Preface of the 2nd Italian Workshop on Artificial Intelligence and Applications for Business and Industries

Francesco Epifania[1], Luca Marconi[1,2], Ricardo Matamoros Aragon[2], Ricardo Rosati[3], Emanuele Frontoni[4] and Luca Romeo[5]

[1]Social Thingum, Milan, Italy
[2]Department of Informatics, Systems and Communication, University of Milano - Bicocca, Milan, Italy
[3]Department of Information Engineering, Polytechnic University of Marche, Ancona, Italy
[4]Department of Political Science, Communication and International Relations, University of Macerata, Macerata, Italy
[5]Department of Economics and Law, University of Macerata, Macerata, Italy

1. Background and Motivations

The 2nd Italian Workshop on Artificial Intelligence and Applications for Busi- ness and Industries (AIABI) is co-located and held within the 21th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022) at Udine and organized by Social Thingum, an Italian notable AI scale-up company and private research and technology-transfer center, located in Milan, Lombardy, University of Milano Bicocca, Polytechnic University of Marche and University of Macerata. The workshop is also sponsored by Assintel, the National Association of ICT Companies of Confcommercio, as well as by InnovUp, the Italian Innovation & Startup Ecosystem. This edition is held in hybrid modality and the program of the meeting is available on the official workshop website https://www.aiabi2022.com/workshop-program/. The workshop is focused on the current technological scenario of Artificial Intelligence (AI) for business in heterogenous fields and industries. Among the editions of the AIxIA annual International Conferences, this edition of the workshop focused on the current technological scenario of AI for business in heterogenous fields and industries. The workshop mainly aims at allowing organizations, academics, researchers and specifically firms, decision-makers and practitioners to share and analyze heterogenous research works and business case studies dealing with AI in business fields. The idea behind this workshop is the opportunity to share knowledge and experience in how AI is actually and currently affecting business cases and intelligence. Companies will share specific case studies as well as their current issues AI is solving in their organizations. Researchers will provide scientific works and studies to contribute in the advancement of the many synergies between AI and business models and organizations. The final aim of the workshop is contributing in depicting the overall scenario and framework of the exploitation, advantages and current issues of AI in business. Artificial Intelligence (AI)

is becoming crucial in every business field. AI is currently reshaping organizations and how technologies affects management and business (Haefner et al., 2021). AI has the power to transform business and society, in a transversal and pervasive way, due to its ability to extract and manage knowledge potentially in every industry. Researchers and scientists are aware that AI is transforming business models of all industries, by reshaping existing organizational processes (Brynjolfsson and McAfee, 2017; von Krogh, 2018). Moreover, AI has the potential to provide higher quality, greater efficiency, and better outcomes than human experts (Agrawal et al., 2018a). AI is actually able to foster evolution in society, emerging as transversal and powerful technological paradigm and giving rise to the so-called fourth industrial revolution. Andrew Ng, former chief scientist at Badu and Cofounder at Coursera, said in a keynote speech at the AI Frontiers conference in 2017 that AI is really the new electricity: a disruptive, pervasive and enabling technology, empowering technologies and processes in potentially any field or domain. In the organizational and business framework, AI can provide assistance to decision-makers and technicians beyond the scope of humans (Groves et al., 2013; Wamba et al., 2017). Indeed, both academics and practitioners agree that AI may substantially impact firms' innovation processes (Bughin et al., 2018; von Krogh, 2018). Organizations have long exploited AI-based solutions to automate routine tasks in operations and logistics. Recent advances in computational power and resources, the exponential increase in data availability, and new machine-learning techniques now allow organizations to also exploit AI-based solutions for managerial tasks (Brynjolfsson & McAfee, 2017). For example, AI-based solutions play important roles in Unilever's talent acquisition process (Marr, 2018), in Netflix's decision-making processes regarding movie plots, directors, and actors (Westcott Grant, 2018), and in Pfizer's drug discovery and scientific development activities (Fleming, 2018). In the industrial field, there is a wide use of vision tools for the automation of quality control procedures by the means of AI tools that focus on the quantitative and deterministic analysis of a product, in order to ensure that it complies with the requirements expressed by the customer. Moreover, there is also the need for software tools which could allow the modeling and generalization of quantitative analyses that aim to determine the value of a product or material according to aesthetic standards. These operations are still carried out by specialized technicians, thereby the traditional process is slowed down by the huge waste of time and human resources required, as well as by a performance limit mainly due to the high intrinsic variability among the different annotators. For these reasons, it is not surprising that the quality control task has rapidly established itself as a relevant use case for AI in the field of Industry 4.0.

2. Topics of Interest

- Artificial Intelligence in Business
- Application of AI in industries and market
- AI use-cases in heterogeneous business contexts
- ML applications to Quality Control
- Transfer learning and domain adaptation for industrial applications
- Anomaly and defect detection
- Zero defect manufacturing

- ML for flexible manufacturing
- Decision support systems to supply chain monitoring
- AI potential in leveraging Education and training of company stakeholders
- Explainable, Interpretable and Trustworthy AI in business
- Strategies to exploit the AI potential to leverage business competitive advantages
- Theoretical aspects of AI potentialities for business
- Evaluating AI Systems and AI impact in real business scenarios
- Ethics for AI in companies and industries

3. Accepted Papers

We believe that the program provides a good balance between the different topics related to the area of AI for Business and Industries. Moreover, the program is further enriched through different chance to interact with the speaker both remotely and in presence. All the speaker provided a real application of AI in the Business and Industry use case.

Also this edition confirms how, among the works accepted, there is a specific focus on technology-transfer projects and positive accounts of fruitful collaborations between universities, research centers and companies. Such projects are particularly fundamental especially for the growth of innovation in the business and industry sectors. Then, this focus on the technology-transfer projects is also an appropriate and key reported result, especially considering public concerns by the Italian government, and also the level of innovation required by Cluster 4 of Horizon Europe: "The overarching vision behind the proposed investments under Cluster 4 is that of Europe shaping competitive and trusted technologies for a European industry with global leadership in key areas, enabling production and consumption to respect the boundaries of our planet, and maximising the benefits for all parts of society in the variety of social, economic and territorial contexts in Europe."

The call for papers attracted 6 submissions by 26 different authors. After the review process, 6 of 6 papers were accepted for publication (acceptance rate: 100%). The accepted papers range from the definition of methodologies or frameworks to apply in AI-based recommender system to empowering organization process to specific machine learning or deep learning approaches applied in the context of predictive maintenance and pollution forecasting. Going into details, accepted papers address several topics from different perspectives. In the following, we provide a short overview of such works, grouping them by topics.

Four papers proposed specific AI tools and applications usually embedded in a recommender system for improving social relationship and business organization. In particular Ali et al. proposed a modified CNN model for age and gender recognition at real time. Age and gender information are essential for many real-world appli- cations, such as social intelligence, biometric identity verification, video surveil- lance, human-computer interaction, digital consumer, crowd behavior analysis, online marketing, item recommendation, and many more. Siváková et. Al, proposed a quantification of user preferences during decision making by using a fully probabilistic design approach. Their paper provide representative evidence behind the quantification of preferences. Guerranti et al. proposed the use of machine learning and artificial intelligence methods to predict the likelihood of resignation of an employee. In their paper

they showed how machine learning models can indeed play a crucial role as decision support systems, identifying the best decisions to be made and providing data-motivated explanations. Marconi et al. provided an overview of Explainable AI in the field of recommender systems. As a general consideration to conclude their study, they definitively agree that the evolution of recommender system systems necessarily involve a synergy between the empowerment of the models' performances and the emergent human-AI in- teraction perspective.

Two paper focused on the Industry process domain. D'Agostino et al. investigated the prediction capabilities of neural sequence models for the prediction of the remaining useful life of a machine component for predictive maintenance task. Their experiments were performed on a public dataset from particle filtration systems. The evidences presented in their paper highlighted the accuracy of some of these models when modelling the evolution of the health state of the analyzed machine. Nonetheless, the qualitative analysis shows that this prediction is less accurate when the fault is far away. Finally Roitero et al. proposed a novel approach based on large language causal models to perform the task of time-series forecasting. In particular they used the proposed approach based on transformers to effectively forecast the concentration of polluting substances in a water treatment plant; they addressed both short- and mid- term forecasting. Their empirical results provided evidence that large language models are more effective than state-of-the-art forecasting systems for solving this type of task, and that they can be practically used in time-series forecasting tasks.

4. Committee

As a final remark, the co-chairs would like to thank all the members of the Program Committee (listed below), the organizers of the AI*IA 2022 Conference, the Italian Association for Artificial Intelligence, the University of Milano – Bicocca, the University of Udine, the Marche Polytechnic University and the University of Macerata as well as the sponsors, Assintel, the Italian National Association of ICT Companies, and InnovUp, the Italian Italian Innovation & Startup Ecosystem.

5. Chairs

Francesco Epifania, Social Thingum (ITALY)
Luca Marconi, Social Thingum, University of Milano - Bicocca (ITALY)
Ricardo Matamoros, Social Thingum, University of Milano - Bicocca (ITALY)
Riccardo Rosati, Marche Polytechnic University (ITALY)
Emanuele Frontoni, University of Macerata, Marche Polytechnic University (ITALY)
Luca Romeo, University of Macerata, Marche Polytechnic University (ITALY)

6. Organizers

Simone Caporale, Profima srl, AI Magister (ITALY)
Ernesto Damiani, University of Milan (ITALY)
Francesco Epifania, Social Thingum (ITALY)

Alberto Fioravanti, Digital Magics (ITALY)
Giacinto Fiore, AI Week Italia (ITALY)
Guido di Fraia, IULM University (ITALY)
Emanuele Frontoni, University of Macerata, Marche Polytechnic University (ITALY)
Iuri Frosio, NVIDIA (USA)
Luca Marconi, Social Thingum, University of Milano - Bicocca (ITALY)
Ricardo Matamoros, Social Thingum, University of Milano - Bicocca (ITALY)
Luca Nardone, Unicredit (ITALY)
Filippo Neri, Università di Napoli Federico II, ProMarket 11 (ITALY)
Luca Romeo, University of Macerata, Marche Polytechnic University (ITALY)
Riccardo Rosati, Marche Polytechnic University (ITALY)
Alessandro Rozza, lastminute.com group (SWITZERLAND)
Alberto Schiaffino, Engitel S.p.A. (ITALY)
Pasquale Viscanti, AI Week Italia (ITALY)

7. Program Committee

Ernesto Damiani, University of Milan (ITALY)
Guido di Fraia, IULM University (ITALY)
Donato De Ieso, dilium srl, Assintel (ITALY)
Francesco Epifania, Social Thingum (ITALY)
Alberto Fioravanti, Digital Magics (ITALY)
Iuri Frosio, NVIDIA (USA)
Altin Kadareja, Cardo AI (ITALY)
Eugenia Kovatcheva, University of Sofia (BULGARY)
Alberto Lucchetti, lastminute.com group (SWITZERLAND)
Luca Marconi, Social Thingum, University of Milano - Bicocca (ITALY)
Giulio Massucci, Wavenure (ITALY)
Ricardo Matamoros, Social Thingum, University of Milano - Bicocca (ITALY)
Luca Nardone, Unicredit (ITALY)
Filippo Neri, Università di Napoli Federico II, ProMarket 11 (ITALY)
Roumen Nikolov, University of Sofia (BULGARY)
Alessandro Rozza, lastminute.com group (SWITZERLAND)

A Modified CNN for Age and Gender Prediction

Danish Ali[1] , Francesco Epifania[2], Naeem Ahmed[3] , Bilal Khan[4] , Luca Marconi[5], Ricardo Matamoros[6]

[1,4] *GPGC Haripur affiliated with University of Haripur, Near new judiciary, Haripur, 22620, Pakistan*
[2,6] *Social Things srl, Italy*
[3] *UET, Taxila, Taxila, Pakistan*
[5] *University of Milano-Bicocca, Milan, Italy*

Abstract

Age and gender information are essential for many real-world applications, such as social intelligence, biometric identity verification, video surveillance, human-computer interaction, digital consumer, crowd behavior analysis, online marketing, item recommendation, and many more. This study intends to employ deep learning technology in the prediction process, effective accuracy, and predictive mining and assess it in order to obtain the best outcomes of prediction and get around the issues of time, accuracy, and processing load. In this multi-task learning problem, age and gender are predicted concurrently with the help of a single Convolutional neural network with two heads (output branches). The model has 95% accuracy for gender classifier and 92% accuracy for age classifier. The pro-posed model uses the computing resources (RAM, CPU, and GPU) in a much more optimized manner and the computing cost is also lower.

Keywords

Convolutional Neural Network, Recognition System, Gender Prediction, Age Prediction

1. Introduction

For many real-world applications, including social intelligence, biometric identity verification, video surveillance, human-computer interface, digital consumer, crowd behavior analysis, online marketing, item suggestion, and many more, age and gen-der data are crucial [1, 14]. No matter how widespread their uses, being able to automatically determine age and gender from face pictures is a very difficult problem [15]. This is especially true given the various sources of intra-class variations at peo-ple's facial images, which restricts the use of these models in real-world programs [2, 16]. In the past several years, a lot of works have been offered for predicting age and gender [3]. Recent research has focused in particular on using a classifier after manually extracting face information from photos [17]. Nonetheless, because to the out-standing success of deep learning models in several computer vision issues over the past few years [18], the majority of the more recent efforts on age and gender predictions have turned toward models based on deep neural networks [4, 19].

As aim to propose a deep learning system in this study to jointly estimate the age and gender from facial images. Given the intuition that a few neighboring parts of the face provide very obvious messages regarding a person's age and gender [20] (inclusive of beard and moustache for male, and wrinkles around eyes and mouth for age) [5]. Employ a single version using a multi-task learning approach to collectively estimate both gender and age bucket since estimating age and gender from faces is highly correlated [6]. Additionally, as knowing a person's gender helps us estimate their age more accurately, add the predicted gender output to the age-prediction branch's feature [21].

In order to accurately anticipate the future and learn more about a specific man or woman, studies in the biometric field, including human face recognition applications, focus on gender and age

[1]2nd Italian Workshop on Artificial Intelligence and Applications for Business and Industries - AIABI | co-located with AI*IA 2022
EMAIL: danishalikhan545@gmail.com (A. 1); francesco.epifania@socialthingum.com (A. 2); Naeem.uoh@gmail.com (A. 3); bkhan66688@gmail.com (A. 4); luca.marconi@socialthingum.com (A. 5); ricardo.matamoros@socialthingum.com (A. 6)
ORCID: 0000-0002-4828-4239 (A. 1); 0000-0003-1839-9366 (A. 2); 0000-0002-1850-5846 (A. 3); [0000-0001-8567-4298 (A. 4); 0000−0002−0236−6159 (A. 5); 0000−0002−1957−2530 (A. 6)

prediction [7, 22]. Different techniques and algorithms are used throughout the process, with deep learning seeing the highest usage rates [8, 23]. In this study, propose a deep learning framework to predict the gender and age group of face images with a high accuracy rate. This framework is built on the ensemble of attention and residual Convolutional networks.

The aims of the proposed model are to use the approach of deep learning technique in the prediction process, the effective accuracy and predictive mining and evaluate it reaching to get best results of prediction and overcome the problems of time, accuracy and processing load. Section 2 presents the related work while Section 3 discusses the proposed methodology. The Section 4 elaborate the results of proposed model while Section 5 conclude the paper.

2. Related Work

Prediction of age and gender from the face photos, as a special problem of face analysis has been attracting attention in recent years [9]. There are many works done so far in the prediction of age and gender from facial images. Here are review of the most promising research work.

Nada et al., [10] conducted research on validation and prediction of gender and age using CNN for one single image. The UP-student's dataset was used in the experiment to evaluate the suggested method. Sadly, age estimation deteriorated due to the pro-posed solution's poor gender prediction performance [24]. Overall, both genders had a gender prediction accuracy of roughly 82%. Additionally, the algorithm performs better when guessing images of male faces (89%, compared to 74% for females). After examining the photographs where the model failed to correctly estimate the gender, there were a number of causes. The primary cause at some ages, the distinction between the facial characteristics of men and women is not always as obvious as it ought to be. Hijab also conceals various facial characteristics in photographs of women. Finally, there is a flaw in the model that was utilized; it did not assign the moustache enough importance in predicting gender. Considering the age prediction findings, it was not so good that the total forecast accuracy for both genders was just 57%.

Al-Azzawi, [11] used Adience Benchmark dataset of face images; it consists of 17603 images of human faces for variance of ages and genders. The ages of the persons in the dataset are classified into 10 groups and the gender binary is classified into two types. The images of the datasets are divided into two sets equally, one for the training phase and another for the testing phase. They used Mean Absolute Error (MAE) function to evaluate the implementation of age prediction, and the accuracy of the gender prediction assessed by the hit ratio to compare the current proposed Deep Multi-tasking CNN [25]. The accuracy for gender detection was (91%) while accuracy in mean absolute error (MAE) for age prediction was (4.00) in CNN and DMTL model.

3. Proposed Method

This section will discuss the design and components used in the proposed system. The dataset and its features are also discussed in this section.

3.1. Proposed Model

The proposed system is being created and developed while keeping in mind all of the shortcomings and restrictions of the current system, thus anticipate that it will be an acceptable system that successfully satisfies all of the goals of the current system.

The reasons behind choosing of this model are to improve accuracy, for both of the classifiers i.e. age and gender, to improve training time for both classifier, and to minimize the resource utilization of computer. A new CNN layer architecture is de-signed which works much better than other Built-in models like VGG, resnet50 and Mobile net etc. Especially for that particular problem. It contains the Convolution layer, Batch Normalization layer, Max and average pooling (at the last) and also dropout layers along with fully connected layers. Here it used ReLU, Elu and Soft-max activation functions. The model takes input layer of image size= (128, 128, 3) and used 3x3 kernel/filter. Also used bias

constraints, filter of 64 to 512 and stride of (1, 1) with same padding. Figure 1 illustrates the proposed model architecture.

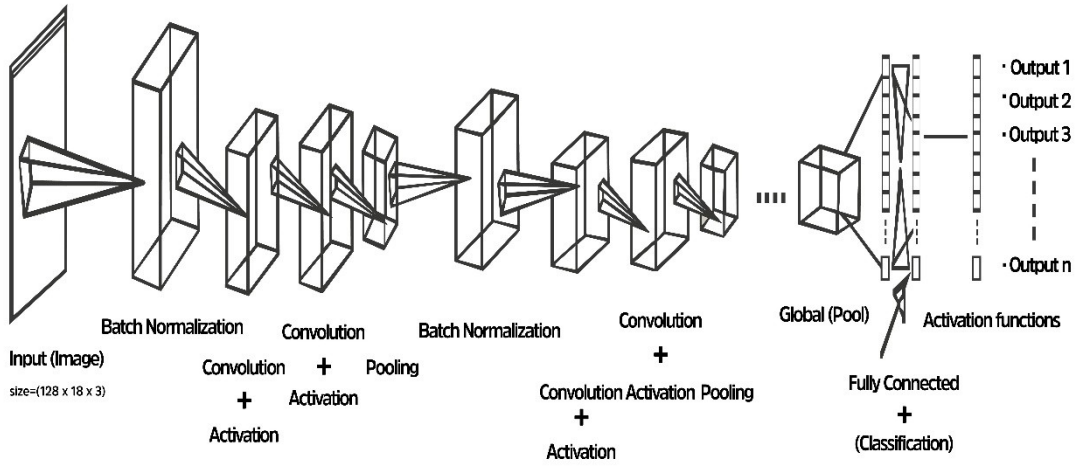


Figure 1: Proposed model architecture

3.2. Data Set

The Adience dataset served as the basis for this study. The fundamental tenet of the data collection is to record the photographs as accurately as possible, taking into account any variances in look, posture, lighting, and image quality, to mention a few. Almost 26000 pictures were used along with labels of age and gender group. The Adience benchmark dataset, which consists of face photographs that are automatically posted from smart phones to Flickr, is made for age and gender categorization. These photos show unfiltered, unedited photographs from the real world and social media. So, need to adjust them according to suitable way by performing preprocessing.

3.3.Structure of Proposed Model

A single Convolutional neural network with two heads (output branches) is utilized in this multi-task learning issue to simultaneously predict age and gender with the following properties.

- Number of Epoch for both Age and Gender module is 12
- Size of image is 128 x 128.
- Total number of parameters 6059022.
- Batch size used is 64.
- Time required for both classifiers is 45 min Approx.
- “Sparse categorical Cross entropy” as loss functions.
- 20% data for testing and 80% for training.
- 25 deep layers.

3.4.Training and Testing

In the training and testing phases of the current network, Adience Benchmark dataset of face images was used; it consists of Approx. 26000 images of human faces for variance of ages and genders. The ages of the persons in the dataset are classified into 8 groups and the gender binary is classified into two types. The images of the datasets are divided into two sets equally, one for the training phase and another for the testing phase. The image size 128x128 was used in the training and testing phase. The Epoch for gender is 15 and Epoch for Age is 12, and the time taken for both classifier to train is 90 min (45 min each).

4. Results and Analysis

The gender classification model is tested and evaluated using machine learning evaluation metrics. The below section will discuss the results of gender classification model.

4.1. Gender Classification model

Gender prediction is viewed as a classification issue, and this network's output layer is a Softmax with two nodes that represent the classifications of male and female. The model is implemented as a network with three layers, two of which are output layers and one of which is a fully linked layer. The anticipated values for each class may be obtained from the gender prediction network by loading this model into memory and sending the output of the face detection process (detected face) through the network. Now that the output has reached its maximum value, that may utilize that number to determine a person's gender. Table 1 presents the classification report of gender model while Figure 2 illustrates the accuracy comparison of gender model with other models.

Table 1
Classification Report for the Gender Model

(Classes)	(Precision)	(Recall)	(F1-Score)	(Support)
Female (0)	0.92	0.92	0.92	2744
Male (1)	0.92	0.91	0.91	2492
Accuracy			0.95	5236

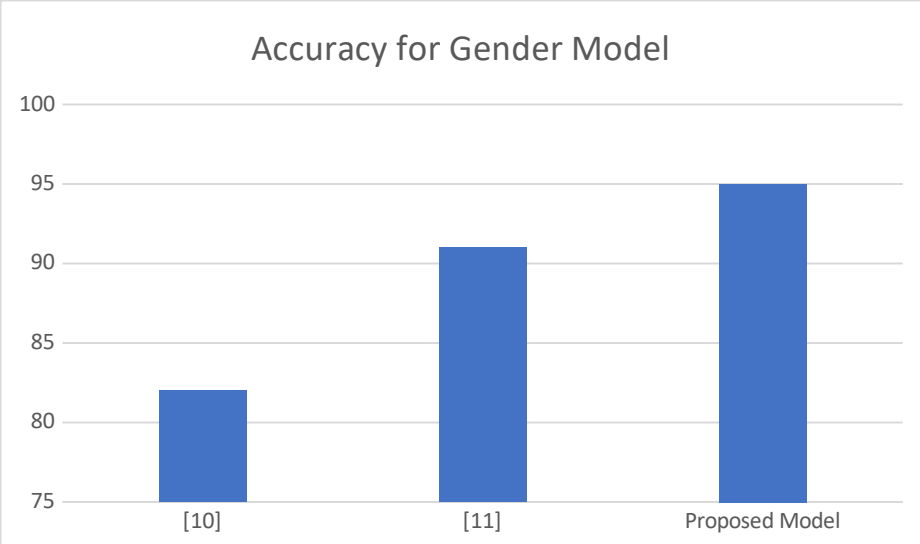


Figure 2: Accuracy comparison between gender models

4.2. Age Classification model

People find it exceedingly difficult to make accurate age predictions by simply looking at a person, but it is possible to do so when estimating a range of ages. Consequently, regarded it as a classification issue using the Adience Benchmark dataset. The predicted values for all training may be obtained from the network by loading this model into memory and running the output of the face detection process (detected face) through the age prediction network. As, utilized the output's maximum value as a forecast age group by taking that value. Table 2 presents the classification report of age model while Figure 3 illustrates the accuracy comparison of age model with other models

Table 2
Classification Report for Age Model

(Classes)	(Precision)	(Recall)	(F1-Score)	(Support)
(0-2)	0.83	0.83	0.83	422
(4-6)	0.69	0.85	0.76	625
(8-13)	0.86	0.63	0.73	662
(15-20)	0.61	0.56	0.59	558
(25-32)	0.71	0.79	0.75	1558
(38-43)	0.59	0.65	0.62	879
(48-53)	0.53	0.38	0.44	267
(60+)	0.91	0.55	0.69	265
Accuracy		0.92	5236	

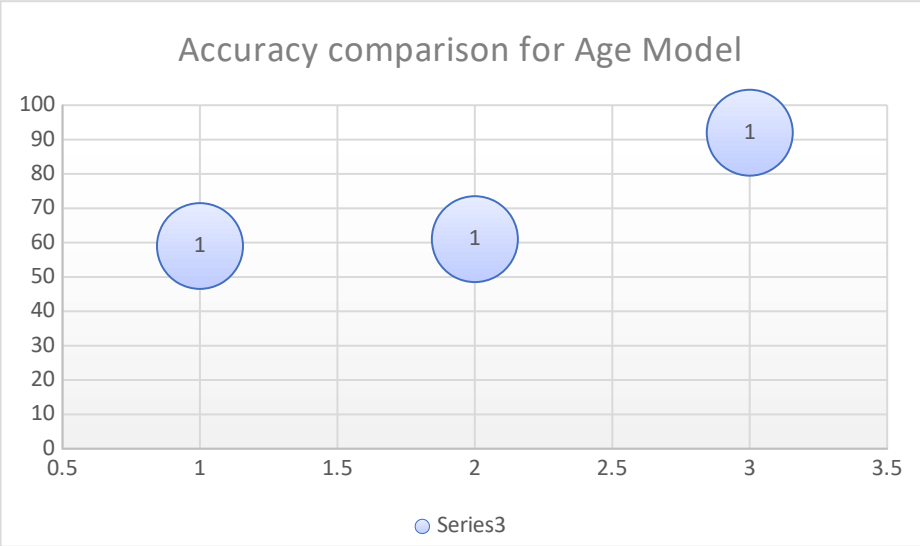


Figure 3: Accuracy comparison of Age Model

The proposed model gives 95% Accuracy for Gender Classifier and 92% for Age Classifier which is highest among all the previous/related work discussed. The model takes much short time to train up for both of the classifier i.e. Age and Gender Classifier. model utilizes the computing resources (RAM, CPU, and GPU) in much optimized manner and the computing cost is also lower for the model as compared to the previous model discussed in existing model session.

5. Conclusion

Age, gender, and the age range of a person's personal photo have recently become crucial pieces of information for many businesses and governments to use for commercial, identity, security, and other purposes [12]. Additionally, because this information was gathered from people using an enterprise system, form validation was suggested as a way to lower user data entry mistakes. The module is mainly designed for biometrics research in social applications for the future where the content is to be shown for some specific gender and age group and it must be possible to predict and disclose information about each person [13, 26]. The experimental investigation found that the suggested CNNs had a fair classification accuracy after being trained quickly with a large number of photos. The proposed CNNs will be utilized in next work for social media statistics and gender categorization in mobile applications. Secondly, the model is just trained on Adience benchmark dataset that means

more sophisticated systems can use more training data. It is possible that the results can be significantly improved in future beyond the results reported here.

6. References

[1] Hizam, S. M., Ahmed, W., Fahad, M., Akter, H., Sentosa, I., & Ali, J. User behavior assessment towards biometric facial recognition system: A SEM-neural network approach. In Future of Information and Communication Conference (pp. 1037-1050). Springer, Cham (2021).

[2] Zhu, J., Luo, B., Zhao, S., Ying, S., Zhao, X., & Gao, Y. IExpressNet: Facial Expression Recognition with Incremental Classes. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 2899-2908) (2020).

[3] Shinde, G. R., Kalamkar, A. B., Mahalle, P. N., Dey, N., Chaki, J., & Hassanien, A. E. Forecasting models for coronavirus disease (COVID-19): a survey of the state-of-the-art. SN Computer Science, 1(4), 1-15 (2020).

[4] Chekroud, A. M., Bondar, J., Delgadillo, J., Doherty, G., Wasil, A., Fokkema, M., ... & Choi, K. The promise of machine learning in predicting treatment outcomes in psychiatry. World Psychiatry, 20(2), 154-170 (2021).

[5] Maior, C. B. S., das Chagas Moura, M. J., Santana, J. M. M., & Lins, I. D. Real-time classification for autonomous drowsiness detection using eye aspect ratio. Expert Systems with Applications, 158, 113505 (2020).

[6] Ma, C., Wei, J. L., Liu, B., Ding, M., Yuan, L., Han, Z., & Poor, H. V. Trusted AI in Multi-agent Systems: An Overview of Privacy and Security for Distributed Learning. arXiv preprint arXiv:2202.09027 (2022).

[7] Alonso-Fernandez, F., Hernandez-Diaz, K., Ramis, S., Perales, F. J., & Bigun, J. Facial masks and soft-biometrics: Leveraging face recognition CNNs for age and gender prediction on mobile ocular images. IET Biometrics, 10(5), 562-580 (2021).

[8] Tufek, N., Yalcin, M., Altintas, M., Kalaoglu, F., Li, Y., & Bahadir, S. K. Human action recognition using deep learning methods on limited sensory data. IEEE Sensors Journal, 20(6), 3101-3112 (2019).

[9] Carletti, V., Greco, A., Percannella, G., & Vento, M. Age from faces in the deep learning revolution. IEEE transactions on pattern analysis and machine intelligence, 42(9), 2113-2132 (2019).

[10] Abu Nada, A.M., Alajrami, E., Al-Saqqa, A.A. and Abu-Naser, S.S., Age and Gender Prediction and Validation Through Single User Images Using CNN (2020).

[11] Al-Azzawi, D.S,. Human Age and Gender Prediction Using Deep Multi-Task Convolutional Neural Network. Journal of Southwest Jiaotong University, 54(4) (2019).

[12] Zeebaree, S., Ameen, S., & Sadeeq, M. Social media networks security threats, risks and recommendation: A case study in the kurdistan region. International Journal of Innovation, Creativity and Change, 13, 349-365 (2020).

[13] Faundez-Zanuy, M., Fierrez, J., Ferrer, M. A., Diaz, M., Tolosana, R., & Plamondon, R. Handwriting biometrics: Applications and future trends in e-security and e-health. Cognitive Computation, 12(5), 940-953 (2020).

[14] Bosman, L., Fernhaber, S., & SpringerLink (Online service). Teaching the entrepreneurial mindset to engineers. Switzerland: Springer International Publishing (2018).

[15] Othmani, A., Taleb, A. R., Abdelkawy, H., & Hadid, A. Age estimation from faces using deep learning: A comparative analysis. Computer Vision and Image Understanding, 196, 102961 (2020).

[16] Korban, M., Youngs, P., & Acton, S. T. TAA-GCN: A Temporally Aware Adaptive Graph Convolutional Network for Age Estimation. Pattern Recognition, 109066 (2022).

[17] Sun, N., Zhang, J., Rimba, P., Gao, S., Zhang, L. Y., & Xiang, Y. Data-driven cybersecurity incident prediction: A survey. IEEE communications surveys & tutorials, 21(2), 1744-1772 (2018).

[18] Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. Deep learning and process understanding for data-driven Earth system science. Nature, 566(7743), 195-204 (2019).
[19] Hosny, A., Parmar, C., Coroller, T. P., Grossmann, P., Zeleznik, R., Kumar, A., ... & Aerts, H. J. Deep learning for lung cancer prognostication: a retrospective multi-cohort radiomics study. PLoS medicine, 15(11), e1002711 (2018).
[20] Tahir, R., Batool, B., Jamshed, H., Jameel, M., Anwar, M., Ahmed, F., ... & Zaffar, M. F. Seeing is believing: Exploring perceptual differences in deepfake videos. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-16) (2021, May).
[21] Khanagar, S. B., Vishwanathaiah, S., Naik, S., Al-Kheraif, A. A., Divakar, D. D., Sarode, S. C., ... & Patil, S. Application and performance of artificial intelligence technology in forensic odontology–A systematic review. Legal Medicine, 48, 101826 (2021).
[22] Benkaddour, M. K., Lahlali, S., & Trabelsi, M. Human Age and Gender Classification using Convolutional Neural Network. In 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH) (pp. 215-220). IEEE (2021, February).
[23] Ghosh, P., Azam, S., Karim, A., Jonkman, M., & Hasan, M. Z. Use of efficient machine learning techniques in the identification of patients with heart diseases. In 2021 the 5th International Conference on Information System and Data Mining (pp. 14-20) (2021, May).
[24] Rai Jain, P., MK Quadri, S., & Lalit, M. Recent Trends in Artificial Intelligence for Emotion Detection using Facial Image Analysis. In 2021 Thirteenth International Conference on Contemporary Computing (IC3-2021) (pp. 18-36) (2021, August).
[25] Ghosh, P. Deep Learning to Diagnose Diseases and Security in 5G Healthcare Informatics. In Machine Learning and Deep Learning Techniques for Medical Science (pp. 279-331). CRC Press (2022).
[26] Shopon, M., Hossain Bari, A. S. M., Bhatia, Y., Narayanaswamy, P. K., Tumpa, S. N., Sieu, B., & Gavrilova, M. Biometric System De-identification: Concepts, Applications, and Open Problems. In Handbook of Artificial Intelligence in Healthcare (pp. 393-422). Springer, Cham (2022).

Estimating the Remaining Useful Life via Neural Sequence Models: a Comparative Study

Giovanni D'Agostino[1], Alex Falcon[1,2,*], Oswald Lanz[3], Giorgio Brajnik[1], Carlo Tasso[1] and Giuseppe Serra[1]

[1]University of Udine, Via delle Scienze, 206, 33100 Italy
[2]Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Italy
[3]Free University of Bozen-Bolzano, Piazza Domenicani, 3, 39100 Italy

Abstract

The prediction of the Remaining Useful Life of a machine component represents a strategic problem in predictive maintenance, which may have important consequences for a company. Recent approaches to this problem leverage data-driven methodologies based on deep learning, achieving impressive results. In particular, due to the temporal nature of the sensor measurements detailing the life of a component, neural sequence models are often chosen to automate the feature extraction process. In this paper, we investigate several of these models on a particle filtration system. The experiments performed present the good prediction capabilities of these models, highlighting some of them for their accuracy. Nonetheless, the qualitative analysis shows that when the fault is farther away, most of these models tend to have unstable predictions. These results motivate some future research directions which are discussed in the conclusions.

Keywords

Remaining useful life estimation, Neural sequence models, Deep learning

1. Introduction

Recently, the United States Department of Energy reported that most of the companies in the USA follow a reactive maintenance strategy, that is they wait for a machine component to fail instead of properly maintaining it [1]. To avoid the replacement of extremely costly components, being able to accurately estimate when a failure is going to happen, that is to estimate its Remaining Useful Life (RUL), represents a strategic problem which is often put at the core of predictive maintenance [2].

Commonly, the methods for the RUL estimation task methods are either model-based or data-driven. In the former case, the predictions are made by physical or mathematical models which simulate the degradation of the machine under analysis [3, 4]; however, because of the need for

*Proceedings of the 2nd Italian Workshop on Artificial Intelligence and Applications for Business and Industries (AIABI 2022) co-located with 21st International Conference of the Italian Association for Artificial Intelligence (AI*IA 2022), Udine, Italy, December 2, 2022.*

*Corresponding author.
dagostino.giovanni@spes.uniud.it (G. D'Agostino); falcon.alex@spes.uniud.it (A. Falcon); oswald.lanz@unibz.it (O. Lanz); giorgio.brajnik@uniud.it (G. Brajnik); carlo.tasso@uniud.it (C. Tasso); giuseppe.serra@uniud.it (G. Serra)
0000-0002-6325-9066 (A. Falcon); 0000-0003-4793-4276 (O. Lanz); 0000-0002-4269-4501 (G. Serra)

domain expertise and extensive verification, they are highly complex, expensive, and need to be designed in a case-by-case manner. The methods developed with the second methodology rely on historical sensor data to build an approximate degradation model by leveraging handcrafted features [5, 6]. Although powerful and often applicable to heterogeneous domains, they still rely on domain expertise in order to perform the feature engineering step. Recently, data-driven methodologies which use deep learning gained a lot of attention [7, 8], thanks to their automatic feature extraction step, which works directly on raw data, and ease of application to different domains. In particular, neural sequence models are often chosen because they inherently discover hidden patterns in temporally-related data [9, 10].
This paper investigates the prediction capabilities of neural sequence models. The experiments are performed on a public dataset from particle filtration systems [11], which are often deployed in manufacturing companies dealing with food and beverage, semiconductor and electronic components, and many more. The evidences presented in this paper highlight the accuracy of some of these models when modelling the evolution of the health state of the analyzed machine. Nonetheless, the qualitative analysis shows that this prediction is less accurate when the fault is far away.

2. Related Work

Model-based and data-driven methods. The problem of correctly estimating the RUL has been strategic for several decades [12, 13]. Traditional approaches can be divided in model-based and data-driven. The former use mathematical or physical models of the degradation phenomena, e.g. [14, 15], thus requiring an in-depth understanding of the underlying system and the failure modes. Instead, data-driven methods build a degradation model solely based on historical sensor data. Methods based on statistics, e.g. [5, 16], and Artificial Intelligence, e.g. [6, 17], are popular examples. Instead of relying on a deep understanding of the underlying system, data-driven methods leverage handcrafted features which are extracted from the raw data. However, such a feature engineering step can be time consuming and may still rely on domain knowledge.
Deep Learning-based methods. A major advantage of deep learning consists in the automatic extraction of the features, as opposed to handcrafted ones. Initial approaches with these techniques used Multilayer Perceptrons (MLP) to estimate the RUL directly from raw data [18]. However, since the sensor measurements are taken periodically, they likely have temporal dependencies, making neural sequence models a more suitable choice. RNNs were used in [18], yet they can fail at remembering information from long time series. To overcome this issue, memory-based networks were used to store key knowledge over time: for instance, LSTMs [7, 19] and GRUs [20, 21] were often used. More recently, NTMs also showed potential in this field by using a memory bank and learnable operations to access and modify it [8, 22].

3. Methodology

An overview of the methodology followed in this study is shown in Figure 1. In particular, the time series are first sliced into shorter windows, normalized through MinMax, and then labelled

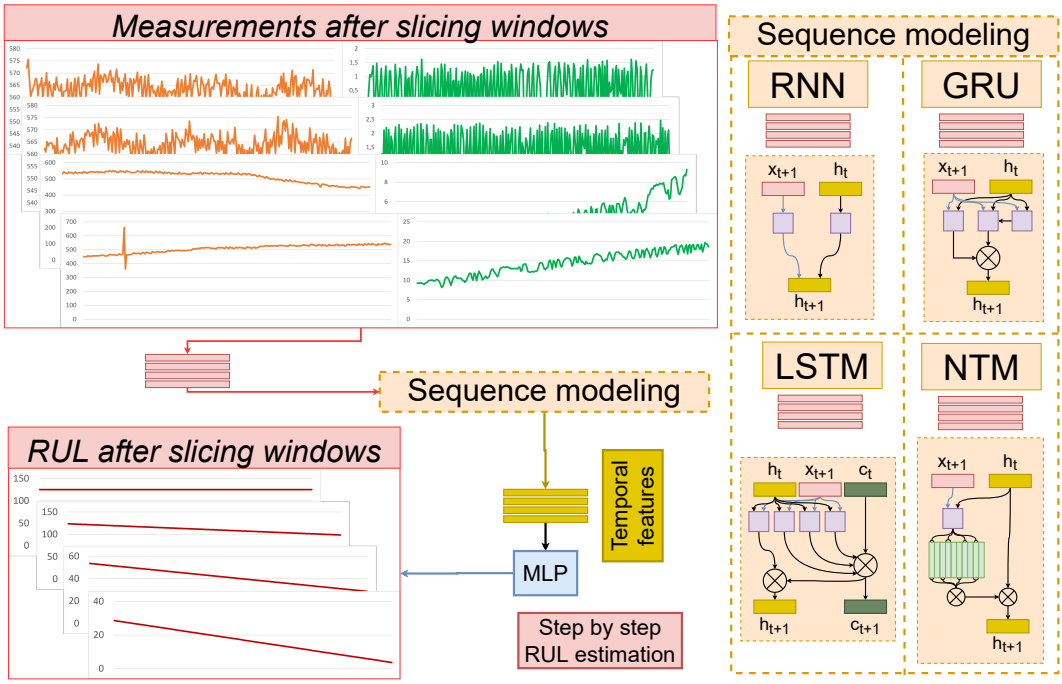


Figure 1: Graphical overview of our methodology. The series of sensor measurements are first sliced into short windows. A sequence model is used to automatically extract relevant features from the raw data. Finally, a MLP is used to estimate the RUL values.

with a piece-wise degradation function [18] with a max RUL of 125, as in [23]. The sequences are then modelled by means of a neural sequence model, including Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU), and Neural Turing Machines (NTM). Then, a mapping function between the automatically extracted features and the RUL values is learned by using an MLP. Finally, the network weights are optimized by using the Mean Square Error of the predictions. In the following, these sequence models are briefly described.

Recurrent Neural Network. The data typically considered in the Prognostics and Health Management field is composed of long time series measurements of sensors data. To model temporally-related sequential data and the evolution of its intrinsic characteristics, RNNs have shown good performance in extrapolating hidden patterns in data. RNNs are a class of artificial neural networks which compute the t-th output by using the $t-1$-th output together with the t-th element of the input sequence. RNNs are affected by two problems when calculating the gradient of the cost function over long input sequences: the vanishing gradient problem, in which the value of the gradient gradually converges to zero (thus “vanishing"), and the exploding gradient problem, in which its value tends to infinite (thus “exploding") [24, 25].

Long Short-Term Memory. Due to the length of the sequences considered in the RUL estimation problem, the gradient issues affecting RNNs need to be paid attention to; the LSTM Networks were introduced to mitigate such issues. The flow of information in an LSTM network is controlled by three gates, called *input*, *output*, and *forget*. Moreover, two memory states,

called *hidden* and *cell*, are recurrently updated by the LSTM. In particular, the input gate decides whether to update the cell state by using the current input, the forget gate decides whether to keep or forget the information from the previous hidden state, and the output gate decides how to update the hidden state given the information stored in the cell state.

Gated Recurrent Unit. GRUs were introduced in [26] as a variant of the LSTM networks architecture; in fact, GRUs have only two gates: the reset and the update gate. Differently from LSTMs, GRUs do not possess a cell state, and the reset gate is applied directly to the previous hidden state, therefore performing a similar task as the input and output gates in LSTMs. It follows that GRUs have less training parameters than LSTMs, thus using less memory and executing faster; nonetheless, LSTMs may be more accurate on larger datasets.

Neural Turing Machine. The NTM was originally proposed in [10] and later applied to the predictive maintenance field in [22, 8]. It is inspired by classical Turing Machines: in fact, it comprises a tape-like memory and updates it by means of read and write operations which are guided by a controller. Differently from LSTMs and GRUs, the NTM has an array of memory vectors, therefore enlarging its mnemonic capabilities and possibly reducing the likelihood of overwriting previously learnt concepts. This is also made possible by the usage of learnable read and write operations, which consider contextual information to decide which locations to use and to which extent the information contained therein should be updated.

4. Experimental Results

4.1. Analyzed dataset

The PHM Society 2020 Data Challenge **(PHM20)** [11] public dataset is used to perform the experiments because it offers sensor measurements comprising failures in a particle filtration system, which is often used in food and beverage manufacturing, pharmaceutical industries, etc. In this dataset, the measurements come from an experimental rig. Contaminants in the liquids passing through the system may clog it, and the challenge objective is to anticipate when such an occurrence will happen. In particular, the clogging can be identified when the pressure difference is higher than 20 psi. Each of the 32 experiments (24 for training, 8 for validation) in the dataset include concentration (40%-47.5%) and size (45-53μm, or 63-75μm) of the contaminant particles, and are thousands of steps long with a sampling rate of 10 Hz. For each time step, three measurements are taken: flow rate, upstream and downstream pressures. In addition, we also consider the concentration value and the size of the particles. In this work, the RUL is 0 when the pressure difference becomes higher than 20 psi for the first time. Finally, we use the validation experiments as the test data, and further split the training data with an 80/20 ratio to create a validation set.

4.2. Training settings and model evaluation

The experiments are performed using PyTorch 1.7.1. With our hardware (RTX A5000 and i7-9700K), a training run takes around 50 minutes for the NTM, for which a CUDNN implementation is not currently available, and 6-8 minutes for the other models. We used the following

Table 1
5-runs average RMSE/MAE values on the test set. Overall best is underlined.

Temp. ctx	30	45	60	70	140	210	280	350
RNN	9.3/6.5	9.4/6.7	8.3/5.7	9.2/6.4	9.8/6.8	11.1/8.5	10.7/8.0	11.1/8.0
LSTM	10.4/6.8	10.3/6.8	7.0/4.8	9.3/6.3	7.5/5.2	8.4/5.6	10.2/7.3	8.1/5.8
GRU	**9.0**/6.2	**6.9**/**4.7**	**6.6**/**4.3**	7.8/5.5	**6.2**/**4.4**	6.9/4.6	5.9/4.5	**6.2**/**4.5**
NTM	**9.0**/**5.8**	7.3/4.6	7.1/4.5	**6.8**/**4.4**	6.7/4.5	**5.5**/**3.7**	**5.4**/**3.7**	6.9/5.0

hyperparameters: batch size 100, learning rate 5e-3, 64 neurons in the MLP, and all the hidden sizes are set to 64.
In this study, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) were chosen to assess the prediction accuracy of each of the sequence models. A key difference consists in the higher sensitivity of RMSE when it comes to prediction errors which highly deviate from the mean value.

4.3. Quantitative comparison

Since sequence models perform the prediction based on a sequence of observations, varying the size of such a temporal context may highly influence the prediction error. Therefore, an experiment is conducted by using different sizes for it. Three observations can be drawn from Table 1. Firstly, by using shorter contexts, e.g. 30-45, which likely lack of crucial information, all the models make unstable predictions, leading to high RMSE. Secondly, by increasing it, more information is likely to be found, and the prediction error steadily decreases. In particular, by modelling the sequences with a GRU or a NTM and using a context of 280 steps, the lowest error is achieved (5.9 RMSE and 4.5 MAE, and 5.4 RMSE and 3.7 MAE). Lastly, while really long sequences may contain additional and potentially useful information, they are also harder to be modelled: as a consequence, the prediction error increases.

4.4. Qualitative analysis

Figure 2 compares the prediction made by the four models (context of 280) on a full experiment from the test set (groundtruth shown in red). It shows that all the models are highly precise when the RUL is close to 0, indicating that the fault is evident by looking at the sensor measurements. Conversely, the farther from the fault, the higher the uncertainty: this clearly indicates the difficulty of anticipating such an event, although the GRU and the NTM are quite precise, especially if compared to the noisy predictions made by the RNN and the LSTM.

5. Conclusions

Being able to predict when a fault may occur in a industrial machine is fundamental. To achieve this goal, a precise predictive model is required and the availability of historical data often shifts the attention to data-driven methodologies, and in particular to the use of deep learning techniques to automatically extract useful features from raw sensor measurements. Given that faults develop over time, in this study we investigated the predictive capabilities of several

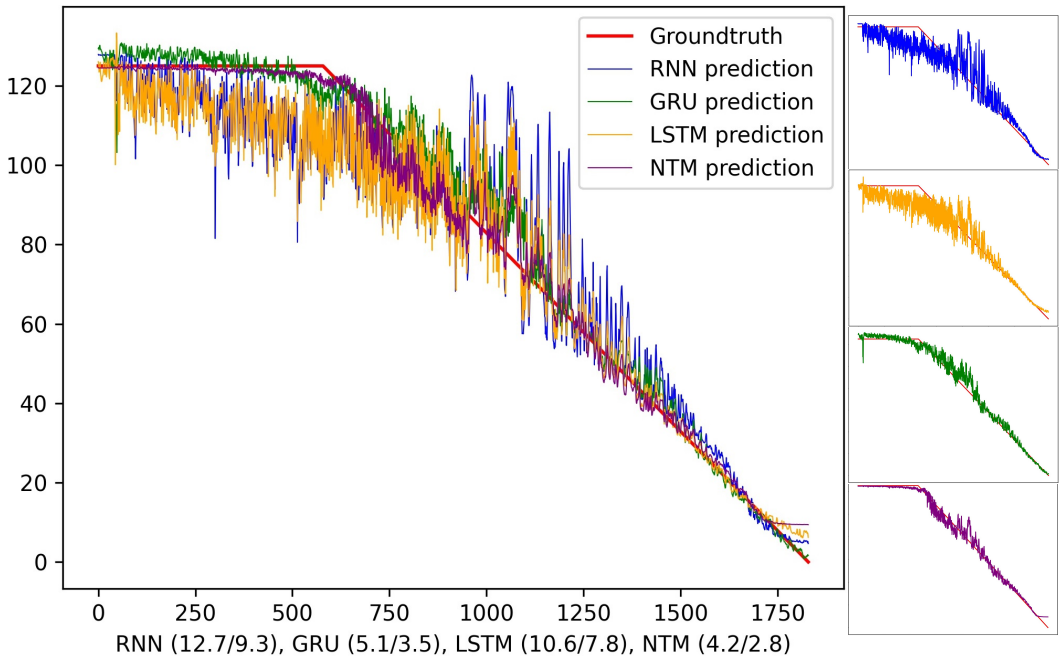


Figure 2: Predictions made by the four models on a full experiment from the test set. RMSE and MAE values are shown below. Best viewed in color.

neural sequence models in a particle filtration system. Quantitatively, we observed that all the models achieve modest prediction accuracy, although the GRU and the NTM perform better than the others. Considering that these models are designed with a lot of care on the technique used to access their memory state, further research is needed to improve the operations used to access and update the memory, while at the same time strive for more attention on the contents put into it. Qualitatively, we presented evidence that all the models are accurate when the fault is close, but they become more and and more uncertain the farther it is. Consequently, neural sequence models may become aware of a fault when it is far too close, therefore it may be difficult to perform a preemptive action. Therefore, future work may also focus on improved training procedures which put more emphasis on detecting when the fault starts to develop, which represents a critical point for a predictive system. Finally, Transformer-based approaches [27] could also be used for future research on RUL estimation.

References

[1] U. S. D. of Energy, Operations & Maintenance Best Practices: A Guide to Achieving Operational Efficiency, Technical Report, 2020. URL: https://www.energy.gov/sites/prod/files/2020/04/f74/omguide_complete_w-eo-disclaimer.pdf.

[2] A. K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, MSSP 20 (2006) 1483–1510.

[3] W. Teng, C. Han, Y. Hu, X. Cheng, L. Song, Y. Liu, A robust model-based approach for bearing remaining useful life prognosis in wind turbines, IEEE Access 8 (2020) 47133–47143.
[4] N. Li, Y. Lei, T. Yan, N. Li, T. Han, A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability, IEEE Transactions on Industrial Electronics 66 (2018) 2092–2101.
[5] C. Ordóñez, F. S. Lasheras, J. Roca-Pardiñas, F. J. de Cos Juez, A hybrid arima–svm model for the study of the remaining useful life of aircraft engines, Journal of Computational and Applied Mathematics 346 (2019) 184–191.
[6] Y. Pan, R. Hong, J. Chen, W. Wu, A hybrid dbn-som-pf-based prognostic approach of remaining useful life for wind turbine gearbox, Renewable Energy 152 (2020) 138–154.
[7] J. Xia, Y. Feng, C. Lu, C. Fei, X. Xue, Lstm-based multi-layer self-attention method for remaining useful life estimation of mechanical systems, Engineering Failure Analysis 125 (2021) 105385.
[8] A. Falcon, G. D'Agostino, O. Lanz, G. Brajnik, C. Tasso, G. Serra, Neural turing machines for the remaining useful life estimation problem, Computers in Industry 143 (2022) 103762.
[9] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[10] A. Graves, G. Wayne, I. Danihelka, Neural turing machines, arXiv preprint arXiv:1410.5401 (2014).
[11] P. Society, Phm society 2020 data challenge, 2020. Https://phm-europe.org/data-challenge-2020 (last accessed: 23 April 2022).
[12] C. S. Byington, S. E. George, G. W. Nickerson, Prognostic issues for rotorcraft health and usage monitoring systems, A Critical Link: Diagnosis to Prognosis (1997) 93.
[13] F. L. Greitzer, L. J. Kangas, K. M. Terrones, M. A. Maynard, B. W. Wilson, R. A. Pawlowski, D. R. Sisk, N. B. Brown, Gas turbine engine health monitoring and prognostics, in: International Society of Logistics (SOLE) 1999 Symposium, Las Vegas, Nevada, volume 30, 1999, pp. 1–7.
[14] N. Bolander, H. Qiu, N. Eklund, E. Hindle, T. Rosenfeld, Physics-based remaining useful life prediction for aircraft engine bearing prognosis, in: Annual Conference of the PHM Society, volume 1, 2009.
[15] A. Coppe, M. J. Pais, R. T. Haftka, N. H. Kim, Using a simple crack growth model in predicting remaining useful life, Journal of Aircraft 49 (2012) 1965–1973.
[16] Y. Zhou, M. Huang, Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and arima model, Microelectronics Reliability 65 (2016) 265–273.
[17] H.-Z. Huang, H.-K. Wang, Y.-F. Li, L. Zhang, Z. Liu, Support vector machine based estimation of remaining useful life: current research status and future trends, Journal of Mechanical Science and Technology 29 (2015) 151–163.
[18] F. O. Heimes, Recurrent neural networks for remaining useful life estimation., in: ICPHM, 2008.
[19] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long short-term memory network for remaining useful life estimation, in: ICPHM, IEEE, 2017, pp. 88–95.
[20] M. Baptista, H. Prendinger, E. Henriques, Prognostics in aeronautics with deep recurrent neural networks, in: PHM Society European Conference, volume 5, 2020.

[21] Q. Luo, Y. Chang, J. Chen, H. Jing, H. Lv, T. Pan, Multiple degradation mode analysis via gated recurrent unit mode recognizer and life predictors for complex equipment, Computers in Industry 123 (2020) 103332.
[22] A. Falcon, G. D'Agostino, G. Serra, G. Brajnik, C. Tasso, A neural turing machine-based approach to remaining useful life estimation, in: ICPHM, 2020, pp. 1–8.
[23] K. Ince, E. Sirkeci, Y. Genç, Remaining useful life prediction for experimental filtration system: A data challenge, in: PHM Society European Conference, volume 5, 2020.
[24] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (1994) 157–166.
[25] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.
[26] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.

Discovering the Unknown Suggestion: a Short Review on Explainability for Recommender Systems

Luca Marconi[1,2], Ricardo A. Matamoros A.[1,2] and Francesco Epifania[2]

[1]Department of Informatics, Systems and Communication, University of Milano - Bicocca, Milan, Italy
[2]Social Things SRL, Italy

Abstract

Artificial Intelligence and in particular machine learning and deep learning models are normally considered to be fast and high performing, but in general there is a lack of transparency and interpretability. The issues related to explainability and its consequences are becoming more and more relevant in the whole broad scenario of Artificial Intelligence. To address this issue, explainable AI emerged, as a set of Artificial Intelligence techniques able to make their own decision more transparent and interpretable, so as to let users understand the specific reasons why the system provided its outcome, decision, or, in the case of recommender systems, its suggestions. Explainable Artificial Intelligence is deeply needed in heterogeneous domains and contexts, as the need for transparency, interpretability and even accountability of the Artificial Intelligence-based systems is a big necessity, as confirmed by the recent right to explanation in the 2018 General Data Protection Regulation by the European Union. Due to the diffusion of recommender systems in many applicative domains and situations in everyday life and business fields, there is an emerging necessity for systems not only able to provide human decision-makers with suggestions and ease the decision-making processes in organizations, but also to give the right motivations of their recommendations. This paper summarizes the results of the study of the state of the art for Explainable Artificial Intelligence for Recommender Systems. We will follow the main reviews in literature to present the main work, kinds of explanainable recommendations and methods.

Keywords

Machine Learning, Recommender Systems, Artificial Intelligence, eXplainable Artificial Intelligence, eXplainable Recommender Systems

1. Introduction

Nowadays, Artificial Intelligence (AI) is becoming more and more important in our professional and personal life. According to the International Data Corporation (IDC) the global investment on AI will reach almost 118 billion U.S. dollars in 2022 and even surpass 300 billion U.S. dollars by 2026 [1]. Moreover, the statistics portal Statista forecasts that revenues from the AI market worldwide will grow from 10.1 billion U.S. dollars in 2018 to 126 billion U.S. dollars by 2026 [2]. Gartner identifies AI as a fundamental technology in most of the the Gartner Top 10 Strategic Technology Trends for 2023 [3]. In the context of the current fourth industrial revolution, overlapping waves of breakthroughs in computing, artificial intelligence, nanotechnology and

l.marconi3@campus.unimib.it (L. Marconi); r.matamorosaragon@campus.unimib.it (R. A. M. A.); r.matamorosaragon@campus.unimib.it (F. Epifania)
0000-0002-0877-7063 (L. Marconi); 0000-0002-1957-2530 (R. A. M. A.); 0000-0002-1957-2530 (F. Epifania)

CEUR Workshop Proceedings (CEUR-WS.org)

material science, 3D-printing, molecular biology (gene sequencing), robotics and other evolving and emergent technologies are reshaping life, business models and ecosystems, according to [4] In this scenario, AI is strongly emerging as transversal and powerful technological paradigm, due to its ability not only to deal with data and big data, but especially because it produces and manages knowledge. Andrew Ng, former chief scientist at Badu and Co-founder at Coursera, said in a keynote speech at the AI Frontiers conference in 2017 that AI is really the new electricity: a disruptive, pervasive and enabling technology, empowering technologies and processes in potentially any field or domain. AI and in particular Machine Learning (ML) and Deep Learning (DL) models are normally considered to be fast and high-performing, but in general there is a lack of transparency and interpretability [5, 6, 7]: it's hard work to get insights from their internal mechanisms when trying to understand why the system provided its outcome or decision. To address this issue, explainable AI (XAI) emerged, as a set of AI techniques able to make their own decision more transparent and interpretable, so as to let users understand the specific reasons why the system provided its outcome, decision, or, in the case of recommender systems, its suggestions [5, 6, 7]. Explainable AI is deeply needed in heterogeneous domains and contexts, as the need for transparency, interpretability and even accountability of the AI-based systems is a big necessity, as confirmed by the recent *right to explanation* in the 2018 General Data Protection Regulation (GDPR) by the European Union [8]. Due to the diffusion of Recommender Systems (RSs) in many applicative domains and situations in everyday life and business fields, there is an emerging necessity for systems not only able to provide human decision-makers with suggestions and ease the decision-making processes in organizations, but also to give the right motivations of their recommendations [9, 10]. A good way to classify eXplainable Recommender Systems (XRSs) was proposed by Zhang et al. in 2014 [11]: it essentially deals with two dimensions: the information source or display style of the explanations (e.g., textual sentence explanation, or visual explanation): it represents the human-computer interaction perspective of explainable recommendation research; the model itself, representing the machine learning perspective of explainable recommendation research. XRSs can be evaluated both by qualitative, user-centered and quantitative evaluation methods. The evaluation can be either related to the performance of the system or to its explainability. In both cases, experiments can be designed wher real users are involved, or without the contribution of human users in the experimental setting. When it comes to evaluating the explainability of the RSs, methods can regard *online*, offline evaluation or *user studies* [12], while other classifications have been proposed in the literature. Overall, the evaluation of explainability suffers from a lack of a unified, precise and widely accepted formal definition of explainability, which implies the use of complementary qualitative and quantitative methodologies to completely strive to evaluate such systems.

This paper summarizes the results of the study of the state of the art for XRSs. We will follow the main reviews in literature to present the main work, kinds of explainable recommendations and methods. Thus, the aim of this paper is to provide a short and compact macro-review of the mostly diffused and used methods and systems reported in the literature. The rise, evolution, adaptation and modifications of models are definitively ongoing processes in the state-of-the-art, thus getting a comprehensive and complete classification is challenging. Given the ongoing evolution of the field, as well as the increasing number of potential applications, the aim of this paper is definitively not to provide a comprehensive and complete review of the large panorama

of such discipline. Rather, we report a limited and carefully circumscribed set of fundamental concepts and methods to get a general picture for later understanding and appreciate the many potential applications and uses of for heterogeneous business and industrial domains. Given the increasing need of explainable, interpretable and thrustworthy systems in business and organizational Therefore, the proposed survey is intended to provide a general overview of the growing scenario of the XRSs, with the aim to help researchers, practitioners and decision-makers to orient themselves to exploit the many potentialities of explainability in recommender systems for business and industrial applications.

2. The context of Explainable AI

Actually, the explanation problem is definitively not new in the literature: the term started to be used in 2004 [13], though the problem itself has existed since the mid-1970s, specifically in the field of expert systems [14], with the first rise of AI in the literature. Though, a greater interest in this theme started to grow with the evolution of machine learning methodologies and techniques, particularly with the growth of its performances in the last years. In the literature, the need for explainable AI is motivated mainly by three reasons: the need for *trust*, for *interaction* and for *transparency* [7]. It's worth to notice that, consequently, explainable AI is strictly related to *responsibility* and transparency [7, 12, 6]. Consequently, explainability is definitively becoming a key conceptual elements for the present and incoming AI systems, as it is also explicitly required in the European General Data Protection Regulation (GDPR) [15], where also the key related concepts of *fairness* and transparency in automated decision-making are highlighted.
In general, XAI is strongly needed for justifying and interpreting the results, so as to ensure that they were not made erroneously [7, 6]. Moreover, the possibility to explain the results would help to improve the way the results are obtained, control the systems dynamics and facilitate new ways to gain knowledge [7].

In a broader perspective, the diffusion of XAI methods and techniques is a crucial step in the current and future evolution of AI systems. Such methods can significantly be grouped into the so-called *third wave* of AI, as defined by DARPA. Thus, XAI strives to realize the big challenge of contextual adaptation, i.e. the construction of progressively explanatory methods for classes of real-world phenomena. The further steps in the design and development of such new and empowered AI systems is the ability to foster continuous learning by the inclusion of synergetic learning techniques, as well as the progressive empowerment of the interaction with human decision-makers [7]. Eventually, the last mile of this ambitious evolution is the quest for reaching or emulating the human intelligence [7].
In the literature, there are different ways to classify the XAI models: among them, there are classifications distinguishing algorithms for their *global* or *local* interpretability, and classifications taking into considerations the differences between *model-specific* or *model-agnostic* methods, thus related with the possibility to apply explainable techniques only to specific models or not. We present the main useful concepts for our work, as well as the main classification reported in the literature, in the case of XRS.

3. Explainable Recommendations

In this context, explainable AI in the field of RSs is aimed at providing intuitive explanations for the suggestions and recommendations provided by the algorithms [12, 19]. Basically they try to address the problem of why certain recommendations are suggested by the models. As they are part of the big world of the XAI, explainable recommendations can either be *model-intrinsic* or *model-agnostic*: in the former case, the output model is intrinsically interpretable, meaning that the decision mechanism is completely transparent providing explainability; in the latter case, instead, the output model provides the so-called *post-hoc* explanations, without any modification of the model itself. It is interesting this two approaches can be conceptually linked to a cognitive psychological root [12]: in this perspective, the model-intrinsic models would be similar to the human minds rational decisions taken after some reasoning process, while the model-agnostic ones would somehow resemble the intuitive ways of deciding, followed by some search of the explanations.

In other words, as in the general case of XAI, XRSs, based on explainability-aware ML techniques, can generally be categorized into two main groups [18]:

1. Systems providing an explanation of their predictions in a way that is interpretable by the user. These types of methods usually only justify their output by the means of an added explanations, but without providing an in depth understanding of the underlying algorithm. This is typical in the case of post-hoc explanations.
2. Explainable systems directly incorporating interpretable models in the construction of the automated systems. Model intrinsic and, specifically, white-box models, such as DTs, can be categorized in this group.

XRSs started formally to be defined, conceived and used in recent years. The term explainable recommendation was formally introduced by Zhang et al. in 2014 [11], but there were earlier works in personalized recommendation research. An extensive review of the first historical stages of explainable recommendation and how it was focused especially on collaborative filtering methods in RSs is in Zhang et al., 2018 [12].

4. Classification of Explainable Recommender Systems

A good way to classify XRS was proposed by Zhang et al. in 2018 [12]: it essentially deals with two dimensions:

1. the *information source* or *display style* of the explanations (e.g., textual sentence explanation, or visual explanation): it represents the human-computer interaction perspective of explainable recommendation research;
2. the model itself, representing the machine learning (ML) perspective of explainable recommendation research.

A somehow generalized taxonomy, focused on the specific classification of interpretability methods, is provided in the review by Linardatos in 2021 [66], which depicts and highlights the major concepts and dimensions involved in the analysis of interpretable models. It proves to be useful to get a complete picture of the most significant conceptual perspectives involved.

4.1. Information Source for Explanations

The first dimension of this classification model is the *information source* for explanations, also called *display style*: namely, explanations are pieces of information related to the recommendations given by the algorithm. Recommendations can come from different information sources and can be displayed in several ways: some examples include textual sentences, word clouds or visual explanations. In the following paragraphs we provide a short summary of the different types of recommendation explanations and we give some examples of relevant related work.

Explanations based on Relevant Users or Items
This comes from the first stages of recommendation explanation research. User-based explanations are especially used by collaborative filtering RSs, thus when the recommendation is based on the ratings or interests of "similar" users. [12] reports the example of Herlocker et al. [20], comparing the effectiveness of different display styles for explanations in user-based collaborative filtering. Instead, for item-based explanations, the measure of similarity comes from the user's past liked items. Zhang and Chen [12] argue that relevant-item explanations are more intuitive for users than user-based explanations due to the familiarity of the user with the items more than with other potential users: nevertheless, this problem could be solved by another kind of explanations, the so-called *social explanation*.

Feature-based Explanations
This kind of explanations are especially related to content-based recommendation methods [12]. CB-RSs elaborate suggestions according to a specific match between users' proles and content features of candidate items. In this case it is more intuitive to base the recommendations on the specific features of the items, and then to display them in the best explanation style: for example in Vig et al. [21] the recommendations are provided adopting movie tags as features.

Textual Sentence Explanations
This kind of explanations is very useful for getting relevant benefits from user-generated content, such as e-commerce reviews and social media posts [12]. Sentences could come from pre-defined templates or be directly generated based on natural language generation models. Zhang and Chen [12] classify such approaches between *aspect-level* and *sentence-level* approaches, based on the display style of the explanations. It is worth to notice some sort of similarity between aspect-level textual explanations and feature-based explanation: though,in

the former case the aspects addressed are usually not directly available in an item or user prole. In fact,they come from textual information usually related to and users opinions or textual feedback about specific items. This is what happens in [11], where explanations are presented as aspect-opinion wordclouds based on large-scale user reviews.

Visual Explanations
They help users to get precise and intuitive suggestions. Visually explainable recommendation are still a relatively new topic in research, thus the integration of visual information and images into recommender systems is far from being optimized in terms of both explainability and performance [12]. For example, in [16] visually explainable recommendation are based on personalized region-of-interest high-lights.

Social Explanations
The involvement of friends in the recommendation process implies a higher level of personalization into the suggestions themselves, while solving the typical trustworthiness and privacy problems of relevant-user explanations. Examples include the studies and applications in music [17], and in product recommendations [22].

4.2. Explainable Recommendation Models

The second dimension of the classification model proposed by Zhang and Chen [12] regards the specific models used for producing the explanations: namely, explanations given by different types of algorithms. As always, explainable recommendations can either be *model-intrinsic* or *model-agnostic*. In the following paragraphs we provide a short summary of the major types of explainable recommendation models and we give some examples of relevant related work.

Factorization Models
Latent Factor Models based on Matrix Factorization is a classical ML model for recommender systems [26]. It learns latent factors to predict the missing ratings in a user-item rating matrix. Factorization models for explainable recommendations have been proposed in order to explain the specific latent factor acting user decisions. As an example, Explicit Factor Models [11] links each latent dimension of matrix factorization with an explicit feature among the users favorite ones. Thus, it can provide explicit recommendations based on the features. Instead, other studies [27], focus on model-based approaches to generate relevant-user or relevant-item explanations based on the user-item rating matrix.

Topic Modeling
This kind of explainable recommendations is still based on text information. Topic modeling

refers to a general methodology to classify semantics in documents according to topics clusters. Explanations are generally displayed in the form of topical word clouds. McAuley and Leskovec [28] proposed to use a model based on latent factor analysis to understand hidden topics learned from reviews. Other studies [29] focused on other probabilistic graphic models. Wu and Ester [29] created an hybrid model based on both collaborative filtering and aspect-based opinion mining. The algorithm analyses users preferences on item aspects according to reviews and then predicts the users ratings on different ones.

Graph-based Models

Graphs help to define relevant relations among information, so they can be specifically useful to represent user-user or user-item relationships, especially in social recommendation scenarios. For example, Park et al. [22] use a graph-based explainable recommendation algorithm for providing interpretable suggestions thanks to rating and similar users. Other authors exploited other kinds of graphs: in He et al. [30] a tripartite graph structure allows to model user-item-aspect relations where an aspect is an item feature generally taken from user reviews. These relations are constructed for the possible recommendations and then aspects are ranked and explanations are given to the top-ranked aspects matching the target user and the recommended item. Heckel et al. [31] instead created explainable recommendations thanks to over-lapping co-clustering based on user-item bipartite graph [12]: this approach allows to exploit both clusters of similar users and of items with similar properties.

Deep Learning

Given the higher and higher importance of deep learning techniques, there are many studies and experiments to adopt a huge variety of them in the explainable recommendations scenario. In Seo et al. [32] user preferences and item properties are represented through convolutional neural networks upon review text, so as to attribute specific weights to words in the text and highlight the relevant ones to provide explainable recommendations. Among the other various typologies of neural networks used, it is worth to cite the work by Chen et al. [33], where explainable sequential recommendation are extracted due to memory networks: they have memory over previous items chosen, so each item in the users interaction history is in a memory slot and predictions of the new behaviors can be made and explained subsequently, so as to directly show the way the users previous choices influenced new predictions. That implies the possibility to set dynamic explainable recommendations. Another interesting approach comes from capsule networks, namely neural networks empowered with capsule structures to manage hierarchies. Li et al. [34] use capsule networks to model item aspects and users viewpoints as logic units, so as to get the users' rating behaviors. Then, the algorithm, for each user-item pair, extracts the informative logic units from the reviews so as to infer their corresponding sentiments.

Knowledge Graph-based
As one of the classical ways to manage knowledge, knowledge graphs can be used for providing better explanations for the recommended items thanks to their information about users and items. Catherine et al. [35] proposed a method to provide explanations and recommendation after producing a rank of the items thanks to information found in knowledge graphs. Instead, Ai et al. [36] constructed a user-item knowledge graph, so as to get recommendations for a user as the most similar item under the ”purchase” relation. In this way, they can establish a series of relations between users and items to orient and explain recommendations.

Data Mining
Among the various possibilities and techniques, Zhang and Chen [12] report that the most frequently used one is association rule mining. As an example, Davidson et al. [23] introduced the YouTube video recommendation system, adopting association rule mining to create associations between couples of videos co-watched within the same session. Then, explanations are given considering the seed video and the the association rules themselves. The approach for transparent, scrutable, and explainable recommendations suggested by Balog et al. [25] is particularly interesting: given a set of tags or keywords characterizing user preferences, they aimed at inferring preferences and recommendations by aggregating over items associated with a tag. Consequently, item recommendations can be both transparent and explainable. They chose to provide recommendations through sentence-level textual explanations, allowing users to provide feedback on clear and scrutable suggestions. It is worth to mention that this approach is a framework, which can be generalized to different machine learning models.

Model Agnostic and Post Hoc
These approaches are typically used when it is to difficult to include the explanability in the recommendation model itself. Then, after the recommendations have been provided, an explanation model generates the explanations according to the previously created recommendations. As an example exploiting a data mining technique (thus related to the previous paragraph), Peake and WanH [37] proposed an association rule mining approach. The method considers the users'transaction history to explain the recommendation: namely, the association rules help to associate the recommendations themselves with the users' previous choices, thus providing explanations to the recommendations.
Overall, the literature makes a clear distinction among models that are interpretable by design, and those that can be explained by means of external XAI techniques. This duality could also be regarded as the difference between interpretable models and model interpretability techniques; a more widely accepted classification is that of transparent models and post-hoc explainability.

In particular, local interpretation methods explain predictions individually from each other. Among these [38] we have:

1. Individual conditional expectation (ICE) [39] curves underlie partial dependence plots (PDPs) and describe how the change in a feature affects the change in the prediction.

2. Local surrogate models, as the Local Interpretable Model-agnostic Explanations model (LIME) [40] explain a prediction by replacing the complex model with an interpretable local surrogate.
3. Scoped rules (anchors) [41] are rules that describe which feature values allow the prediction to be fixed.
4. Counterfactual explanations [42, 43] explain a prediction by examining which features should be changed to achieve the desired prediction.
5. Shapley values [44] are an attribution method that assigns prediction equally to individual features.
6. SHAP (SHapley Additive exPlanations) [45] is another computation method for Shapley values, but unlike these it proposes global interpretation methods based on combining Shapley values across data.

4.3. Intrinsic, interpretable, white-box models

Hereinafter, we recall the main interpretable models. We focus on such macro-category of models due to the fact that the chosen approach for our XRS, DT models, is actually interpretable. Therefore, we synthetically show the main characteristics of these models, as well as their main advantages and disadvantages. Finally, we sketch the main motivations that lead us to orient ourselves towards a decision-tree approach.

Linear regression

A linear regression model predicts the target as a weighted sum of the feature inputs [38]. Linear regression are particularly useful and significant in practice for their linearity. They have long been used by statisticians, computer scientists, mathematicians and practitioners in general [38]. They are usually exploited to model the dependence of a regression target y on some features x, and the predicted outcome of an instance is a weighted sum of its features, where the optimal weights can be estimated by several methods.
The main advantages of such methods are its linearity and the modeling of the predictions as a weighted sum makes it transparent how predictions are produced [16]. The modeling of the predictions as a weighted sum makes guarantees transparency on how predictions are created. From the mathematic point of view, they are widely accepted and diffused methods among practitioners, and high level of collective experience and expertise is available in the scientific community [38].
Nevertheless, the are only useful for representing linear relationships, while any required nonlinearity or interaction has to be hand-crafted and explicitly provided to the model [38]. Moreover, they often have no good predictive performance, due to restricted ability to represent reality in a purely linear way [38]. Finally, there is a possible unintuitive interpretation of weights, due to the correlations and interactions with all the other involved features [38].

Logistic regression

Linear regression models the probabilities for classification problems with two outcomes. It's

an extension of the linear regression model for classification problems [38]. Therefore, it shows similar advantages and disadvantages than the linear regression models [38]. Also logistic regression has been widely used by practitioners in different domains and application fields, and it has issues with restrictive expressiveness and with dealing with interactions, as well as with limitations in predictive performance. Moreover, logistic regression can suffer from *complete separation*, namely the impossibility tobe trained in the case where there is a feature that would perfectly separate the two classes.

GLM and GAM

Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) are heterogeneous generalization models of regression, useful for modeling real-life situations. They can be applied in situations where the classical regression approaches fail or its assumptions are violated [38]. In the case of GLM, they can be applied where the input features do not follow a Gaussian distribution, which concretely happens in many cases in reality [38]. Instead, GAMs deal with the cases of nonlinearities, not tackled by the classical linear models. GAMs relax the restriction that the relationship must be a linear weighted sum, assuming that the outcome can be modeled by arbitrary functions that can be involved for each features [38]. Then, such models are generally .

In general, these models are highly flexible and useful for making predictions and inferences in many application cases and contexts. These methods are highly diffused in the scientific community and updated methods are often released allowing to make inferences for heterogeneous problems and applications [38]. Though, such models suffer from a significant reduced interpretability, as compared with the classical linear models, and they strongly rely on assumptions about the data generating process, which have to be respected for the validity of the model and its interpretation of the weights [38].

Decision-trees

Already previously introduced in this Chapter, DT models are useful for solving many of the presented issues, especially in the case of linear regression and logistic regression models, which have problems in situations where the relationship between features and outcome is nonlinear or where features interact with each other [38]. Tree-based models work through an iterative process of multiple splitting of the dataset, according to certain cutoff values in the features. Thus, they are inherently interpretable due to the tree structure itself, while they are also able to capture interactions between features in the data, as well as to effectively explain and visualize their output results. The main disadvantages are related to their inability to deal with linear relationships, as well as their lack of smoothness and unstability [38]. Moreover, their interpretability is reduced in the case of a significant increase in the tree depth [38].

Decision rules

Decision rules are probably the most interpretable models. IF-THEN statement consist of a

condition (antecedent) and a prediction and, in simple cases, they semantically resembles natural language [16]. Then, they are usually easy to interpret, expressive, robust and compact [38]. Nevertheless, in the literature they are used only for classification [38], resulting in applications for restricted classes of problems. Moreover, they necessarily require categorical features and, as in the case of decision-trees, they have issues in describing linear relationships.

5. Conclusions

In this study we strived to provide an overview of Explainable AI in the field recommender systems. We are aware that many other issues could have been addressed, specifically regarding the pros and cons of the wide set of methods in the literature, as well as the many evaluation techniques of both RS and explainability. As a general consideration to conclude our study, we definitively agree that the evolution of such systems necessarily involve a synergy between the empowerment of the models' performances and the emergent human-AI interaction perspective. We also conclude that much more work and effort should be dedicated to search and adopt a widely accepted, pre-defined and formally circumscribed definition of explainability and its related concepts. While there are several studies proposing both qualitative and quantitative definitions, it should be necessary to both choose and apply them to the field of recommender systems: this investigation could be the aim of a future study. Indeed, there are many further challenges and possible future directions to explore for this fascinating topic: among them, the issues related to the difficulties in quantiatively and formally measuring explainability, which will be a key step to reach and exploit the full potentialities of explainable and interpretable recommender systems for heterogeneous business and industrial domains. Moreover, the rise of explainable intelligent recommender systems will increasingly require to further investigate the broader impact of explainability on decision-making processes, so as to understand their full influence in organizational context and applications. We hope that our work can contribute to help researchers, scholars and practitioners to understanding the concept of explainable recommendation, the main approaches in the literature and their potentialities for business or industrial applications.

References

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[2] Liu, Shanhong. "Revenues from the Artificial Intelligence (AI) Software Market Worldwide from 2018 to 2025." (2020).

[3] CeArley, David, et al. "Top 10 strategic technology trends for 2018." The Top 10 (2016): 1-246.

[4] Schwab, Klaus. The fourth industrial revolution. Currency, 2017.

[5] Gunning, David. "Explainable artificial intelligence (xai)." Defense advanced research projects agency (DARPA), nd Web 2.2 (2017): 1.

[6] Došilović, Filip Karlo, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A

survey." 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO). IEEE, 2018.
[7] Adadi, Amina, and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)." IEEE access 6 (2018): 52138-52160.
[8] Selbst, Andrew, and Julia Powles. "“Meaningful Information” and the Right to Explanation." Conference on Fairness, Accountability and Transparency. PMLR, 2018.
[9] Vultureanu-Albişi, Alexandra, and Costin Bădică. "Recommender systems: an explainable AI perspective." 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). IEEE, 2021.
[10] Sardianos, Christos, et al. "The emergence of explainability of intelligent systems: Delivering explainable and personalized recommendations for energy efficiency." International Journal of Intelligent Systems 36.2 (2021): 656-680.
[11] Zhang, Yongfeng, et al. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis." Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. 2014.
[12] Zhang, Yongfeng, and Xu Chen. "Explainable recommendation: A survey and new perspectives." Foundations and Trends® in Information Retrieval 14.1 (2020): 1-101.
[13] Van Lent, Michael, William Fisher, and Michael Mancuso. "An explainable artificial intelligence system for small-unit tactical behavior." Proceedings of the national conference on artificial intelligence. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.
[14] Moore, Johanna D., and William R. Swartout. Explanation in expert systemss: A survey. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 1988.
[15] Goodman, Bryce, and Seth Flaxman. "European Union regulations on algorithmic decision-making and a “right to explanation”." AI magazine 38.3 (2017): 50-57.
[16] Chen, Xu, et al. "Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation." Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019.
[17] Sharma, Amit, and Dan Cosley. "Do social explanations work? Studying and modeling the effects of social explanations in recommender systems." Proceedings of the 22nd international conference on World Wide Web. 2013.
[18] Abdollahi, Behnoush, and Olfa Nasraoui. "Transparency in fair machine learning: the case of explainable recommender systems." Human and machine learning. Springer, Cham, 2018. 21-35.
[19] Biran, Or, and Courtenay Cotton. "Explanation and justification in machine learning: A survey." IJCAI-17 workshop on explainable AI (XAI). Vol. 8. No. 1. 2017.
[20] Herlocker, Jonathan L., Joseph A. Konstan, and John Riedl. "Explaining collaborative filtering recommendations." Proceedings of the 2000 ACM conference on Computer supported cooperative work. 2000.
[21] Vig, Jesse, Shilad Sen, and John Riedl. "Tagsplanations: explaining recommendations using tags." Proceedings of the 14th international conference on Intelligent user interfaces. 2009.
[22] Park, Haekyu, et al. "Uniwalk: Explainable and accurate recommendation for rating and

network data." arXiv preprint arXiv:1710.07134 (2017).
[23] Davidson, James, et al. "The YouTube video recommendation system." Proceedings of the fourth ACM conference on Recommender systems. 2010.
[24] Hussein, Doaa Mohey El-Din Mohamed. "A survey on sentiment analysis challenges." Journal of King Saud University-Engineering Sciences 30.4 (2018): 330-338.
[25] Balog, Krisztian, Filip Radlinski, and Shushan Arakelyan. "Transparent, scrutable and explainable user models for personalized recommendation." Proceedings of the 42nd international acm sigir conference on research and development in information retrieval. 2019.
[26] Mehta, Rachana, and Keyur Rana. "A review on matrix factorization techniques in recommender systems." 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA). IEEE, 2017.
[27] Abdollahi, Behnoush, and Olfa Nasraoui. "Using explainability for constrained matrix factorization." Proceedings of the eleventh ACM conference on recommender systems. 2017.
[28] McAuley, Julian, and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text." Proceedings of the 7th ACM conference on Recommender systems. 2013.
[29] Wu, Yao, and Martin Ester. "Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering." Proceedings of the eighth ACM international conference on web search and data mining. 2015.
[30] He, Xiangnan, et al. "Trirank: Review-aware explainable recommendation by modeling aspects." Proceedings of the 24th ACM international on conference on information and knowledge management. 2015.
[31] Heckel, Reinhard, et al. "Scalable and interpretable product recommendations via overlapping co-clustering." 2017 IEEE 33rd International Conference on Data Engineering (ICDE). IEEE, 2017.
[32] Seo, Sungyong, et al. "Interpretable convolutional neural networks with dual local and global attention for review rating prediction." Proceedings of the eleventh ACM conference on recommender systems. 2017.
[33] Chen, Xu, et al. "Sequential recommendation with user memory networks." Proceedings of the eleventh ACM international conference on web search and data mining. 2018.
[34] Li, Chenliang, et al. "A capsule network for recommendation and explaining what you like and dislike." Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. 2019.
[35] Catherine, Rose, et al. "Explainable entity-based recommendations with knowledge graphs." arXiv preprint arXiv:1707.05254 (2017).
[36] Ai, Qingyao, et al. "Learning heterogeneous knowledge base embeddings for explainable recommendation." Algorithms 11.9 (2018): 137.
[37] Peake, Georgina, and Jun Wang. "Explanation mining: Post hoc interpretability of latent factor models for recommendation systems." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018.
[38] Molnar, Christoph. Interpretable machine learning. Lulu. com, 2020.
[39] Goldstein, Alex, et al. "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." journal of Computational and Graphical Statis-

tics 24.1 (2015): 44-65.
[40] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
[41] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations." Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.
[42] Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." Harv. JL & Tech. 31 (2017): 841.
[43] Dandl, Susanne, et al. "Multi-objective counterfactual explanations." International Conference on Parallel Problem Solving from Nature. Springer, Cham, 2020.
[44] Shapley, Lloyd S. "A value for n-person games." Classics in game theory 69 (1997).
[45] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Advances in neural information processing systems 30 (2017).
[46] Gorwa, Robert, Reuben Binns, and Christian Katzenbach. "Algorithmic content moderation: Technical and political challenges in the automation of platform governance." Big Data & Society 7.1 (2020): 2053951719897945.
[47] Gillespie, Tarleton. "Content moderation, AI, and the question of scale." Big Data & Society 7.2 (2020): 2053951720943234.
[48] Bonhard, Philip, and Martina Angela Sasse. "'Knowing me, knowing you'—Using profiles and social networking to improve recommender systems." BT Technology Journal 24.3 (2006): 84-98.
[49] Subramanyam Kalyan, Katikapalli, Ajit Rajasekharan, and Sivanesan Sangeetha. "AMMUS: A Survey of Transformer-based Pretrained Models in Natural Language Processing." arXiv e-prints (2021): arXiv-2108.
[50] Nuruzzaman, Mohammad, and Omar Khadeer Hussain. "A survey on chatbot implementation in customer service industry through deep neural networks." 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE). IEEE, 2018.
[51] Garg, Vishal, and Ritu Tiwari. "Hybrid massive open online course (MOOC) recommendation system using machine learning." International Conference on Recent Trends in Engineering, Science & Technology-(ICRTEST 2016). IET, 2016.
[52] Young, Tom, et al.. "Recent trends in deep learning based natural language processing." ieee Computational intelligenCe magazine 13.3 (2018): 55-75.
[53] Hu, Dichao. "An introductory survey on attention mechanisms in NLP problems." Proceedings of SAI Intelligent Systems Conference. Springer, Cham, 2019.
[54] Alyafeai, Zaid, Maged Saeed AlShaibani, and Irfan Ahmad. "A survey on transfer learning in natural language processing." arXiv preprint arXiv:2007.04239 (2020).
[55] Vedantam, Vamsi Krishna. "The Survey: Advances in Natural Language Processing using Deep Learning." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12.4 (2021): 1035-1040.
[56] Zaslavsky, Arkady, Charith Perera, and Dimitrios Georgakopoulos. "Sensing as a service and big data." arXiv preprint arXiv:1301.0159 (2013).
[57] Bala, K., et al.. "Chat-bot for college management system using AI." International Research

Journal of Engineering and Technology 4.11 (2017): 2030-2033.
[58] O'Leary, Daniel E. "Using AI in knowledge management: Knowledge bases and ontologies." IEEE Intelligent Systems and Their Applications 13.3 (1998): 34-39.
[59] Lalwani, Tarun, et al.. "Implementation of a Chatbot System using AI and NLP." International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3 (2018).
[60] Mendes Rodrigues, Eduarda, and Natasa Milic-Frayling. "Socializing or knowledge sharing? Characterizing social intent in community question answering." Proceedings of the 18th ACM conference on Information and knowledge management. 2009.
[61] Rajabion, Lila, et al.. "A new model for evaluating the effect of cloud computing on the e-learning development." Journal of Workplace Learning (2019).
[62] Gjorgjevska, Emilija, and Georgina Mirceva. "Content Engineering for State-of-the-art SEO Digital Strategies by Using NLP and ML." 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2021.
[63] Lee, Gichang, et al.. "Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network." Knowledge-Based Systems 152 (2018): 70-82.
[64] Lee, Joohong, Sangwoo Seo, and Yong Suk Choi. "Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing." Symmetry 11.6 (2019): 785.
[65] Wang, John, et al.. "Cloud Computing, Intelligent Business Process Management and Artificial Intelligence." international journal of data analysis and information systems 9 (2017): 1-12.
[66] Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. Entropy, 23(1), 18.

Causal Text-to-Text Transformers for Water Pollution Forecasting

Kevin Roitero[1,*], Cristina Gattazzo[2], Andrea Zancola[2], Vincenzo Della Mea[1] and Stefano Mizzaro[1]

[1]*University of Udine, Italy*
[2]*AcegasApsAmga SpA, Hera Group, Italy*

Abstract

We propose a novel approach based on large language causal models to perform the task of time-series forecasting, and we use the proposed approach to effectively forecast the concentration of polluting substances in a water treatment plant; we address both short- and mid-term forecasting. As opposed to the classical state-of-the-art approaches for time-series forecasting, that handle numerical and categorical features following a standard deep learning approach, we transform the input features into a textual form and we then feed them to a standard causal model pre-trained on natural language tasks. Our empirical results provide evidence that large language models are more effective than state-of-the-art forecasting systems, and that they can be practically used in time-series forecasting tasks. We also show promising results on zero-shot learning. The results of this study open up to a wide range of works aimed at predicting future temporal values by leveraging natural language paradigms and models.

Keywords

Deep learning, Time-series forecast, Language models

1. Introduction

Water treatment plants, and in particular drinking water systems make use of different water treatment methods in order to serve safe drinking water to the population. Such systems use a series of treatments steps that transform the source water that enters the systems from river, lakes, etc. to tap water. To ensure that the water that leaves the system is drinkable and safe for the population, water treatment plants constantly monitor the concentration of polluting substances into the water, making use of specific instruments and techniques, such as the ion chromatography, an analytical separation technique based on ionic interactions. Such a technique separates ions and polar molecules based on their affinity and is able to carry out both qualitative and quantitative determinations. The field of application of ion chromatography is very broad, and the most common analyses with this technique concern water related analysis such as drinking water, sea water, waste water, rain water, determination of traces in electronics and power plants, quality control and analysis of impurities, etc.

AIABI'22: 2nd Italian Workshop on Artificial Intelligence and Applications for Business and Industries, November 28 – December 2, 2022, University of Udine, Udine, Italy

*Corresponding author.
kevin.roitero@uniud.it (K. Roitero); cgattazzo@acegasapsamga.it (C. Gattazzo); azancola@acegasapsamga.it (A. Zancola); vincenzo.dellamea@uniud.it (V. D. Mea); stefano.mizzaro@uniud.it (S. Mizzaro)

In this paper we deal with the analysis carried out by a ion chromatograph instrument located in the water treatment plant of Randaccio, which serves the city of Trieste. The instrument we deal with is managed by the Laboratory of AcegasApsAmga which makes the data available through the company data transmission network. At the laboratory the data are: downloaded, validated, uploaded to the internal system, used to create a report, evaluated. The created reports are then made available.

The instrument analyzes different substances; in this paper we focus on three of them which are important for the water treatment system: chloride, nitrate, and sulfate. The instrument monitors the concentration values of such substance approximately every 1h 30min, and collects a total of approximately 14 samples per day. Multiple samples are then joined together to form a time-series. The trend of the measured values in the time-series is constantly monitored and, if predefined patterns emerge (e.g., the value of a polluting substance increases), practical countermeasures are applied to the water plant, as for example the decision to exclude an intake point from the system and switch to another one where pollution levels are lower. It must be noted that such practical counter measures require a certain amount of time to be implemented. For this reason, the domain experts are interested in predicting in advance future values and trends for the observed substances.

In this paper we propose an effective practical methodology to reliably forecast the concentration of the polluting substances monitored by the ion chromatograph in the water treatment plant; our approach is based on transforming the input features from the time-series into a textual form and we then feed them to a standard causal model pre-trained on natural language tasks and asking the model to forecast the concentration of the substances for subsequent time steps. We validate our approach on real data coming from the treatment plant, providing also promising results on domain adaptation via zero-shot learning. Empirical evidence shows that our approach is more effective than state-of-the-art approaches for both short- and mid-term forecast.

2. Dataset

In the following we detail the dataset considered for the experimental part, used to validate the proposed approach. We consider the three substances (i.e., chloride, nitrate, and sulfate) monitored by the ion chromatography system which are modeled in the form of a time-series. It should be noted that the instrument monitors more than 3 substances, but those can not be interpreted as time-series, since their values assume the value of 0 for more than 95% of the observations. Our dataset is composed by observations made over a one year period, specifically between May, 2021 and May, 2022. A sample of the time-series for the three substances used in this work is shown in Figure 1 (first row). By inspecting the time-series behavior for those substances, we notice some interesting patterns.

First, we see that there are non negligible missing observations. The law requires minimum quality and safety levels, which are verified both internally by the company and externally by the health authority. The chromatograph used for collecting the dataset is not used for the production of required data, but it is part of an experimental setup aimed at verifying its usefulness in addition to formal measurements. As such, it is not always working, and this

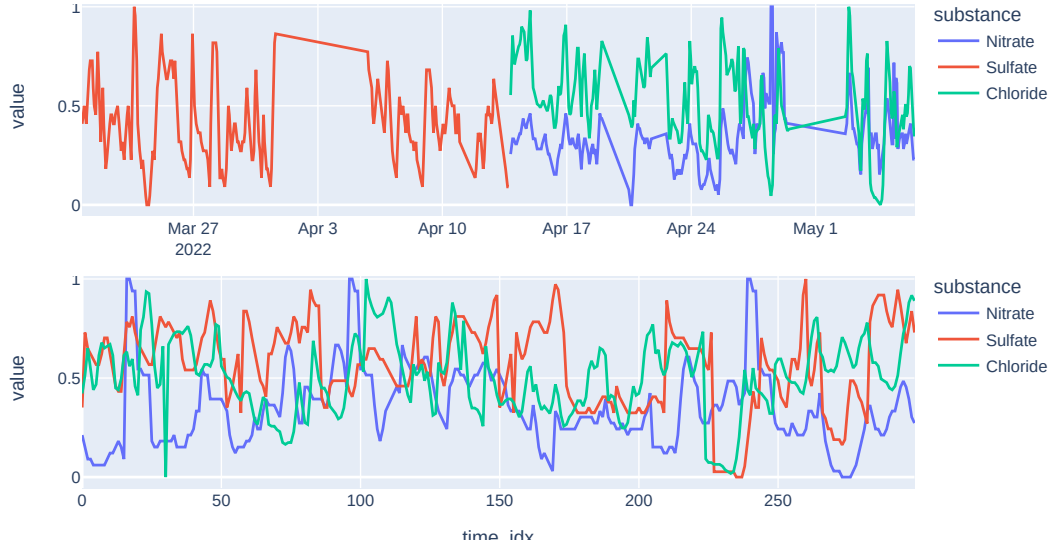


Figure 1: Time-series for the three substances before sampling (first row), and after the sampling process (second row). X-axis has been cut, and values are scaled in 0–1.

justifies missing samples. Then, we also notice that the monitoring period is not the same for all three substances, and in some periods the overlap is minimal or not-existent. In other words, when an observation is made for a substance, there is not guarantee that an observation will be available for one or both of the other substances for the corresponding time.

To overcome these issues, and transform the input time-series into a set of new ones without gaps, in a first pre-processing step we simply remove the missing observations, ending up with a smaller dataset having about $2,800$ observations for each substance, on average 14 per day. Then, we check for seasonality effects by running both the seasonal decomposition using moving averages and Season-Trend decomposition using LOESS[1] [1] analyses. We found no evidence of seasonality or significative trend effects. This is also confirmed by the domain experts, which also confirmed that there is no interaction or dependence between the three substances (e.g., the pattern of chloride is not influenced by the temporal pattern of nitrate and sulfate, and the same holds for the other substances); thus, it does not make sense to use one time-series as feature to predict the others. In other words, we can frame the context as being a univariate time-series.

Then, to remove the bias introduced by the removal of missing values, we transform the dataset as follows. First, we compute for each substance the set of dates for which we have observations. Then, we random sample with replacement from the set of days and we concatenate the result. Let us make it clear by providing an example; if we suppose to have 10 days (i.e., $d_1, \ldots, d_{10}$) and having missing values for days 2, 6, 7, and 9, the initial dataset can be represented as: $d_1, d_3, d_4, d_5, d_8, d_{10}$, while the resulting dataset can be represented as: $d_1, d_3, d_4, d_3, d_1, d_8, \ldots, d_4$. Then, we form a training, validation, and test sets, by paying attention that if a day is present in the training set it can not be included in the test set. The final

[1] see https://www.statsmodels.org/dev/tsa.html.

dataset is obtained by sampling approximately observations from 8600 days, and is composed as follows: $93,183$ observations in the training set, $4,905$ in the validation set, and $24,522$ in the test set. It should be noted that the sampling process performed is used only as a data augmentation technique to train the considered algorithms, and it does not affect the practical application of the proposed approach. A sample of the resulting dataset is shown in Figure 1 (second row).

3. Related Work

3.1. Time Series Forecast

The forecast of substances concentration that we deal with in the paper is related to general time-series forecasting research. State-of-the-art deep learning approaches designed for time-series forecasting are based on Recurrent Neural Networks (RNN) and their variations such as Long Short Term Memory (LSTM) networks [2] and Gated Recurrent Units (GRU) [3]. RNNs are a particular neural network architecture where the output of previous steps is fed as input to the current step. Such architecture is well suited to model scenarios where the prediction of the current value (e.g., the next word in a sentence or the next value of a time-series), is dependent on previous observations. More recently, architectures based on transformers as addition to classical architectures [4, 5] have been proposed [6].

While some successful attempt of adopting vanilla transformer architectures standalone [7] or in conjunction with other architectures [8] has been made in the setting of human mobility forecast where many contextual features are available, plain transformers and in particular causal models are quite new to the task of time-series forecasting, especially in the univariate setting and/or when there is a lack of context features, such as in the case investigated in this paper. This is primarily due to two main reasons [7], the absence of large-scale training data needed to develop pre-trained models, and the requirement for unique designs needed to capture domain-specific time-series features, such as seasonality effects.

In this work we propose an approach based on causal language models, and compare the proposed approach to state-of-the-art time-series forecasting models.

3.2. Large Language Models

In recent years, rapid advancements in the self-supervised learning paradigm joint with the success of the transformer-based architectures [9] contributed to the spread of general pre-trained and domain-specific fine-tuned models that demonstrated their effectiveness on a large variety of natural language processing (NLP) tasks; famous examples include BERT [10], a large masked language model pre-trained on English and Multi-language corpora which can be fine-tuned to a huge variety of tasks due to the learned language understanding ability. Masked language models are trained by randomly masking a percentage (e.g., 15%) of the input tokens and training the model to predict the masked tokens. The model loss is computed by considering the cross entropy loss between the logits of the model and the vocabulary tokens.

Opposed to masked language models, another popular set of transformer based models are causal models, as for example T5 [11]. Masked language models are trained to predict the

masked tokens in a sentence, and by doing so they leverage a bidirectional representation schema, because the representation of the masked tokens is learned based on the tokens that occur to the left and to the right of the masked part; the analogy for this representation schema is a "fill-in-the-blanks" problem statement. On the contrary, causal models predict the masked token in a given sentence but, unlike masked models, a causal model is allowed to just consider tokens that occur to the left of the masked set of tokens, thus leveraging a unidirectional representation schema. As result, such models are used in the case of generative tasks, where they are trained to predict the next token (or set of tokens) in a sentence based on the previous observed ones. As well as masked language models, the causal loss is computed by considering the cross entropy loss between the predicted token against the tokens in the vocabulary.

In this paper, due to the their intrinsic nature of being trained to predict the next value in a sequence based on the occurrence of past values, i.e., being that exactly the classical way of representing and modeling a time-series, in the following we base our solution on causal models, and specifically on the T5 model.

4. Methodology

4.1. Problem Formulation

We are interested, given a set of past observations of the substance concentration as measured by the ion chromatography, to predict the value for the substance for the subsequent timestamps. More in detail, we feed the models with 56 past timestamps, corresponding approximately to the measures obtained in the past 4 days, and we forecast two different future time steps: the next value in the time-series (t+1) which corresponds to a short-term prediction, as well as a mid-term prediction that allows domain experts to take practical countermeasures and apply them to the clean water plant, t+14 (i.e., one day forecast).

4.2. Metrics

To evaluate the effectiveness of the proposed approach, we rely on the following metrics used to evaluate the effectiveness of time-series forecasting methods: Mean Absolute Error (MAE), defined as the sum of absolute errors divided by the sample size, Max Error (ME), computed by considering the maximum of all absolute differences between the target and the prediction, and Root Mean Squared Error (RMSE), computed by considering the standard deviation of the residuals (i.e., prediction errors).

4.3. Deep Learning Methods

We consider the following state-of-the-art deep learning based methods: Long Short-Term Memory network (LSTM) [12], a sequence to sequence model which employs an architecture that allows the network to remember values over arbitrary intervals, thus showing a relative insensitivity to gap length between observations. Gated Recurrent Unit network [13] (GRU), a LSTM variation designed to solve the vanishing gradient problem, which makes use of the update gate and the reset gate to decide which part of information should be passed trough

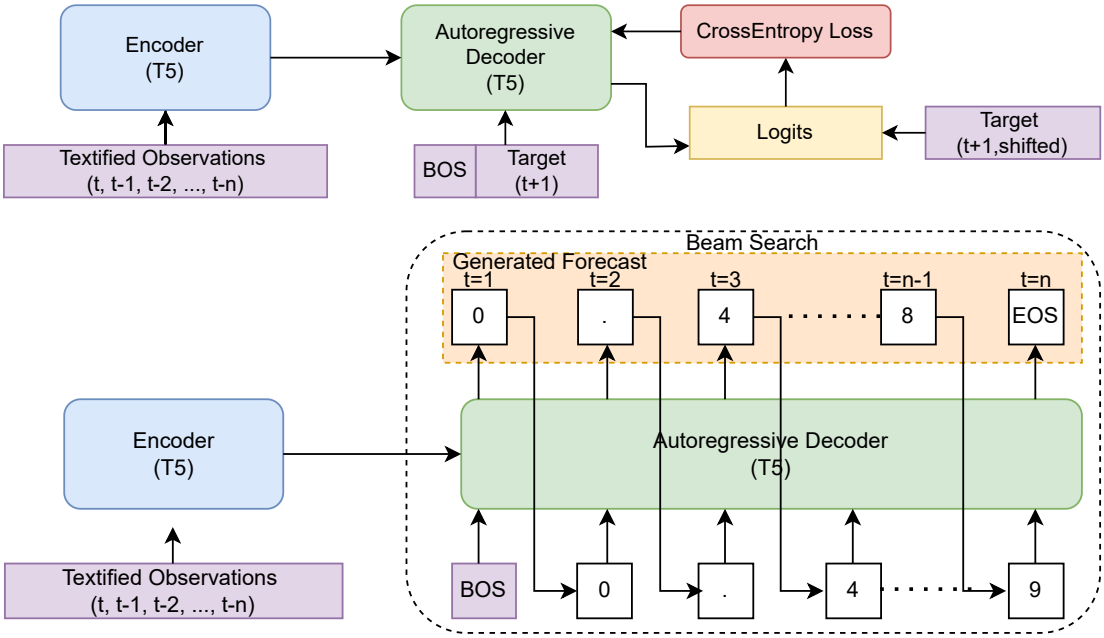


Figure 2: Training and inference phases for the transformer based model.

the network to compute the output. Neural Basis Expansion Analysis For Interpretable Time Series Forecasting [4] (NBeats), a deep neural architecture which is based on a set of backward and forward residual link and a deep stack of fully connected layers arranged in a doubly-residual stacking manner, and bases the predictions on a lookback and forecast period. Deep Autoregressive model [5] (DeepAR), an algorithm based on recurrent neural networks (RNN) which learns successive approximations of the target time-series. Temporal Fusion Transformer [6] (TFT), an attention-based neural network which leverages the recently developed transformer architecture [9] to identify important long-range patterns in the time-series and prioritizes the most relevant patterns.

4.4. Text-to-Text Transformer Model

To be able to train our model based on natural language processing, we first need to describe the input features i.e., the past observations of the time-series in a natural language form. To this aim, we leverage a process denoted as "textification" or "prompting" of the input features and that has been proven to be effective in the context of diagnostic texts [14, 15, 16] as well as in forecasting of human mobility [8]. Such approach takes in input the past observations of the time-series (i.e., the input features) and translate them into a string, which is then used as input to the NLP-based model. In this case we only rely on the array of floating point values corresponding to the past values of each time-series (called lags). We can denote our prompting schema as follows:

> contextual information: {contextual features}.
> previous observations: {time-series features}

More in detail, if we consider a set of k previous values (i.e., lags), the prompt is as follows:

> contextual information: {contextual features}.
> previous observations: {value} at time t-1, . . ., {value} at time t-k.

A real example of the prompt applied to the dataset is reported in the following, considering $k = 56$.

> contextual information: the month is 4, the day is 9 (5 day of the week), 14 week of the year. the time is 08:14.
> previous observations are: 9.8 at time t-1, 9.8 at time t-2, 9.8 at time t-3, 9.8 at time t-4, 9.6 at time t-5, 9.8 at time t-6, . . . [features from time t-7 to time t-54] . . ., 8.7 at time t-55, 9.2 at time t-56.

We develop and train our model using the PyTorch[2] and HuggingFace[3] frameworks. We rely on the `T5-base` model[4], which was trained on a mixture of unsupervised and supervised tasks [11, Appendix Section]. The considered model is composed of an encoder decoder stack including 12 blocks, each comprising self-attention, optional encoder-decoder attention, and a feed-forward network. The attention is of dimension 64, while embeddings have 768 dimensions. The final model has about 220 million parameters.

We initialized the model with the pre-trained weights. We feed the textual input to the model by using custom prefixes "predict:", "input:", and "target:". The experiments have been carried put on a Linux server equipped with 16x Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz, 70GB of RAM, and 2x Nvidia Geforce RTX 3090 GPUs for 3 epochs. As loss we use the conventional multi-class cross entropy loss, where the number of classes is equal to the size of the vocabulary, defined as $\mathcal{L} = -\frac{1}{B}\sum_{b=1}^{B}\sum_{k=1}^{|V|} y_k^b \log(\hat{y}_k^b)$ where the superscript b represents the current batch and B is the batch size, $|V|$ is the size of the vocabulary, y represents the true token, and $\hat{y}_k$ is the output probability distribution over the vocabulary for each time-step.

To perform inference we generate text using beam search, thus generating the output sequence token-by-token by leveraging the cross-attention layers while passing the input to the decoder, and we generate auto-regressively the output of the decoder. We implement early stopping by setting the corresponding parameter to true. We found that our fine-tuned model generates floating point numbers for each beam, so we had no need to leverage constrained search strategies. The training and inference phases for our model are summarized in Figure 2.

Table 1
Metrics for chloride, nitrate, and sulfate test sets. We consider a lag of 4 days (14 observations per day x 4 days = 56), and we forecast the next value in the series (t+1), the subsequent day (t+14). We highlight in bold the most effective method for each section.

		Chloride			Nitrate			Sulfate		
Model	**Pred**	**MAE**	**ME**	**RMSE**	**MAE**	**ME**	**RMSE**	**MAE**	**ME**	**RMSE**
LSTM	t+1	.1572	.8278	.2003	.1086	**.6468**	.1419	.1893	.8857	.2381
GRU	t+1	.1577	.8109	.2007	.1090	.6484	.1424	.1888	.8869	.2375
DeepAR	t+1	.1533	.7839	.1949	.1058	.6473	.1377	.1851	.8391	.2324
NBeats	t+1	.1592	.8477	.2030	.1095	.6518	.1435	.1910	.9027	.2406
TFT	t+1	.1589	.8548	.2027	.1114	.6576	.1456	.1918	.9112	.2413
T5	t+1	**.0316**	**.6027**	**.0674**	**.0121**	.8163	**.0402**	**.0182**	**.6596**	**.0579**
LSTM	t+14	.1212	.7075	.1543	.0899	.5939	.1156	.1526	.6838	.1912
GRU	t+14	.1207	.6948	.1533	.0888	.6093	.1145	.1555	.7099	.1932
DeepAR	t+14	.1208	.6103	.1534	.0881	.6318	.1143	.1481	**.6018**	.1817
NBeats	t+14	.1278	.6482	.1620	.0934	.6958	.1209	.1575	.7043	.1954
TFT	t+14	.1260	.6246	.1594	.0853	**.5909**	.1114	.1457	.6236	.1792
T5	t+14	**.1176**	**.6027**	**.1506**	**.0762**	.6122	**.1068**	**.1292**	.6170	**.1697**

5. Results

Table 1 and Figure 3 show the results for the three substances for the short- and mid-term predictions. Let us start by inspecting the predictions for the subsequent timestamp. As we can see from the first section of the table, it is almost always the case that the proposed approach achieves higher effectiveness than the state-of-the-art approaches, with the only exception of the maximum error for the nitrate substance. Similarly, our model outperforms state-of-the-art models when performing predictions for the mid-term, that is predicting the substance concentration for the subsequent day, with the two only exceptions. This is an important result; in fact, having a reliable prediction for the subsequent day allows domain experts to plan and implement effective countermeasures for the drinking water plant.

Besides providing quantitative results, we also perform qualitative ones. Figure 4 shows the prediction for the sulfate substance when predicting the subsequent value in the time-series (i.e., t+1) for the best method (i.e., T5) and the second best (i.e., DeepAR) according to the effectiveness metrics as in Table 1. The results for the other two substances are identical and thus not reported. As we can see from the plot, both approaches approximate the real time-series. Nevertheless, by inspecting the two series closely we can find an important difference; the DeepAR algorithm (as we well as the other deep-learning based methodologies) tends to predict accurate values of the time-series, but they also tend to provide those forecasts with a certain time-lag; in other words, it predicts accurate values with a (mostly) fixed time delay, noticeable by inspecting the x-axis of the plot and comparing the pace of the two series, the real and the predicted one. Thus, if we select a real value in the y-axis, we see that the same value is

[2] https://pytorch.org/
[3] https://huggingface.co/
[4] https://huggingface.co/t5-base

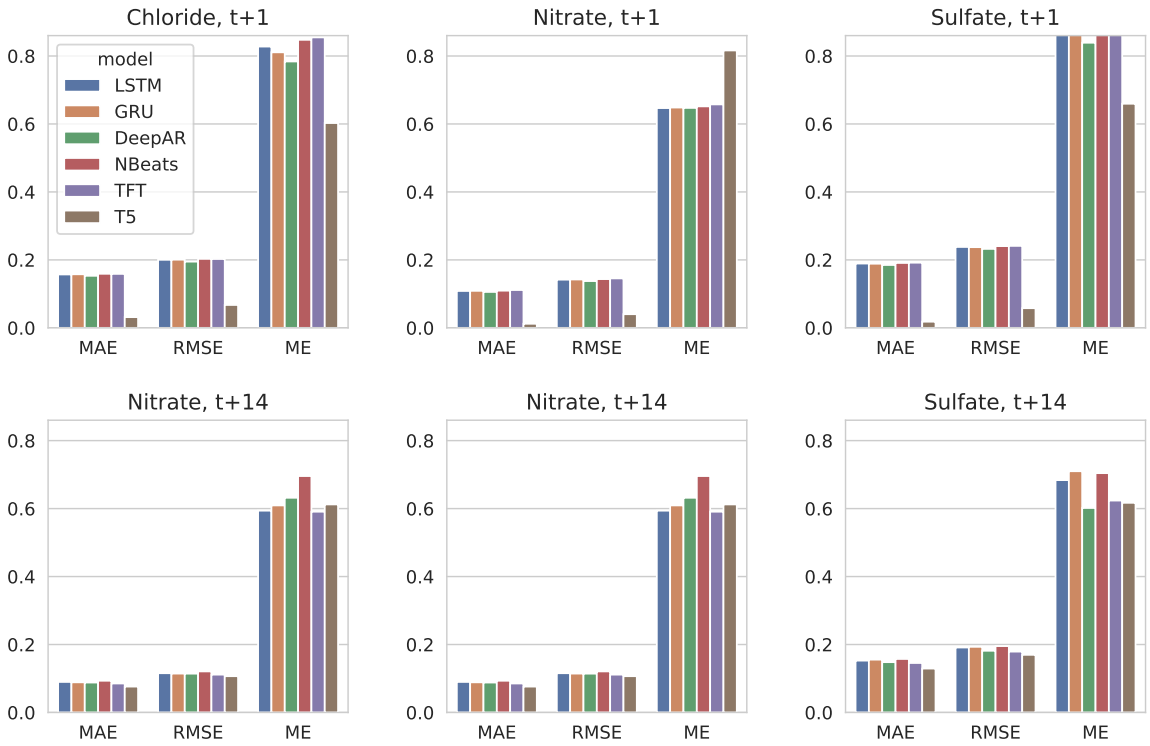


Figure 3: Metrics for chloride, nitrate, and sulfate test sets.

predicted by the algorithm in a time frame around t+1. This is a well documented effect in time-series forecasting literature and it is known to affect both machine and deep learning approaches. On the contrary, possibly due to the different modeling approach adopted by the natural language approach, we see that T5 does not suffer, or suffers in a limited form, from such effect. In fact, it tends to make different kind of errors, distributed mostly with shifts on the y-axis (i.e., prediction errors) rather than on the x-axis (i.e., delayed forecasts).

Figure 5, similarly to Figure 4, shows the prediction for the sulfate substance when predicting the value in the time-series for the next day (i.e., t+14) for the best method (i.e., T5) and the second best (i.e., DeepAR) according to the effectiveness metrics as in Table 1. The results for the other two substances are very similar and thus not reported. As we can see from the plot, the models make very different prediction errors, analogously to what observed in the previous result for t+1. In this case, while the DeepAR algorithm prediction follows a sort of moving average computed for the different time stamps, T5 successfully predicts some of the peaks present in the time-series, and makes errors distributed mostly around the y-axis.

6. Zero-Shot Capabilities

One of the documented advantages of large pre-trained natural language models is that they carry the ability of zero- and few-shot leaning [17, 18] i.e., the ability of solving a task for a

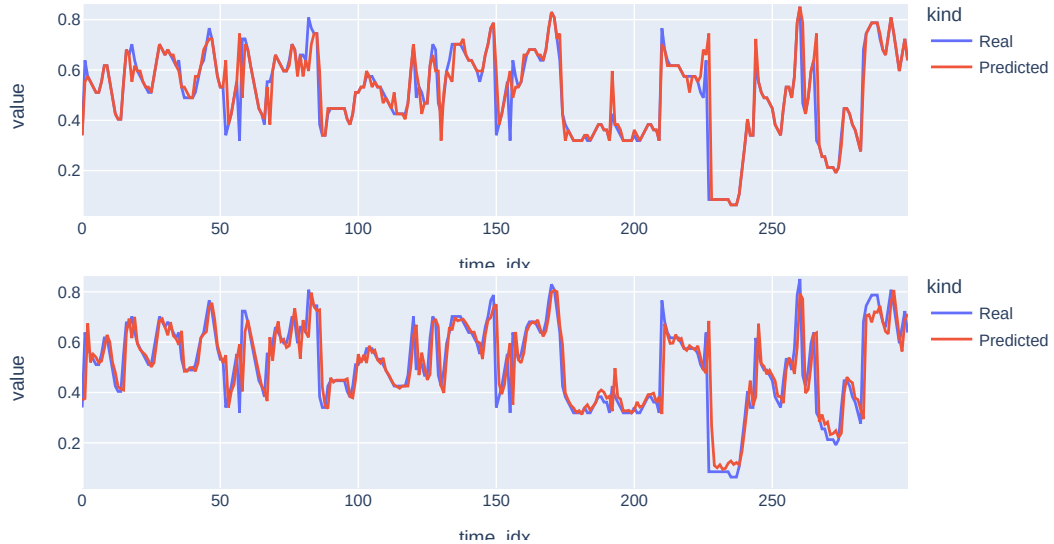


Figure 4: Prediction for the sulfate substance at t+1 for the T5 (best) and DeepAR (second best) method. X-axis has been cut, and values are scaled in 0–1.

Figure 5: Prediction for the sulfate substance at t+14 for the T5 (best) and DeepAR (second best) method. X-axis has been cut, and values are scaled in 0–1.

domain without receiving any, or just few, examples of that task or for that domain at training phase. To further investigate the effectiveness of the T5 model to forecast the concentration of polluting substances in a water treatment plant, we conduct an experiment under the zero-shot paradigm. More in detail, we train each model on a substance and we test the trained model on the set of other substances which are different from the training one (i.e., we use the model trained on chloride to forecast the sulfate substance).

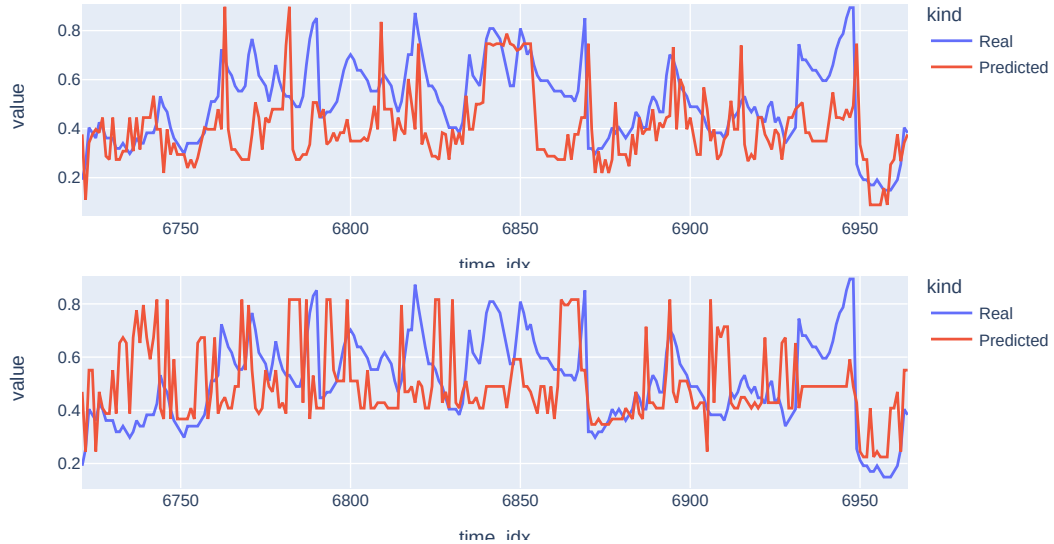


Figure 6: Prediction for the sulfate substance at t+1 performed using the T5 model trained on the chloride substance (above), and the nitrate substance (below). X-axis has been cut, and values are scaled in 0–1.

Figure 6 shows the qualitative prediction for the sulfate substance at t+1 performed using the T5 model trained on either the chloride or the nitrate substance. As we can see from the plots, while the model predictions are far from the ones computed with the corresponding model and test set (i.e., T5 trained on sulfate), they are not random either, and we can see that the predictions tend to follow the real time-series and correctly approximate some of the series peaks.

We also computed the effectiveness metrics for the zero-shot scenario: the model trained on chloride and nitrate achieves on sulfate respectively a MAE of 0.1717 and 0.1808 (T5 had 0.0182 and DeepAR 0.1377), a ME of 0.7368 and 0.8298 (T5 had 0.6596 and DeepAR 0.8391), and a RMSE of 0.2095 and 0.2182 (T5 had 0.0579 and DeepAR 0.2324). By looking at the metrics, we found that while the zero-shot model effectiveness is far the one obtained with the T5 model trained on domain specific data, the zero-shot models are almost as effective as, and for RMSE even more effective than, state-of-the-art deep learning approaches.

Although using the T5 model does not demonstrate optimal performances for the zero-shot task, this experiment show that causal models have promising generalization abilities for time-series forecast. Thus, we believe that further research is this direction, with the help of domain specific pre-trained models would improve the effectiveness and generalization abilities of those models.

7. Discussion and Conclusion

We studied the capabilities of causal language models (especially T5) for the task of forecasting the concentration of polluting substances in a water treatment plant, addressing both short- and

mid-term forecasting. To this end, we applied transformation to the input features to translate them into a textual form and feed them to the natural language model. The results show that our approach could improve state-of-the-art algorithms for forecasting on both the short and mid-term.

Given that the application of language models for the task of time-series forecasting might appear counter-intuitive at a first sight, let us make some remarks on why such approach works in practice. As we have seen, recent research showed that transformer based models are suitable and effective on a variety of tasks which are not related to the NLP paradigm, from images [19, 20] to videos [21] and even reinforcement learning [22] and graphs [23]. All the transformers based models rely on the attention mechanism which, joint with the training procedure that always consist in reconstructing a masked or perturbed part of the input, allow them to learn latent relationship in input sequences and between the input and output ones. For textual tasks they learn to reconstruct missing tokens, for visual ones they learn to reconstruct missing or altered frames, but they also showed the ability to learn and reconstruct complex structures such as (sub) graphs. For the same reason, we believe that the textual description of the time-series allows the model to form an accurate latent representation of it, which is then leveraged, jointly with the causal training modality (i.e., predict the next item in a sequence), to make accurate forecasting predictions. We plan to provide further insights on this by leveraging interpretability frameworks [24].

The results of this paper opens for a wide range of applications of language models to time-series forecasting problems. Future work aims at validating predictions with domain experts to understand to what extent the predicted values allow for practical and effective countermeasures to be applied in the treatment plant. Furthermore, we plan to improve zero-shot effectiveness by deepening the study on domain-invariant features.

Acknowledgments

This work was partially supported by the REACT-EU project "Data-Driven Multiutility Grid: Supporto alle Decisioni per Garantire la Sostenibilità dal Real Time al Lungo Termine" with "PON 2014-2020 AZIONE IV.6 GREEN".

References

[1] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, Stl: A seasonal-trend decomposition, Journal of Official Statistics 6 (1990) 3–73.

[2] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[3] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[4] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, arXiv preprint arXiv:1905.10437 (2019).

[5] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting

with autoregressive recurrent networks, International Journal of Forecasting 36 (2020) 1181–1191.
[6] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting 37 (2021) 1748–1764.
[7] H. Xue, B. P. Voutharoj, F. D. Salim, Leveraging language foundation models for human mobility forecasting, arXiv preprint arXiv:2209.05479 (2022).
[8] H. Xue, F. D. Salim, Y. Ren, C. L. Clarke, Translating human mobility forecasting through natural language generation, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1224–1233.
[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., Exploring the limits of transfer learning with a unified text-to-text transformer., J. Mach. Learn. Res. 21 (2020) 1–67.
[12] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: Lstm cells and network architectures, Neural computation 31 (2019) 1235–1270.
[13] R. Dey, F. M. Salem, Gate-variants of gated recurrent unit (gru) neural networks, in: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, 2017, pp. 1597–1600.
[14] M. H. Popescu, K. Roitero, S. Travasci, V. Della Mea, Automatic assignment of ICD-10 codes to diagnostic texts using transformers based techniques, in: 2021 IEEE 9th International Conference on Healthcare Informatics (ICHI), IEEE, 2021, pp. 188–192.
[15] K. Roitero, B. Portelli, M. H. Popescu, V. Della Mea, DiLBERT: Cheap embeddings for disease related medical NLP, IEEE Access 9 (2021) 159714–159723.
[16] V. Della Mea, M. H. Popescu, K. Roitero, Underlying cause of death identification from death certificates using reverse coding to text and a nlp based deep learning approach, Informatics in Medicine Unlocked 21 (2020) 100456.
[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
[18] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, arXiv preprint arXiv:2205.11916 (2022).
[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[20] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 22–31.
[21] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, in: Proceedings of the IEEE/CVF Conference on Computer

Vision and Pattern Recognition, 2021, pp. 8741–8750.
[22] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, I. Mordatch, Decision transformer: Reinforcement learning via sequence modeling, Advances in neural information processing systems 34 (2021) 15084–15097.
[23] V. P. Dwivedi, X. Bresson, A generalization of transformer networks to graphs, arXiv preprint arXiv:2012.09699 (2020).
[24] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, et al., Captum: A unified and generic model interpretability library for pytorch, arXiv preprint arXiv:2009.07896 (2020).

Experiments with the User's Feedback in Preference Elicitation

Tereza Siváková[1], Miroslav Kárný[1]

[1]*The Czech Academy of Sciences, Institute of Information Theory and Automation*
182 00 Prague 8, Czech Republic

Abstract

This paper deals with user's preferences (wishes). Common users are uneducated in the decision-making (DM) theory and present their preferences incompletely. That is why we elicit them from such a user during the DM. The paper works with the DM theory called fully probabilistic design (FPD). FPD models closed DM loop, made by the user and the system, by the joint probability density (pd, real pd). A joint ideal pd quantifies the user's preferences. It assigns high probability values to preferred closed-loop behaviors and low values to undesired behaviors. The real pd should be kept near the ideal pd. By minimizing the Kullback-Leibler divergence of the real and ideal pds, the optimal decision policy is found. The presented algorithmic quantification of preferences provides ambitious but potentially reachable DM aims. It suppresses demands on tuning preference-expressing parameters. The considered ideal pd assigns high probabilities to desired (ideal) sets of states and actions. The parameters of the ideal pd (tuned during the DM via the user's feedback) are: ▶ relative significance of respective probabilities; ▶ a parameter balancing exploration with exploitation. Their systematic tuning solves meta-DM level task, which observes the agent's satisfaction expressed humanly by "school-marks". It opts free parameters to reach the best marks. A formalization and solution of this meta-task were recently done, but experience with it is limited. This paper recalls the theory and provides representative samples of extensive up to now missing simulations.

Keywords

Preference elicitation, Adaptive agent, Decision making, Bayes rule

Motivation Our results contribute to long-term research that tries to create a normative theory of dynamic decision making applicable by imperfect decision makers, [12, 13], [14, 16]. Its aims are close to the quest for universal artificial intelligence, [9, 15, 21].

1. Introduction

Decision making (DM) is the everyday activity of every human. It is important to make the right decisions to achieve the goal. DM is described by a closed-loop formed by an agent (the person, who makes decisions) and an environment. The environment of the agent is usually called a system and its dynamics is unknown. It is described by transition probability density (pd) between its states conditioned by the agent's actions. The agent observes the state s of the system and makes an action a to meet their wishes, ideally, to move the system to the desired state. The actions are chosen via the agent's policy π. It consists of decision rules r, which determine what

Italian Workshop on Artificial Intelligence and Applications for Business and Industries 2022, Udine, Italy
sivakter@cvut.cz (T. Siváková); school@utia.cas.cz (M. Kárný)
https://www.utia.cas.cz/AS (M. Kárný)
0000-0002-7440-6041 (M. Kárný)

action should be chosen in each time epoch depending on the system's state and the model of the system. The model m expresses the agent's beliefs about the dynamics of the real system.

The main task of DM is to select the optimal policy. This paper uses a fully probabilistic design (FPD), which introduces an ideal probability density

$$\mathsf{c}^i(b) = \prod_{t\in\mathbb{T}} \mathsf{m}^i(s_t|a_t, s_{t-1})\mathsf{r}^i(a_t|s_{t-1}),$$

which expresses the desired pd of behavior $b \equiv (s_0, a_1, s_1, a_2, s_2, \ldots, a_T, s_T) \in \mathbb{B}$. It sets high probability values to preferred behaviors and low probability values to unwanted behaviors. It consists of an ideal model m^i of the system and of an ideal decision rule r^i. The real pd $\mathsf{c}^\pi(b)$ depends on the model m of the system and decision rules r forming the policy π.

$$\mathsf{c}^\pi(b) = \prod_{t\in\mathbb{T}} \mathsf{m}(s_t|a_t, s_{t-1})\mathsf{r}(a_t|s_{t-1}).$$

This paper exploits Bayes' learning to get m relating (the observed state, the used action, the next state). The optimal policy π^o in a set Π minimizes the Kullback-Leibler divergence (KLD) of the pd c^π to the ideal pd c^i

$$\pi^o \in \mathsf{Arg}\min_{\pi\in\Pi} \mathsf{D}(\mathsf{c}^\pi||c^i) = \mathsf{Arg}\min_{\pi\in\Pi} \int_{b\in\mathbb{B}} \mathsf{c}^\pi(b)\ln\left(\frac{\mathsf{c}^\pi(b)}{\mathsf{c}^i(b)}\right)\mathsf{d}b.$$

Theorem 1. ***(FPD, [22])*** *Decision rules, which constitute the optimal decision policy π^o, are computed for $t = T, T-1, \ldots, 1$ and with $\mathsf{h}(s_T) \equiv 1$ as follows*

$$\mathsf{r}^o(a_t|s_{t-1}) \equiv \mathsf{r}^i(a_t|s_{t-1})\frac{\exp[-\mathsf{d}(a_t, s_{t-1})]}{\mathsf{h}(s_{t-1})},$$

$$\mathsf{d}(a_t, s_{t-1}) \equiv \int_{s_t\in\mathbb{S}} \mathsf{m}(s_t|a_t, s_{t-1})\ln\left[\frac{\mathsf{m}(s_t|a_t, s_{t-1})}{\mathsf{h}(s_t)\mathsf{m}^i(s_t|a_t, s_{t-1})}\right]\mathrm{d}s_t \tag{1}$$

$$\mathsf{h}(s_{t-1}) \equiv \int_{a_t\in\mathbb{A}} \mathsf{r}^i(a_t|s_{t-1})\exp[-\mathsf{d}(a_t, s_{t-1})]\,\mathrm{d}a_t \in \mathsf{h}(s_t) \in [0, 1].$$

$$\textit{The attained minimum is } \min_{\pi\in\Pi} \mathsf{D}(\mathsf{c}^\pi||\mathsf{c}^i) = -\ln(\mathsf{h}(s_0)). \tag{2}$$

We focus on the preference quantification, on finding the c^i. The preference specification is mostly incomplete due to the agent's imperfections. This means that

$$\mathbb{C}^i \equiv \{\text{ideal pds } \mathsf{c}^i(b), b \in \mathbb{B}, \text{respecting the agent's wishes}\} \tag{3}$$

includes several pds. It can be also empty because of the agent's inconsistencies. The agent's preferences can be in contradiction or the agent can have un-achievable goals. The preference elicitation (PE) consists of the choice of: ▶ the non-empty set $\mathbb{C}^i$ that overcomes the agent's inconsistencies ▶ the optimal ideal pd c^{io} from the set (3).

The PE principle from [18] recommends to choose as the optimal ideal pd

$$\mathsf{c}^{io} \in \mathsf{Arg}\min_{\mathsf{c}^i\in\mathbb{C}^i}\min_{\pi\in\Pi}\mathsf{D}(\mathsf{c}^{\pi}||\mathsf{c}^i). \tag{4}$$

Its use in FPD ensures that no preferences are added to the agent's. Theorem 1 describes the 1^{st} minimization over π. The 2^{nd} minimization over c^i is harder and it can be done over individual factors of c^i for each already observed state.

Then, cf. (1), (2), (4), the optimal closed-loop ideal pd c^{io} in the last step reads

$$\mathsf{c}^{io} \equiv \mathsf{m}^{io}\mathsf{r}^{io} \overset{(1),(2)}{\in} \mathsf{Arg}\max_{\mathsf{r}^i\in\mathbb{R}^i}\left[\max_{\mathsf{m}^i\in\mathbb{M}^i}\int_{a_1\in\mathbb{A}}\mathsf{r}^i(a_1|s_0)\exp[-\mathsf{d}(a_1,s_0)]\mathsf{d}a_1\right]$$

$$\mathsf{d}(a_1,s_0)=\int_{s_1\in\mathbb{S}}\mathsf{m}(s_1|a_1,s_0)\ln\left(\frac{\mathsf{m}(s_1|a_1,s_0)}{\mathsf{h}(s_0)\mathsf{m}^i(s_1|a_1,s_0)}\right)\mathsf{d}s_1, \tag{5}$$

$\mathsf{h}(s_0)$ comes from the backward recursion via step (1). The minimization over a c^i- factor ($\mathsf{c}^i(s_t|a_t,s_{t-1})=\mathsf{m}^i(s_t|a_t,s_{t-1})\mathsf{r}(a_t|s_{t-1})$) in any decision epoch $t\in\mathbb{T}$ and for any realized state s_{t-1} are formally identical. Therefore, we can suppress t and $s_{t-1}\in\mathbb{S}$ and deal with $\mathsf{m}(s|a)\equiv\mathsf{m}(s_t=s|a_t=a,s_{t-1})$, $\mathsf{m}^i(s|a)\equiv\mathsf{m}^i(s_t=s|a_t=a,s_{t-1})$, $\mathsf{r}(a)\equiv\mathsf{r}(a_t=a|s_{t-1})$, $\mathsf{r}^i(a)\equiv\mathsf{r}^i(a_t=a|s_{t-1})$ and $\mathsf{h}(s)=\mathsf{h}(s_t=s)$. The optimization (5) uses the given $\mathsf{h}(s)$ and runs over $\mathbb{M}^i$ (a set of m^i-s) while $\mathbb{C}^i$ is determined by a given r^i and chosen from the set $\mathbb{R}^i$ (a set of r^i-s). For then $\mathsf{c}^i=\mathsf{m}^i\mathsf{r}^i$- factors are in

$$\{\mathsf{c}^i(s,a):\mathsf{c}^i(s,a)=\mathsf{m}^i(s|a)\mathsf{r}^i(a),s\in\mathbb{S},a\in\mathbb{A},\text{respecting the agent's wishes}\}. \tag{6}$$

2. Preference Quantification

We first perform the optimization for a quite general choice of sets $\mathbb{M}^i,\mathbb{R}^i$. Then, we specialize it to a specific but still general case.

2.1. The generic choice of optimal ideal model of the system

Theorem 2. *(**Optimal** m^{io}**-factor, [19])** Let $\mathsf{r}^i\in\mathbb{R}^i$ be a fixed ideal decision rule, which defines a non-empty cross-section $\mathbb{M}^i\equiv\{\mathsf{m}^i:\mathsf{m}^i\mathsf{r}^i\in$ set $(6)\}$. Let $\mathsf{m}^i(s|a)\in\mathbb{M}^i$ exist such that $\mathsf{d}(a)<\infty,\forall a\in\mathbb{A}$ (1) t and s_{t-1} suppressed. Then, the optimal ideal $\mathsf{m}^{io}-$factor minimises $\mathsf{d}(a),s\in\mathbb{S},a\in\mathbb{A}$, i.e.*

$$\mathsf{m}^{io}(s|a)\in\mathsf{Arg}\max_{\mathsf{m}^i\in\mathbb{M}^i}\int_{\mathbb{A}}\mathsf{r}^i(a)\exp[-\mathsf{d}(a)]\mathsf{d}a=\mathsf{Arg}\min_{\mathsf{m}^i\in\mathbb{M}^i}\mathsf{d}(a). \tag{7}$$

2.2. The generic choice of optimal ideal decision rule

The decision rules work on the set of admissible actions. Thus, the support of an admissible r-factor must be included in the set of possible actions i.e. $\text{supp}[\mathsf{r}]\subset\mathbb{A}$. The form of the FPD-optimal r^o-factor, Theorem 1, implies that $\text{supp}[\mathsf{r}^o]\subseteq\text{supp}[\mathsf{r}^i]$. Therefore, only the ideal r^i-factors

$$\mathsf{r}^i\in\mathbb{R}^i\equiv\left\{\mathsf{r}^i:\text{supp}[\mathsf{r}^i]=\mathbb{A}\right\} \tag{8}$$

keep actions $a \in \mathbb{A}$ and exclude none. Thus, (8) is the generic constraint and

$$\mathbb{R}^i \equiv \{\mathsf{r}^i : \mathsf{m}^{io}\mathsf{r}^i \in (6) \text{ while } \mathsf{m}^{io} \text{ is given by } (7)\}.$$

Theorem 3. ***(Optimal r^{io}-factor meeting (8), [19])*** *Let assumptions of Theorem 2 hold and for a scalar $p > 1$*

$$\mathbb{R}^i \equiv \left\{ \mathsf{r}^i : supp[\mathsf{r}^i] = \mathbb{A},\ ||\mathsf{r}^i||_p \equiv \left[\int_{\mathbb{A}} (\mathsf{r}^i(a))^p \mathsf{d}a\right]^{1/p} < \infty \right\},\ |\mathbb{A}| \equiv \int_{\mathbb{A}} \mathsf{d}a < \infty. \tag{9}$$

Then, the optimal ideal r^{io}-factor reads, cf. (1), (7),

$$\begin{aligned} \mathsf{r}^{io} &\propto \chi_{\mathbb{A}}(a) \exp[-\nu \mathsf{d}^o(a)], \quad \nu \equiv \frac{1}{p-1},\ \chi_{\mathbb{A}}(a) \textit{ is the indicator function of } \mathbb{A} \\ \mathsf{d}^o(a) &\equiv \int_{\mathbb{S}} \mathsf{m}(s|a) \ln\left(\frac{\mathsf{m}(s|a)}{\mathsf{h}(s)\mathsf{m}^{io}(s|a)}\right) \mathsf{d}s \overset{(7)}{\leq} \mathsf{d}(a). \end{aligned} \tag{10}$$

The r^{io}-factor (10) belongs to (9) and meets (8).

Remarks ▶ The generic constraint (8) implies that the ideal r^i-factors support exploration, which makes the Bayesian learning efficient. ▶ The parameter ν controls exploration. Every action from the set of possible actions can be tried with almost the same probability if the parameter ν is close to 0. If ν gets bigger the exploration declines, cf. form of r^{io} in (10).

2.3. The specific choice of $\mathbb{M}^i$ making $\mathbb{C}^i \neq \emptyset$

The optimal ideal r^{io}-factor is uniquely given by the choice of m^{io} (and by the opted ν) via (10). The description of the agent's preferences only guarantees a non-empty set $\mathbb{M}^i$. A wide range of practical cases can be covered with a few additional PE-oriented queries. Our specific elaborated case concerns the next agent's general wish.

$$\begin{gathered} \textit{The agent wants to reach given sets of ideal states } \mathbb{S}^i \textit{ and ideal actions } \mathbb{A}^i, \\ \emptyset \neq \mathbb{S}^i \subset \mathbb{S},\ \emptyset \neq \mathbb{A}^i \subseteq \mathbb{A}. \end{gathered} \tag{11}$$

This is quantified as the wish to assign the highest probability to the set of ideal states $\mathbb{S}^i$ and to the set of ideal actions $\mathbb{A}^i$ (11) by closing the loop of the given model m and of the optimal ideal decision rule r^{io}. So we choose as the maximized functional

$$\int_{\mathbb{A}} \rho(a)\mathsf{r}^{io}(a)\,\mathrm{d}a \equiv \int_{\mathbb{A}} \left[(1-w)\int_{\mathbb{S}} \chi_{\mathbb{S}^i}(s)\mathsf{m}(s|a)\,\mathrm{d}s + w\chi_{\mathbb{A}^i}(a)\right] \mathsf{r}^{io}(a)\,\mathrm{d}a. \tag{12}$$

The introduced weight $w \in \mathbb{W} \equiv [0,1]$ parameterizes how much the agent prefers to stay in the set of ideal actions $\mathbb{A}^i$ relative to being in the set of ideal states $\mathbb{S}^i$.

The inspected problem has a meaningful solution if

$$\rho(a) = (1-w)\int_{\mathbb{S}} \chi_{\mathbb{S}^i}(s)\mathsf{m}(s|a)\,\mathrm{d}s + w\chi_{\mathbb{A}^i}(a) > 0, \text{ on } \mathbb{A}. \tag{13}$$

If the functional (12) is large, then the probabilities of the preferred sets are large. The part $(1-w)\int_{\mathbb{S}}\chi_{\mathbb{S}^i}(s)\mathsf{m}(s|a)\,\mathrm{d}s$ forces the highest probability to the set $\mathbb{S}^i$. And the part $w\chi_{\mathbb{A}^i}(a)\mathsf{r}^{io}(a)$ should guarantee that the ideal decision rule will often choose the actions from the set $\mathbb{A}^i$. The weight w balances these probabilities.

Remarks ▶ The weight is fixed. Its fine-tuning is controlled by additional queries. ▶ The function determining $\rho(a)$ qualitatively plays the role of the reward. ▶ Our construction of the optimal ideal pd c^{io} quantifies the agent's preferences in an ambitious but realistic way. ▶ Maximization of (12) with r^{io} given by (10) rely on:

Theorem 4. ***(Optimal value of*** d^o***, [19])*** *Under assumptions of Theorem 3, covering those of Theorem 2, and under (13), the optimal ideal model* m^{io} *fulfilling (12) determines* $\mathsf{d}^o(a)$*, giving* $\mathsf{r}^{io}=\mathsf{r}^i(\mathsf{m}^{io})$ *(10),* $a\in\mathbb{A}$*, as the function*

$$\mathsf{d}^o(a) \;\equiv\; \mathsf{d}^o(\bar{a})+\ln\left(\frac{\max_{a\in\mathbb{A}}(\rho(a))}{\rho(a)}\right),\ \bar{a}\in\mathrm{Arg}\max_{a\in\mathbb{A}}(\rho(a)). \tag{14}$$

Theorem 5. ***(Solvability of (14), [19])*** *Under (13) and* $|\mathbb{A}|<\infty$*, the smallest* $\mathsf{d}^o(\bar{a})$ *exists such that (14) has a solution* $\mathsf{m}^{io}(s|a)$*,* $s\in\mathbb{S}$*,* $\forall a\in\mathbb{A}$*. Thus, the smallest* $\mathsf{d}^o(\bar{a})$ *guaranteeing solvability of (14)* $\forall a\in\{a\}$ *is*

$$\mathsf{d}^o(\bar{a})=\max\left[0,\max_{a\in\mathbb{A}}\int_{\mathbb{S}}\mathsf{m}(s|a)\ln\left[\frac{\rho(a)}{\rho(\bar{a})\mathsf{h}(s)}\right]\mathrm{d}s\right]. \tag{15}$$

The ideal m^{io} gives $\mathsf{d}^o(a)$ (1) and $\mathsf{r}^{io}(\mathsf{m}^{io})$ via (7). The next proposition provides it for generic pds $\mathsf{m}(s|a)$. It requires to find m^{io} giving d^o (14) on $\mathbb{A}$.

Theorem 6. ***(*** m^{io} ***meeting (12), generic*** $\mathsf{m}(s|a)$***, [19])*** *Let* $\mathsf{m}(s|a)$*, for some* $a\in\mathbb{A}$*, be non-uniform on* $\mathbb{S}$ *and Theorem (3) hold. Then, the* m^{io}*−factor meeting (12) reads*

$$\mathsf{m}^i(s|a) \;=\; \frac{\mathsf{m}(s|a)\exp(-\mathsf{e}(a)\mathsf{m}(s|a))}{\int_{\mathbb{S}}\mathsf{m}(s|a)\exp(-\mathsf{e}(a)\mathsf{m}(s|a))\,\mathrm{d}s},\ \textit{while}\ |\mathbb{S}|\equiv\int_{\mathbb{S}}\mathrm{d}s<\infty. \tag{16}$$

The real valued $\mathsf{e}(a)$ *in (16) is the existing solution of* $\mathsf{L}(\mathsf{e}(a))=\mathsf{R}(a)$*. For* $\mathsf{d}^o(\bar{a})$ *meeting (15) with* $\bar{a}\in\mathrm{Arg}\max_{a\in\mathbb{A}}\rho(a)$*, the left- and right-hand sides of this equation are*

$$\begin{aligned}\mathsf{L}(\mathsf{e}(a)) \;&\equiv\; \mathsf{e}(a)\Lambda(a)+\ln\left(\int_{\mathbb{S}}\mathsf{m}(s|a)\exp[-\mathsf{e}(a)\mathsf{m}(s|a)]\,\mathrm{d}s\right),\ \Lambda(a)\equiv\int_{\mathbb{S}}\mathsf{m}^2(s|a)\,\mathrm{d}s\\ \mathsf{R}(a)\equiv-\int_{\mathbb{S}}\mathsf{m}(s|a)\ln\left(\frac{\mathsf{m}(s|a)}{\mathsf{h}(s)}\right)\mathrm{d}s+\mathsf{d}^o(\bar{a})+\ln\left(\frac{\rho(\bar{a})}{\rho(a)}\right),\ \bar{a}\in\mathrm{Arg}\max_{a\in\mathbb{A}}\rho(a).\end{aligned} \tag{17}$$

The uniform case was solved similarly, see [19].

3. On algorithmization

In the considered case with the discrete-valued states and actions, the found solution can be directly converted into a compact algorithm. It is done in [19]. Here, we just stress that it uses

the Bayesian estimation of unknown but time-invariant values of the transition probabilities Θ. The gained parametric model $\mathsf{m}(s_t|a_t, s_{t-1}, \Theta)$ belongs to the exponential family [1] and makes Dirichlet's prior pd self-reproducing. Its degrees of freedom counting the observed transitions $s_{t-1} = \tilde{s} \in \mathbb{S}$, $a_t = a \in \mathbb{A}$ to $s_t = s \in \mathbb{S}$ form the sufficient statistic for learning unknown $\Theta_{s|a,\tilde{s}} \equiv \mathsf{m}(s|a, \tilde{s}, \Theta)$ [3].

4. Dialogue with the user

The agent specifies the preferred states $\mathbb{S}^i$ and preferred actions $\mathbb{A}^i$ before the beginning DM. A problem arises as the agent[1] wishes concern two usually contradiction things. In this case, we need to choose the weight w in (12), which determines how much the user prefers to stay in the set $\mathbb{A}^i$ relative to being in $\mathbb{S}^i$. But they are unable to express how much they prefer it before they will observe how the closed loop behaves. That is why we added a dialogue with the user during the DM. The user will express their preferences and next they will control the results of the DM during the DM. The DM solved in Section 3, referred to as the basic DM, deals with two types of inputs:

- ✓ those directly describing the basic DM, which include: ▶ the state $\mathbb{S}$ and action $\mathbb{A}$ sets; ▶ the wishes-expressing ideal sets $\mathbb{S}^i \subset \mathbb{S}$ and $\mathbb{A}^i \subseteq \mathbb{A}$;
- ✓ more technical, policy-influencing, inputs that include: ▶ the weight $w \in [0, 1]$ balancing the relative importance of ideal sets, see (12); ▶ the scalar $\nu > 0$, see (10), balancing exploration with exploitation (duality, [10, 20]).

Fine variations of ideal sets $\mathbb{S}^i$, $\mathbb{A}^i$ or the design horizon $|T|$ are potential inputs of the preference processing but they are here fixed. Thus, the paper focuses just on the pair w, ν. Its optimal choice depends on: ▶ subjective user's preferences; ▶ the user's attitude to the basic DM; ▶ emotions, etc., i.e. on the user's mental state. The dependence is complex and the mental state can hardly be directly measured and quantified. Two users can have the same preferences expressed by the sets $\mathbb{S}^i, \mathbb{A}^i$, but their responses differ.

In our solution, the user is asked to judge the DM quality reached for various choices of w, ν. This is the domain of classical PE [8] that often elicits preferences about a static DM and interactively queries the user. Even advanced versions, represented by [4, 5, 7], become cumbersome in the targeted basic *dynamic* DM. This makes us adopt the next user-driven way that consists of solving an appropriate FPD meta-task, whose description uses capital versions of all functions and parameters entering it, cf. [11].

The user assigns (satisfaction) marks. Their changes during the dialog serve as the (meta-)state $S_T \in \bar{\mathbb{S}}$, to the behaviour caused by the policy, designed for trial values of the optional inputs here, (w, ν). Their changes A_T are the (meta-)actions. They are generated by (meta-)policy gained by the same algorithm as that used at the basic level[2]. It runs more slowly than the basic DM, $T \in \{\bar{T}, 2\bar{T}, \dots, \} \subset \mathbb{T}$ given by $\bar{T} > 1$.

This simple idea has to cope with the possible infinite regress, i.e. DM at meta-level needs meta-inputs opted via a meta-PE, etc. Also, the curse of dimensionality [2] endangers applicability as the opted inputs are multiple and continuous-valued. Our way counteracts both obstacles. We

[1] The agent will be called user as it is usual for preference elicitation.
[2] In harmony with the quest for a universal DM.

decided to ask queries after every time epoch $\bar{T} > 1$, but the queries can be answered irregularly after some multiples of the $\bar{T}$. The use of zero-order holder copes with the expected irregularity of user's responses. It makes realistic the time-invariance of the model $\mathsf{M}(S_T|A_T, S_{T-\bar{T}}, \Theta) := \Theta_{S_T|A_T,S_{T-\bar{T}}}$ needed for learning this meta-model, cf. the beginning of Sec. 3.

The set of possible meta-states is $\bar{\mathbb{S}} := \{-1, 0, 1\}$. It is implied by a difference of the current mark and previous mark[3] i.e. $\Delta g = g_T - g_{T-1}$. If $\Delta g < 0 \Longrightarrow \bar{\mathbb{S}} = \{-1\}$, if $\Delta g = 0 \Longrightarrow \bar{\mathbb{S}} = \{0\}$ and if $\Delta g > 0 \Longrightarrow \bar{\mathbb{S}} = \{1\}$.

The choice of the ordinal scale of marks $g \in \bar{\mathbb{G}} \equiv \{1, \ldots, |\bar{\mathbb{G}}| \equiv 5\}$ suffices for expressing "satisfaction degree". A rich, cross-domain, experience, e.g. in marketing [6] or in European Credit and Accumulation System, confirms this. The mark $g = 1$ is taken as the best one. The ideal set of meta-states is then $\bar{\mathbb{S}}^i \equiv \{-1\}$.

By construction, the outcomes of the basic DM depend smoothly on the discussed inputs. Thus, changes $A \equiv (\Delta w, \Delta\nu)$ of inputs (w, ν) can be selected in a finite set $\bar{\mathbb{A}} := \{(\Delta w, \Delta\nu)\}$ of discrete values. The natural flexible options are

$$\Delta w \in \{-\bar{w}, 0, \bar{w}\},\ \Delta\nu \in \{-\bar{\nu}, 0, \bar{\nu}\}, \quad \bar{w}, \bar{\nu} > 0. \tag{18}$$

The meta-policy is to guarantee that its actions stay within their allowed ranges ($w \in [0, 1]$, $\nu > 0$). The used simple clipping at boundaries of (18) seems to suffice. We have no other demands on the actions. Thus, $\bar{\mathbb{A}} = \bar{\mathbb{A}}^i$ and $W = 0$ (meta-twin to w in (12)).

The last input to the meta-DM is the parameter of exploration ν. It makes no sense to choose a different value at the meta-level: the meta-action is its common value.

The appearance of $\bar{T}$, $\bar{w}$, $\bar{\nu}$ still preserves the danger of infinite regress. At present, it is cut by force and they are chosen heuristically. They, however, offer, the first step in a conceptual solution that: ▶ lets appear only meta-inputs that have a weak influence on results; ▶ tunes them via an adaptive minimization of miss-modelling error [17].

5. Experiments

This core section presents experiments. We have chosen a DM example with a heating system.

Common simulation options The simulated system is Markov with $|\mathbb{S}| = 15$ and $|\mathbb{A}| = 7$. It is created by learning the transition pd $\mathsf{p}(s_t|a_t, s_{t-1})$ on the simulated system generating 10^6 real values y_t stimulated by independently generated discrete actions in $\mathbb{A} := \{1, \ldots, 7\}$. The states $s_t \in \mathbb{S} := \{1, \ldots, 15\}$ are gained via an affine mapping of discretized values of the real-valued y_t generated by the equation ($y_0 = 1$)

$$y_t = 0.028y_{t-1} + 1.81y_{t-2} - 0.817y_{t-3} + 0.1a_t - 0.16a_{t-1} + 0.05\varepsilon_t.$$

There, ε_t is the white, zero-mean, normal noise with a unit variance. In all experiments with the Markov chain, the number of simulated epoch was 800. The seed of the random generator was fixed, and the initial state $s_0 = 1$. The initial guess of the entries of the array e (17) was 1.2. The horizon for dynamic programming is $h = 2$, which suffices when taking the outcome from the previous epoch as the initial guess of the stationary value function.

[3] We decided to note marks with a symbol g as grade, as m for mark is already used.

Experiments We present DM results without and with the user's control. DM without the user's control, it is the basic DM with no meta-level and preferences expressed by the ideal sets $\mathbb{S}^i$, $\mathbb{A}^i$ and by fixed options w, ν. DM with the user's control solves the basic DM supported by the second-layer implementing the solution of the meta-DM task with the dialogue with the user. The DM with the user's control gives the user the chance to express their satisfaction every ten steps, $\bar{T} = 10$. The satisfaction is quite subjective. It is demonstrated by presenting selected results for different users. We also present results with different fixed parameters w, ν to show how these parameters influence DM. In experiments with the user's control, these parameters are free and they are changed by the responses of the user. The changes of the free parameters w, ν are $\bar{w} = 0.1$ and $\bar{\nu} = 0.3$ (18). To compare the results impartially we use prices paid for deviation from the preferred behavior. The agreed prices are in Table 1 (common to all experiments) and Table 2 that suits to the preferred state $\mathbb{S}^i = \{8\}$. Other preferred states are priced similarly.

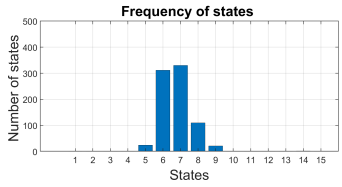
Table 1
The price paid for the individual action values

action	1	2	3	4	5	6	7
price	3	2	1	0	1	2	3

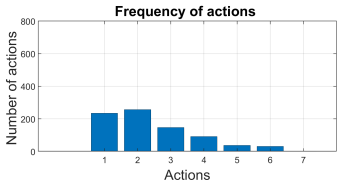
Table 2
The price paid for the individual state values when $\mathbb{S}^i = \{8\}$

state	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
price	4	3	3	2	2	1	1	0	1	1	2	2	3	3	4

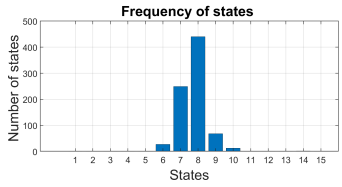
Experiment 1. It shows the results for the preferred state $\mathbb{S}^i = \{6\}$ and then for $\mathbb{S}^i = \{8\}$. No action is preferred, $\mathbb{A}^i = \mathbb{A}$. The free parameters are fixed, $w = 0, \nu = 1$.



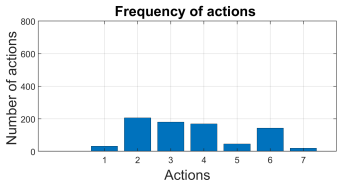
(a) States for $\mathbb{S}^i = \{6\}$



(b) Actions for $\mathbb{S}^i = \{6\}$



(c) States for $\mathbb{S}^i = \{8\}$



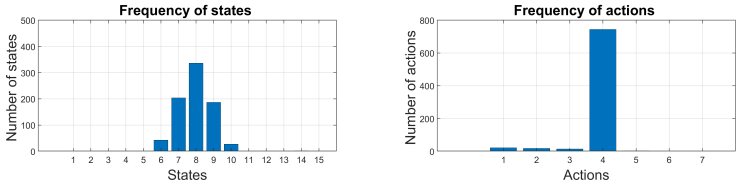
(d) Actions for $\mathbb{S}^i = \{8\}$

Figure 1: Exp. 1: states and actions in the basic DM for different preferences of states without any additional preference of actions $\mathbb{A}^i = \mathbb{A}$

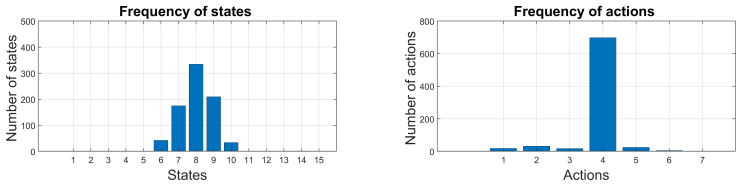
Discussion In the Fig. 1, we can see that the frequency of the preferred state $\mathbb{S}^i = \{8\}$ is pretty high. It occurs the most often. On the other hand the preferred state $\mathbb{S}^i = \{6\}$ does not occur the most often and its frequency is low. It is hard for the system to get the state $\mathbb{S}^i = \{6\}$. We will try to change the free parameters to improve results.

Experiment 2. It shows the results for $\mathbb{S}^i = \{8\}$ and $\mathbb{S}^i = \{6\}$ with the extra preference of actions $\mathbb{A}^i = \{4\}$. The weight $w = 0.3$ and the value $\nu = 1$ are fixed.

Discussion With the extra preference on actions the preferred states appear less often for both preferences, but still the preferred state appears the most often for the preference $\mathbb{S}^i = \{8\}$. On



(a) States for $\mathbb{S}^i = \{6\}$ (b) Actions for $\mathbb{S}^i = \{6\}$

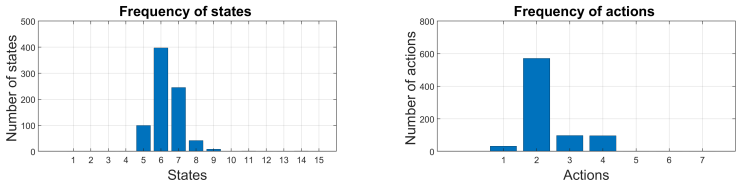


(c) States for $\mathbb{S}^i = \{8\}$ (d) Actions for $\mathbb{S}^i = \{8\}$

Figure 2: Exp. 2: states and actions in the basic DM for different preferences of states with an extra preference on actions $\mathbb{A}^i = \{4\}, w = 0.3$

the other hand the results for actions are pretty good for both. The preferred state $\mathbb{S}^i = \{6\}$ with the extra preference of action occurs even much less often, as expected, because these preferences contradict. But still the results are not bad. The user can be satisfied with the results because they could prioritized the results of actions over the poorer results concerning states.

Experiment 3. We would like to show how the parameter ν influence the results. We try improve our results for the preferred state $\mathbb{S}^i = \{6\}$, which gives worse results. We should be able to improve the results when there is no additional preference of actions. If the exploration parameter ν will be bigger, the selection of the action will not be uniform, but will be concentrated on the action, which guarantees the preferred state.

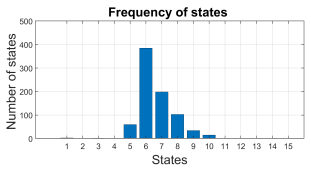


(a) States (b) Actions

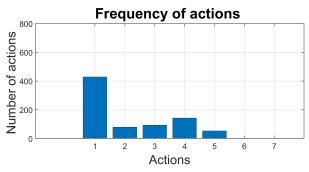
Figure 3: Exp. 3: states and actions in the basic DM for $\mathbb{S}^i = \{6\}, \mathbb{A}^i = \mathbb{A}, \nu = 5$

Discussion We can see, that we can improve the previous results via the parameter ν. We tried many values of ν, and present the best of for which the preferred state occurs the most often. The actions that cause the state 6 are around the action 2.

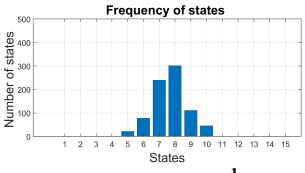
Experiment 4. We showed that the results of the experiments are influenced by the parameters w and ν. That is why we left these parameters to be free for the dialogue with the user. We choose their values according to the responses of the users. We would like to show, that they can get the desired results without any knowledge of DM and PE theories just using our algorithm. The users were instructed to want $\mathbb{S}^i = \{6\}$ without an additional preference of actions $\mathbb{A}^i = \mathbb{A}$ and then $\mathbb{S}^i = \{8\}$ with their preference of actions $\mathbb{A}^i = \{4\}$.



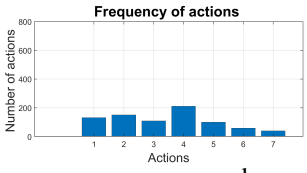
(a) States for the 1st user



(b) Actions for the 1st user

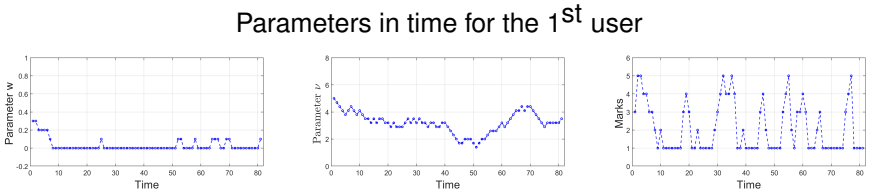


(c) States for the 2nd user

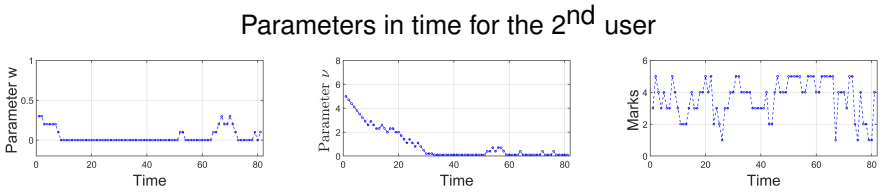


(d) Actions for the 2nd user

Figure 4: Exp. 4: states and actions for $\mathbb{S}^i = \{6\}, \mathbb{A}^i = \mathbb{A}$ in DM for the users



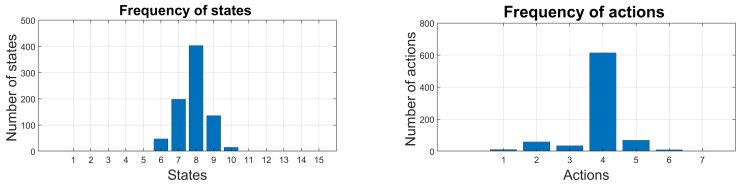
(a) Parameter w in time (b) Parameter ν in time (c) Marks in time



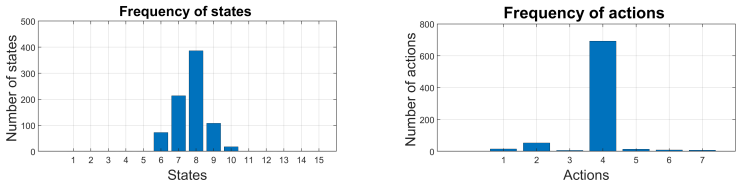
(d) Parameter w in time (e) Parameter ν in time (f) Marks in time

Figure 5: Exp. 4: The evolution of parameters for users with preferences $\mathbb{S}^i = \{6\}, \mathbb{A}^i = \mathbb{A}$.

Discussion In the Fig. 4 we can see that the results for $\mathbb{S}^i = \{6\}, \mathbb{A}^i = \mathbb{A}$ are much better for the 1st user. The occurrence of the preferred state is pretty high and it appears the most often. For the 2nd user the results are worse. The preferred state does not appear as much often and its occurrence is low. If we look at the evolution of the free parameters and marks Fig. 5, we can see that the courses of weight w are very similar and the weight is zero most of the time, which we assumed, because there is no additional preference. The courses of parameter ν are pretty different. The 1st user's ν is almost all the time between 2 and 5 and the 2nd user's declines to the value 0.1. That is because the 2nd user was strict with their marking, they were not so satisfied and that's why the algorithm tries to increase the exploration and find the results to satisfy the user. Because of that the results got worse and we got into dead end. So it really depends on the user's strategy. Thanks to the 1st user we can see that we can get good results but also thanks to the 2nd user we can see that if the algorithm gets bad feedback, it will worsen the results.

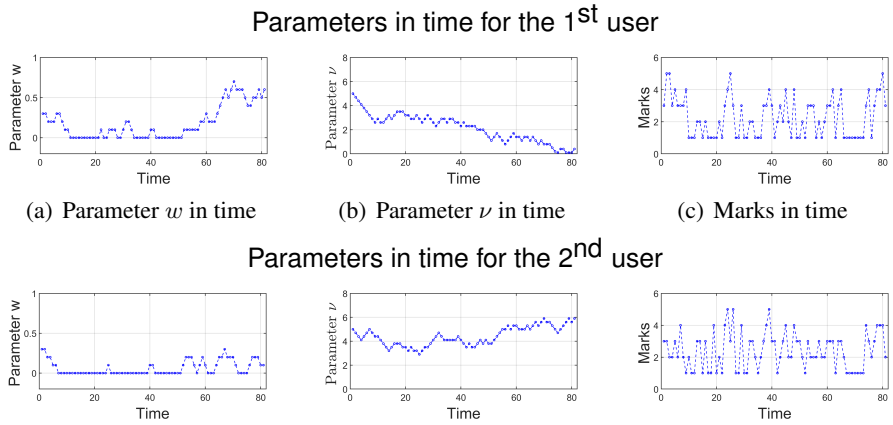


(a) States for the 1st user (b) Actions for the 1st user



(c) States for the 2nd user (d) Actions for the 2nd user

Figure 6: Exp. 4: states and actions for $\mathbb{S}^i = \{8\}, \mathbb{A}^i = \{4\}$ in DM for the user



(a) Parameter w in time (b) Parameter ν in time (c) Marks in time

(d) Parameter w in time (e) Parameter ν in time (f) Marks in time

Figure 7: Exp. 5: The evolution of parameters with preferences $\mathbb{S}^i = \{8\}, \mathbb{A}^i = \{4\}$.

Discussion For the preferences $\mathbb{S}^i = \{8\}, \mathbb{A}^i = \{4\}$, Fig. 5 the both users got great results. These preferences are not in a contradiction and as we could see above, it is easy for the system to reach this state. We can see from marking that both users were satisfied. The courses of the weight and ν differ. The 1st user's weight increase more and parameter ν decrease more than for the 2nd user. The 2nd user's parameters are more consistent but the frequencies of preferred state and action do not differ much.

6. Numerical results

Table 3 shows the prices paid for actions and states for $\mathbb{S}^i = \{8\}, \mathbb{A}^i = \{4\}$. For $\mathbb{S}^i = \{6\}, \mathbb{A}^i = \mathbb{A}$ the results can be judged in the same way.

We can see that the total price is the best (the lowest) for the 2nd user because they had the

Table 3
The price paid for actions and states in all experiments

Exp. no	Opted parameters	The price of actions	The price of states	Total price	Number of the preferred actions	Number of the preferred states
1.	$w = 0.0,\ \nu = 1$	1086	370	1456	170	440
2.	$w = 0.3,\ \nu = 1$	181	475	656	698	335
3.	1^{st} user	281	403	684	614	403
4.	2^{nd} user	219	420	639	692	386

top number of selections of the preferred action and the preferred state occurs also very often. They were satisfied as can be seen on the evolution of marks Fig 7. So we can say that it is the best result of our experiments. But the user are different so for someone it is a good results and for someone else not, because the prices that the users are willing to pay are individual. That we should keep in mind. The "objective" numerical comparison is of secondary importance. We also repeat that the users got the results they want without any knowledge of DM theory. It is also less time demanding to find a good policy via their feedback during the DM.

7. Concluding remarks

The paper presents the quantification of preferences within the fully probabilistic design of decision results. It provides the user's feedback that optimizes free parameters ν and w. It presents the experiments which show how the fixed parameters influence the DM. It compares the DM with and without the user's control. The algorithm does not need users any additional knowledge of the DM and PE theories.

The further research should:

- ✓ care about dimensionality curse connected with other wishes;
- ✓ add more free parameters, e.g., extensions of preferred sets of states and actions;
- ✓ address continuous systems;
- ✓ more specific application and real-system cases; etc.

These are hard tasks requiring more research to fill the gaps in the built universal DM theory, cf. **Motivation**.

Acknowledgement: *We would like to thank for the support of this paper by grant MŠMT SGS21/165/OHK4/3T/14 and by the department of Adaptive Systems of Institute of Information Theory and Automation of the Czech Academy of Sciences.*

References

[1] Barndorff-Nielsen, O.: Information and Exponential Families in Statistical Theory. Wiley, N.Y. (1978)
[2] Bellman, R.: Adaptive Control Processes. Princeton U. Press, NJ (1961)
[3] Berger, J.: Statistical Decision Theory and Bayesian Analysis. Springer (1985)

[4] Boutilier, C.: A POMDP formulation of preference elicitation problems. In: Proc. of the 18th National Conf. on AI, AAAI-2002. pp. 239–246. Edmonton, AB (2002)
[5] Boutilier, C., et al: A constraint-based approach to preference elicitation and decision making. In: Proc. of AAAI Spring Symp. on Qualitative Decision Theory, pp. 19–28. AAAI Press (1997)
[6] Brace, I.: Questionnaire design. How to plan, structure and write survey material for effective market research. Kogan Page, London (2004)
[7] Braziunas, D., Boutilier, C.: Elicitation of factored utilities. AI Magazine **29**(4), 79–91 (2008)
[8] Chen, L., Pu, P.: Survey of preference elicitation methods. Tech. Rep. IC/2004/67, HCI Group Ecole Politechnique Federale de Lausanne, Switzerland (2004)
[9] Dwivedi, Y., et al: Artificial intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy
[10] Feldbaum, A.: Theory of dual control. Autom. Remote Control **22**, 3–19 (1961)
[11] Greco, C., Suglia, A., Basile, P., Semeraro, G.: Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems. In: Conf. of the Italian association for artificial intelligence. pp. 372–386. Springer (2017)
[12] Guy, T., Kárný, M., Rios-Insua, D., Wolpert, D. (eds.): Proc. of the NIPS 2016 Workshop on Imperfect Decision Makers, vol. 58. JMLR (2016)
[13] Guy, T., Kárný, M., Wolpert, D.: Decision Making and Imperfection, vol. 474. Springer, Berlin (2013)
[14] Guy, T., Kárný, M., Wolpert, D.: Decision Making: Uncertainty, Imperfection, Deliberation and Scalability, vol. 538. Springer, Switzerland (2015)
[15] Hutter, M.: Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability. Springer, Berlin, Heidelberg, N.Y. (2005)
[16] Kárný, M.: Fully probabilistic design unifies and supports dynamic decision making under uncertainty. Inf. Sci. **509**, 104–118 (2020)
[17] Kárný, M.: Towards on-line tuning of adaptive-agent's multivariate meta-parameter. Int. J. of Machine Learning and Cybernetics **12**(9), 2717–2731 (2021)
[18] Kárný, M., Guy, T.: Preference elicitation within framework of fully probabilistic design of decision strategies. In: IFAC Int. Workshop on Adaptive and Learning Control Systems. vol. 52, pp. 239–244 (2019)
[19] Kárný, M., Siváková, T.: Agent's feedback in preference elicitation. In: 2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS). pp. 421–429 (2021). https://doi.org/10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00073
[20] Mesbah, A.: Stochastic model predictive control with active uncertainty learning: A survey on dual control. Annual Reviews in Control **45**, 107 – 117 (2018)
[21] Müller, V., Bostrom, N.: Future Progress in Artificial Intelligence: A Survey of Expert Opinion, pp. 555–572. Springer Int. Pub. (2016)
[22] Šindelář, J., Vajda, I., Kárný, M.: Stochastic control optimal in the Kullback sense. Kybernetika **44**(1), 53–60 (2008)