer. The first session consists of data captured on both the laptop and the mobile phone, while the remaining ones include data from the latter only.

- The *VoxCeleb-1/2* data sets [239, 240] contain over 1 million utterances for 7,205 celebrities, extracted from videos uploaded to YouTube. The datasets are gender balanced. The speakers span a wide range of different ethnicities, accents, professions and ages. The videos are shot in a large number of challenging multi-speaker acoustic environments. These include outdoor stadiums, quiet studio interviews, speeches given to large audiences, excepts from professionally shot multimedia. Crucially, all are degraded with background chatter, laughter, overlapping speech, room acoustics, and there is a range in the quality of recording equipment and channel noise. Such data sets can serve for both online and onlife scenarios. To the best of our knowledge, they represent the unique data sets large enough for training a deep neural network.

5.5 The Proposed Face-Touch Verification Approach

In this section, we propose a *transparent continuous authentication* approach that integrates physical (*face*) and behavioral (*touch* and *hand motion*) biometrics to control the learner's identity during *point-and-click activities* on mobile devices.

5.5.1 Methodology

The reference architecture underlying the proposed approach is depicted in Fig. 5.3. The implementation of each biometric module was properly carried out to ensure both *efficacy* and *efficiency*. It aims to provide a good trade-off between *security* and *usability*.

The approach works as follows. Starting from the login step, for each touch stroke, a tracking module gets all the data about the *touch stroke* together with the image captured from the *camera* and the data from *Inertial Measurement Unit* (IMU) sensors, close to the time the user touches the screen. The sample is sent to the continuous authenticator. Inside this, each biometric authenticator individually performs authentication by comparing the features extracted from the *face image*, *touch data*, or *motion sensors data* with the corresponding templates and returns a *matching score*. Then, the matching scores are fused by the *trust manager* which decides if the user can continue the session.

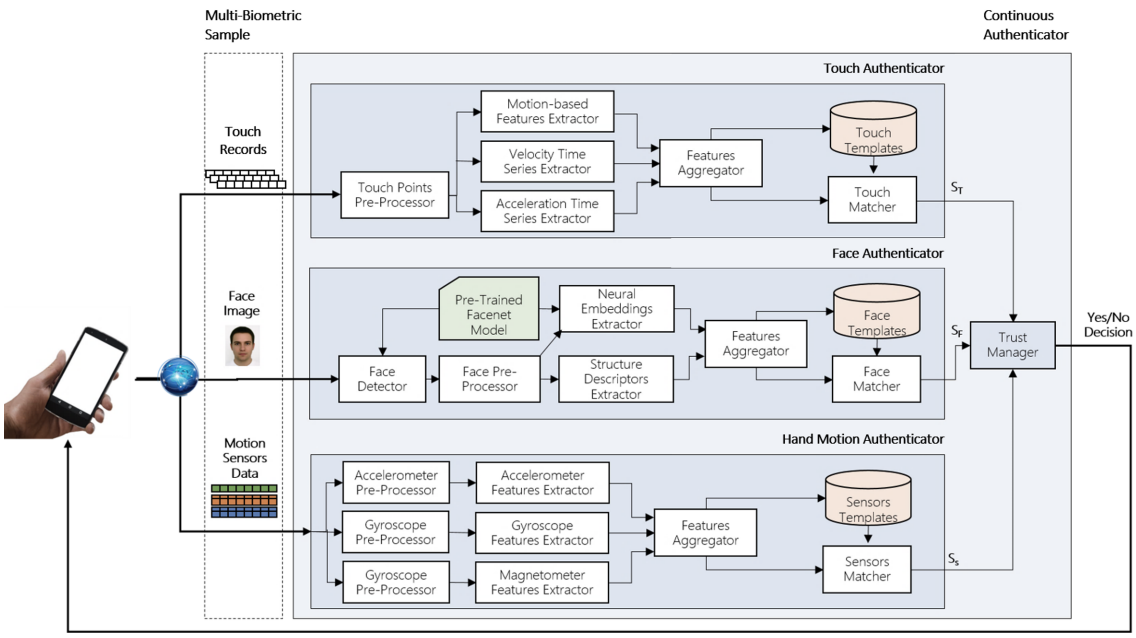


Fig. 5.3: The proposed face-touch verification approach.

Face Authenticator

The module receives an image as input. It implements the *FaceNet* algorithm for the localization of the relevant sub-region of the image containing the face [232]. *FaceNet* has been proven to work well when face samples are collected in the wild, including illumination changes, occlusion and wide variation in poses and expressions. A correction routine is used to normalize *illumination* on the image. Then, the module detects the points of interest of the face, i.e., *landmarks*: the left eyebrow, the right eyebrow, the left eye, the right eye, the nose, and the mouth. The landmark detector is based on a regression trees ensemble trained to estimate facial key points from pixel intensity [270].

The face region is cropped and aligned in pre-processing based on the eye regions. This leads to a *normalized*, *rotated*, *translated*, and *scaled* representation of the face. Landmarks are also cropped and manipulated separately. From the extracted image parts, the module extracts different types of features:

- Pre-processed face converted to grayscale and re-scaled to 64 × 64 pixels.
- *Local Binary Patterns*[12] (LBPs) extracted for cell size of 8 × 8 from 64 × 64 re-scaled grayscale images.
- Eye-based, nose, and mouth bounding boxes extracted from the 64 × 64 re-scaled grayscale face and resized to 16×20, 28×17 and 22×46 pixels, respectively.
- LBPs obtained for a cell size of 12 × 12 pixels from each one of the resized landmarks bounding boxes computed during the previous point.
- *FaceNet neural embeddings* resulting from the face and the individual landmarks.

The computed features are *vectorized* as uni-dimensional vector, and the features vectors extracted from multiple training samples are *averaged*, then *normalized* in the range [0,1] by using *z-score normalization*. The resulting features vector constitutes the *user's template*. The matching score between the template and the probe is computed by using *cosine similarity*, and returned by the authenticator.

Touch Authenticator

The module receives a set of data records related to an individual *touch stroke* as input. Each set of records is composed by *one finger-down* record, N *consecutive moving* records, and *one finger-up* record. Every record r_i is encoded as quadruplet: $r_i = (x_i, y_i, c_i, t_i)$ $\forall i \in 1, \ldots, N$, where (x_i, y_i) are the location coordinates, c_i the contact size applied at time t_i, and N is the number of records captured during the current touch stroke. The module is set to process only strokes containing more than *three data records*.

[12] LBP is a texture operator which labels pixels of an image by thresholding the neighborhood of each pixel and considers the result as binary number.

From each set of records, different types of feature are extracted in relation to the *contact*, the *velocity* and the *acceleration* during the touch. More precisely, 5 *statistical features* (i.e., mean, standard deviation, maximum, minimum) are computed. In addition, 6 *frequency-based features* are extracted from time series representing contact, velocity, and acceleration (i.e., spectral centroid, spectral energy, spectral spread, spectral skewness, spectral kurtosis, and spectral flatness).

Features are *concatenated*, and the features vectors extracted from multiple training samples are features-wise *averaged*, then *normalized* in the range [0,1] by using *z-score normalization*. Each feature is scaled per user using the related standard deviation. *Cosine similarity* between the probe feature vector and the user's template is computed, then the similarity is returned as a *matching score* by the authenticator.

Hand-Motion Authenticator

The module receives *three sequences of observations*, one sequence for each motion sensor. In a sequence, each observation represents the values from the three axes of that sensor at a given time in the form $< \mathtt{timestamp}, x, y, z >$. To get rotation-invariant features, each observation is enriched with the magnitude m of the combined vector, i.e., the square root of the sum of the squares of the homologous values in the three axis. Hence, each observation is in the form $< \mathtt{timestamp}, x, y, z, m >$.

For each motion sensor, the module separately analyzes x, y, z, and m values as time series, and extracts from each one 10 *time-based features* (i.e., mean, standard deviation, average deviation, skewness, kurtosis, root mean square energy, minimum, maximum, non-negative count, zero-crossing rate) and 6 *frequency-based features* (i.e., spectral centroid, spectral energy, spectral spread, spectral skewness, spectral kurtosis, and spectral flatness). Therefore, from each sensor sequence, a 40-dimensional feature vector is extracted, i.e., 4 time series $\times$ 10 features per time series.

Features vectors extracted from multiple training samples are *averaged*. Features vectors coming from the three motion sensors data are *vectorized*, and *concatenated* to form a unique feature vector. The features vector is *normalized* in the range [0,1] by using *z-score normalization*, and each feature is scaled per user using the related standard deviation. During verification, the module uses *cosine similarity* to calculate how much the feature vector from the probe and the template are similar. The resulting score is returned as a *matching score* by the authenticator.

Trust Manager

The module is responsible of *the fusion of the matching scores*, and of the computation of the *global trust level* in the *user's genuiness*. Such a score is used to decide whether the current user will be successfully authenticated.

For each biometric authenticator, an *individual trust level* in user genuiness is kept updated considering past values in the session. The parameters have been optimized for each authenticator during preliminary stages. For each biometric authenticator, the manager rewards or penalizes the trust in the user's genuiness based on the amount of the variation between the genuine user template and the current user probe. If there is a small deviation for that biometric module, then the trust of the system in user's genuiness for that biometric module increases. For large deviation, the trust decreases.

The trust levels are fused through a *weighted sum*. The values assigned to the weights are based on the *equal error rate* (i.e. the value when the false acceptance rate and false rejection rate are the same) of the corresponding authenticator. If the global trust is over a given threshold, the user continues the session; otherwise, they are locked out.

5.5.2 Evaluation

We evaluated the effectiveness of our approach on a chimeric data set that includes *H-MOG* and *MOBIO* multi-modal data. For each person, assuming face and touch plus sensors biometrics are statistically independent, 100 chimeric users were built from these two datasets. It is accepted that results obtained in this way are worthy of full reliability.

Given a user u_i from *MOBIO* and a user u_j from *H-MOG*, a chimeric user c_{ij} is defined as (u_i, u_j). Since H-MOG contains 100 users, 100 subjects out of the 150 of *MOBIO* were randomly chosen. Then, 100 chimeras were built by randomly associating users. Since the number of sessions per user is the same in the data sets, we chronologically sorted sessions, and we paired them using their position in the sorted lists.

For each user's session, we created the biometric sample by counting and sampling the touch strokes the user has done during the session; then, we associated each touch stroke to a single biometric sample. Each biometric sample is enriched with the data from each motion sensor from 100ms before and after the touch stroke.

Finally, we merged all the videos of the current user in the current session, and we counted the total number of frames of the merged video. We get as video frames as the number of touch strokes in the session at equally-spaced intervals. We chronologically assigned the extracted video frames to touch strokes.

Evaluation Metrics and Protocols

The performance of our approach was measured with the following metrics:

- *False Acceptance Rate* (FAR) measures the chance that the biometric system incorrectly accepts an attempt of an impostor user. FAR results as dividing the number of false acceptances by the number of impostor attempts. The lower the better.

- *False Rejection Rate* (FRR) measures the chance that the biometric system incorrectly fails to authenticate a legitimated user. FRR is calculated as the ratio between the number of false rejects and the number of genuine attempts. The lower the better.
- *Equal Error Rate* (EER) is the value obtained at the threshold level used by the biometric system where FAR and FRR are equal. The lower the better.

Results and Discussion on Short-term Authentication

The first experimental setup aims to evaluate the performance of our approach when the authentication is performed closely to the enrolment. In this setting, the influence of the interaction context and the effect of aging should be negligible.

For each user session, we trained the system with data recorded on the first 70% of the session, and tested it on data tracked on the remaining 30% data. Going over the scope of this experimental setup, the simulated scenario could also serve as continuous authentication within one session of device utilization: assuming that the genuine user logins in the application by entry-point authentication, and the enrolment starts with the session, after having observed few samples, the device turns to authentication mode. In this way, it could detect impersonation when the genuine user interacts with the device and, after a moment, they leave the device unattended. We assume that the login and the first part of the session are entirely carried out with the genuine user.

The EER of the individual biometric subsystems (i.e., face, touch and motion sensors matchers) and of the bi-modal and tri-modal fusion matchers obtained during this experimental setup are reported in Table 5.3. From the results with uni-modal verification, we observed that the motion sensor subsystem achieved the lowest performance (EER=12.45%) with respect to the other combinations. Since data tracked by motion sensors is exposed to large variations during the session, it is reasonable that the resulting EER was higher. The touch subsystem achieved promising but not exceptional results. The authentication capability mainly depends on the duration of the touch stroke and

System Type	Biometrics Combination	Authentication Setting	
		Short-term	Mid-term
Uni-modal	Face	3.21	5.95
	Touch	7.80	11.23
	Hand Motion	12.45	17.56
Bi-modal	Face + Touch	1.53	3.92
	Face + Hand Motion	2.81	5.69
	Touch + Hand Motion	6.78	10.52
Tri-modal	Face + Touch + Hand Motion	0.84	3.56

Table 5.3: The EER obtained by the proposed face-touch verification approach.

the amount of data they consequently provide. For instance, swipe gestures enable to get more data in terms of touch records with respect to tap gestures. The face subsystem reached good performance when it is used alone.

The multi-modal subsystems got lower EERs than single-modality ones, except for the bi-modal approach combining touch and motion sensors; the latter performed worse than the individual face matcher (3.57% EER). By fusing touch and face, the EER of the individual face matcher was reduced of 1.68%. Integrating all the modalities allows to further decrease it of 0.69% with respect to the best bi-modal subsystem (i.e., face plus touch). The achieved EER is reasonable.

Results and Discussion on Mid-term Authentication

The second experimental setup aims to evaluate the performance of our approach when the authentication is performed some weeks after the enrolment; thus, the aging effect and the different interaction context could influence recognition capabilities.

This scenario assumes that the genuine user trains the approach during enrolment, then the template stays the same over many sessions. During the session, the system authenticates the user by comparing the probe with the template built during old sessions. Even though the short-term authentication scenario embraces a lot of real situations, it does not work when the mobile device has been stolen in advance. Improper access is detected only if the impostor enters the session after the real user. To overcome this limitation, we simulated mid-term authentication by training the system with data from the first 70% sessions, and testing it on data from 30% later ones.

The EERs obtained in the second experimental setup are reported in Table 5.3. The EERs varied between 5.95% and 17.56% for single-modality approaches. The motion sensors subsystem reported the largest decline in performance with respect to the ones under short-term evaluation, while they remained more stable for face and touch.

Integrating multi-modal solutions improved the authentication performance, but the combination touch plus motion sensors performed worse than face-only recognition in this experimental setup. The bi-modal recognition performance resulted in higher EERs in comparison with the EERs obtained under short-term evaluation. Moreover, the contribution of touch and motion sensors subsystems to improve the face recognition capabilities is less in this scenario. This is reasonable since behavioral biometrics usually tend to exhibit large variations among different sessions. Furthermore, some variables of the interaction context could be varied between enrolment and authentication (e.g., emotional state, hand posture). The aging effect could also influence performance.

The tri-modal fusion enabled to reach reasonable, but higher EERs values with respect to the one on short-term evaluation. The total number of scores being tested was big, and those correctly matched were relevant. The EERs can be considered reasonable.

To have a more detailed picture of the proposed approach, we also tracked the computational time required by each biometric module. From such an analysis, we observed that strengthening face biometrics with touch and hand-motion does not compromise the stringent real-time requirements for continuous authentication. The computational time of the multi-biometric verification is still dominated by the time required for processing face biometrics (i.e., face verification is 60% slower than touch or hand-motion verification). The difference in computational time between touch and hand-motion sub-modules is negligible (i.e., less than 5%). This might derive from the higher complexity of processing images, i.e., faces, than plain time-series, i.e., touch and hand-motion. Furthermore, verification is run in parallel over sub-modules, and inserting additional sub-modules does not impact the overall computation time.

5.6 The Proposed Face-Voice Verification Approach

In this section we propose a *transparent continuous authentication* approach that capitalizes on *vocal* and *facial* data to control the learner's identity while *speaking*. Each uni-biometric model helps each other to recognize learners when trained jointly.

5.6.1 Methodology

The proposed fusion strategy that jointly learns voice and face feature representations, including input data formats, architectures, and training details, is depicted in Fig. 5.4. The core idea is to leverage the morphological relations existing between voice and face biometrics in order to investigate a cross-modal training, where each uni-biometric model is supported by the biometric model of the other modality in improving the effectiveness of its feature representations. Differently from other intermediate fusion approaches, such a multi-biometric fusion might happen (i) on training to develop better uni-biometric models and/or (ii) on deployment to exploit joint evidence.

Face Backbone

Let $A_f \subset \mathbb{R}^{m \times n \times 3}$ denote the domain of RGB images with $m \times n \times 3$ size. Each image $a_f \in A_f$ is pre-processed in order to detect the bounding box and key points (two eyes, nose and two mouth corners) of the face. The affine transformation is used to align the face. The image is then resized and each pixel value is normalised in the range [0,1]. The resulting intermediate facial image, defined as $S_f \subset \mathbb{R}^{m \times n \times 3}$, is used as input of the visual modality branch of our model. In this branch, an explicit feature extraction produces fixed-length representations in $D_f \subset \mathbb{R}^e$. We denote such a stage as $\mathscr{D}_{f\theta_f} : A_f \rightarrow D_f$. Its output is referred to as face feature vector.

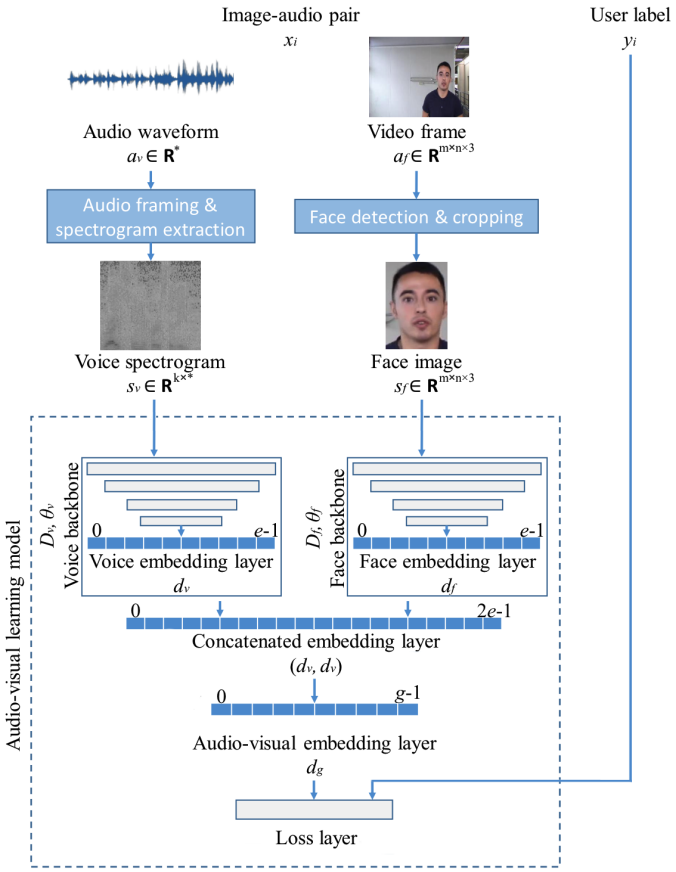


Fig. 5.4: The proposed neural architecture for intermediate multi-biometric fusion.

Voice Backbone

Let $A_v \subset \mathbb{R}^*$ denote the domain of waveforms digitally represented by an intermediate visual acoustic representation $S_v \subset \mathbb{R}^{k \times *}$, such as a spectrogram or a filter-bank. Each audio $a_v \in A_v$ is converted to single-channel. The spectrogram is then computed in a sliding window fashion using a Hamming window, generating an acoustic representation s_v that corresponds to the audio a_v. Mean and variance normalisation is performed on every frequency bin of the spectrum. The resulting representation is used as input of the acoustic modality branch of our model. In this branch, an explicit feature extraction produces fixed-length representations in $D_v \subset \mathbb{R}^e$. We denote such a stage as $\mathscr{D}_{v\theta_v} : S_v \rightarrow D_v$. Its output is named voice feature vector.

Fusion Backbone

Let $D^{2 \times e}$ be the domain of audio-visual feature vectors generated by a plain concatenation of the sparse representation from the face and voice backbones, i.e., d_f and d_v.

We denote as $\mathscr{C}_\theta : (D_f, D_v) \rightarrow D^{2\times e}$ such a concatenation stage of both modalities applied after the representation layer of each single modality branch. Then, an additional feature vector learning step is applied to the concatenated vector $d \in D^{2\times e}$ to get a single feature vector of size g jointly learned from d_f and d_v. This extra layer aims to (i) keep independent the multi-biometric embedding size from the uni-biometric embedding sizes and (ii) learn more compacted and flexible representations. Moreover, by setting $g = e$, reasonable comparisons between uni-biometric and multi-biometric sparse representations of the same size can be performed. We denote such an extra step as $\mathscr{D}_{fv\theta_{f,v}} : D^{2\times e} \rightarrow D^g$. Its output is named as audio-visual feature vector.

Combining both modalities might generate a better sparse representation of the individual, and enrich the feature representation of a single modality. This is due to the relations of voice to genre and facial morphology of people, e.g., male people commonly have a tone lower than female people. Therefore, by leveraging the fusion backbone, the uni-biometric backbones help each other to better recognize people. Our hypothesis is that the embeddings of each backbone should perform better when trained jointly.

Backbones Instantiation

The proposed approach makes use of existing neural network architectures, slightly arranged to accommodate the modality digested by each of the above-mentioned backbones and the subsequent fusion purposes.

Two instances of the residual-network (*ResNet-50*) architecture are used as feature vector extractors $\mathscr{D}_{f\theta_f}$ and $\mathscr{D}_{v\theta_v}$ within face and voice backbones, respectively [271]. Such a network, well known for good classification performance on visual and acoustic modalities [230, 240], is similar to a multi-layer convolutional neural network, but with added skip connections such that the layers add residuals to an identity mapping on the channel outputs. The input layers of the original *ResNet-50* architecture are adapted to the modality associated to the corresponding backbone. Moreover, the fully-connected layer at the top of the original network is replaced by two layers: a flatten layer and a fully-connected layer giving the embedding of the modality, i.e., d_f or d_v.

The fusion backbone $\mathscr{D}_{fv\theta_{f,v}}$ is instantiated by a concatenation layer stacked into the model to combine face and voice feature vectors in $D^{2\times e}$ domain, and an additional fully-connected layer where the significant features of video and audio modality are jointly embedded. The latter output represents the audio-visual feature vector $d \in D^g$ previously formalized. Moreover, for each fully-connected layer, batch normalization has been set before the activation function to regularize the outputs, and a dropout layer is inserted after activation to prevent model over-fitting. Finally, an output layer depending on the applied loss function is posed at the top of the network during training.

Training Process Description

The training data is composed by N tuples $\{(x_i, y_i)\}_{i=1}^{N}$ where each multi-biometric sample x_i corresponds to a person associated with the class $y_i \in 1, ..., I$, being I the number of different identities depicted in N samples. Each sample x_i is defined as a pair $x_i = (a_{v_i}, a_{f_i})$ such that a_{v_i} is a utterance and a_{f_i} is a visual frame. The elements of each pair are randomly chosen among face and voice samples from the same user; then, they are sequentially fed into the multi-biometric model. Such a model can be integrated with any existing loss function. Additionally, a hold-out validation set consisted of all the speech and face segments from a single randomly-selected video per user.

5.6.2 Evaluation

Experiments aimed to assess the reliability of the proposed uni-/multi-biometric feature representations. The instances of the model involved in these experiments were trained on VoxCeleb1-Dev and tested on AveRobot, MOBIO, and VoxCeleb1-Test. This is motivated by the fact that VoxCeleb1 has been known for being the largest dataset for audio-visual verification, and this is more suited for training a deep neural network.

Evaluation Data Format

For the face branch, each frame is analyzed in order to detect the face area and landmarks through MTCNN [267]. The five facial points (two eyes, nose and two mouth corners) are adopted by such an algorithm to perform face alignment. The faces are then resized to 112×112 pixels in order to fit in our branch and each pixel in *[0, 255]* in RGB images is normalized by subtracting *127.5* then dividing by *128*. The resulting images are then used as input to the face branch.

For the voice branch, each audio is converted to single-channel, *16*-bit streams at a *16* kHz sampling rate for consistency. The spectrograms are then generated in a sliding window fashion using a Hamming window of width 25ms and step 10ms. This gives spectrograms of size 512×300 for three seconds of speech. Mean and variance normalisation is performed on every frequency bin of the spectrum. No other speech-specific pre-processing is used. The spectrograms are used as input to the voice branch.

Evaluated Feature Representations

The evaluation involved uni-biometric and multi-biometric feature representations obtained from backbone networks trained on top of *VoxCeleb1-Dev*. In order to optimize model weights, several instances of the network were independently trained through

different loss functions from various families: *Softmax* loss [230], *Center* loss [233], *Ring* loss [234], and *AM-Softmax* loss [235]. More precisely, for each training loss, we trained appropriate models to learn the following feature representations:

- *Uni-Modal Voice* representations extracted from d_v when the voice branch is trained alone (baseline).
- *Uni-Modal Face* representations extracted from d_f when the face branch is trained alone (baseline).
- *Multi-Modal Voice* representations extracted from d_v when the voice branch is trained jointly with the face branch (introduced in this paper).
- *Multi-Modal Face* representations extracted from d_f when the face branch is trained jointly with the voice branch (introduced in this paper).
- *Multi-Modal Face+Voice* representations extracted from d_g when the face branch and the voice branch are jointly trained (introduced in this paper).

Each model was initialised with weights pre-trained on ImageNet. Stochastic gradient descent with a weight decay set to *0.0005* was used on mini-batches of size *64* along *40* epochs. The initial learning rate was *0.1*, and this was decreased with a factor of *10* after *20*, *30* and *35* epochs. The training procedure was coded in Python using Keras.

Evaluation Metrics and Protocols

For each testing data set, the protocol aims to evaluate how the learned representations are capable of verifying, given a pair of test frames/spectrograms, whether the faces/voices come from the same person. For each testing data set, we randomly selected 40 users every time in order to (i) keep constant the number of considered users and (ii) maintain comparable the results across the different data sets (VoxCeleb1-Test has the minimum number of participants among the considered data sets, i.e., 40). Then, we randomly created a list of *20* videos (with repetitions) for each selected user and, from each one of them, we randomly created *20* positive frame/spectrogram pairs and *20* negative frame/spectrogram pairs. The above-mentioned feature representations were considered as feature vector associated to each frame/spectrogram. We used euclidean distances to compare probes and templates. *Equal Error Rate* (EER) was computed to evaluate the performance of the models on the test pairs. The lower the EER, the higher the performance. Lastly, the experiment was repeated, and the results were averaged.

Results on Verification

Fig. 5.5-5.7 plot the results achieved by the learned representations on verification. The loss and the modality performance vary across settings.

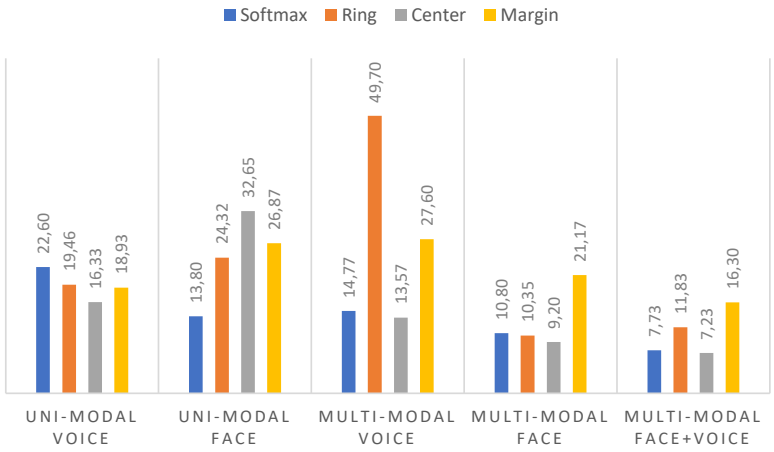


Fig. 5.5: EER verification results on VoxCeleb1-Test.

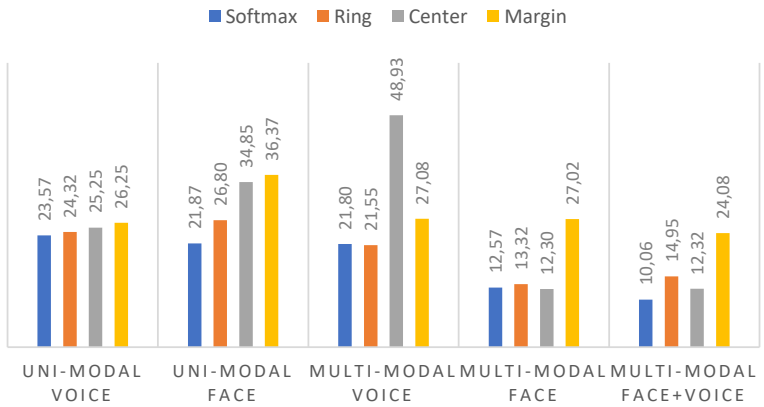


Fig. 5.6: EER verification results on MOBIO.

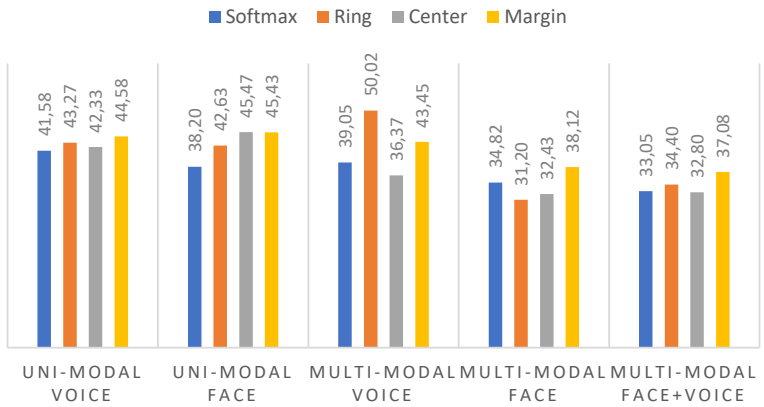


Fig. 5.7: EER verification results on Averobot.

It can be observed that multi-modal face representations achieve lower *EER* than uni-modal face representations with all the dataset and training losses. This means that deep fusion significantly helps to create better sparse representations for the face modality. More precisely, *EER* obtained by representations learned through *Ring* and *Center* losses can be improved of around *50%*, while we observed an improvement of around *25%* thanks to representations learned through *Softmax* and *Margin* losses. It follows that multi-modal face representations better separate genuine and impostor pairs.

Comparable results are obtained by multi-modal voice representations, even though the improvement w.r.t. the uni-modal voice representations is lower, i.e. among *5%* and *10%*. Interestingly, multi-modal voice representations learned through *Ring* loss do not work well. This revealed that *Ring* loss suffers from the deep fusion approach. Our results suggest that such a loss has a minor impact in deep audio-visual fusion settings.

By merging face and voice embeddings into a single representation, the verification performance improves on all the datasets, with all the losses. It can be observed an improvement of around *50%* on all the settings. Face-voice fused representations work well also when learned through *Ring* loss; hence, the deficiencies of multi-modal voice representations learned through this loss are mitigated by fusing voices and faces.

The results across the testing datasets confirm that the context has a relevant impact on the absolute performance of the models, moving from *VoxCeleb1-Test* to *AveRobot* by increasing challenging level. In particular, the verification results on *AveRobot* pairs are *4/5* times worse than the ones achieved on *MOBIO* pairs. This large difference could be related to the more uncontrolled conditions, with dark faces and highly-noisy scenarios.

It is worth noticing that, to get full advantage of combining highly-variant touch and hand-motion biometrics with face biometrics, the multi-biometric verification protocols presented in Section 5.5 required to collect data over multiple sessions before building meaningful multi-biometrics templates. On the other hand, combining voice and face biometrics led to reasonable accuracy just after having observed one evidence of the targeted user; therefore, the multi-biometrics verification protocol we adopted for face-voice authentication focused on one-shot evidence for building templates. Consequently, the results presented in this chapter should not mislead the reader towards considering face, touch, and hand-motion combination has more accurate than face and voice combination, as they were tested on different context-specific verification protocols.

5.7 The Proposed Attack on Uni-modal Verification

In this section, to raise awareness on multi-biometrics, we demonstrate the feasibility of dictionary attacks on otherwise-developed uni-biometrics, using the voice modality as a use case. We targeted large populations without specific knowledge of the individuals or their speech models. Such attacks, recently demonstrated for fingerprints [272, 273],

rely on the necessary usability-security trade-offs in mass deployments (e.g., only partial scans of multiple independent finger impressions), and stand in stark contrast to the prevailing individual-targeted attacks [274]. The widely-known menagerie analysis already hints at this vulnerability - it shows that certain individuals act as *wolves* and can match a lot of users. This suggests the existence of a large family of *Master Voices* (MVs), which match large populations by chance with high probability.

More precisely, building on top of the formalization provided by Section 5.3, finding master voices becomes an optimization problem, which aims to find audio waveforms maximizing the following objective function:

$$\tilde{a} = \underset{a}{\operatorname{argmax}} \underset{u \in U}{\mathbb{E}} \nu_{p,\tau}\left(\mathscr{D}(\mathscr{F}(a)), D_u^N\right) \tag{5.4}$$

5.7.1 Methodology

Verification System and Data Sets

In this study, we leveraged the VoxCeleb data sets [239, 240], two of the largest corpora for speaker verification and identification. They include unconstrained speech of celebrities, extracted from public videos, and featuring diverse acoustic environments.

All waveforms were single-channel, 16-bit recordings sampled at 16 kHz. As acoustic feature extraction $\mathscr{F}$, we computed magnitude spectrograms with a sliding window of size $k = 512$ samples, and a stride of 160 samples. We applied the Hamming window of 512 samples. Then, mean and variance normalisation was performed on every frequency bin. As feature extractor $\mathscr{D}$, we used the VGGVox model [239] pre-trained on the train portion of the first version of the data set $A_{\text{VC1-Train}}$ (1,211 speakers) and validated on the test portion of the same data set $A_{\text{VC1-Test}}$. The model extracts a $e = 1024$-dimensional representation from each $512 \times *$ spectrogram. As similarity function $\mathscr{S}$, we used the cosine similarity. When we applied a policy, we sampled $N = 10$ utterances per user for enrollment. The selected value N can represent a good trade-off between long and short enrolment lists employed by the existing literature. We consider two verification policies p, *Any*N [275] and *Avg*N [236, 237], which rely on a similarity function $\mathscr{S} : D \times D \to \mathbb{R}$:

$$\nu_{p,\tau} = \begin{cases} \text{any}\left(\{\mathscr{S}(d, d_u^i) > \tau : i \in 1, \ldots, N\}\right) & \text{if } p = \textit{Any}N \\ \mathscr{S}\left(d, \frac{1}{N}\sum_{i=1}^{N} d_u^i\right) > \tau & \text{if } p = \textit{Avg}N \end{cases}$$

For our experiments on master voices, we used the second version of VoxCeleb [240]. From it, we sampled two disjoint populations, i.e., $A_{\text{VC2-Train}}$ for exploration and training, and $A_{\text{VC2-Test}}$ for testing. Each included 1,000 speakers, equally divided between the sexes. Each individual was represented by 50 utterances, leading to the total of 50,000 utterances with duration between 4 and 10 seconds for each population.

The verification pipeline achieves an Equal Error Rate (EER) of 8% on utterance-utterance pairs from the validation set $A_{\text{VC1-Test}}$ (consistent with results in [239]), which increases to 11.2% on our sampled population $A_{\text{VC2-Train}}$. Based on the latter evaluation, we chose two global thresholds: $\tau_{\text{EER}} = 0.53$ and $\tau_{\text{FAR1\%}} = 0.74$, which correspond to the Equal Error Rate and False Acceptance Rate (FAR) of 1%, respectively.

Exploratory Analysis

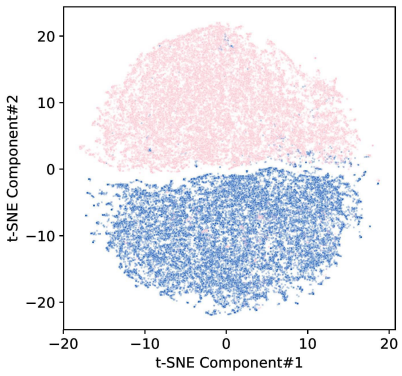
Our first step was to perform exploratory menagerie analysis to assess prevalence of naturally occurring wolves [276] - potential candidates for master voices. We conducted the experiments on $A_{\text{VC2-Train}}$ for male and female speakers separately, since they exhibit distinct characteristics, which is confirmed in feature domain $\mathscr{D}$ (see Fig. 5.8a).

For each user, we first computed the average *genuine score*, which represents how well they match against themselves, and the average *imposter score* indicating how well they match against others (Fig. 5.8b and 5.8c). Each point in the scatter plots represents a user. Intuitively, to find good master voices, we are interested in people in the top of the graphs since they cause a disproportionate number of false acceptances. Interestingly, the model revealed an undesired effect against women, who exhibit significantly greater intra-sex average imposter similarity than men.

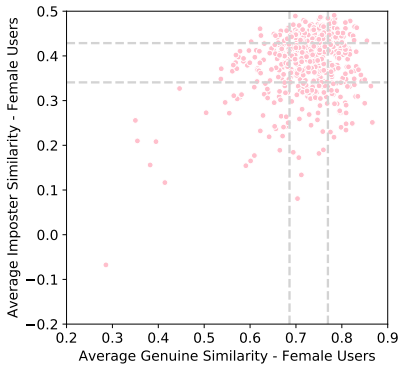
To investigate this phenomenon, we resorted to utterance-level analysis. In Fig. 5.8d, we show false acceptance rates for utterance-to-utterance matching targeting a specific gender. The x axis is ordered by FAR, which leads to nearly no errors at the beginning, and a sudden deterioration in the middle, which corresponds to unlikely erroneous intra-sex matches. On the end, the plot clearly shows the existence of outliers, and significant differences between males and females. The plot also reveals that, with the commonly used equal-error-rate threshold, the female wolves match over 60% of peers' utterances.

We also assessed the consistency of individual speakers to produce easily confusible utterances. The results are shown in Fig. 5.8e and 5.8f for female and male speakers, respectively. For each gender, we ranked the utterances belonging to that gender by decreasing FAR and we grouped them in groups of 500 utterances based on their position in the ranking (i.e., the top-500 utterances with highest FAR belong to the first group and so on). For each user, we then counted how many of her/his utterances belong to each group. Users who generate high FARs have a lot of utterances in the top groups (bottom-left of the plots), while users with less impersonation power have utterances in the last groups (top-right of the plots). It can be observed that, while some users are prone to produce high FARs, their utterances are scattered across the groups. Hence, we can conclude that while the individual speaker properties matter, there is a strong content-related component which may inhibit attacks when what it is said is important.

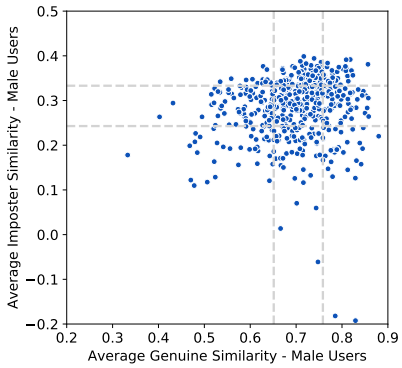
Finally, we assessed the impact of different verification policies (Fig. 5.8g and 5.8h). For each enrolled utterance and user u, we match other users based on their enrollment



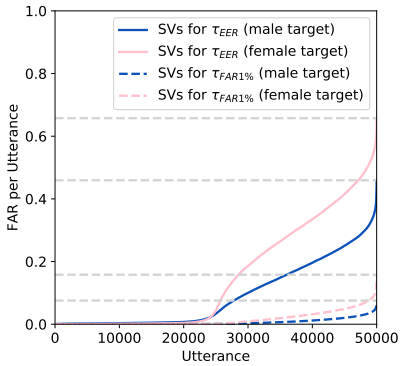
(a) 2-D t-SNE projection of VGGVox vectors.



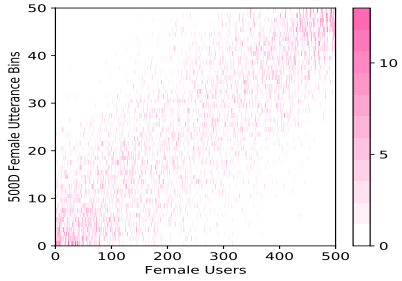
(b) Menagerie analysis within women.



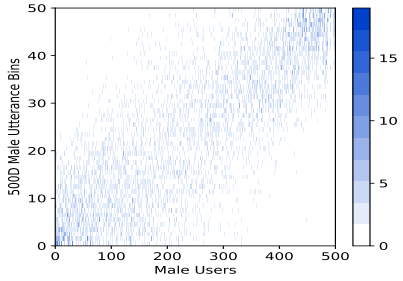
(c) Menagerie analysis within men.



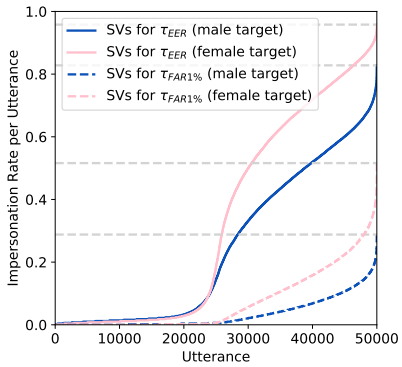
(d) False accept rate per utterance.



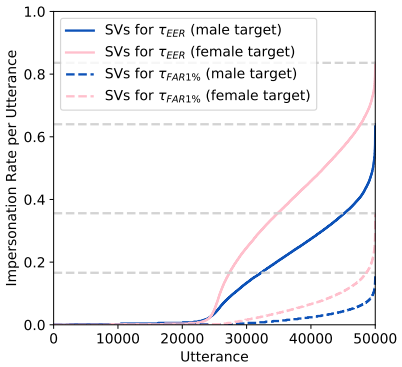
(e) Women ranking based on false accept rates.



(f) Male ranking based on false accept rates.



(g) Impersonation rate with Any10 policy.



(h) Impersonation rate with Avg10 policy.

Fig. 5.8: Exploratory menagerie analysis.

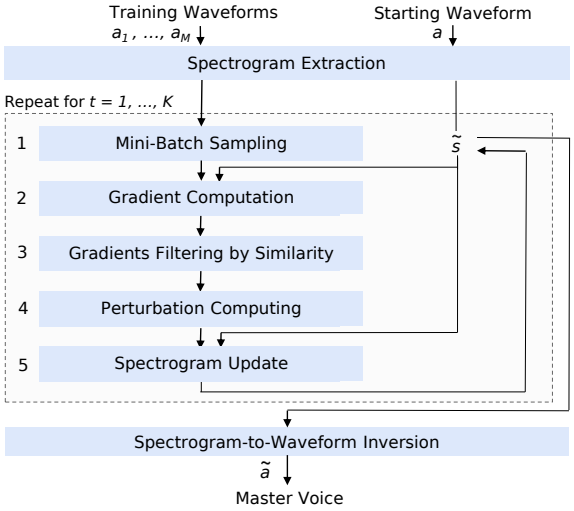


Fig. 5.9: The proposed approach for generating a master voice.

set, verification policy, and decision threshold τ. With the *Any*10 policy, we observed utterances capable of impersonating between 80% and 90% of the users for τ_{EER} and between 20% and 35% for $\tau_{\text{FAR1\%}}$. The results were only slightly worse for the *Avg*10 policy, where we observed impersonation rates between 60% and 80% for τ_{EER} and between 10% and 25% for $\tau_{\text{FAR1\%}}$.

The impersonation results indicate that naturally occurring wolves are good candidates for master voices given existing verification policies[13]. However, speech content and background noise have strong influence as well, leading to difficulties in finding a large collection of master voices for a successful dictionary attack. Hence, in the following section, we explore adversarial perturbations for seeking effective master voices.

The Proposed Optimization

The goal of the optimization process is to generate a *master voice waveform* which maximizes the expected FAR (5.4). Given an existing *seed waveform* a and a set of M training waveforms A_{Train} of a large user population U, we seek $\tilde{a}$, which maximizes $\sum_{u \in U} v_{p,\tau}\left(\mathscr{D}(\mathscr{F}(\tilde{a})), D_u^N\right)$. We will later show empirically that, although the optimization relies on a given population, the results are fully generalizable to unseen individuals.

Our optimization process seeks adversarial perturbations $\tilde{s}$ of the selected *input spectrogram* $s = \mathscr{F}(a)$, and relies on an iterated stochastic gradient descent (Fig. 5.9). By slightly perturbing the input spectrogram, we are able to make it more and more similar to an increasing number of training spectrograms, and bias the verification choices the system makes towards higher FAR.

[13] We acknowledge huge impact of the decision thresholds. However, while a 1% FAR may already seem to be excessive, even state-of-the-art models cannot guarantee acceptable TPRs for stricter thresholds.

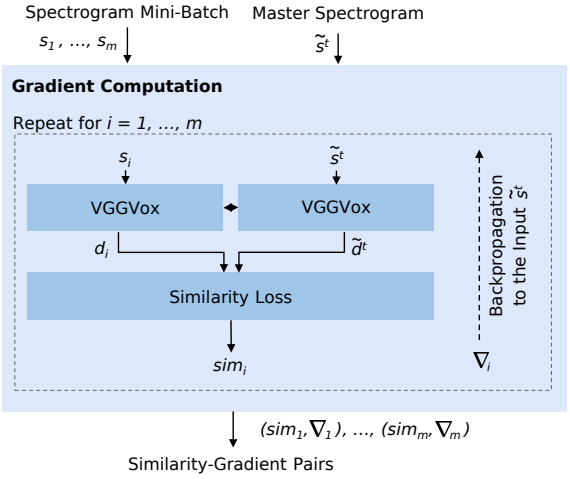


Fig. 5.10: The core of the *Gradient Computation* module.

The steps below are repeated in each iteration t:

1. **Mini-Batch Sampling**. We sample a batch of m spectrograms $S_{\text{batch}} \leftarrow \{\mathscr{F}(a) : a \in A_{\text{Train}}\}$ with $m << M$.

2. **Gradient Computation**. We pair the current iteration of the input spectrogram $\tilde{s}^t$ and the batch spectrograms $\{(\tilde{s}^t, s_i) : s_i \in S_{\text{batch}}\}$ and feed them to the Siamese network for comparison (Fig. 5.10). We compute gradients w.r.t. $\tilde{s}^t$ and feed them to the next step for filtering.

3. **Gradients Filtering by Similarity**. We discard gradients obtained from target examples with similarity outside a certain range $[\mathscr{S}_{min}, \mathscr{S}_{max}]$. This prevents seeking futile optimization directions, i.e., users who we already match, who we have negligible matching chances.

4. **Perturbation Computation**. We compute the adversarial perturbation $\tilde{\nabla}^t$ as:

$$\tilde{\nabla}^t = \max\left(\frac{\alpha}{N}\sum_i \nabla_i, \tilde{\nabla}_{min}\right) \tag{5.5}$$

   where $\tilde{\nabla}_{min}$ is the minimum perturbation, α is the learning rate, and ∇_i is the gradient from i-th filtered pair.

5. **Spectrogram Update**. We update the current estimate of the input spectrogram as:

$$\tilde{s}^{t+1} = \tilde{s}^t + \tilde{\nabla}^t \tag{5.6}$$

The process is repeated until the gain in FAR is higher than γ. We then get a *master voice waveform* $\tilde{a}$ by inverting its optimized spectrogram via the Griffin-Lim algorithm [277].

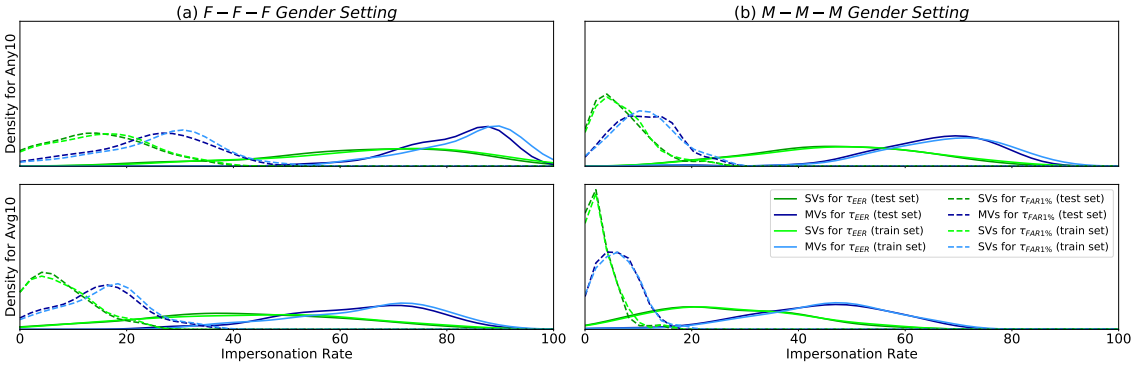


Fig. 5.11: The distribution of impersonation rates for the best-performing settings.

Seed	Gender Train	Test	EER Threshold				FAR1% Threshold			
			Any10		Avg10		Any10		Avg10	
			SVs	MVs	SVs	MVs	SVs	MVs	SVs	MVs
M	**M**	**M**	**47.2**	**65.0**	**27.2**	**44.9**	**6.7**	**16.3**	2.9	**9.6**
M	**F**	**M**	46.6	17.0	26.3	6.9	6.8	1.9	**3.5**	0.7
F	**M**	**M**	3.4	29.3	0.9	14.8	0.3	2.6	0.1	1.2
F	**F**	**M**	3.5	1.5	0.9	0.4	0.3	0.3	0.1	0.1
M	**M**	**F**	4.7	2.1	1.3	0.4	0.3	0.2	0.1	0.0
M	**F**	**F**	5.2	41.9	1.7	23.6	1.8	6.1	1.4	3.3
F	**M**	**F**	62.7	28.3	41.5	15.2	**15.4**	5.6	**8.1**	2.5
F	**F**	**F**	**63.4**	**80.9**	**41.6**	**62.6**	14.4	**34.2**	7.2	**20.8**

Table 5.4: Average impersonation rates for seed and master voices.

5.7.2 Evaluation

Our optimization starts from a seed voice (SV) and aims to improve its impersonation capabilities. To measure the improvement that our optimization can achieve from an arbitrary SV, we sample SVs from the population $A_{\text{VC2-Train}}$ while controlling their initial impersonation power. For each gender, we ordered the corresponding 25,000 utterances according to their inherent impostor score for the $Any10$ verification policy and threshold τ_{EER}. We sampled 200 utterances from uniformly distributed percentiles in the population. Finally, we used our optimization procedure, to yield 100 master voices (50:50 for males and females, respectively) optimized for intra-sex matching (i.e. training only on utterances from the same gender in $A_{\text{VC2-Train}}$), and 100 master voices optimized for inter-sex matching (i.e. training only on utterances from the opposite gender in $A_{\text{VC2-Train}}$). We assess their impersonation rates on a disjoint population of 1,000 people $A_{\text{VC2-Test}}$.

Table 5.4 compares the impersonation rates for Seed Voices (SVs) and master voices (MVs) under different training and testing genders, verification policies, and thresholds. The reported results correspond to the test population $A_{\text{VC2-Test}}$, with people unseen at the time of optimization. On average, we can improve the seed impersonation rate by 20 and 10 percentage points for τ_{EER} and $\tau_{\text{FAR1\%}}$, respectively. For the least secure setting

with the *Any*10 policy and threshold τ_{EER}, on average, a MV can impersonate 80% of females and 65% of males. In the most secure configuration, the *Avg*10 policy and $\tau_{\text{FAR1\%}}$, on average, a MV can still impersonate 20% of females and 10% of males.

Regarding gender, we observed a significant improvement in the impersonation rates when the same sex is chosen for seed, training and testing samples (i.e., M-M-M and F-F-F settings). Moreover, when the training and the testing gender are the same (i.e., F-M-M and M-F-F settings), the results seem to be good, independent from the seed gender - except for *Avg*10 policy at $\tau_{\text{FAR1\%}}$. This means that the added noise makes it possible to use perturbed utterances to impersonate users of the opposite gender. In contrast, settings with different training and the testing genders (i.e., M-F-M, F-F-M, M-M-F, F-M-F) led to poor results, highlighting how relevant the training gender is on the optimization. Female MVs seem to be more powerful than male MVs.

In Fig. 5.11 we show the distribution of impersonation rates in the populations of seed and master voices for F-F-F and M-M-M gender settings. The probability of finding an utterance with high impersonation rate is low in SVs (green lines), while it significantly increases in MVs (blue lines). This means that MVs produce high impersonation rates independently from the starting utterance and, thus, from the speaker, the speech and the environment. The difference in impersonation rates between train (lighter colors) and test populations (darker colors) is negligible, so MVs generalize well across populations.

5.8 Findings and Recommendations

In this chapter, we provided data and approaches for performing learners' authentication in ubiquitous settings. More precisely, we presented an approach for integrating face authentication together with behavior authentication to secure the user access to online learning platforms, going beyond one-off authentication. We also proposed a multi-biometric model that digests face and voice traits in parallel, and we explored how it helps to improve recognition performance in learners' verification scenarios, both online and onlife. Furthermore, we performed the first analysis of speaker verification systems in the context of recently reported dictionary attacks on biometrics. Such attacks might put at risk current uni-modal verification approaches.

Based on the obtained results, we can conclude that:

1. Strengthening face biometrics with touch and hand-motion biometrics leads to higher authentication accuracy, without compromising the stringent real-time requirements for continuous authentication. The computational time of the multi-biometrics verification is still dominated by the time required for processing the original face biometrics, and verification is run in parallel over sub-modules to avoid additional overhead.

2. Even though touch and hand-motion exposed a low persistence over time, they can

be effectively used to recognize people within few weeks.

3. Face and voice models can benefit from deep intermediate fusion, and the recognition improvement depends on the modality, the loss, and the context.
4. Uni-biometric face models exhibit higher accuracy improvements than uni-biometric voice models, after being jointly trained at intermediate level.
5. Merging face and voice into a single embedding vector at intermediate level positively impacts the accuracy of multi-biometric audio-visual models.
6. Face and voice models jointly trained at intermediate level generalize well across populations, and are more robust when applied in challenging contexts.
7. Speech seems susceptible to dictionary attacks, and both speaker and speech content affect impersonation rates.
8. Dictionary attacks based on adversarial optimization can be used to significantly increase impersonation capabilities of arbitrary inputs.
9. The gender bias of a verification model increases the difference in exposure to dictionary attacks across populations based on the gender.

In next steps, we plan to investigate other approaches to improve the recognition scores returned by individual biometric subsystems, the adoption of additional biometrics, the way the system trusts the user's genuiness, and the capability of working well when one motion signal lacks. Moreover, we will investigate master voice transferability and generative models usage for master voice generation. Finally, we plan to build proper countermeasures to the uncovered attack.

Chapter 6

Machine Learning Models for Sentiment Analysis

Research Highlights
- The *COCO-SA* data set enables deep sentiment analysis in education.
- Our *neural network* has been proved to be accurate for sentiment analysis.
- Context-trained embeddings perform better than general-purpose ones.
- Context-trained *Word2Vec* embeddings show the lowest error score.

6.1 Introduction

Individuals use *online social platforms* to express opinions on products and services. Such user-generated data, generally a *text* (e.g., reviews, tweets, wikis, blogs), is often characterized by a *positive or negative polarity* according to the satisfaction of people who write it. Understanding individual's satisfaction is a key element for companies and institutions to do *collective reasoning*. The automatic investigation performed on person's opinions usually relies on *Sentiment Analysis* (SA). Techniques and systems in this field aim to extract and classify emotions and sentiments by combining *Natural Language Processing* (NLP), *Text Mining* and *Computational Linguistics* [278]. Exploiting this artificial intelligence makes it possible to complement common practices (e.g., focus groups or surveys), but still presents challenges because of context dependencies of data.

Online educational platforms deployed at large scale are earning more and more attention as *social spaces* where students can consume content and *share opinions regarding their experience* [279]. For instance, such a collective intelligence might be useful for peers who are planning to attend a given course, instructors interested in improving their teaching practices, and providers who can get benefits from learners' feedback to refine the platform itself. Therefore, these environments can be envisioned as *dedicated*

social channels where discussions are focused on specific topics concerning course content quality, teachers' skills, and so on [280]. SA approaches on students' opinions have recently started to gain attention [281], and their design is still an open challenge.

Prominent SA solutions use *word embeddings* as feature vectors, i.e., distributed representations that model words properties in vectors of real numbers which capture syntactic features and semantic word relationships [282, 283, 46, 284, 285]. Generating word embeddings is based on distributional co-occurrences of adjacent words able to model words meanings not visible from their surface. This exploits the fact that words with a similar meaning tend to be connected by a given relation. For instance, the verbs *utilize* and *use*, synonym although syntactically different, present similar sets of co-occurring words and can be considered similar, while a third verb, such as *play*, has different co-occurrences and should be considered different from both *utilize* and *use*.

Training *deep neural networks* on *word embeddings* is making it possible to learn complex data representation, increasingly higher-level features, and correctly measure properties held by data, especially in SA tasks [286, 282, 283]. However, the literature acknowledges that models generated from general-purpose data sets, independently from any specific domain, under-perform the ones built on data from the target context of the SA task. This happens because *context-trained models* may capture specific patterns from the target context, while generic-trained ones might learn patterns acting as noise for it. Hence, using *deep neural networks powered by word embeddings* learned from e-learning data can enable improving the effectiveness of the dedicated SA systems.

In this chapter, we aim to investigate deep learning and word embeddings for sentiment analysis in educational contexts. To this end, we propose a deep learning model, trained on word embedding representations coming from the e-learning context, that is able to predict a sentiment score for reviews posted by learners. We also report its experimental evaluation on a large-scale dataset of online course reviews. More precisely, we show how word embeddings trained on smaller context-specific textual resources are more effective with respect to those trained on bigger general-purpose resources. Moreover, we highlight the benefits derived from combining word embeddings and deep learning approaches for sentiment analysis in education.

The contribution of this chapter is threefold:

- We arranged a *large-scale data set* of textual reviews extracted from *Udemy*, including over 16K instructors and 2, 5M learners who provided 4.5M reviews over 43k courses.
- We propose a *fully-automated approach* for effective and efficient classification of textual reviews based on educational-specific word embeddings as features and on top of a deep learning architecture for prediction, going over traditional term-based methods.
- We provide an *extensive evaluation* in terms of effectiveness and efficiency for review classification, comparing our approach with different combinations of feature types and algorithms to assess which one is the best at classifying learners' reviews.

The rest of this chapter is structured as follows: Section 6.2 describes the most representative techniques for sentiment analysis, going deeply on those tested in education. Then, Section 6.3 formalizes the problem we seek to investigate. Section 6.4 introduces the learners' review data set we collected. Section 6.5 describe and evaluate the classification model we trained on such a data. Finally, Section 6.6 concludes the chapter.

6.2 Related Work

6.2.1 Sentiment Analysis Techniques

Common SA approaches can be classified in *supervised* and *unsupervised* [287]. Supervised approaches require a training data set annotated with numerical values left by users or inferred from the content in the text, and leverage it to build a classifier that predicts a sentiment score for unseen data. Common supervised pipelines require the extraction of features from the text. Such features are then fed into an algorithm that assigns a positive or negative polarity to the text. On the other hand, unsupervised approaches rely on lexicons associated with sentiment scores in order to model the polarity.

ML has been extensively adopted for SA tasks. For instance, the authors in [288] used *Maximum Entropy* (ME) and *Naive Bayes* (NB) algorithms, adopting syntactical and semantic patterns extracted from words on *Twitter*. Their method relies on the concept of contextual semantic i.e. considering word co-occurrences in order to extract the meaning of words [289]. When evaluated on nine Twitter data sets, their method showed better performance than baselines. More recently, the authors in [290] applied *Stochastic Gradient Descent* (SGD) and *Support Vector Machine* (SVM) algorithms to classify movies reviews in a binary classification problem (i.e. positive or negative). They showed that the use of a n-gram model to represent reviews obtains higher levels of accuracy when the value n was small. Moreover, they showed that combining uni-gram, bi-gram, and tri-gram features was better than using a single representation at once.

The SA domain also experienced the influence of the wide spread of *deep learning* approaches. For instance, in [46], a *Convolutional Neural Network* (CNN) was designed to capture features from character to sentence level. An ensemble neural network was proposed by [291], where various sentiment classifiers trained on different sets of features were combined. They performed experiments on six different datasets coming from Twitter and movies reviews, showing improvements when compared with state-of-art ML baselines. Another approach combines various ML classifiers with DL ones [292]. A SVM classifier was mounted on top of a 7-layer CNN in order to complement the characteristics of each other and obtain a more advanced classifier. They were able to obtain more than 3% of improvement compared to a classic DL model. To learn continuous representations of words for SA, a combination of CNN with a *Long-Short Term*

Memory (LSTM) was exploited by [293]. They were able to assign fixed-length vectors to sentences of varying lengths, showing how DL outperform common ML algorithms.

Emerging deep learning SA approaches have been usually fed with word embeddings, improving accuracy of baseline methods not using word embeddings. For instance, the authors in [284] incorporated prior knowledge at both word and document level with the aim to investigate how contextual sentiment was influenced by each word. On the same direction, other researchers employed sentiment of text for the generation of words embeddings [294]. They joined context semantics and sentiment characteristics, so that neighboring words have both a similar meaning and sentiment. Similarly, the authors in [295] augmented sentiment information into semantic word representations and extended the Skip-gram model, coming up with two sentiment word embedding models. The learned sentiment Word Embeddings were able to represent sentiment and semantics. Furthermore, the authors in [296] presented a model that uses a mix of unsupervised and supervised techniques to learn word vector representations, including semantic term-document features. The model showed performances higher than several ML methods adopted for sentiment detection. Focusing on Twitter sentiment classification, the authors in [297] trained sentiment-sensitive words embeddings through the adoption of three neural networks. Their experiments showed it outperformed the competitors.

Several challenges have been created to solve SA polarity detection task and several resulting winning systems employed word embeddings within their core. For instance, the *Semantic Sentiment Analysis* challenge [298, 299, 300, 301], held within the ESWC 2018 conference, included a polarity detection task where participants were asked to train their systems by using a combination of word embeddings. The annual *SemEval* workshop usually includes SA polarity detection, and most prominent solutions employ neural network architectures with various types of input features, including word embeddings. *Kaggle*[1] hosts several challenges, and some were related to SA. For instance, the *Sentiment Analysis on Movie Reviews* challenge[2] asked participants to label the movie reviews collected in the Rotten Tomatoes data set [302] on a scale of five values: negative, somewhat negative, neutral, somewhat positive, positive. One recent challenge, namely *Bag of Words Meets Bags of Popcorn*[3], looked for deep learning models combined with word embeddings for polarity detection on movie reviews [296].

6.2.2 Sentiment Analysis in Education

Education has recently gained the attention of SA in order to get knowledge from dynamics that online platforms enable. More precisely, studying sentiments in education can support learning and teaching evaluation, assessing the quality of the involved tools, increasing students' motivation, and refining learning processes [303].

[1] https://www.kaggle.com/

[2] https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews

[3] https://www.kaggle.com/c/word2vec-nlp-tutorial

Based on recent surveys [279], it can be possible to identify two main sentiment analysis approaches used in this domain: the *machine learning* and *lexicon-based* approaches. On the one hand, machine learning approaches can be divided into *supervised* and *unsupervised* machine learning approaches. Regarding supervised machine learning approach, there are several classifiers used in education domain such as *decision tree*, *linear*, *rule-based*, and *probabilistic classifiers*. On the other hand, the lexicon-based approach uses techniques such as *dictionary-based* and *corpus-based* approaches. For instance, one work embraces the use of Sentiment Analysis to mine students' interactions in collaborative tools, guaranteeing the students' communication privacy in their opinions [304]. Another relevant area that exploited SA was teachers' assessment. For example, the authors in [281] adopted a Support Vector Machine to evaluate the teachers' performance using 1,040 comments of systems engineering students as a dataset. The evaluation of the teaching-learning process was also object of study through SA on comments posted by both students and teachers [305]. Similarly, the authors in [306] studied sentiments to build an adaptive learning environment with recommendations.

6.3 Problem Formalization

The sentiment polarity detection task can be formalized as follows. Given N tuples $\mathbb{D} = (x_i, y_i)_{i=1}^{N}$ where each $x_i \subset \mathbb{X}^*$ corresponds to a textual comment of unknown length composed by words from a dictionary $\mathbb{X}$, and each $y_i \subset \mathbb{Y}$ corresponds to the polarity of the comment from a pre-defined set of polarities. We consider an explicit feature extraction step which produces fixed-length representations in $\mathbb{F} \subset \mathbb{R}^e$. We denote the stage as $\mathscr{F} : \mathbb{X} \rightarrow \mathbb{F}$. Given a feature vector $f \subset \mathbb{F}$, we seek to create a regression model $\mathscr{G}_\theta$ that maps f into a continuous value closed to y. Namely, we want to find a function $\mathbb{G}_\theta : \mathbb{F} \rightarrow \mathbb{Y}$ that outputs the floating-point polarity of the comment x whose feature vector is $f = \mathscr{F}(x)$. Finding such a regression model implies to minimize the following objective function:

$$\mathscr{G}_\theta = \operatorname{argmin} \mathop{\mathbb{E}}_{(x,y)\in\mathbb{D}} |\mathscr{G}_\theta(\mathscr{F}(x)) - y| \tag{6.1}$$

6.4 The Proposed COCO-SA Data Set

COCO-SA contains over 1M reviews from more than 4.5M learners, extracted from courses delivered on *Udemy*. Unlike academic-oriented platforms driven to traditional coursework, Udemy enables experts in various areas to offer courses at no charge or for tuition fees. In comparison with other online course platforms, no third-party control on reliability, validity, accuracy or truthfulness of the course content is performed. All copyright and registered trademarks remain property of their owners. To the best of our

Data Set	#Reviews	Context
Stanford Movies [285]	10,605	Movie Reviews
IMDB [164]	25,000	Movie Reviews
Sentiment140 [307]	160,000	Twitter Posts
Dranziera [308]	1,000,000	Amazon Reviews
Stanford Amazon [309]	34,686,770	Amazon Reviews
ForumSent [310]	1,075	Online Learning Forum Posts
FacebookEdu [311]	10,000	Facebook Education Posts
TwitterEdu [312]	19,997	Twitter Education Posts
CourseraC3 [313]	35,590	Online Learning Forum Posts
COCO-SA	**1,396,312**	**Online Course Reviews**

Table 6.1: Representative data sets for textual sentiment analysis.

knowledge, there exists no other benchmark large-scale data set that includes educational ratings from online courses. Table 6.1 summarises other representative data sets used for sentiment analysis. Besides lacking online courses conditions, most of them include only few reviews and the set of review polarities is limited (e.g., two or five).

6.4.1 Collection Methodology

This section describes our *multi-stage approach* for collecting a large educational review data set. The pipeline is summarised in Fig. 6.1, and key stages are discussed below:

1. **Candidate Courses Retrieval**: The first stage is to obtain a list of courses from *Udemy*. We start from the list of courses that are returned by *Udemy APIs*[4], that exposes functionalities to help developers accessing content and building external applications. The script retrieved 43,023 courses, dumped in November 2017.

2. **Course Reviews Download**: To retrieve the reviews released by learners, the script uses the *Udemy APIs* method aimed to return course reviews given the course identifier. The query result gave 6M rows, each with the course id, the timestamp, the rating between 0 and 5 with step 0.5, and the comment.

3. **Data Cleaning and Pruning**: To ensure that the data set can be used for benchmarking sentiment analysis, we took only reviews whose comment has at least 20 characters. The re-sampled data set includes 1,396,312 reviews from 30,599 courses.

4. **Semantic Features Enrichment**: We enriched each review with Part-of-Speech (PoS) tags computed by the Natural Language Toolkit[5] (NLTK), and keywords and concepts computed by the IBM Watson Natural Language Understanding APIs[6].

[4] https://www.udemy.com/developers/

[5] http://www.nltk.org/

[6] https://www.ibm.com/watson/services/natural-language-understanding/

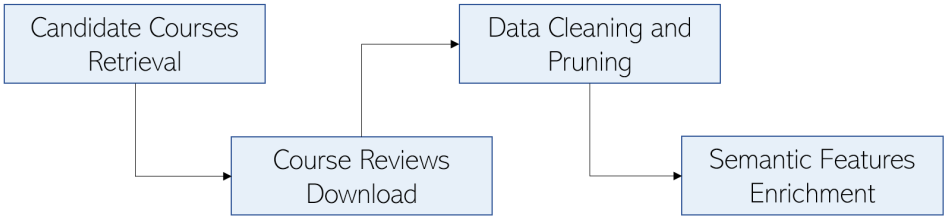


Fig. 6.1: Collection pipeline for COCO-SA data set.

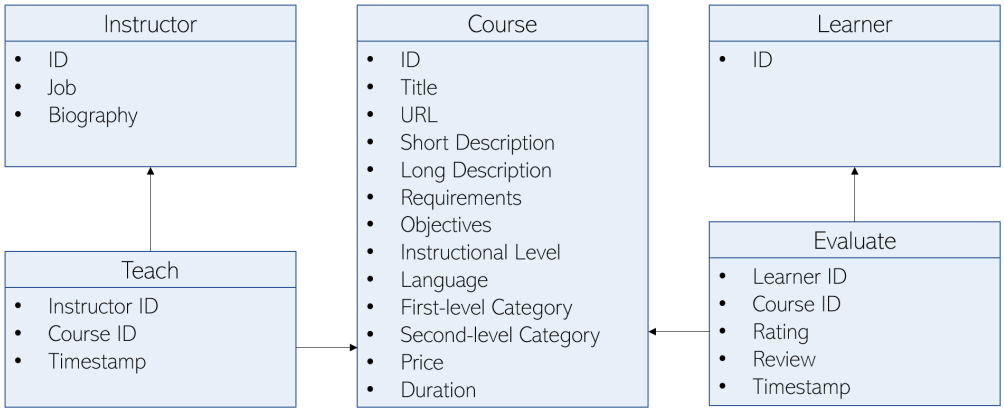


Fig. 6.2: Structure of the COCO-SA data set.

6.4.2 Structure and Statistics

COCO-SA is a *CSV*-based collection whose structure in terms of entities and associations is depicted in Fig. 6.2. Text attributes have *Unicode* coding, while languages and timestamps hold *ISO639-1* and *ISO8601* standards, respectively.

Course is the most informative entity. First, *id* and *course URL* provide unique identification attributes. Then, the course is described by *short* and *long descriptions*. *Requirements* and *objectives* list technical and pedagogical needs at the beginning and expected learner skills at the end, respectively. The *language*, the *instructional level* (*beginner*, *intermediate*, *expert*), *first/second-level categories*, and *tags* are listed. Each course has only one first-level category and one second-level category. Other course fields identify the current *price* and the *discount*. Numerical attributes list the *course duration* in hours.

The *Instructor* and *Learner* entities only include information available on the corresponding public profiles. Each entity instance is uniquely identified by a fake *id*, so that the id stored into the data set does not correspond to the real id of the user. Each instructor is described by the job title and biographic keywords. Regarding relationships, in *Teach*, the pairs of *instructor id* and *course id* model the association among instructors and the courses they teach. One instructor can teach more than one course and the same course can have one or more instructors. *Evaluate* contains *learner id*, *course id*, the [0-5] *rating* with step 0.5, the *comment*, and the *timestamp*.

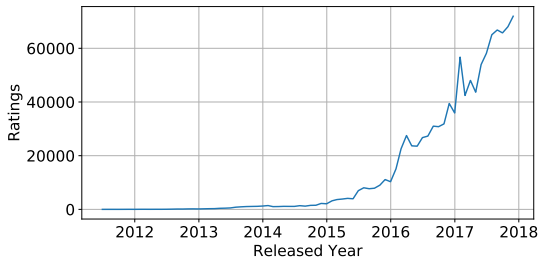


Fig. 6.3: Reviews/year on COCO-SA.

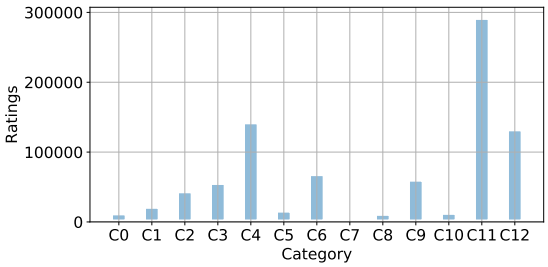


Fig. 6.4: Reviews/category on COCO-SA.

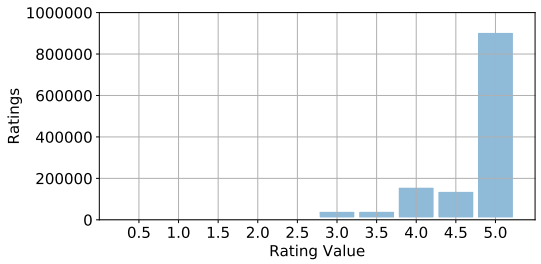


Fig. 6.5: Polarity distribution on COCO-SA.

By counting the comments released over time, it can be observed that their number grows exponentially in recent years (Fig. 6.3). The data set includes 30,599 courses, distributed over 15 first-level categories (see Fig. 6.4), that received 1,396,312 comments.

The comment distribution is unbalanced across categories (avg. 67,755; st.dev. 78,049; min 2,968; max 292,693). Similarly, the languages distribution along courses follows such a trend. Only 21% of courses do not use English as primary language. The reviews are non-uniformly distributed across the range of values they can hold (Fig. 6.5). Most of the polarity ratings have a value of 5. Only few courses received lots of reviews. The distribution of the number of reviews per courses (avg. 45; st.dev. 265; min. 1; max. 17,695) shows a downward trend, but there is a large number of courses with a lot of comments. Each review is usually short in terms of number of characters (avg 141; st.dev. 195; min 1; max 65,000). This makes the data set challenging since the prediction algorithms will rely on only few words for classifying the sentiment behind the text.

6.5 The Proposed Sentiment Prediction Approach

6.5.1 Methodology

In this section, we describe our deep learning approach for performing regression in a polarity detection task, through word embeddings. Fig. 6.6 depicts its main components.

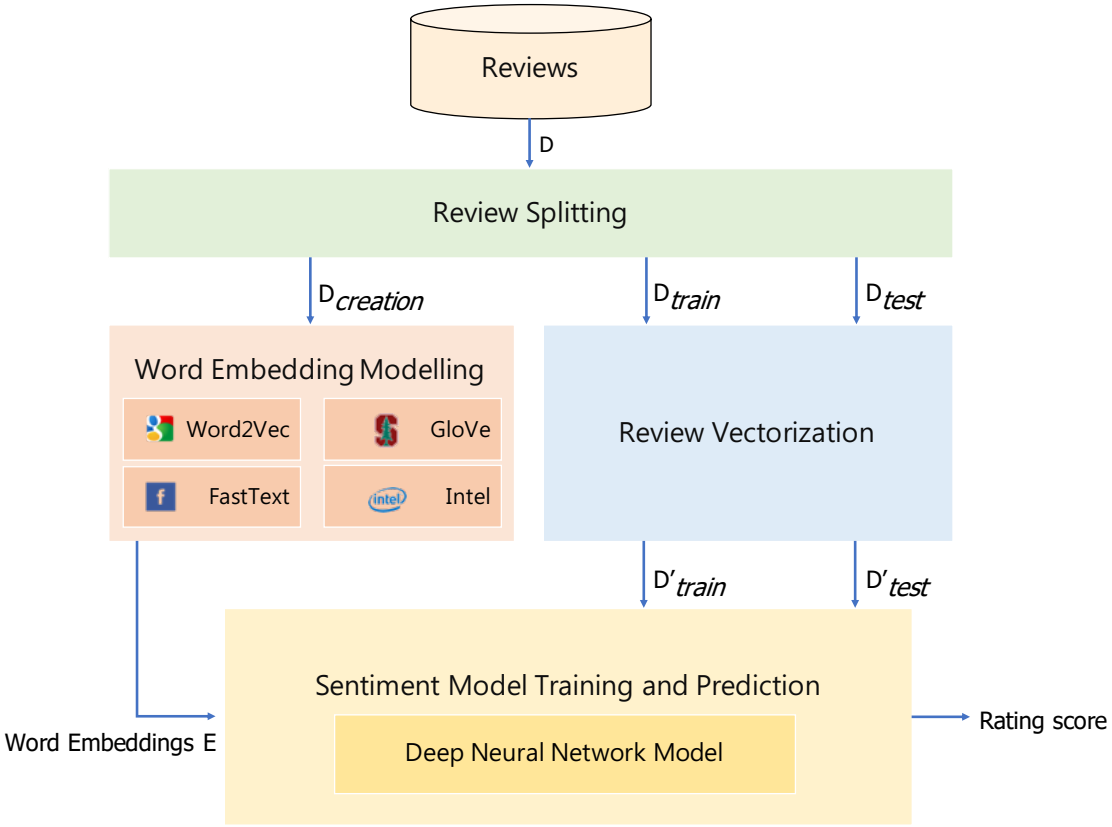


Fig. 6.6: The proposed approach for sentiment prediction.

The approach is designed to be modular and extensible. The addition and the update of word embedding algorithms or neural networks involve almost no changes. Moreover, each component is properly parametrized to facilitate its configuration. All the modules are described in the following sections.

Review Splitting

The *review splitting* step serves to define the various dataset splits over the model development. The system digests a dataset D of N reviews organized as follows:

$$D = \{(text_1, score_1), ..., (text_N, score_N)\} \tag{6.2}$$

where $text_i$ is a textual review and $score_i$ is an integer label that belongs to the set $C = \{score_1, ..., score_M\}$.

During this step, we thus need to split input data D in three subsets, each for a specific phase of the development:

1. $D_{creation}$: the sample of data used to create word embeddings.
2. D_{train}: the sample of data used to fit the model (i.e., weights and biases).
3. D_{test}: the sample of data used as the gold standard to evaluate the model.

In order to do this, we set up two split ratios and we assign the text-score pairs in D to the different subsets $D_{creation}, D_{train}, D_{test}$ according to them:

1. $s_{creation} \in [0, 1]$: the percentage of reviews for each class $c \in C$ that are randomly chosen from the set D to create word embeddings, yielding $D_{creation}$.
2. $s_{training} \in [0, 1]$: the percentage of reviews for each class $c \in C$ that are randomly chosen from $D \setminus D_{creation}$ to train the model, yielding D_{train}.

The remaining reviews represent D_{test}. The overall procedure ensures that the subsets are disjoint and their union covers the entire dataset D.

Word Embedding Modelling

The state-of-the-art method to model a word with a vector is using word embeddings; it is common to see word embeddings that are 256-dimensional, 512-dimensional, or 1,024-dimensional when dealing with very large vocabularies. There are two ways to generate and leverage word embeddings:

1. Learn word embeddings jointly with the same context we are interested in by starting with random word vectors and, then, learning word vectors along the process.
2. Load into the sentiment prediction model the word embeddings pre-computed using a different ML task than the one we are interested in. If the amount of training data in D_{train} is small, this is the common solution.

To the best of our knowledge, no word embedding database specifically targets the e-learning context. Therefore, this step goes through the first most general solution of learning word embeddings from scratch on top of the COCO-SA data set, while we also use word embedding pre-computed on other contexts for comparison along the chapter.

In order to generate word embeddings from scratch, the subset of pre-processed reviews $D_{creation}$ was employed. We concatenated them into a large corpus and this corpus was fed into a given word embedding generation algorithm selected among the following ones: *Word2Vec* [314], *GloVe* [315], *FastText* [316], or *Intel* [317]. Each of them outputs a set of feature vectors E for words in that corpus. The feature values are non-negative real numbers. For each distinct word w in the vocabulary in $D_{creation}$, there exists a corresponding feature vector $e \in E$ which represents the word embedding for that word. All the feature vectors share the same size. The size of the word embeddings and of the window where the word embeddings generator looks at contextual words can be arbitrarily selected. These two values are parametrized on our approach.

Review Vectorization

Review vectorization is the process of transforming each review into a numeric sequence. This can be done in multiple ways (e.g., segment text into words and transform each word into a vector, segment text into characters and transform each character into a vector, extract n-grams of words or characters, and transform each n-gram into a vector). The different units into which the text is broken (words, characters, or n-grams) are called *tokens*, and breaking text into such tokens is called *tokenization*. The process consists of applying some tokenization schemes and then associating numeric vectors with the generated tokens. These vectors, packed into sequences, are needed for manipulating text during sentiment model training and inference.

In order to be treated by machines, we need to turn the reviews within D_{train} and D_{test} into a set of integer-encoded pre-processed reviews defines as follows:

$$D'_{train} = \{(text'_1, score_1), ..., (text'_K, score_K)\} \forall (text_i, score_i) \in D_{train} \tag{6.3}$$

$$D'_{test} = \{(text'_1, score_1), ..., (text'_J, score_J)\} \forall (text_i, score_i) \in D_{test} \tag{6.4}$$

where each pair $(text'_i, score_i)$ includes an integer encoding of the text comment $text_i$ and the original polarity $score_i$ from D_{train} and D_{test}, respectively.

The process for generating D'_{train} and D'_{test} works as follows. Each word is associated to a unique integer value chosen from a range $[0,|V|-1]$, where V is the vocabulary in D. For each input review $(text_i, score_i)$, we build an integer-encoded vector $text'_i$ from $text_i$, where an integer value at position j in $text'_i$ represents the mapped value for word w for that position in $text_i$. The sets D'_{train} and D'_{test} are vectorized.

Deep Neural Network Modelling

This step is necessary for defining the architecture and the optimization components of the deep neural network that (i) takes pairs of integer-encoded texts and sentiment scores, (ii) maps such texts into word embeddings, and (iii) predicts the sentiment scores.

The proposed architecture tailored for sentiment score prediction is shown in Fig. 6.7. Given that our training process requires running the network on a rather large corpus, our design choices are mainly driven by the computational efficiency of the network. Hence, differently from [286], which presents an architecture with two Bidirectional LSTM layers, we adopt a single Bidirectional LSTM layer architecture. Moreover, we configure the last layer to return a single continuous value, i.e., the predicted sentiment

score. Therefore, our network is composed by an Embedding layer followed by a Bidirectional LSTM layer, a Neural Attention mechanism, and a Dense layer:

1. **Embedding Layer** takes a two-dimensional tensor of shape (N, M) as input, where N represents the number of integer-encoded text comment samples, while M the maximum sequence length of such samples. Each entry is a sequence of integers passed by the Input Layer. The output of the Embedding layer is a two-dimensional vector with one embedding for each word w in the input sequence of words of each text comment t. Before receiving data, the Embedding Layer loads the pre-trained word embeddings computed during the previous step as weights. Such weights are frozen, so that the pre-trained parts are not updated during training and testing.

2. **Bidirectional LSTM Layer** is an extension of the traditional LSTM that generally improves model performance on sequence classification problems. It trains two LSTM instead of just one: the first is trained on the input sequence as it is and the second on a reversed copy of the input sequence. The forward and backward outputs are then concatenated before being passed on to the next layer, and this is the method often used in studies of bidirectional LSTM. Through this layer, the model is able to analyze a review as a whole, binding first and last words coming up with a more precise score. Moreover, by exploiting the bidirectional version of a LSTM, the model is able to get patterns that better model the learners' writing style.

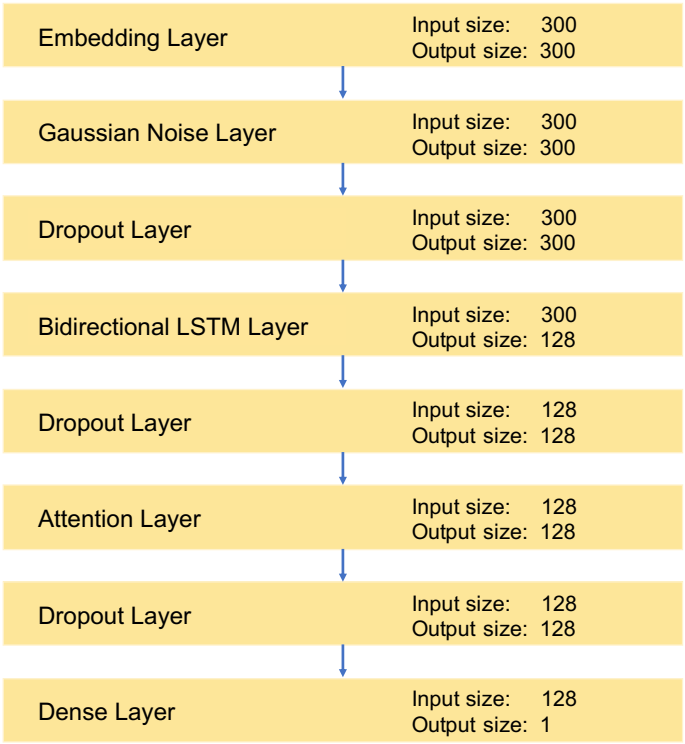


Fig. 6.7: The proposed deep learning architecture for sentiment prediction.

3. **Attention Layer** enables the network referring back to the input sequence, instead of forcing it to encode all the information forward into one fixed-length vector. It takes n arguments $y_1, ..., y_n$ and a context c. It returns a vector z which is supposed to be the summary of the y_i, focusing on information linked to the context c. More specifically, in our model it returns a weighted arithmetic mean of the y_i, and the weights are chosen according to the relevance of each y_i given the context c. This step can improve performance, detecting which words more influence the sentiment.
4. **Dense Layer** is a regular densely-connected layer implementing a function output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function, while kernel and bias are a weights matrix and a bias vector created by the layer, respectively. The layer uses a linear activation $a(x) = x$ and provides a single output unit representing the sentiment score.

To mitigate overfitting, the network augments the cost function within layers with l_2-norm regularization terms for the parameters of the network. It also uses Gaussian Noise and Dropout layers to prevent feature co-adaptation.

Sentiment Model Training and Prediction

The fresh instance of the sentiment model takes a set of neural Word Embeddings E together with a set of pre-processed reviews D'_{train}, as input. With these embeddings and reviews, the component trains the deep neural network. As an objective, the network measures the MSE (Mean Squared Error) of the predicted sentiment score against the gold standard value for each input sequence. Parameters are optimized using RMSProp (Root Mean Square Propagation) [318] with $learning\_rate = 0.001$. The network was configured for training on batches of size 128 along 20 epochs, shuffling batches between consecutive epochs. The trained neural network takes a set of unseen reviews D'_{test} and returns the sentiment score $score'$ predicted for that comment $text'$.

6.5.2 Evaluation

In this section, we describe the experiments we conducted to explore the effectiveness and efficiency of our approach over the *COCO-SA* data set. More precisely, we used textual reviews in *COCO-SA* as an input to our approach, while the corresponding ratings were considered as ground-truth sentiment labels for the input reviews.

In our experiments, we considered reviews with non-empty English text comments. They are 1,396,312 in COCO-SA. Each review includes a rating ranging from 0.5 to 5 with step of 0.5. Considering that our approach supports only integer ratings, we mapped COCO-SA ratings on a scale from 0 to 9 with steps of 1. The dataset D included 1.396.312 reviews and the split ratios were $s_{creation} = s_{train} = 0.90$. Those values were selected

since we wanted to keep both training and testing sets with balanced rating distributions. Moreover, we performed 10-fold stratified cross validation. Hence, 1,396,312 - 6,500 * 10 reviews were put in $D_{creation}$ to create embeddings, while 5,850 * 10 were put in D_{train} for training the model and 650 * 10 were put in D_{test} for testing.

The approach was developed in Python using Keras on top of a TensorFlow backend. The experiments were carried out on a NVIDIA Titan XGPU equipped with 16 GB RAM.

Evaluation Metrics and Protocols

In order to evaluate the performance of our model, we measured the MSE (Mean Squared Error) and the MAE (Mean Absolute Error) scores, defined as follows:

$$MAE(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \tag{6.5}$$

$$MSE(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \tag{6.6}$$

where y_i is a true target value, $\hat{y}_i$ is a predicted target value, and n is the number of samples for both (6.5) and (6.6). It follows that given two algorithms a_1 and a_2, a_1 is better than a_2 if its MSE and MAE are lower than those of a_2. During experiments, each of the 10 polarity labels has been equally represented in the testing data set [92].

Baselines

We compared our approach with the following common ML algorithms:

- *Support Vector Machine (SVM).* Support Vector Machines aim to build hyperplanes among data samples in such a way that the separation between classes is as large as possible. In the regression variant, named SVR (Support Vector Regressor), the algorithm tries to find hyperplanes that predict the distribution of data.
- *Random Forests (RF).* Random Forests is based on an ensemble of decision trees, where each tree is trained and votes for a class for the presented data [319]. We use a Random Forest with 10 trees with depth 20. Essentially, each decision tree splits data into smaller data groups based on the features of the data until there are small enough sets of data that only have data points with the same label. These splits are chosen according to a purity measure and, at each node, the algorithm tries to maximize the gain on it. For our regression problem, we consider Mean Squared Error (MSE).

- *Feed-forward Neural Network (FF).* We used a common fully-connected network with 10 hidden layers. It represents one of the simplest neural network architectures.

To feed data into these baseline models, we compute the average of word embeddings for each review. More specifically, given a review r with terms $\{t_0, ..., t_{n-1}\}$, we took the associated word embeddings $\{w_0, ..., w_{n-1}\}$ and computed their average w.

Regressors Effectiveness

Figure 6.8 and Figure 6.9 report MAE and MSE of the compared regressors. First of all, our results confirm that Neural Networks, both using a single Feed-forward layer and using our model, perform better than common ML algorithms, showing a lower error.

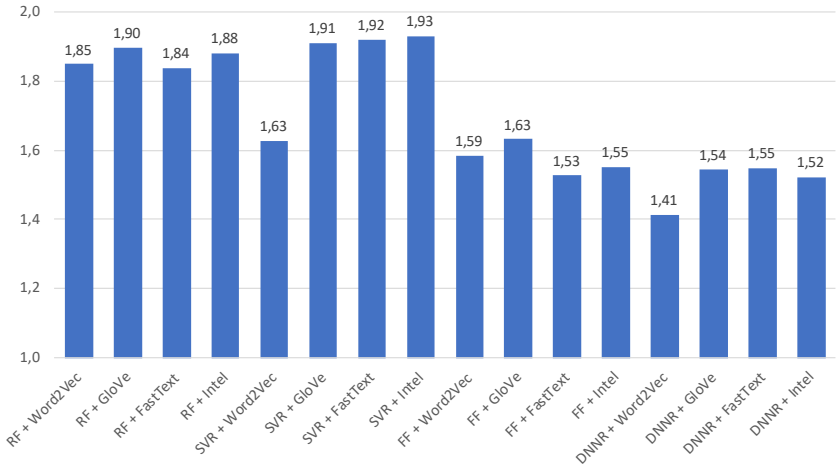


Fig. 6.8: Mean absolute error on sentiment regression.

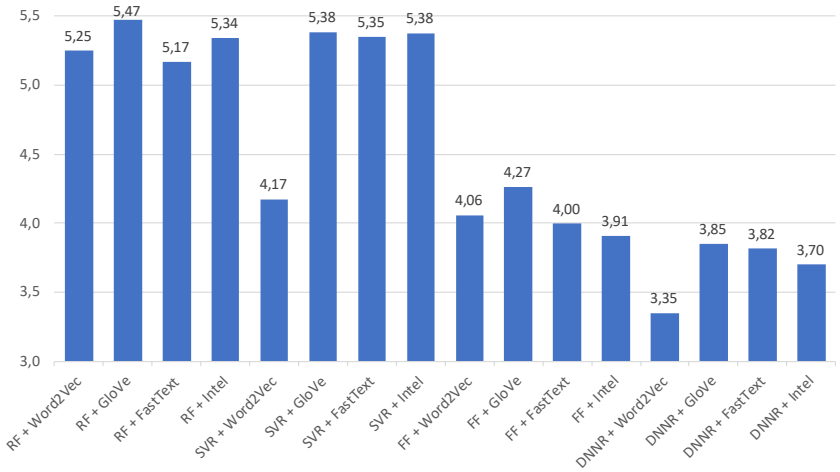


Fig. 6.9: Mean squared error on sentiment regression.

Comparing the Feed-forward baseline with our Deep Neural Network model, there is a little error difference. It is possible to note that the combination *FF + FastText* obtains similar performances of both *DNNR + GloVe* and *DNNR + FastText*. The best performance was obtained by *DNNR + Word2Vec*. Similar considerations also apply when analyzing the MSE. In fact the *DNNR* model gets best performance as well. In contrast with the *MAE*, no baseline obtains performances similar to our model.

Embeddings Effectiveness

This further experiment aims to show how the context-trained Word Embeddings we generated have advantage over reference generic-trained Word Embeddings, when they are fed into our deep neural network as frozen weights of the Embedding Layer. In order to evaluate the effectiveness of our approach, we performed experiments using embeddings of size 300 trained on COCO-SA online course reviews. We compared them against the following reference generic-trained Word Embeddings of size 300:

- The *Word2Vec*[7] Word Embeddings trained on a part of the Google News dataset including 100 billion words with a vocabulary of 3 million words.
- The *GloVe*[8] Word Embeddings trained on a Wikipedia dataset including one billion words with a vocabulary of 400 thousand words.
- The *FastText*[9] Word Embeddings trained on a Wikipedia dataset including four billion words with a vocabulary of 1 million thousand words.

Context-trained *Intel* word embeddings are compared with generic *Word2Vec* word embeddings because i) there are not public generic *Intel* word embeddings, and ii) the *Intel* algorithm is an evolution of the *Word2Vec* algorithm.

Figure 6.10 and Figure 6.11 show that there is no significant difference between context-trained word and generic-trained embeddings when the MAE is used for the comparison. Nevertheless, it is worth underling how the type of embeddings enables to obtain better results in the E-learning domain. Context-trained *Word2Vec* embeddings show the lowest values of MAE compared to other embeddings types. In contrast, when the MSE is considered, context-trained embeddings perform better, as shown in Fig. 6.11. In this case, context-trained embeddings have low values of MSE in almost all cases except for the *GloVe* Word Embeddings. The best performance was obtained by context-trained *Word2Vec* embeddings, proving that i) *Word2Vec* is the best algorithm to learn word representations from our dataset, and ii) context-trained Word Embeddings are able to capture specific patterns in education. This makes possible to adapt our DL model on the E-learning domain and improve results in sentiment prediction.

[7] https://code.google.com/archive/p/word2vec/

[8] https://nlp.stanford.edu/projects/glove/

[9] https://s3-us-west-1.amazonaws.com/fasttext-vectors/wiki.en.vec

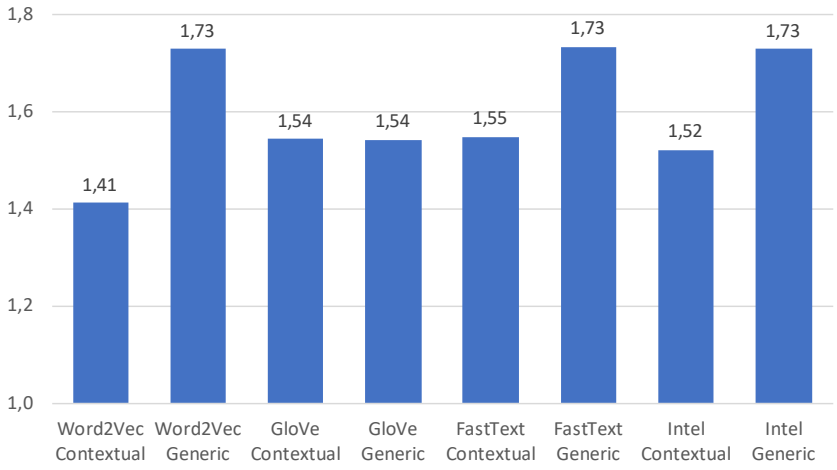


Fig. 6.10: Mean Absolute Error (MAE) for contextual word embeddings.

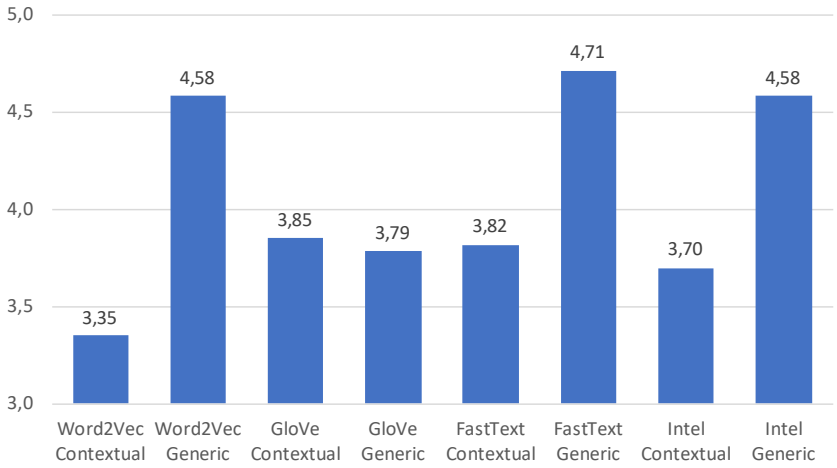


Fig. 6.11: Mean Squared Error (MSE) for contextual word embeddings.

6.6 Findings and Recommendations

This chapter was structured around a case study on SA within the e-learning context. By combining state-of-the-art deep learning methodologies and word embedding text representations, we introduced and described a deep neural network aimed to predict a sentiment score for text reviews left by learners after attending online courses, as a targeted educational context. The chapter guides the readers on how to address this task by providing an overview of the common building blocks of a SA system, from input text representations to neural network components, followed by a recipe which combines them into a model able to outperform common ML baselines.

Based on the obtained results, we can conclude that:

1. Collecting reviews from online learning platforms makes it possible to leverage deep learning models powered by word embeddings for sentiment prediction in education.
2. Simple and advanced deep learning models perform better than common machine learning algorithms, such as SVM and Random Forest, on learners' reviews.
3. Integrating Bidirectional LSTM and Attention Layers enables the neural network to perform more accurate predictions with respect to using only fully-connected layers.
4. The use of word embeddings generated from e-learning resources enables the model to capture more peculiarities from this target context.
5. Education-specific *Word2Vec* word embeddings can better represent the semantics behind e-learning data and help the model to well predict the sentiment score.

Possible future work can involve (i) the analysis of other ways to generate context and sentiment-aware word embeddings, (ii) the exploration of the difference among context/general-trained embeddings, which words their embeddings change significantly and have strong impact in the polarity, and (iii) the investigation on the social effects generated by reviews and their predicted sentiment labels. In order to get insightful knowledge on how people perceive each review, more research taking into account various, potentially related aspects discussed within a single review is needed. Moreover, advanced SA strategies can integrate opinions conveyed by other mediums (e.g., videos, images, audios) in addition to those released through textual reviews.

Chapter 7

Conclusions

In this thesis, we investigated the design, evaluation and application of machine-learning technology aimed to support educators and learners over the instructional pipeline. To this end, we designed and analyzed several data sets and models, ranging from micro-learning video classifiers to educational recommender systems, from learner's biometric verifiers to education-specific sentiment predictors. Natural Language Processing, Computer Vision, and Machine Perception methods have made it possible to automatically interpret human language and behaviour in learning-related data.

7.1 Contribution Summary

The research under this thesis has been proved to provide the following contributions:

- The four data sets we proposed outrun existing educational data sets in terms of scale, completeness, and comprehensiveness. To the best of our knowledge, these collections are among the first ones that enable machine/deep learning in educational platforms.
- The categorization models proposed in Chapter 3 can support users in mapping video-lessons on pre-existing categories. Differently from most of the competitive baselines, they exploit semantics rather than term frequency for accurate predictions.
- The educational recommendation models proposed in Chapter 4 mitigate the effects of biases existing in data while still being effective. These models differ from other existing educational recommenders that are often optimized only over accuracy.
- The multi-biometric identity verification models proposed in Chapter 5 can support educators, when learners must be recognized. With respect to other systems for learners' proctoring, our models target ubiquitous scenarios and transparent verification.
- The sentiment prediction models proposed in Chapter 6 can help monitoring learners' opinions. It differs from other state-of-the-art approaches as it embeds education-specific text representations and deep neural networks that led to accurate predictions across the sentiment scale.

It is worth noticing that several models primarily tested in laboratory experiments have been successfully transferred into a real platform under the "*iLearnTV, Anywhere, Anytime*" project. Nonetheless, the models presented in this thesis contain limitations, and the most relevant ones have been mentioned in the respective chapters.

7.2 Take-home Messages

While each chapter has provided several insights specific to the models involved in a given step of the instructional pipeline, the contributions presented in this thesis allows us to make also broader conclusions that are commonly shared:

1. Collecting and processing large-scale data sets from online learning platforms allows researchers to shape effective educational machine-learning models. In support to this point, this thesis brings to the public four data sets and several use cases.

2. Leveraging word semantics rather than word frequencies in learning-related text analysis makes it possible to develop more accurate and efficient machine learning models. This thesis proved this point via semantics-powered models for micro-learning video classification and sentiment prediction in learners' reviews.

3. Combining multiple data sources at different levels of the prediction pipeline often leads to higher performance of the corresponding educational machine-learning model. Several solutions in this thesis merge diverse data, e.g., concepts and keywords for video classification, physical and behavioral biometrics for identity verification.

4. Even though educational machine learning models usually propagate biases, this phenomenon can be mitigated by regularizing their optimization function, with a minimum loss in effectiveness. The exploratory analysis conducted in this thesis and the resulting algorithms (e.g., recommenders) are just an example of this point.

5. Effective machine learning models can be successfully trained even on data transparently collected during the interaction of users with the platform. This cost-effective property has made it possible to design verification algorithms that identify learners continuously as well as recommendation algorithms that rely only on implicit data.

6. Shaping proper deep learning architectures and training them on the collected large-scale data sets has improved the effectiveness of the resulting models, while keeping them efficient. This win-win situation has made it possible to integrate several of the models proposed in this thesis into a real-world educational platform.

7.3 Future Research Directions

Our research on machine learning in education has produced a variety of methods, but still poses some interesting challenges that require further investigation:

- **Public Data Sets and Models**. Most studies in education have still used rather small data sets which were not made publicly available. This makes it difficult to train and test ML models. As education is an heterogeneous field, a larger collection with more diverse data sets is needed. Furthermore, sharing code and pre-trained models should become a common practice. Code for existing models is not often made public, while, for others, it is needed to re-implement models from scratch. More data sets and models should be shared.
- **Transferability across Contexts**. Existing models tend to target large-scale online learning environments and appear to be sensitive to such a context targeted by the underlying data. This has favored the creation of ML models that, after being trained with data from a given context (e.g., online teaching), do not generalize well in other contexts (e.g., face-to-face teaching). With the availability of new public data sets and pre-trained models, it will become easier to plug them into a task different from the original one. Researchers could fine-tune pre-trained models with few contextual data.
- **Robustness in Uncontrolled Environments**. Devising ML models that can properly operate in real-world environments is another open issue. Most models either implicitly or explicitly impose some constraints and suffer from low efficiency while operating. Such limits should be reduced in order to seamlessly make decisions, e.g., during the interaction between an individual and a proctoring system. This requires innovative, robust, and efficient algorithms.
- **Robustness against Spoofing Attacks**. Synthetically generated data or maliciously modified data are used to circumvent ML models. For instance, this could affect educational recommender systems fed with artificial data that boost popularity of courses irrespective of their quality as well as identity verification systems that are fooled by impostor biometric data. The challenge is to develop counter-measures that are applicable also to unseen or unknown attacks. Investigating how the deployment of ML models might help to face this challenge requires further research.
- **Explainability and Interpretability**. ML models embedded in educational artificial intelligence systems might suffer from low explainability (e.g., on the reason those particular recommendations are provided to the user). Hence, it becomes important to understand how we can explain the output of a model and how it varies based on changes in input or algorithmic parameters. Moreover, it requires attention how internal mechanics can be explained in human terms.
- **Fairness, Transparency and Accountability**. With the advent of AI-based education, addressing bias within ML models will be a core priority due to several rea-

sons. Some biases can be introduced by using training data which is not an accurate sample of the population (e.g., more men than women) or is influenced by sociocultural stereotypes (e.g., popularity). Moreover, advanced properties built on top of biases, e.g., fairness, transparency, and accountability, require attention. Future research should control these properties in the developed models.

Based on the rapid evolution of machine learning in education over these years, we are hopeful that this technology will positively impact real-world education more and more. In addition to the use cases presented in this thesis, other possible applications of our studies include but are not limited to: classification of other resources (e.g., pptx, books), feeds for learning material suitable for life-long learning, evaluation of the teaching activities, recognition of learners and teachers for access control to both restricted areas (e.g., classrooms) and to computers and mobile devices.

Methodological and technological shifts often go hand-in-hand and, as online education become increasingly ubiquitous, an increasing level of data on learners and their context can be automatically and continuously obtained. This enhances the richness of the collected contextual data, removes the requirement for active and passive sensing to take place on a single device, and can further optimise the models that support stakeholders along the instructional pipeline. We argue that these circumstances have the opportunity to further strengthen the reliability of machine learning in education.

References

[1] Technavio, *Professional Development Market in Europe 2016-2020*. Available at https://www.technavio.com/report/europe-k12-and-higher-education-professional-development-market-europe-2016-2020 (2016).

[2] F. J. García-Peñalvo, Á. Hernández-García, M. Á. Conde, Á. Fidalgo-Blanco, M. L. Sein-Echaluce, M. Alier-Forment, F. Llorens-Largo, and S. Iglesias-Pradas, "Enhancing education for the knowledge society era with learning ecosystems," in *Open Source Solutions for Knowledge Management and Technological Ecosystems*, pp. 1–24, IGI Global, 2017.

[3] D. Shah, *ClassCentral Reports: By The Numbers - MOOCs in 2018*. Available at https://www.classcentral.com/report/mooc-stats-2018/ (2018).

[4] "Google classroom." Available at https://edu.google.com/products/classroom/ (2019).

[5] "Coursera." Available at https://www.coursera.org (2019).

[6] "Udemy." Available at https://www.udemy.com/ (2019).

[7] M. Schophuizen, K. Kreijns, S. Stoyanov, and M. Kalz, "Eliciting the challenges and opportunities organizations face when delivering open online education: A group-concept mapping study," *The Internet and Higher Education*, vol. 36, pp. 1–12, 2018.

[8] I. Tuomi, *The Impact of Artificial Intelligence on Learning, Teaching, and Education*. Available at https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/impact-artificial-intelligence-learning-teaching-and-education (2018).

[9] EuropeanCommission, *Artificial Intelligence*. Available at https://ec.europa.eu/digital-single-market/en/artificial-intelligence (2018).

[10] ThomsonReuters, *Artificial Intelligence in Education Market 2019*. Available at https://www.reuters.com/brandfeatures/venture-capital/article?id=93893 (2019).

[11] EuropeanCommission, *Ethics for trustworthy AI*. Available at https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai (2019).

[12] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[13] S. Palvia, P. Aeron, P. Gupta, D. Mahapatra, R. Parida, R. Rosner, and S. Sindhi, "Online education: Worldwide status, challenges, trends, and implications," 2018.

[14] G. Fenu, M. Marras, S. Barra, F. Giorgini, D. Zucchetti, and F. Chesi, "Ilearntv: Un ecosistema di conoscenza condivisa e produzione collaborativa per innovare la formazione," *Mondo Digitale*, p. 2, 2019.

[15] D. Dessì, G. Fenu, M. Marras, and D. R. Recupero, "Leveraging cognitive computing for multi-class classification of e-learning videos," in *European Semantic Web Conference*, pp. 21–25, Springer, 2017.

[16] D. Dessì, G. Fenu, M. Marras, and D. R. Recupero, "Bridging learning analytics and cognitive computing for big data classification in micro-learning video collections," *Computers in Human Behavior*, vol. 92, pp. 468–477, 2019.

[17] L. Boratto, G. Fenu, and M. Marras, "The effect of algorithmic bias on recommender systems for massive open online courses," in *European Conference on Information Retrieval*, pp. 457–472, Springer, 2019.

[18] D. Dessì, G. Fenu, M. Marras, and D. Reforgiato, "Coco: Semantic-enriched collection of online courses at scale with experimental use cases," in *World Conference on Information Systems and Technologies*, pp. 1386–1396, Springer, 2018.

[19] G. Fenu, M. Marras, and L. Boratto, "A multi-biometric system for continuous student authentication in e-learning platforms," *Pattern Recognition Letters*, vol. 113, pp. 83–92, 2018.

[20] G. Fenu and M. Marras, "Leveraging continuous multi-modal authentication for access control in mobile cloud environments," in *International Conference on Image Analysis and Processing*, pp. 331–342, Springer, 2017.

[21] S. Barra, M. Marras, and G. Fenu, "Continuous authentication on smartphone by means of periocular and virtual keystroke," in *International Conference on Network and System Security*, pp. 212–220, Springer, 2018.

[22] G. Fenu and M. Marras, "Controlling user access to cloud-connected mobile applications by means of biometrics," *IEEE Cloud Computing*, vol. 5, no. 4, pp. 47–57, 2018.

[23] M. Marras, P. A. Marín-Reyes, J. Lorenzo-Navarro, M. Castrillón-Santana, and G. Fenu, "AveRobot: An audio-visual dataset for people re-identification and verification in human-robot interaction," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, pp. 255–265, 2019.

[24] M. Marras, P. A. Marín-Reyes, J. Lorenzo-Navarro, M. Castrillón-Santana, and G. Fenu, "Deep multi-biometric fusion for audio-visual user re-identification and verification," in *International Conference on Pattern Recognition Applications and Methods*, pp. 136–157, Springer, 2019.

[25] M. Marras, P. Korus, N. Memon, and G. Fenu, "Adversarial optimization for dictionary attacks on speaker verification," *Proc. Interspeech 2019*, pp. 2913–2917, 2019.

[26] D. Dessì, M. Dragoni, G. Fenu, M. Marras, and D. R. Recupero, "Evaluating neural word embeddings created from online course reviews for sentiment analysis," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 2124–2127, ACM, 2019.

[27] D. Dessí, M. Dragoni, G. Fenu, M. Marras, and D. R. Recupero, "Deep learning adaptation with word embeddings for sentiment analysis on online course reviews," in *Deep Learning-Based Approaches for Sentiment Analysis*, pp. 57–83, Springer, 2020.

[28] N. J. Nilsson, *Principles of artificial intelligence*. Morgan Kaufmann, 2014.

[29] P. Smolensky, "Connectionist ai, symbolic ai, and the brain," *Artificial Intelligence Review*, vol. 1, no. 2, pp. 95–109, 1987.

[30] J. G. Carbonell, T. M. Mitchell, and R. S. Michalski, *Machine learning: An artificial intelligence approach*. Springer-Verlag, 1984.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[32] P. Lison, "An introduction to machine learning," *Language Technology Group (LTG), 1*, vol. 35, 2015.

[33] F. Chollet, "Deep learning with python (2017)."

[34] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[35] V. Orazio, S. T. Landis, G. Palmer, and P. Schrodt, "Separating the wheat from the chaff: Applications of automated document classification using support vector machines.," *Political analysis*, vol. 22, no. 2, 2014.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[38] H. G. Hosseini, D. Luo, and K. J. Reynolds, "The comparison of different feed forward neural network architectures for ecg signal diagnosis," *Medical engineering & physics*, vol. 28, no. 4, pp. 372–378, 2006.

[39] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[40] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[41] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, IEEE, 2013.

[42] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[43] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, IEEE, 2015.

[46] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78, 2014.

[47] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21–29, 2016.

[48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489, 2016.

[49] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification.," in *ACL (1)*, pp. 1014–1023, 2015.

[50] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, pp. 2177–2185, 2014.

[51] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *arXiv preprint arXiv:1511.06807*, 2015.

[52] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, pp. 1050–1059, 2016.

[53] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv preprint arXiv:1702.05659*, 2017.

[54] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *Technical Report.*, 2017.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[57] C. D. Kloos, P. Jermann, M. Pérez-Sanagustín, D. T. Seaton, and S. White, *Digital Education: Out to the World and Back to the Campus.* Springer, 2017.

[58] D. Coakley, R. Garvey, and Í. O'Neill, "Micro-learning—adopting digital pedagogies to facilitate technology-enhanced teaching and learning for cpd," in *Empowering 21st Century Learners Through Holistic and Enterprising Learning*, pp. 237–242, Springer, 2017.

[59] G. Sun, T. Cui, W. Guo, G. Beydoun, D. Xu, and J. Shen, "Micro learning adaptation in mooc: A software as a service and a personalized learner model," in *International Conference on Web-Based Learning*, pp. 174–184, Springer, 2015.

[60] M. Leach and S. M. Hadi, "Supporting, categorising and visualising diverse learner behaviour on moocs with modular design and micro-learning," *Journal of Computing in Higher Education*, vol. 29, no. 1, pp. 147–159, 2017.

[61] M.-C. Valiente, M.-A. Sicilia, E. Garcia-Barriocanal, and E. Rajabi, "Adopting the metadata approach to improve the search and analysis of educational resources for online learning," *Computers in Human Behavior*, vol. 51, pp. 1134–1141, 2015.

[62] S. Basu, Y. Yu, and R. Zimmermann, "Fuzzy clustering of lecture videos based on topic modeling," in *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pp. 1–6, IEEE, 2016.

[63] P. Ristoski and H. Paulheim, "Semantic web in data mining and knowledge discovery: A comprehensive survey," *Web semantics: science, services and agents on the World Wide Web*, vol. 36, pp. 1–22, 2016.

[64] Y. Song and D. Roth, "On dataless hierarchical text classification.," in *AAAI*, vol. 7, 2014.

[65] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1937–1946, 2014.

[66] Z. Ren, M.-H. Peetz, S. Liang, W. Van Dolen, and M. De Rijke, "Hierarchical multi-label classification of social text streams," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 213–222, ACM, 2014.

[67] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.

[68] C. Xing, D. Wang, X. Zhang, and C. Liu, "Document classification with distributions of word vectors," in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pp. 1–5, IEEE, 2014.

[69] J. R. Quinlan, *C4. 5: Programs for machine learning.* Elsevier, 2014.

[70] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[71] S. P. Algur and P. Bhat, "Web video mining: Metadata predictive analysis using classification techniques," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 8, no. 2, p. 69, 2016.

[72] N. Gkalelis and V. Mezaris, "Video event detection using generalized subclass discriminant analysis and linear support vector machines," in *Proceedings of international conference on multimedia retrieval*, p. 25, ACM, 2014.

[73] L. Yang and X. Wang, "Online appearance manifold learning for video classification and clustering," in *International Conference on Computational Science and Its Applications*, pp. 551–561, Springer, 2016.

[74] I. Feki, A. B. Ammar, and A. M. Alimi, "Automatic environmental sound concepts discovery for video retrieval," *International Journal of Multimedia Information Retrieval*, vol. 5, no. 2, pp. 105–115, 2016.

[75] J. G. Ellis, B. Jou, and S.-F. Chang, "Why we watch the news: A dataset for exploring sentiment in broadcast video news," in *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 104–111, ACM, 2014.

[76] J. Kim, P. J. Guo, C. J. Cai, S.-W. D. Li, K. Z. Gajos, and R. C. Miller, "Data-driven interaction techniques for improving navigation of educational videos," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pp. 563–572, ACM, 2014.

[77] H. Chen, M. Cooper, D. Joshi, and B. Girod, "Multi-modal language models for lecture video retrieval," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1081–1084, ACM, 2014.

[78] S. Wang and Q. Ji, "Video affective content analysis: a survey of state-of-the-art methods," *IEEE Transactions on Affective Computing*, vol. 6, no. 4, pp. 410–430, 2015.

[79] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann, "Bridging the ultimate semantic gap: A semantic search engine for internet videos," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 27–34, ACM, 2015.

[80] S. Oh, S. McCloskey, I. Kim, A. Vahdat, K. J. Cannons, H. Hajimirsadeghi, G. Mori, A. A. Perera, M. Pandey, and J. J. Corso, "Multimedia event detection with multimodal feature fusion and temporal concept localization," *Machine Vision and Applications*, vol. 25, no. 1, pp. 49–69, 2014.

[81] J. Dalton, J. Allan, and P. Mirajkar, "Zero-shot video retrieval using content and concepts," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 1857–1860, ACM, 2013.

[82] Z. Chen, J. Cao, Y. Song, Y. Zhang, and J. Li, "Web video categorization based on wikipedia categories and content-duplicated open resources," in *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, (New York, NY, USA), pp. 1107–1110, ACM, 2010.

[83] A. S. Imran, L. Rahadianti, F. A. Cheikh, and S. Y. Yayilgan, "Semantic tags for lecture videos," in *2012 IEEE Sixth International Conference on Semantic Computing*, pp. 117–120, IEEE, 2012.

[84] A. S. Imran, S. Chanda, F. A. Cheikh, K. Franke, and U. Pal, "Cursive handwritten segmentation and recognition for instructional videos," in *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, pp. 155–160, IEEE, 2012.

[85] N. Sharma, P. Shivakumara, U. Pal, M. Blumenstein, and C. L. Tan, "A new method for arbitrarily-oriented text detection in video," in *2012 10th IAPR International Workshop on Document Analysis Systems*, pp. 74–78, IEEE, 2012.

[86] N. Sharma, P. Shivakumara, U. Pal, M. Blumenstein, and C. L. Tan, "Piece-wise linearity based method for text frame classification in video," *Pattern Recognition*, vol. 48, no. 3, pp. 862–881, 2015.

[87] S. Abdelali *et al.*, "Education data mining: mining moocs videos using metadata based approach," in *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp. 531–534, IEEE, 2016.

[88] E. H. Othman, G. Abderrahim, and E. B. Jaber, "Mining moocs videos metadata using classification techniques," in *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, p. 94, ACM, 2017.

[89] Z. Chen, B. Feng, H. Xie, R. Zheng, and B. Xu, "Video to Article Hyperlinking by Multiple Tag Property Exploration," in *MultiMedia Modeling*, pp. 62–73, 2014.

[90] S. Basu, Y. Yu, and R. Zimmermann, "Fuzzy clustering of lecture videos based on topic modeling," in *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pp. 1–6, IEEE, 2016.

[91] D. Marković, J. Popat, F. Antonacci, A. Sarti, and T. K. Kumar, "An informed separation algorithm based on sound field mapping for speech recognition systems," in *Acoustic Signal Enhancement (IWAENC), 2016 IEEE International Workshop on*, pp. 1–5, IEEE, 2016.

[92] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[93] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: introduction and challenges," in *Recommender Systems Handbook*, pp. 1–34, Springer, 2015.

[94] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič, *Group Recommender Systems: An Introduction.* Springer, 2018.

[95] S. Hajian, F. Bonchi, and C. Castillo, "Algorithmic bias: From discrimination discovery to fairness-aware data mining," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2125–2126, ACM, 2016.

[96] J. Wasilewski and N. Hurley, "Are you reaching your audience?: Exploring item exposure over consumer segments in recommender systems," in *Proc. of the 26th Conference on User Modeling, Adaptation and Personalization*, pp. 213–217, ACM, 2018.

[97] D. Jannach, I. Kamehkhosh, and G. Bonnin, "Biases in automated music playlist generation: A comparison of next-track recommending techniques," in *Proc. of the 2016 Conference on User Modeling Adaptation and Personalization*, pp. 281–285, ACM, 2016.

[98] K. Nagatani and M. Sato, "Accurate and diverse recommendation based on users' tendencies toward temporal item popularity," 2017.

[99] F. Guo and D. B. Dunson, "Uncovering systematic bias in ratings across categories: A bayesian approach," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 317–320, ACM, 2015.

[100] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver, "Exploring author gender in book rating and recommendation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 242–250, ACM, 2018.

[101] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. of the fourth ACM conference on Recommender systems*, pp. 39–46, ACM, 2010.

[102] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling popularity bias in learning-to-rank recommendation," in *Proc. of the Eleventh ACM Conference on Recommender Systems*, pp. 42–46, ACM, 2017.

[103] P. Cremonesi, F. Garzotto, and R. Turrin, "User-centric vs. system-centric evaluation of recommender systems," in *IFIP Conference on Human-Computer Interaction*, pp. 334–351, Springer, 2013.

[104] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, and L. C. Jain, "Recommender systems in e-learning environments," in *E-Learning Systems*, pp. 51–75, Springer, 2017.

[105] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization," *Computers in Human Behavior*, vol. 58, pp. 119–129, 2016.

[106] X. Amatriain, "Big & personal: data and models behind netflix recommendations," in *Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pp. 1–6, ACM, 2013.

[107] T. Berka, W. Behrendt, E. Gams, and S. Reich, "A trail based internet-domain recommender system using artificial neural networks," in *Int. Conf. on Adaptive Hypermedia and Adaptive Web Based Systems*, 2002.

[108] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[109] W. Cheng, J. Hühn, and E. Hüllermeier, "Decision tree and instance-based learning for label ranking," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 161–168, ACM, 2009.

[110] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[111] N. Golbandi, Y. Koren, and R. Lempel, "Adaptive bootstrapping of recommender systems using decision trees," in *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 595–604, ACM, 2011.

[112] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, no. 1, p. 117, 2008.

[113] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert Systems with Applications*, vol. 23, no. 3, pp. 329–342, 2002.

[114] M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. McGrath, "Racofi: A rule-applying collaborative filtering system," 2003.

[115] M. Brand, "Fast online svd revisions for lightweight recommender systems," in *Proceedings of the 2003 SIAM International Conference on Data Mining*, pp. 37–46, SIAM, 2003.

[116] X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver, "The wisdom of the few: a collaborative filtering approach based on expert opinions from the web," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 532–539, ACM, 2009.

[117] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple bayesian classifier," in *Pacific Rim International conference on artificial intelligence*, pp. 679–689, Springer, 2000.

[118] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proc. of the 26th International Conference on World Wide Web*, pp. 173–182, International World Wide Web Conferences Steering Committee, 2017.

[119] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proc. of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, AUAI Press, 2009.

[120] W. Niu, J. Caverlee, and H. Lu, "Neural personalized ranking for image recommendation," in *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 423–431, ACM, 2018.

[121] Y. Shi, M. Larson, and A. Hanjalic, "List-wise learning to rank with matrix factorization for collaborative filtering," in *Proc. of the fourth ACM Conference on Recommender systems*, pp. 269–272, ACM, 2010.

[122] S. Huang, S. Wang, T. Liu, J. Ma, Z. Chen, and J. Veijalainen, "Listwise collaborative filtering," in *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 343–352, ACM, 2015.

[123] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver, "Rate it again: increasing recommendation accuracy by user re-rating," in *Proceedings of the Third ACM Conference on Recommender Systems*, pp. 173–180, ACM, 2009.

[124] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[125] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge & Data Engineering*, no. 6, pp. 734–749, 2005.

[126] A. Ahmed and E. P. Xing, "Scalable dynamic nonparametric bayesian models of content and users," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[127] K. Masters, "A brief guide to understanding moocs," *The Internet Journal of Medical Education*, vol. 1, no. 2, p. 2, 2011.

[128] W. Greller and H. Drachsler, "Translating learning into numbers: A generic framework for learning analytics," pp. 42–57, 2012.

[129] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, "Panorama of recommender systems to support learning," in *Recommender Systems Handbook*, pp. 421–451, Springer, 2015.

[130] S. B. Aher and L. Lobo, "Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data," *Knowledge-Based Systems*, vol. 51, pp. 1–14, 2013.

[131] T. Tang and G. McCalla, "Smart recommendation for an evolving e-learning system: Architecture and experiment," *International Journal on E-learning*, vol. 4, no. 1, pp. 105–129, 2005.

[132] S. Fazeli, B. Loni, H. Drachsler, and P. Sloep, "Which recommender system can best fit social learning platforms?," in *European Conference on Technology Enhanced Learning*, pp. 84–97, Springer, 2014.

[133] P. Karampiperis, A. Koukourikos, and G. Stoitsis, "Collaborative filtering recommendation of educational content in social environments utilizing sentiment analysis techniques," in *Recommender Systems for Technology Enhanced Learning*, pp. 3–23, Springer, 2014.

[134] M. Bieliková, M. Šimko, M. Barla, J. Tvarožek, M. Labaj, R. Móro, I. Srba, and J. Ševcech, "Alef: From application to platform for adaptive collaborative learning," in *Recommender systems for technology enhanced learning*, pp. 195–225, Springer, 2014.

[135] O. C. Santos, M. Saneiro, S. Salmeron-Majadas, and J. G. Boticario, "A methodological approach to eliciting affective educational recommendations," in *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pp. 529–533, IEEE, 2014.

[136] A. Fernández, M. Erdt, I. Dackiewicz, and C. Rensing, "Recommendations from heterogeneous sources in a technology enhanced learning ecosystem," in *Recommender Systems for Technology Enhanced Learning*, pp. 251–265, Springer, 2014.

[137] S. Nowakowski, I. Ognjanović, M. Grandbastien, J. Jovanovic, and R. Šendelj, "Two recommending strategies to enhance online presence in personal learning environments," in *Recommender Systems for Technology Enhanced Learning*, pp. 227–249, Springer, 2014.

[138] A. Kaklauskas, E. K. Zavadskas, M. Seniut, V. Stankevic, J. Raistenskis, C. Simkevičius, T. Stankevic, A. Matuliauskaite, L. Bartkiene, L. Zemeckyte, *et al.*, "Recommender system to analyze student's academic performance," *Expert Systems with Applications*, vol. 40, no. 15, pp. 6150–6165, 2013.

[139] O. C. Santos, M. Saneiro, J. G. Boticario, and M. C. Rodriguez-Sanchez, "Toward interactive context-aware affective educational recommendations in computer-assisted language learning," *New Review of Hypermedia and Multimedia*, vol. 22, no. 1-2, pp. 27–57, 2016.

[140] H. Drachsler, H. Hummel, B. Van den Berg, J. Eshuis, W. Waterink, R. Nadolski, A. Berlanga, N. Boers, and R. Koper, "Effects of the isis recommender system for navigation support in self-organised learning networks," *Journal of Educational Technology & Society*, vol. 12, no. 3, pp. 115–126, 2009.

[141] K. I. Ghauth and N. A. Abdullah, "The effect of incorporating good learners' ratings in e-learning content-based recommender system," *Journal of Educational Technology & Society*, vol. 14, no. 2, pp. 248–257, 2011.

[142] X. Jing and J. Tang, "Guess you like: course recommendation in moocs," in *Proceedings of the International Conference on Web Intelligence*, pp. 783–789, ACM, 2017.

[143] "Course talk." Available at https://www.coursetalk.com (2019).

[144] "Class central." Accessed: 2017-11-20.

[145] D. Nikolov, M. Lalmas, A. Flammini, and F. Menczer, "Quantifying biases in online information exposure," *Journal of the Association for Information Science and Technology*, vol. 70, no. 3, pp. 218–229, 2019.

[146] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, "What recommenders recommend: an analysis of recommendation biases and possible countermeasures," *User Modeling and User-Adapted Interaction*, vol. 25, no. 5, pp. 427–491, 2015.

[147] Y. Park and A. Tuzhilin, "The long tail of recommender systems and how to leverage it," in *Proc. of ACM Conference on Recommender Systems*, pp. 11–18, ACM, 2008.

[148] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Missing data modeling with user activity and item popularity in recommendation," in *Asia Information Retrieval Symposium*, pp. 113–125, Springer, 2018.

[149] J. Oh, S. Park, H. Yu, M. Song, and S. Park, "Novel recommendation based on personal popularity tendency," in *2011 IEEE 11th International Conference on Data Mining*, pp. 507–516, IEEE, 2011.

[150] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Correcting popularity bias by enhancing recommendation neutrality.," in *RecSys Posters*, 2014.

[151] L. Hou, X. Pan, and K. Liu, "Balancing popularity bias of object similarities for personalised recommendation," *European Physical Journal*, vol. 91, no. 3, p. 47, 2018.

[152] H. Abdollahpouri, R. Burke, and B. Mobasher, "Popularity-aware item weighting for long-tail recommendation," *arXiv preprint arXiv:1802.05382*, 2018.

[153] H. Abdollahpouri, R. Burke, and B. Mobasher, "Managing popularity bias in recommender systems with personalized re-ranking," *arXiv preprint arXiv:1901.07555*, 2019.

[154] Z. Zhu, X. Hu, and J. Caverlee, "Fairness-aware tensor-based recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, pp. 1153–1162, ACM, 2018.

[155] B. Rastegarpanah, K. P. Gummadi, and M. Crovella, "Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 11-15, 2019*, pp. 231–239, ACM, 2019.

[156] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 2921–2930, 2017.

[157] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow, "Fairness in recommendation ranking through pairwise comparisons," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, pp. 2212–2220, ACM, 2019.

[158] B. Edizel, F. Bonchi, S. Hajian, A. Panisson, and T. Tassa, "Fairecsys: mitigating algorithmic bias in recommender systems," *International Journal of Data Science and Analytics*, pp. 1–17, 2019.

[159] W. Liu and R. Burke, "Personalizing fairness-aware re-ranking," *CoRR*, vol. abs/1809.02921, 2018.

[160] N. Sonboli and R. Burke, "Localized fairness in recommender systems," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, UMAP*, pp. 295–300, ACM, 2019.

[161] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz, "Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM*, pp. 2243–2251, ACM, 2018.

[162] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[163] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th International Conference on World Wide Web*, pp. 22–32, ACM, 2005.

[164] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 17–24, ACM, 2007.

[165] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," 2011.

[166] M. Harper and J. Konstan, "The movielens datasets: History and context," *ACM transactions on interactive intelligent systems (TIIS)*, vol. 5, no. 4, p. 19, 2016.

[167] V. Estivill-Castro, C. Limongelli, M. Lombardi, and A. Marani, "Dajee: A dataset of joint educational entities for information retrieval in technology enhanced learning," in *Proceedings of the 39th International ACM SIGIR*, pp. 681–684, ACM, 2016.

[168] N. Pappas and A. Popescu-Belis, "Combining content with user preferences for ted lecture recommendation," in *Proceedings of the 11th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 47–52, IEEE, 2013.

[169] "Merlot." Available at https://www.merlot.org/merlot/index.htm (2019).

[170] "Mace." Available at http://ea-tel.eu/tel-research/mace/ (2019).

[171] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems.," in *UMAP Workshops*, vol. 4, 2015.

[172] C. Cechinel, M.-Á. Sicilia, S. Sánchez-Alonso, and E. GarcíA-Barriocanal, "Evaluating collaborative filtering recommendations inside large learning object repositories," *Information Processing & Management*, vol. 49, no. 1, pp. 34–50, 2013.

[173] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, ACM, 2008.

[174] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets.," in *ICDM*, vol. 8, pp. 263–272, Citeseer, 2008.

[175] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 273–282, ACM, 2014.

[176] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.

[177] T. Griffiths, "Gibbs sampling in the generative model of latent dirichlet allocation," 2002.

[178] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, and M. S. Pera, "All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness," in *Conference on Fairness, Accountability and Transparency*, pp. 172–186, 2018.

[179] G. Adomavicius, J. Bockstedt, S. Curley, and J. Zhang, "De-biasing user preference ratings in recommender systems," in *Joint Workshop on Interfaces and Human Decision Making in Recommender Systems*, p. 2, 2014.

[180] S. Kopeinik, D. Kowald, and E. Lex, "Which algorithms suit which learning environments? a comparative study of recommender systems in tel," in *European Conference on Technology Enhanced Learning*, pp. 124–138, Springer, 2016.

[181] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, R. Vuorikari, and E. Duval, "Dataset-driven research for improving recommender systems for learning," in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pp. 44–53, ACM, 2011.

[182] N. Manouselis, R. Vuorikari, and F. Van Assche, "Collaborative recommendation of e-learning resources: an experimental investigation," *Journal of Computer Assisted Learning*, vol. 26, no. 4, pp. 227–242, 2010.

[183] A. G. and G. S., "Evaluating recommender systems," in *Recommender Systems Handbook* (F. Ricci, L. Rokach, and B. Shapira, eds.), pp. 265–308, Springer, 2015.

[184] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 335–344, ACM, 2017.

[185] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems.," in *IJCAI*, pp. 3203–3209, 2017.

[186] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," *arXiv preprint arXiv:1708.04617*, 2017.

[187] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[188] A. Bellogín, P. Castells, and I. Cantador, "Statistical biases in information retrieval metrics for recommender systems," *Information Retrieval Journal*, vol. 20, no. 6, pp. 606–634, 2017.

[189] A. Vegendla and G. Sindre, "Mitigation of cheating in online exams: Strengths and limitations of biometric authentication," in *Biometric Authentication in Online Learning Environments*, pp. 47–68, IGI Global, 2019.

[190] M. Wang and W. Deng, "Deep face recognition: A survey," *arXiv preprint arXiv:1804.06655*, 2018.

[191] D. Peralta, M. Galar, I. Triguero, D. Paternain, S. García, E. Barrenechea, J. M. Benítez, H. Bustince, and F. Herrera, "A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation," *Information Sciences*, vol. 315, pp. 67–87, 2015.

[192] K. W. Bowyer and M. J. Burge, *Handbook of Iris Recognition.* Springer, 2016.

[193] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[194] Y. Zhong and Y. Deng, "A survey on keystroke dynamics biometrics: approaches, advances, and evaluations," *Recent Advances in User Authentication Using Keystroke Dynamics Biometrics*, pp. 1–22, 2015.

[195] D. Dasgupta, A. Roy, and A. Nag, "Continuous authentication," in *Advances in User Authentication*, pp. 235–279, Springer, 2017.

[196] T. Kawamata, T. Ishii, S. Fujimori, and T. Akakura, "Student authentication by updated facial information with weighting coefficient in e-learning," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 551–555, IEEE, 2016.

[197] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi, "Student attendance system in classroom using face recognition technique," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1032–1035, IEEE, 2016.

[198] J. R. Young, R. S. Davies, J. L. Jenkins, and I. Pfleger, "Keystroke dynamics: Establishing keyprints to verify users in online courses," *Computers in the Schools*, vol. 36, no. 1, pp. 48–68, 2019.

[199] A. Morales and J. Fierrez, "Keystroke biometrics for student authentication: A case study," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 337–337, ACM, 2015.

[200] I. Traoré, Y. Nakkabi, S. Saad, B. Sayed, J. D. Ardigo, and P. M. de Faria Quinan, "Ensuring online exam integrity through continuous biometric authentication," in *Information Security Practices*, pp. 73–81, Springer, 2017.

[201] S. Asha and C. Chellappan, "Authentication of e-learners using multimodal biometric technology," in *2008 International Symposium on Biometrics and Security Technologies*, pp. 1–6, IEEE, 2008.

[202] B. A. Kumar and S. S. Chand, "Mobile learning adoption: A systematic review," *Education and Information Technologies*, vol. 24, no. 1, pp. 471–487, 2019.

[203] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[204] E. Schiavone, A. Ceccarelli, and A. Bondavalli, "Continuous biometric verification for non-repudiation of remote services," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, p. 4, ACM, 2017.

[205] M. E. Fathy, V. M. Patel, and R. Chellappa, "Face-based active authentication on mobile devices," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1687–1691, IEEE, 2015.

[206] K. A. Rahman, R. Moormann, D. Dierich, and M. S. Hossain, "Continuous user verification via mouse activities," in *International Conference on Multimedia Communications, Services and Security*, pp. 170–181, Springer, 2015.

[207] M. L. Ali, J. V. Monaco, C. C. Tappert, and M. Qiu, "Keystroke biometric systems for user authentication," *Journal of Signal Processing Systems*, vol. 86, no. 2-3, pp. 175–190, 2017.

[208] P. S. Teh, N. Zhang, A. B. J. Teoh, and K. Chen, "A survey on touch dynamics authentication in mobile devices," *Computers & Security*, vol. 59, pp. 210–235, 2016.

[209] M. De Marsico, C. Galdi, M. Nappi, and D. Riccio, "Firme: Face and iris recognition for mobile engagement," *Image and Vision Computing*, vol. 32, no. 12, pp. 1161–1172, 2014.

[210] M. De Marsico, M. Nappi, D. Riccio, and H. Wechsler, "Mobile iris challenge evaluation (miche)-i, biometric iris dataset and protocols," *Pattern Recognition Letters*, vol. 57, pp. 17–23, 2015.

[211] M. Lastra, J. Carabaño, P. D. Gutiérrez, J. M. Benítez, and F. Herrera, "Fast fingerprint identification using gpus," *Information Sciences*, vol. 301, pp. 195–214, 2015.

[212] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain, "Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data," in *2015 International Conference on Biometrics (ICB)*, pp. 135–142, IEEE, 2015.

[213] G. Kambourakis, D. Damopoulos, D. Papamartzivanos, and E. Pavlidakis, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security and Communication Networks*, vol. 9, no. 6, pp. 542–554, 2016.

[214] D. Buschek, A. De Luca, and F. Alt, "Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1393–1402, ACM, 2015.

[215] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2015.

[216] R. Kumar, V. V. Phoha, and A. Serwadda, "Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, IEEE, 2016.

[217] A. Alzubaidi and J. Kalita, "Authentication of smartphone users using behavioral biometrics," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1998–2026, 2016.

[218] P. Kang and S. Cho, "Keystroke dynamics-based user authentication using long and free text strings from various input devices," *Information Sciences*, vol. 308, pp. 72–93, 2015.

[219] J.-h. Roh, S.-H. Lee, and S. Kim, "Keystroke dynamics for authentication in smartphone," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1155–1159, IEEE, 2016.

[220] V.-D. Stanciu, R. Spolaor, M. Conti, and C. Giuffrida, "On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 105–112, ACM, 2016.

[221] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, "Continuous user authentication using multi-modal biometrics," *Computers & Security*, vol. 53, pp. 234–246, 2015.

[222] M. Smith-Creasey and M. Rajarajan, "A continuous user authentication scheme for mobile devices," in *14th Annual Conference on Privacy, Security and Trust, PST 2016, Auckland, New Zealand, December 12-14, 2016*, pp. 104–113, 2016.

[223] A. Buriro, S. Gupta, and B. Crispo, "Evaluation of motion-based touch-typing biometrics for online banking," in *International Conference of the Biometrics Special Interest Group, BIOSIG 2017, Darmstadt, Germany, September 20-22, 2017*, pp. 219–226, 2017.

[224] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location," *IEEE Systems Journal*, vol. 11, no. 2, pp. 513–521, 2017.

[225] F. Lin, C. Song, Y. Zhuang, W. Xu, C. Li, and K. Ren, "Cardiac scan: A non-contact and continuous heart-based user authentication system," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, MobiCom 2017, Snowbird, UT, USA, October 16 - 20, 2017*, pp. 315–328, 2017.

[226] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2011, Shanghai, China, October 10-12, 2011*, pp. 141–148, 2011.

[227] Y. Wang, J. Shen, S. Petridis, and M. Pantic, "A real-time and unsupervised face re-identification system for human-robot interaction," *Pattern Recognition Letters*, 2018.

[228] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[229] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, vol. 7, 2017.

[230] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.

[231] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[232] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[233] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European Conference on Computer Vision*, pp. 499–515, Springer, 2016.

[234] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5089–5097, 2018.

[235] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[236] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4052–4056, IEEE, 2014.

[237] Y.-h. Chen, I. Lopez-Moreno, T. N. Sainath, M. Visontai, R. Alvarez, and C. Parada, "Locally-connected and convolutional neural networks for small footprint speaker recognition," in *Proc. Interspeech 2015*, 2015.

[238] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, IEEE, 2018.

[239] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[240] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

[241] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, IEEE, 2016.

[242] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[243] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[244] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, "I-vector based speaker recognition on short utterances," in *Proc. Interspeech 2011*, pp. 2341–2344, 2011.

[245] A. Abozaid, A. Haggag, H. Kasban, and M. Eltokhy, "Multimodal biometric scheme for human authentication technique based on voice and face recognition fusion," *Multimedia Tools and Applications*, pp. 1–17, 2018.

[246] Q. Memon, Z. AlKassim, E. AlHassan, M. Omer, and M. Alsiddig, "Audio-visual biometric authentication for secured access into personal devices," in *Proceedings of the 6th International Conference on Bioinformatics and Biomedical Science*, pp. 85–89, ACM, 2017.

[247] G. Sell, K. Duh, D. Snyder, D. Etter, and D. Garcia-Romero, "Audio-visual person recognition in multimedia data from the iarpa janus program," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3031–3035, IEEE, 2018.

[248] L. Huang, C. Yu, and X. Cao, "Bimodal biometric person recognition by score fusion," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1093–1097, IEEE, 2018.

[249] A. Chowdhury, Y. Atoum, L. Tran, X. Liu, and A. Ross, "Msu-avis dataset: Fusing face and voice modalities for biometric recognition in indoor surveillance videos," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3567–3573, IEEE, 2018.

[250] M. Singh, R. Singh, and A. Ross, "A comprehensive overview of biometric fusion," *Information Fusion*, vol. 52, pp. 187–205, 2019.

[251] J. Geng, X. Liu, and Y.-m. Cheung, "Audio-visual speaker recognition via multi-modal correlated neural networks," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, pp. 123–128, IEEE, 2016.

[252] A. Brutti and A. Cavallaro, "Online cross-modal adaptation for audio–visual person identification with wearable cameras," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 40–51, 2016.

[253] A. Cavallaro and A. Brutti, "Audio-visual learning for body-worn cameras," in *Multimodal Behavior Analysis in the Wild*, pp. 103–119, Elsevier, 2019.

[254] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson, "3d convolutional neural networks for cross audio-visual matching recognition," *IEEE Access*, vol. 5, pp. 22081–22091, 2017.

[255] S. Shon, T.-H. Oh, and J. Glass, "Noise-tolerant audio-visual online person verification using an attention-based neural network fusion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3995–3999, IEEE, 2019.

[256] J. Zhang, K. Richmond, and R. B. Fisher, "Dual-modality talking-metrics: 3d visual-audio integrated behaviometric cues from speakers," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3144–3149, IEEE, 2018.

[257] X. Liu, J. Geng, H. Ling, and Y.-m. Cheung, "Attention guided deep audio-face fusion for efficient speaker naming," *Pattern Recognition*, vol. 88, pp. 557–568, 2019.

[258] S. Borale, M. P. G. Chaudhari, M. V. B. Patil, M. A. D. Shingne, and G. Dhoot, "Fingerprint based attendance management system with sms alert to parents," in *Int J Res Adv Technol (IJRAT)(E-ISSN: 2321-9637) Special issue national conference convergence*, pp. 06–07, 2016.

[259] P. Wagh, R. Thakare, J. Chaudhari, and S. Patil, "Attendance system based on face recognition using eigen face and pca algorithms," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 303–308, IEEE, 2015.

[260] E. Ghaleb, M. Popa, E. Hortal, S. Asteriadis, and G. Weiss, "Towards affect recognition through interactions with learning materials," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 372–379, IEEE, 2018.

[261] J. Khalfallah and J. B. H. Slama, "Facial expression recognition for intelligent tutoring systems in remote laboratories platform," *Procedia Computer Science*, vol. 73, pp. 274–281, 2015.

[262] M. Correa, G. Hermosilla, R. Verschae, and J. Ruiz-del Solar, "Human detection and identification by robots using thermal and visual information in domestic environments," *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1-2, pp. 223–243, 2012.

[263] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. Van Gool, "One-shot person re-identification with a consumer depth camera," in *Person Re-Identification*, pp. 161–181, Springer, 2014.

[264] S. Ouellet, F. Grondin, F. Leconte, and F. Michaud, "Multimodal biometric identification system for mobile robots combining human metrology to face recognition and speaker identification," in *2014 RO-MAN: The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 323–328, IEEE, 2014.

[265] H. Liu, L. Hu, and L. Ma, "Online RGB-D person re-identification based on metric model update," *CAAI Transactions on Intelligence Technology*, vol. 2, no. 1, pp. 48–55, 2017.

[266] B. Irfan, N. Lyubova, M. G. Ortiz, and T. Belpaeme, "Multi-modal open-set person identification in hri," in *Proceedings of the International Conference on Human-Robot Interaction Social Robots in the Wild*, 2018.

[267] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, pp. 1499–1503, Oct 2016.

[268] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa, "Active user authentication for smartphones: A challenge data set and benchmark results," in *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, IEEE, 2016.

[269] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matejka, J. Cernockỳ, N. Poh, J. Kittler, A. Larcher, C. Levy, *et al.*, "Bi-modal person recognition on a mobile phone: using mobile phone data," in *2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 635–640, IEEE, 2012.

[270] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[271] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[272] A. Roy, N. Memon, and A. Ross, "Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2013–2025, 2017.

[273] P. Bontrager, A. Roy, J. Togelius, and N. Memon, "Deep master prints: Generating master-prints for dictionary attacks via latent variable evolution," in *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, IEEE, 2018.

[274] A. Poddar, M. Sahidullah, and G. Saha, "Speaker verification with short utterances: a review of challenges, trends and opportunities," *IET Biometrics*, vol. 7, no. 2, pp. 91–101, 2017.

[275] M. Ivanova, S. Bhattacharjee, S. Marcel, A. Rozeva, and M. Durcheva, "Enhancing trust in eassessment-the tesla system solution," in *Proceedings of 2018 Technology Enhanced Assessment Conference*, 2018.

[276] N. Yager and T. Dunstone, "The biometric menagerie," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 220–230, 2010.

[277] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[278] A. Dridi and D. Reforgiato Recupero, "Leveraging semantics for sentiment polarity detection in social media," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 2045–2055, Aug 2019.

[279] K. Mite-Baidal, C. Delgado-Vera, E. Solís-Avilés, A. H. Espinoza, J. Ortiz-Zambrano, and E. Varela-Tapia, "Sentiment analysis in education domain: A systematic literature review," in *International Conference on Technologies and Innovation*, pp. 285–297, Springer, 2018.

[280] K. L. Cela, M. Á. Sicilia, and S. Sánchez, "Social network analysis in e-learning environments," *Educational Psychology Review*, vol. 27, no. 1, pp. 219–246, 2015.

[281] G. Esparza, A. de Luna, A. O. Zezzatti, A. Hernandez, J. Ponce, M. Álvarez, E. Cossio, and J. de Jesus Nava, "A sentiment analysis model to analyze students reviews of teacher performance using support vector machines," in *Int. Symp. on Distributed Computing and Artificial Intelligence*, pp. 157–164, Springer, 2017.

[282] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, pp. 1188–1196, 2014.

[283] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, pp. 214–224, 2017.

[284] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, 2017.

[285] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, Association for Computational Linguistics, 2011.

[286] M. Atzeni and D. Reforgiato, "Deep learning and sentiment analysis for human-robot interaction," in *Europ. Semantic Web Conference*, pp. 14–18, Springer, 2018.

[287] D. Tang and M. Zhang, "Deep learning in sentiment analysis," in *Deep Learning in Natural Language Processing*, pp. 219–253, Springer, 2018.

[288] H. Saif, Y. He, M. Fernandez, and H. Alani, "Semantic patterns for sentiment analysis of twitter," in *International Semantic Web Conference*, pp. 324–340, Springer, 2014.

[289] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188, 2010.

[290] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Systems with Applications*, vol. 57, pp. 117–126, 2016.

[291] O. Araque, I. Corcuera-Platas, J. F. Sanchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[292] S. Poria, E. Cambria, and A. Gelbukh, "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2539–2544, 2015.

[293] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1422–1432, 2015.

[294] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 496–509, 2016.

[295] Z. Zhang and M. Lan, "Learning sentiment-inherent word embedding for word-level and sentence-level sentiment analysis," in *2015 International Conference on Asian Language Processing (IALP)*, pp. 94–97, Oct 2015.

[296] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. of the Annual Meeting of the Assoc. for Computational Linguistics: Human Language Technologies - Vol. 1*, pp. 142–150, 2011.

[297] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proc. of the Annual Meeting of the Association for Computational Linguistics*, pp. 1555–1565, 2014.

[298] D. Reforgiato Recupero and E. Cambria, "Eswc'14 challenge on concept-level sentiment analysis," in *Semantic Web Evaluation Challenge*, (Cham), pp. 3–20, Springer International Publishing, 2014.

[299] D. Reforgiato Recupero, M. Dragoni, and V. Presutti, "Eswc 15 challenge on concept-level sentiment analysis," in *Semantic Web Evaluation Challenges*, (Cham), pp. 211–222, Springer International Publishing, 2015.

[300] M. Dragoni and D. Reforgiato Recupero in *Semantic Web Challenges*, (Cham).

[301] D. Reforgiato Recupero, E. Cambria, and E. Di Rosa, "Semantic sentiment analysis challenge eswc2017," in *Semantic Web Challenges*, pp. 109–123, Springer, 2017.

[302] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *CoRR*, vol. abs/cs/0506075, 2005.

[303] M. W. Rodrigues, L. E. Zárate, and S. Isotani, "Educational data mining: a review of evaluation process in the e-learning," *Telematics and Informatics*, 2018.

[304] F. Clarizia, F. Colace, M. De Santo, M. Lombardi, F. Pascale, and A. Pietrosanto, "E-learning and sentiment analysis: a case study," in *Proceedings of the 6th International Conference on Information and Education Technology*, pp. 111–118, ACM, 2018.

[305] G. S. Chauhan, P. Agrawal, and Y. K. Meena, "Aspect-based sentiment analysis of students' feedback to improve teaching–learning process," in *Information and Communication Technology for Intelligent Systems*, pp. 259–266, Springer, 2019.

[306] P. Rodriguez, A. Ortigosa, and R. M. Carro, "Extracting emotions from texts in e-learning environments," in *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 887–892, IEEE, 2012.

[307] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[308] M. Dragoni, A. Tettamanzi, and C. da Costa Pereira, "DRANZIERA: An evaluation protocol for multi-domain opinion mining," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 267–272, European Language Resources Association (ELRA), May 2016.

[309] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 165–172, ACM, 2013.

[310] N. Altrabsheh, M. Cocea, and S. Fallahkhair, "Learning sentiment from students' feedback for real-time interventions in classrooms," in *International Conference on Adaptive and Intelligent Systems*, pp. 40–49, Springer, 2014.

[311] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of facebook statuses using naive bayes classifier for language learning," in *IISA 2013*, pp. 1–6, IEEE, 2013.

[312] X. Chen, M. Vorvoreanu, and K. Madhavan, "Mining social media data for understanding students' learning experiences," *IEEE Transactions on Learning Technologies*, vol. 7, no. 3, pp. 246–259, 2014.

[313] M. Wen, D. Yang, and C. Rose, "Sentiment analysis in mooc discussion forums: What does it tell us?," in *Educational data mining 2014*, Citeseer, 2014.

[314] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[315] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[316] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv:1612.03651*, 2016.

[317] S. Ji, N. Satish, S. Li, and P. Dubey, "Parallelizing word2vec in multi-core and many-core architectures," *arXiv preprint arXiv:1611.06172*, 2016.

[318] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[319] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.