Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2018

Zvezek A

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY - IS 2018

Volume A

Slovenska konferenca o umetni inteligenci

Slovenian Conference on Artificial Intelligence

Uredili / Edited by
Mitja Luštrek, Rok Piltaver, Matjaž Gams

http://is.ijs.si

8.–12. oktober 2018 / 8–12 October 2018
Ljubljana, Slovenia

Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2018

Zvezek A

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY – IS 2018

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredili / Edited by

Mitja Luštrek, Rok Piltaver, Matjaž Gams

http://is.ijs.si

8.–12. oktober 2018 / 8–12 October 2018
Ljubljana, Slovenia

Uredniki:

Mitja Luštrek
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Rok Piltaver
Celtra, d. o. o. in
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Matjaž Gams
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2018

Informacijska družba
ISSN 2630-371X

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=31856679
ISBN 978-961-264-135-1 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2018

Multikonferenca Informacijska družba (http://is.ijs.si) je z enaindvajseto zaporedno prireditvijo osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Letošnja prireditev se ponovno odvija na več lokacijah, osrednji dogodki pa so na Institutu »Jožef Stefan«.

Informacijska družba, znanje in umetna inteligenca so še naprej nosilni koncepti človeške civilizacije. Se bo neverjetna rast nadaljevala in nas ponesla v novo civilizacijsko obdobje ali pa se bo rast upočasnila in začela stagnirati? Bosta IKT in zlasti umetna inteligenca omogočila nadaljnji razcvet civilizacije ali pa bodo demografske, družbene, medčloveške in okoljske težave povzročile zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so notranji in zunanji konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali 11 odličnih neodvisnih konferenc. Predstavljenih bo 215 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica, ki se ponaša z 42-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2018 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Kognitivna znanost
- Odkrivanje znanja in podatkovna skladišča – SiKDD
- Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI
- Delavnica AS-IT-IC
- Soočanje z demografskimi izzivi
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Delavnica za elektronsko in mobilno zdravje ter pametna mesta
- Vzgoja in izobraževanje v informacijski družbi
- 5. študentska računalniška konferenca
- Mednarodna konferenca o prenosu tehnologij (ITTC)

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, Slovensko društvo za umetno inteligenco (SLAIS), Slovensko društvo za kognitivne znanosti (DKZ) in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in institucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V letu 2018 bomo šestič podelili nagrado za življenjske dosežke v čast Donalda Michieja in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel prof. dr. Saša Divjak. Priznanje za dosežek leta bo pripadlo doc. dr. Marinki Žitnik. Že sedmič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono letos prejme padanje državnih sredstev za raziskovalno dejavnost, jagodo pa Yaskawina tovarna robotov v Kočevju. Čestitke nagrajencem!

Mojca Ciglarič, predsednik programskega odbora

Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2018

In its 21st year, the Information Society Multiconference (http://is.ijs.si) remains one of the leading conferences in Central Europe devoted to information society, computer science and informatics. In 2018, it is organized at various locations, with the main events taking place at the Jožef Stefan Institute.

Information society, knowledge and artificial intelligence continue to represent the central pillars of human civilization. Will the pace of progress of information society, knowledge and artificial intelligence continue, thus enabling unseen progress of human civilization, or will the progress stall and even stagnate? Will ICT and AI continue to foster human progress, or will the growth of human, demographic, social and environmental problems stall global progress? Both extremes seem to be playing out to a certain degree – we seem to be transitioning into the next civilization period, while the internal and external conflicts of the contemporary society seem to be on the rise.

The Multiconference runs in parallel sessions with 215 presentations of scientific papers at eleven conferences, many round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal, which boasts of its 42-year tradition of excellent research publishing.

The Information Society 2018 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Cognitive Science
- Data Mining and Data Warehouses - SiKDD
- International Conference on High-Performance Optimization in Industry, HPOI
- AS-IT-IC Workshop
- Facing demographic challenges
- Collaboration, Software and Services in Information Society
- Workshop Electronic and Mobile Health and Smart Cities
- Education in Information Society
- 5th Student Computer Science Research Conference
- International Technology Transfer Conference (ITTC)

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, Slovenian Artificial Intelligence Society (SLAIS), Slovenian Society for Cognitive Sciences (DKZ) and the second national engineering academy, the Slovenian Engineering Academy (IAS). On behalf of the conference organizers, we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the sixth year, the award for life-long outstanding contributions will be presented in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to Prof. Saša Divjak for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to Assist. Prof. Marinka Žitnik. The information lemon goes to decreased national funding of research. The information strawberry is awarded to the Yaskawa robot factory in Kočevje. Congratulations!

Mojca Ciglarič, Programme Committee Chair

Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Blaž Mahnič
Jani Bizjak
Tine Kolenik

Programme Committee

Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Mitja Luštrek
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah

KAZALO / TABLE OF CONTENTS

Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2018

Zvezek A

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY – IS 2018

Volume A

Slovenska konferenca o umetni inteligenci
Slovenian Conference on Artificial Intelligence

Uredili / Edited by

Mitja Luštrek, Rok Piltaver, Matjaž Gams

http://is.ijs.si

11.–12. oktober 2018 / 11–12 October 2018
Ljubljana, Slovenia

PREDGOVOR

V letu 2018 smo ponovno priča neverjetnim dosežkom umetne inteligence. Tako je bila letos poleti v Stockholmu največja svetovna konferenca na področju umetne inteligence IJCAI združena z evropsko ECAI, s čimer je imela 37 % več prispevkov kot prejšnje leto. Združeni konferenci sta skupno pritegnili preko 6.000 udeležencev. Približno polovica vseh prispevkov je bila kitajskih, pol manj je bilo evropejskih in ameriških. Velesile se zavedajo, da je področje umetne inteligence eno izmed ključnih, zato tako Putin kot Trump in Ši Džinping intenzivno povečujejo sredstva za njen razvoj, Evropa pa jih bo v prihodnjih letih nekajkrat povečala.

Dnevno umetna inteligenca sprejme neverjetnih 10 bilijonov odločitev. Samo v lanskem letu je bilo dosežkov umetne inteligence toliko, da jih lahko omenimo le majhen delež. Na področju varnosti po svetu uporabljajo sistem, ki vsak dan izdela nov urnik obhodov varnostnikov po letališčih, pristaniščih in podobnih okoljih. Kjer so ti sistemi uporabljeni, je izmerjena bistveno večjo učinkovitost. V skrbi za okolje so raziskovalci pod vodstvom prof. Tambeja (med njimi je bil tudi naš doktorand dr. Kaluža) tovrstne sisteme podarili 60 rezervatom po svetu, da se bodo uspešneje upirali krivolovcem. Leta 2015 so sistemi na osnovi globokih nevronskih mrež začeli dohitevati ljudi pri prepoznavanju vidnih nalog in danes jih že prekašajo, npr. pri prepoznavanju malignih tkiv. Pri nekaterih nalogah, recimo pri ostrenju slike (zaradi dežja, megle, snega itd.), so sistemi osemkrat boljši od ljudi. Ob tem se seveda pojavlja tudi strah, vendar če na diagnozo, ali imate raka ali ne, čakate nekaj tednov, v ZDA pa to diagnozo postavi umetna inteligenca v nekaj minutah in to bolje kot katerikoli zdravnik – kaj pravite, ali bi jo uvedli tudi pri nas? V Sloveniji potrebno znanje že imamo, zatika se le pri vpeljavi. Pri nekaterih posegih, kot je recimo presaditev organov, so sistemi umetne inteligence že desetletja v uporabi in so rešila na tisoče življenj. Nekateri sistemi so tudi novejši – letos so tako vpeljali prvi inteligentni sistem, ki ugotavlja diabetes na podlagi pregleda oči, prav tako tudi prvi program za ugotavljanje abnormalnosti prsnega koša pri slikanju. Hiter razvoj je najbolj znan pri avtonomni vožnji – danes imajo povprečni avtomobili kar nekaj avtonomnih inteligentnih funkcij, modernejši (npr. Tesla) pa vozijo praktično sami in jih nadziramo samo še v nenavadnih situacijah. Nesreč avtonomnih vozil je približno stokrat manj kot tistih človeških voznikov, medijski odziv nanje pa je pogosto veliko bolj poročan in zato napihnjen.

Mnoge zanimive dosežke umetne inteligence predstavljamo tudi na Slovenski konferenci o umetni inteligenci (SKUI), ki je naslednica konference Inteligentni sistemi in je sestavni del multikonference Informacijska družba že od njenega začetka leta 1997. Slovensko društvo za umetno inteligenco, ki letos praznuje že 26. obletnico, SKUI šteje za svojo konferenco. Letos je bilo sprejetih 17 prispevkov. Kot pretekla leta jih je največ z Instituta »Jožef Stefan«, nekaj pa jih je prispevala Fakulteta za računalništvo in informatiko, ki ima skupaj z Institutom vodilno vlogo pri raziskavah umetne inteligence v Sloveniji. Upamo, da bo prispevkov iz industrije in nasploh izven Instituta prihodnja leta še več, saj je ključen cilj SKUI povezovanje vseh slovenskih raziskovalcev umetne inteligence, čeprav na konferenci niso nič manj dobrodošli tudi prispevki iz tujine.

Mitja Luštrek, Rok Piltaver, Matjaž Gams

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Mitja Luštrek

Rok Piltaver

Matjaž Gams

Marko Bohanec

Tomaž Banovec

Cene Bavec

Jaro Berce

Marko Bonač

Ivan Bratko

Dušan Caf

Bojan Cestnik

Aleš Dobnikar

Bogdan Filipič

Nikola Guid

Borka Jerman Blažič

Tomaž Kalin

Marjan Krisper

Marjan Mernik

Vladislav Rajkovič

Ivo Rozman

Niko Schlamberger

Tomaž Seljak

Miha Smolnikar

Peter Stanovnik

Damjan Strnad

Peter Tancig

Pavle Trdan

Iztok Valenčič

Vasja Vehovar

Martin Žnidaršič

Monitoring Bumblebee Daily Activities Using Microphones

Anton Gradišek
Jožef Stefan Institute
Ljubljana, Slovenia
anton.gradisek@ijs.si

Nicolas Cheron
Polytech Paris Sorbonne
Paris, France

David Heise
Lincoln University
Jefferson City, MO, United States

Candace Galen
University of Missouri
Columbia, MO, United States

Janez Grad
Faculty of Administration,
University of Ljubljana
Ljubljana, Slovenia

ABSTRACT

We present initial results of the study where we used microphones, placed in front of nest boxes, to monitor daily foraging activity of bumblebees. Sound recordings were analyzed using a custom-made computer algorithm which detects flight buzzing sounds coming from arrivals or departures of individual bees. In addition, the algorithm distinguishes between arrivals and departures. We show examples of daily activities for three species (*B. pascuorum*, *B. humilis* and *B. hypnorum*), each was monitored over the course of one day. This paper presents initial results of a longer study where we plan to systematically investigate the activities of bumblebees in various circumstances.

Keywords

Bumblebees, foraging activity, sound analysis

1. INTRODUCTION

Bumblebees (genus *Bombus* from the bee family Apidae) are an important group of wild pollinators. Due to different morphology and lifestyle, when pollinating plants, they are often more effective than honeybees – they are able to go foraging in rainy and cold weather, and in addition, they use a special technique, called buzz-pollination, to extract pollen from plants such as tomatoes. In addition to pollination in the wild, this makes bumblebees important players in greenhouse agriculture.

Pollinator monitoring, as well as monitoring of wild pollinators, is of high interest to agronomists, ecologists, and experts in the field of conservation. In studies of bumblebee activity, currently the most typical approaches are observations and capturing. Capturing is problematic as it includes removal of individuals from the environment. Sometimes, bumblebees are also studied in laboratory conditions, by raising an entire colony in a lab, which typically involves commercially available bumblebee species. One can expect that the behavior in a laboratory is not identical to that in a natural environment. A better approach to controlled studies is introducing the bumblebees into special nest boxes outside. This allows us to monitor them in a near-natural environment.

In this paper, we present the first results of a study where we used microphones to monitor bumblebee daily foraging activities. These activities are important to monitor as they provide a direct insight into pollination service. Using a microphone (recording sounds) is clearly advantageous from personal monitoring (such as in Grad et al. [1]) since it is continuous and allows us to monitor several sites simultaneously (using several microphones). Bumblebee buzzing sounds have been studied before, though with a different focus. Gradišek et al. [2] developed a machine-learning-based algorithm to recognize individual species and types (queen or worker) of bumblebees based on flight buzzing sound. Heise et al. [3] developed an algorithm to detect bee buzzes from field recordings. In our case, the task was to detect arrivals and departures of bumblebees from the nest boxes (both of which result in buzzes recorded by the microphone), therefore the algorithm was optimized for this task. We discuss the algorithm and show some initial results.

2. MATERIALS AND METHODS

2.1 Data Collection

USB stick microphones dB9PRO VR1.0 [4] were used for sound recordings. Each microphone has 8 GB of flash memory, which gives it nominal storage capacity above 90 h. Sound is recorded at 48 kHz with a 192 Kbps bit rate. After each charging, a microphone can record for around 10 h. Microphones were placed in front of nest box entrances in order to record arrivals and departures. In the following, we demonstrate the results for three different bumblebee families, each of them monitored over the course of one day. The details of the investigated families are listed in Table 1. In all cases, the microphones were set around 8 am. For *B. pascuorum*, the microphone kept recording until the battery lasted while for the other two families, on the following day, the microphones were collected around 6 pm as the weather deteriorated.

Table 1. Bumblebee families studied

Species	Date	No. of workers	Weather
B. pascuorum	28 May 2018	10	14 – 28 °C, morning fog, sunny during the day, storms in the evening
B. humilis	29 May 2018	20	16 – 26 °C, morning partially cloudy, light rain after 4 pm, heavy rain after 6 pm
B. hypnorum		30	

2.2 Sound Recording Analysis

The flowchart of the algorithm is shown in Figure 1. The algorithm was inspired by that of Heise et al. [3], but simplified in order to work faster as recordings of arrivals and departures in front of a single nest box are typically cleaner than those from a microphone located in the field. Our preliminary analysis was carried out using the Audacity software while a more detailed analysis was done in Matlab, using in-built packages and own code. In each recording, we manually labelled around 10 buzzes at the beginning in order to optimize the thresholds for the algorithm (described in the following). In addition, we manually

labelled the entire recording of *B. pascuorum* in order to evaluate the performance of the algorithm.

Preliminary inspection showed that the microphones recorded bumblebee buzzes well, while also recording a series of noises from the environment, such as passing traffic or human speech. Sometimes, these sounds can be louder than the buzzes themselves. The task of our algorithm is therefore to detect loud events and to check whether they are buzzes or noise. For positively identified buzzes, we next determine whether they correspond to arrival or departure of the bumblebee.

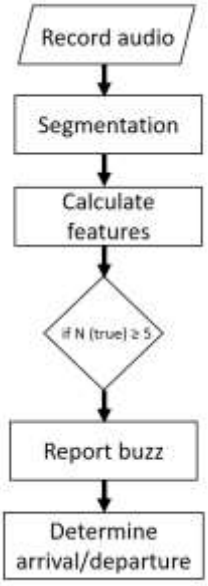


Figure 1. Flowchart of the buzz detecting algorithm

The algorithm is the following:

1. The recording, typically several hours long, is cut to segments of 5 seconds. This length was determined to be sufficiently long to contain the entire event while short enough to contain only a single event.

As the recording is cut into segments, there is a possibility that the cutting occurs in the middle of a buzz. To identify such cases, a special function first checks whether the peak amplitude occurs in the first second of the segment. If it does, it checks the last second of the previous sample, if the peak amplitude occurs in the last second there (indicating that the buzz was cut to two segments). In such cases, both segments are merged into a single segment and the analysis proceeds as described in the following (with only one buzz counted).

2. For each segment, we calculate the Fourier transform of the time-domain signal to obtain the spectra (frequency domain, spectrum amplitude as a function of frequency)

3. For each segment, we calculate seven independent Boolean features that we use to decide whether the segment contains an event or not. In the following, the natural frequencies (the frequency that a bumblebee flaps its wings during flight) are optimized for *B. pascuorum*. For species with significantly different natural frequencies (see [2]), we modify the boundaries. The feature thresholds are set for each family as well, based on some manually inspected events (about 10 – 15 events at the beginning of each recording).

a) We calculate the average amplitude of the segment (which can generally be done either in time or in frequency domain). If the amplitude is larger than a manually determined threshold value, this is a possible event (e.g. true).

b) The natural frequency is f = 180 Hz. We count the number of peaks between 160 and 200 Hz (using the *findpeaks* function). If the number of peaks is smaller than the threshold, this is considered a buzz, otherwise we are dealing with noise.

c) We calculate the ratio of average amplitudes around the proposed peak (average amplitude value on the interval 160 – 200 Hz) and below it (60 – 120 Hz). If the ratio is larger than the threshold, this can be a true buzz, otherwise it is likely to be noise.

d) Similar to feature c), we check the ratio of the average amplitude around the proposed peak and above it (220 – 280 Hz).

e) Similar to feature b), we look for a peak at double natural frequency (first harmonic), looking at the interval (2 * f – 20 Hz, 2 * f + 20 Hz).

f, g) We follow the same procedure as for features c) and d), just at the frequency of first harmonic and correspondingly higher interval boundaries.

If five or more features return "true", we consider the segment to contain a buzz. This criterion was determined on a series of manually labelled events in order to maximize the accuracy.

Once we know that a segment contains a buzz, we can determine whether it corresponds to arrival or departure. This part is carried out using signal in time domain. Figure 2 shows examples of both events. They are roughly symmetric in shape, which is reasonable given the dynamics of the process. When a bumblebee arrives to the nest box, it is initially far from the microphone and then gets closer – resulting in an increasing signal amplitude. When it lands, it stops buzzing, thus a sharp drop in signal. For departure, the bumblebee starts flying (sharp jump) and then flies away from the microphone (gradual drop in amplitude).

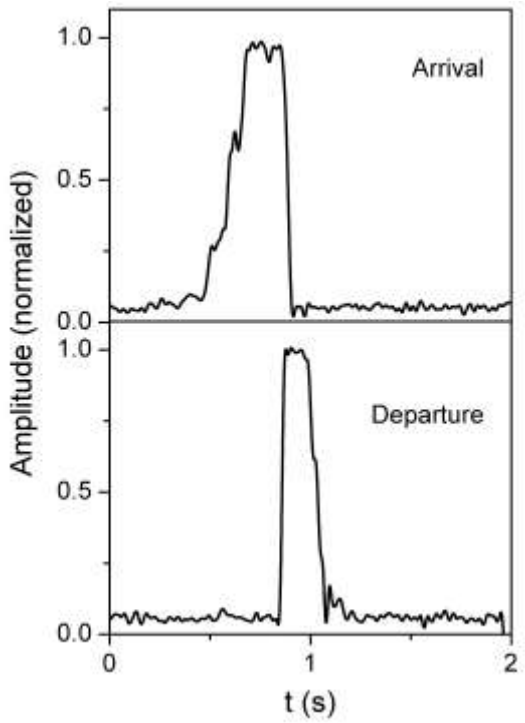


Figure 2. Signal envelope in time domain for arrivals (top) and departures (bottom)

To classify the event as an arrival or a departure, we do the following. First, we calculate the signal envelope and smooth it to reduce the noise. Such envelopes can be seen in Figure 2. Next, we use the *findpeaks* function to identify peaks and we calculate the maximum absolute difference between two consecutive peaks. We call this a "drop". Looking at Figure 2, we see that the drop appears at the end of arrival and at the beginning of departure. By integrating the area before and after the drop over a chosen interval, we can determine the arrival or departure.

As each segment has a timestamp, we are able to plot histograms of either arrivals or departures of bumblebees throughout several hours.

3. RESULTS AND DISCUSSION

Figure 3 shows three histograms for bumblebee departures on a chosen day, on hourly basis.

Figure 3 only shows the number of departures. The numbers for arrivals are very similar, as is to be expected. These three histograms provide a good insight into the daily dynamics of each family. Different species have different foraging habits, for example, *B. pascuorum* were mostly active around noon and in the afternoon while less active in the morning. On the other hand, *B. hypnorum* were more active after 3 pm. Light rain at 4 pm made the *B. humilis* workers stay inside but it did not affect *B. hypnorum*. Of course, as these are initial results on limited datasets, a longer data collection will be required to investigate the dynamics as the family develops over the course of several months.

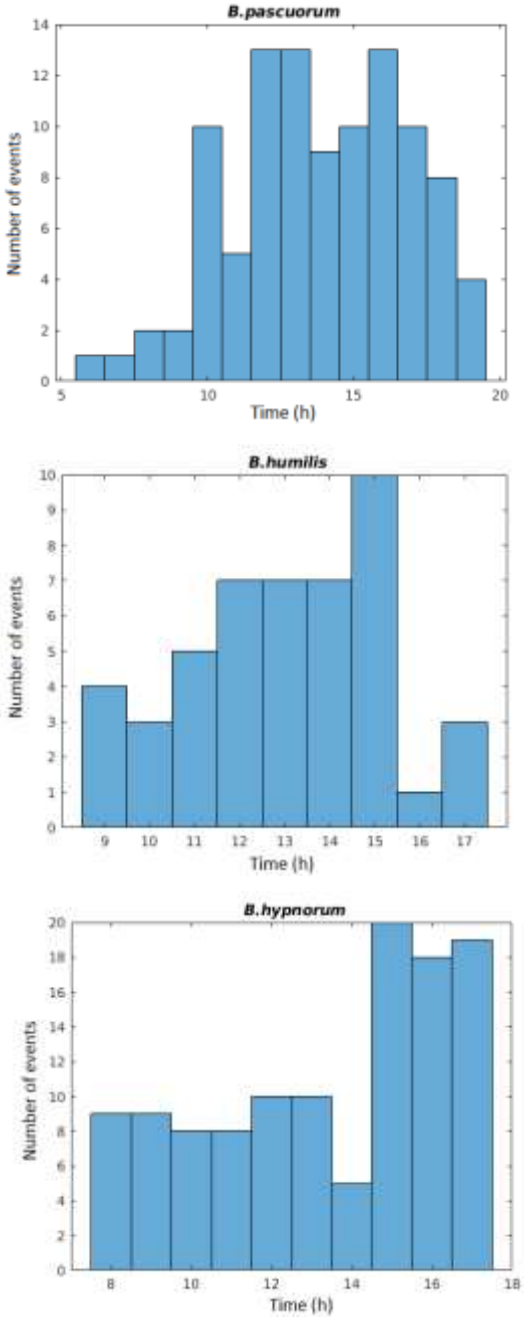


Figure 3. Histograms for number of departures (reflecting daily activity) for three bumblebee families, as described in Table 1.

To check the accuracy of the algorithm, we tested it on the manually-labelled recording (*B. pascuorum*). Out of 180 actual events (counting arrivals and departures together, P), our algorithm correctly detected 171 events (TP), 9 detected events were not buzzes (FP), and 9 events were missed (FN). Based on this, we can determine the algorithm sensitivity, TP / P = TP / (TP + FN) = 0,95 and precision TP / (TP + FP) = 0,95. Clearly, this estimate is based on a single long recording and may vary for other conditions (different species or different structure of noise).

4. CONCLUSION

We demonstrate that microphones can be used as a simple tool to study bumblebee foraging activity, as opposed to personal monitoring. The algorithm we developed detects potential buzzes and classifies them as either arrivals or departures. Compared to the performance of a human manually labelling the buzzes in the recording, the algorithm works with 95 % sensitivity and 95 % precision, which we consider sufficient for meaningful results. In future, we plan to study several bumblebee families throughout the year to investigate the effects of the weather, temperatures, family size, and other parameters on foraging activity.

5. ACKNOWLEDGMENTS

The research is partially funded by National Geographic Grant NGS-282-18 (to C. Galen).

6. REFERENCES

[1] Grad, J., Gradišek, A., Gams, M., Čmrlji: pašna dejavnost in zvok brenčanja, Poklukarjevi dnevi, Book of abstracts, 2016

[2] Gradišek, A., Slapničar, G., Šorn, J., Luštrek, M., Gams, M. and Grad, J. 2016: Predicting species identity of bumblebees through analysis of flight buzzing sounds. Bioacoustics, 26 (1), 63-76

[3] Heise, D., Miller-Struttmann, N., Galen, C., Schul, J. 2017: Acoustic Detection of Bees in the Field Using CASA with Focal Templates, 2017 IEEE Sensors Applications Symposium (SAS), Glassboro, NJ, pp. 1-5.

[4] http://db9pro.com/

Reconstructing PPG Signal from Video Recordings

Gašper Slapničar
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
gasper.slapnicar@ijs.si

Andrejaana Andova
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
andrejaana.andova@ijs.si

Erik Dovgan
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
erik.dovgan@ijs.si

Mitja Luštrek
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT

Physiological signals give important insight regarding someone's health. It would be in the interest of people to monitor such signals without any wearable devices. We used RGB camera recordings of faces to reconstruct the PPG signal, which can be used to monitor many physiological signals such as heart rate, breathing rate, blood pressure, etc. A deep learning method was developed to enhance existing state-of-the-art methods. This method uses the output of an existing method as an input into a LSTM neural network, which substantially improves the reconstruction of PPG.

Keywords

remote PPG, signal processing, deep learning

1. INTRODUCTION

Physiological signals, such as photoplethymogram (PPG), are traditionally measured using wearable devices like cuffs and wristbands. While such devices are rather unobtrusive, it would be preferable to omit them completely. This can be achieved with the use of contact-free devices such as RGB cameras, which can blend into the environment, allowing for remote physiological signal reconstruction. An example group for whom such a system would be useful are people with profound cognitive impairment, who are the subjects participating in the INSENSION project[1], for which our system is being developed.

This paper aims to compare and enhance existing approaches for reconstructing PPG from video data, i.e., remote PPG (rPPG). The PPG signal describes the changes of blood volume in the skin tissue, which corresponds to the heart periodically pushing the blood towards the periphery of the body with each beat. When skin tissue gets filled with blood, it becomes slightly darker and absorbs more light. The light source in contact sensors (e.g., wristbands) is concentrated and constant (a light emitting diode – LED) and enables high-quality PPG reconstruction. Reconstructing rPPG from a camera recording is more difficult as the source of light is most commonly the sun or the lighting of a room. This makes such an approach more sensitive to environmental conditions and less accurate compared to contact sensors. rPPG reconstruction would allow for estimation of several physiological parameters, such as heart rate (HR), breathing rate, heart-rate variability and blood pressure, without a wearable device.

The rest of the paper is organized as follows. Section 2 reviews the related work. The methods for reconstructing the PPG signal are described in Section 3, while the experiments and results are discussed in Section 4. Finally, Section 5 concludes the paper with ideas for future work.

2. RELATED WORK

There are two main approaches for reconstructing rPPG, which are based on different underlying physiological phenomena.

The first approach focuses on variations in blood volume, which is reflected in the changes of the skin color. To detect the variations of blood volume using non-contact sensors (camera), tiny changes in RGB intensity of the skin pixels between two sequential video frames are analyzed. For example, Poh et al. [9, 10] applied independent component analysis (ICA) on the RGB color signals, which were computed as the average of the red, green and blue intensity of all the skin pixels over time. They then chose the most PPG-like resulting signal. Lewandowska et al. [5] used principal component analysis (PCA) instead of ICA to reconstruct the PPG signal. Haan et al. [2] reconstructed the PPG signal simply by calculating a specific linear combination of the obtained RGB traces. Other approaches do not calculate the average of all skin pixels, but treat each skin pixel independently. For example, Wang et al. [11] tracked the variation of color in each skin pixel independently and chose the most PPG-like signal afterwards. The changes of the skin pixel values were also tracked to reconstruct the PPG signal [12]. Petil et al. [7] used the basic RGB signals as inputs to a

[1] http://www.insension.eu

neural network to reconstruct various physiological signals. Another example by Wu et al. [13] amplified all the color changes of the facial pixels to follow the blood flow in these pixels. Although the presented methods seem promising, an independent evaluation conducted by Heusch et al. [3] on a publicly available dataset showed that they are not accurate enough to be used in real-world scenarios. More precisely, this evaluation re-implemented three state-of-the-art methods for reconstructing PPG from RGB cameras, and the results showed that there is a very low correlation between the reconstructed and ground-truth PPG.

The second approach for PPG reconstruction from video analyzes the small head movements that are induced by the blood being pumped into the head. Such a study was conducted by Balakrishnan et al. [1], however, it should be noted that such movements are very subtle and might not be detectable with a low quality camera, imposing an additional hardware requirement on this approach.

3. RECONSTRUCTING PPG WITH VISION-BASED METHODS

This section presents the developed deep-learning-based method for reconstructing the PPG signal from video data. This method enhances the signal reconstruction of an existing state-of-the-art method, as none of them were satisfactory. We first present the state-of-the-art methods used in the evaluation. All of these methods have a similar preprocessing step, which is presented in Section 3.1. The steps specific for each ofcoh these methods are presented in Section 3.2. Finally, in Section 3.3 we present the developed method that takes as input the PPG reconstructed with an existing method and returns an enhanced reconstruction of PPG.

3.1 Preprocessing of Video Data

The first preprocessing step consists of the detection of the subject's face as the "region of interest" (ROI). For detecting the face, we used the Haar cascades, implemented in the OpenCV library[2]. More precisely, the video was segmented into individual frames and only the selected face ROI was cropped from each frame.

The second step of video preprocessing aims at discriminating between skin and non-skin pixels. For this purpose, we implemented two classification methods. The first method transforms the RGB color space into the YCbCr color space, which contains less redundant information. Pixel values are then classified as either skin or non-skin using thresholds. This method is fast, simple and works well on the test dataset, however it probably does not generalize well to datasets where the degree of variation of skin colors and shades is higher. The second method applies one-class support vector machines (SVM) to classify the skin pixels. It learns a decision function for novelty detection from positive examples (corresponding to skin pixels), which are obtained from the forehead region of the first three frames of each video. New data is then classified as similar (skin) or different (not skin) to the training set. The forehead region is detected as the facial area of fixed dimensions above the eyes, which can be easily detected using OpenCV. The SVM method produces worse results than the threshold-based method, but generalizes well for various skin colors and shades. Both skin classification methods incorrectly classify some of the non-skin pixels as skin. To avoid false positives, we selected only the pixels that are most likely to actually be skin. We did this by calculating the mean value of all the skin pixels returned by the classifier and then removing the outlier pixels with respect to the mean in the YCbCr color space.

[2] `https://opencv.org`

3.2 State-of-the-Art Methods

We have evaluated a set of state-of-the-art methods. These methods can be classified as color-based or movement-based as described in Section 2.

Poh-et-al method: This is a color-based method that sequences the mean value of the red, green and blue intensity of all the skin pixels to create three different color traces. Since all the traces contain some information about the blood flow, it first normalizes them and then transforms them with ICA using the FastICA algorithm [4, 9, 10]. This method returns three signals, so we choose the one with most frequencies in the range [0.6 Hz, 4 Hz], i.e., the frequency range of PPG. This is done by analyzing the power spectrum of each output signal.

Haan-et-al method: This is also a color-based method which, similarly to the previous method, uses the mean of the red, green and blue intensity of all the skin pixels [2]. It then creates a linear combination from the red, green and blue traces, resulting in two new traces X and Y, calculated as: $X = 3R - 2G$; $Y = 1.5R + G - 1.5B$. The X and Y traces are then filtered and combined to reconstruct the PPG signal. In our experiments, we used the method implementation from the BOB library[3].

Wang-et-al method: This color-based method uses all the skin pixels from an individual frame to define the color space of frames [12]. By tracking the changes in this space, we reconstruct the PPG signal. To accomplish this, a covariance matrix is computed. This covariance matrix changes for each frame due to the blood flowing into the skin. By calculating the eigenvectors of the original frame and the eigenvalues of the covariance matrix, we get a representation of the color space for the skin pixels. The rotation between two eigenvectors of sequential frames represents the changes of the color space. This rotation is also related to different relative PPG contributors. Therefore, by concatenating the rotation between the first opposing to the second and the third eigenvector, PPG-like traces are retrieved. The eigenvalues are also influenced by the pulsatile blood and are thus used to normalize the PPG-like signals. As for the previous method, we also used the method implementation from the BOB library[3].

Balakrishnan-et-al method: In contrast to the previously presented methods, this is a motion-based method, since it focuses on the oscillations of the head [1]. This method does not need to detect skin pixels, therefore, the second step of data preprocessing is skipped. To reconstruct the oscillations, the Lucas-Kanade flow-tracking al-

[3] `https://pypi.org/project/bob.rppg.base`

gorithm [6] is applied, which tracks the flow of the head movements in the vertical direction. The oscillation signals are then filtered using a band-pass filter with the frequency interval [0.6 Hz, 4 Hz], i.e., the frequency range of PPG. Afterwards, PCA is applied to select the most PPG-like signal.

3.3 Deep-Learning-Based Method

We developed a new method for reconstructing the PPG signal, which takes the PPG reconstructed by an existing method as the input, and outputs an improved reconstruction of the PPG signal. To achieve this, it applies deep learning, which has recently shown superior performance in machine learning on many domains compared to traditional approaches.

To build the deep learning model, we used a Long-Short Term Memory (LSTM) network [8]. Its architecture comprised two LSTM layers and one fully-connected layer. The window length was set to 100 samples, i.e., five seconds. Each layer had 50 LSTM units, each taking input of length 100 and returning output of the same length, as shown in Figure 1. The output of the Wang-et-al method [12] has been selected as the input to the LSTM network.

4. EXPERIMENTS AND RESULTS

In order to evaluate the quality of methods described in Section 3.2 and our method, the reconstructed PPG was compared with the ground truth obtained with a fingertip PPG sensor.

4.1 Materials and Experimental Setup

The existing methods and the developed method have been evaluated on the COHFACE dataset[4]. This dataset consists of 160 videos from 40 different subjects with corresponding synchronized PPG collected with a fingertip device. The mean value of heart rate over the whole dataset is 70.25 beats per minute (BPM) with the corresponding standard deviation of 11.56.

A preliminary test has been done to select the best skin classification method. The evaluated skin classification methods were threshold-based method and SVM-based method as described in Section 3.1. Examples of the masks returned by both methods are shown in Figure 2. The results show that the threshold-based method is better than SVM-based. However, it should be noted that the selected thresholds were fitted to the selected dataset, therefore, the method might not generalize well to other data.

To evaluate the developed method, a leave-one-subject-out experiment was conducted with the aim of testing its predictive performance and generalization capability. To this end, mean absolute error (MAE) and mean squared error (MSE) were used as metrics. Additionally, to evaluate the quality of HR predictions, we computed the number of peaks in the reconstructed signals and compared it to the number of peaks in the ground truth PPG.

4.2 Experimental Results

Results of all the evaluated methods, as well as the developed method, are given in Table 1. All three previously

[4]https://www.idiap.ch/dataset/cohface

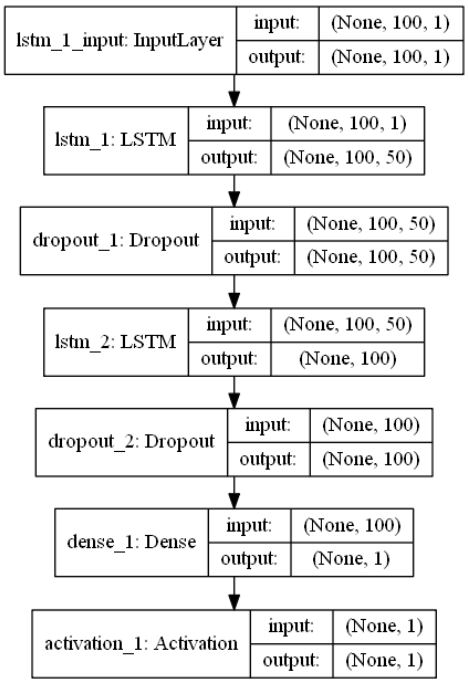


Figure 1: The architecture of the network used in the Deep-Learning-Based method.

(a) **(b)**

Figure 2: Classified skin using the (a) threshold method, and (b) machine learning method.

mentioned metrics are reported, i.e., MAE, MSE and HR MAE. Note that the heart rate MAE of the baseline is 9.67 BPM, while our method achieves 8.75 BPM. In addition, the first 10 seconds of the reconstructed PPG using each of the methods on a subset of videos are shown in Figures 3–5.

The results show that the Deep-Learning-Based method produces better reconstruction of the PPG signal, as the error between the estimated and actual HR is the lowest. Table 1 also shows that the developed method outperforms state-of-the-art methods on the COHFACE dataset by a notable margin.

5. CONCLUSIONS

We presented a new approach for the reconstruction of the PPG signal from video data. This approach enhances state-of-the-art methods with a deep learning model. It has been evaluated on the COHFACE dataset and the results show that it improves the PPG reconstruction with respect to the state-of-the-art methods.

However, the reconstructed PPG signal is still noisy and it

Table 1: Comparison between state-of-the-art methods and the developed method

Method	MAE (signals)	MSE (signals)	MAE [BPM] (heart rates)
Poh-et-al	0.04	0.15	42.00
Haan-et-al	0.16	0.04	20.75
Wang-et-al	0.16	0.40	11.73
Balakrishnan-et-al	0.16	0.04	39.00
Deep-Learning	**0.04**	**0.01**	**8.75**

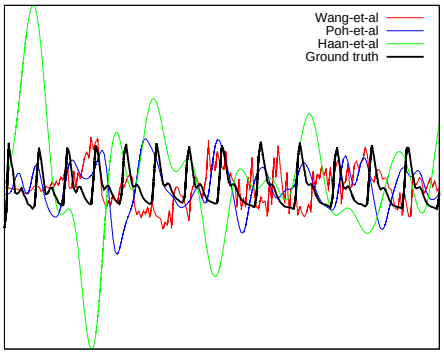


Figure 3: First 10 seconds of the color-based methods.

would thus be difficult to estimate any physiological parameters from it, which will need to be improved in future work. Additionally, higher quality of the recordings, especially regarding the lighting conditions, will be evaluated with the aim of obtaining better results.

6. ACKNOWLEDGMENTS

This work is part of a project that has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No. 780819.

7. REFERENCES

[1] G. Balakrishnan, F. Durand, and J. Guttag. Detecting pulse from head motions in video. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3430–3437, June 2013.

[2] G. de Haan and V. Jeanne. Robust pulse rate from chrominance-based rPPG. *IEEE Transactions on Biomedical Engineering*, 60(10):2878–2886, Oct 2013.

[3] G. Heusch, A. Anjos, and S. Marcel. A reproducible study on remote heart rate measurement. *CoRR*, abs/1709.00962, 2017.

[4] A. Hyvarinen. Fast ICA for noisy data using Gaussian moments. In *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI*, volume 5, pages 57–61, May 1999.

[5] M. Lewandowska, J. Rumiński, T. Kocejko, and J. Nowak. Measuring pulse rate with a webcam — A non-contact method for evaluating cardiac activity. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 405–410, Sept 2011.

[6] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence – Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[7] O. R. Patil, Y. Gao, B. Li, and Z. Jin. CamBP: A camera-based, non-contact blood pressure monitor. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, UbiComp '17, pages 524–529, New York, NY, USA, 2017. ACM.

[8] J. Patterson and A. Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly, Sebastopol, 2017.

[9] M. Poh, D. J. McDuff, and R. W. Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE Transactions on Biomedical Engineering*, 58(1):7–11, Jan 2011.

[10] M.-Z. Poh, D. J. McDuff, and R. W. Picard. Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. *Opt. Express*, 18(10):10762–10774, May 2010.

[11] W. Wang, S. Stuijk, and G. de Haan. Exploiting spatial redundancy of image sensor for motion robust rPPG. *IEEE Transactions on Biomedical Engineering*, 62(2):415–425, Feb 2015.

[12] W. Wang, S. Stuijk, and G. de Haan. A novel algorithm for remote photoplethysmography: Spatial subspace rotation. *IEEE Transactions on Biomedical Engineering*, 63(9):1974–1984, Sept 2016.

[13] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph.*, 31(4):65:1–65:8, July 2012.

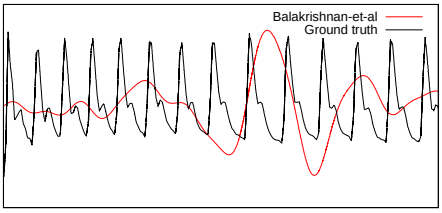


Figure 4: First 10 seconds of the motion-based method.

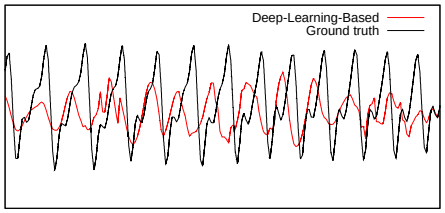


Figure 5: First 10 seconds of the Deep-Learning-Based method.

The Influence of Communication Structure on Performance of an Agent-based Distributed Control System

Andreja Malus, Rok Vrabič, Dominik Kozjek, Peter Butala
Faculty of Mechanical Engineering,
University of Ljubljana
Aškerčeva 6, 1000 Ljubljana
andreja.malus@fs.uni-lj.si

Matjaž Gams
Jožef Stefan Institute,
Department of Intelligent Systems
Jamova cesta 39, 1000 Ljubljana
matjaz.gams@ijs.si

ABSTRACT
Industrial Internet of Things (IIoT) is a new concept denoting extensive use of ubiquitous connected devices on the manufacturing shop floor. While most recent research in this area focuses on the monitoring capabilities of IIoT and on the resulting data analysis, IIoT also presents an opportunity from the perspective of distributed control. The paper suggests that agent-based control of an industrial process can be realized by a multi-agent system in which each agent is able to learn the influence of its actions on the behaviour of the system and to communicate with other agents in its proximity. Influence of the communication structure on performance, robustness, and resilience are analysed for a case of an industrial compressed air system. The simulation results suggest that in such systems, communication between the supply and the demand side improves resilience, while the robustness is improved through learning.

KEYWORDS
Industrial Internet of Things, multi-agent system, distributed control, machine learning, robustness

1. INTRODUCTION
Cyber-physical systems (CPS) represent an emerging paradigm integrating computational, networking and physical processes to address requirements of future industrial systems [1]. Often the interfaces between the physical and the virtual worlds are realized using connected intelligent sensing and actuating devices. Their use in manufacturing environment is the basis of the concept of Industrial Internet of Things (IIoT). Numerous connected IIoT devices enable acquisition and sharing of large amounts of data, promising time and cost savings, scalability and efficiency. However, as IIoT systems grow in size and complexity, their response times and computational complexity outgrow traditional centralized control systems. Researchers work on enhancing flexibility, robustness, adaptability and reconfigurability of CPS by employing concepts of distributed and autonomous control in dynamic environments of flexible manufacturing systems.

A common way to implement distributed control in manufacturing systems is by using autonomous computational entities called agents. Agents acquire information about their environment and take actions to influence the environment. They can exhibit various levels of intelligence depending on the method used to select an action based on the state of environment. Reflex agents passively react to signals from environment, while pro-active agents select their actions continuously to achieve a goal or utility. An agent that has the capacity to adapt its operation based on feedback from the environment is called learning agent [2].

In some applications, multiple software agents are used to collectively solve problems by interacting with each other and reaching mutual agreements through negotiations, bidding and other communication mechanisms, enabling reconfigurability and scalability [3]. Agents acting in parallel results in the system's global behaviour that may include emergent phenomena and is often difficult to predict in advance. Design of interactions between elements is therefore mostly performed using simulations to obtain the desired behaviour of the system in a trial and error manner. Although the ideas of agent-based systems originated more than two decades ago and much research has been done on intelligence of software agents and coordination mechanisms, not many examples of real-life implementations can be found in the manufacturing industry [4].

Despite the challenges of implementation in industrial applications, emerging computing paradigms, developing communication protocols, and decreasing cost of computing power and network communications are suggesting that the research field should be revisited in context of industrial applications.

In [5] a control method using rationally bounded learning agents was proposed. This paper extends this work by analysing the influence of different connection schemes on performance in normal and adverse conditions. The response of the system in adverse conditions can be evaluated from the point of view of robustness and resilience. Robustness measures the extent to which a disturbance affects the system's performance while resilience represents the system's ability of restoring normal operation. Using a simulation of an industrial compressed air system it is shown that (1) the communication between the agents representing the supply (compressors) and the agents representing the demand (consumers) improves the response of the system to repeated disturbance, (2) full connectedness is not necessary as additional connections beyond a certain point do not contribute to improved system performance, and that (3) the communication structure influences resilience, but not robustness.

2. DISTRIBUTED CONTROL WITH RATIONALLY BOUNDED AGENTS
A perfectly rational agent makes decisions under the assumptions that (1) it has complete knowledge of the problem space and is aware of all its available actions, (2) the preferences of actions are known and (3) it has the ability to discover the optimal policy regardless of the necessary computational demand [6]. Absence of any of the three assumptions makes an agent rationally bounded [7]. In the engineering field, the use of the term bounded rationality refers to limited calculation time and computational

capacity. Design of artificial agents strives for optimization under time and capacity constraints.

The presented agent model assumes that in a truly distributed system none of the agents has an overview of the state of the whole system and that each agent can communicate only with elements of the system (i.e. sensors, actuators, and other agents) in their proximity. Based on these limitations, the agent model defines what agents observe, how they learn from the observations, and how they communicate with other agents.

3. AGENT MODEL

A multi-agent system is described as a network $N = (A, CA, S, CS, T, CT)$ where A represents a set of agents and CA a set of communication channels between pairs of the agents. S is a set of sensors, and CS a set of measurement channels between the agents and the sensors, where one agent is connected and reads measurement values from at least one sensor and each sensor is connected to one or more agents. Sensors and communication with other agents enable the agents to gain information about the system but each agent can only have a partial view and no agent has an overview of the whole state of the system. Each agent is connected to exactly one actuator from the set of actuators T, the connections are represented in set CT.

All agents in the system have the same structure, regardless of their function in the system. The components of the model are shown in Figure 1. Agent $a_i \in A$ has some belief b_i about its environment based on measurements of the sensors it has access to S_i and predictions received from other agents P_{ji} as shown in Eq. (1).

$$b_i = f(S_i, P_{ji}) \tag{1}$$

The agent model also includes an environment model M_E that is used to make a set of predictions P_i about future sensor values based on current belief b_i and actions available U_i as shown in Eq. (2).

$$P_i = M_E(b_i, U_i) \tag{2}$$

Based on predictions P_i and the sensor goal values G_i the agent selects its next action u_i, as shown in Eq. (3), that minimizes the selection criterion (e.g. an error estimate) in the form of a function f_A.

$$u_i = \underset{U_i}{\operatorname{argmin}}\left(f_A(M_E(b_i, U_i), G_i)\right) \tag{3}$$

The environment model M_E is learned by observing the control actions u_i taken to influence the environment and the environment's response to the actions, captured by the agent's belief b_i. In absence of communication with other agents the environmental model represents the physical model of the system implicitly including other agents' influence on the state of system. When connected to other agents, an agent receives predictions of future states of environment. These predictions include the knowledge of other agents about their influence on the environment implicitly captured by their environmental models.

The agent acquires the environmental model using random forest algorithm [8]. The algorithm is a highly accurate ensemble learning method, resistant to overfitting and easy to use, since no scaling and normalization of the data is required.

4. EXPERIMENTAL CASE

Compressed air is widely used in industrial systems as a medium for energy transfer to various systems, for example power equipment, spraying tools, conveyers, and power controls. It is safe, easy to use and maintain. However, more than 75 % of the life-cycle costs of compressed air system are accounted for by energy consumption [9] and reports estimate that only about 10–20% of the total input energy is utilized for useful work [10]. In the European Union compressed air systems are reported to consume 80 TWh of electricity [9] or 10% of industrial electricity consumption [10] but at the same time potential economic savings of more than 30 % are estimated [9]. Inefficiencies can be attributed to many reasons, the most important being leakages and inefficient control [10].

Few compressed air systems operate at full-load all of the time. Part-load performance is therefore critical and is primarily influenced by compressor type and control strategy. The choice of the type of control depends largely on the type of compressor being used and the demand profile of the system. For a system with a single processor and mostly steady demand, a simple control system may be appropriate. Simple control approach most often uses two pressure thresholds; when pressure drops below the lower threshold the compressor is turned on and if pressure exceeds the upper threshold the compressor is turned off. However, a complex system with multiple compressors, varying demand, and many types of end-uses requires a more sophisticated control strategy.

This developed distributed control model aims to present a robust and scalable alternative approach for a compressed air system with multiple compressors and compressed air storage tanks using autonomous switches for turning the compressors on or off and autonomous valves for controlling the transport paths.

5. SIMULATION

The simulated compressed air system, shown in Figure 3, consists of 2 compressors, 2 compressed-air tanks, 4 consumers, 3 smart valves, and piping. Each compressor supplies air to one tank and each tank has a safety valve to prevent the tank from becoming over-pressurized. Pressure sensors are installed on both compressed-air tanks and consumers.

The position of the valves in the system enables flow control of the air in the system. Smart valves and compressors' on/off switches are controlled by software agents, that are also connected to corresponding sensors as depicted in the figure. The parameters of the model, e.g. the pressures, time constants and leakage rates, are chosen to correspond to values that are commonly found in real systems. Agents have set target values for their corresponding sensors and the goal is to keep the air pressure values as close to the target values as possible all the time, regardless of consumption.

The agent has one random forest regressor for learning the environmental model for each of its connected sensors. The number of estimators is set to 100.

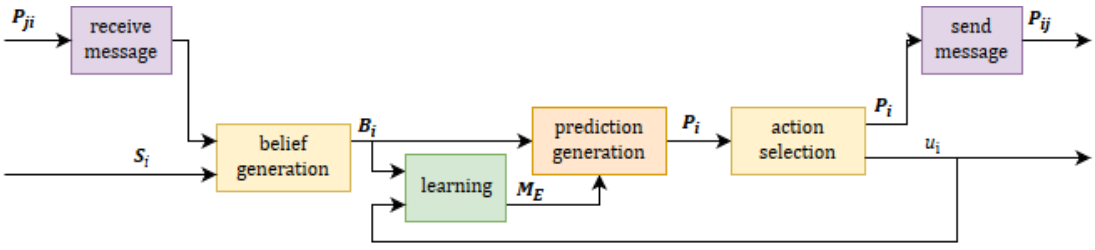


Figure 2. Communication schemes for different scenarios

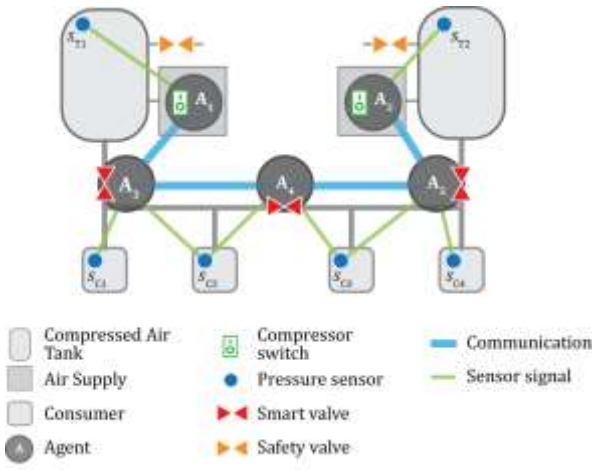


Figure 3. Simulated compressed air system

The input features for the regressor for the $k-th$ sensor consist of the last action u_i taken by the agent, the observed sensor values $s_{ik} \in S_i$, the differences ΔS of each sensor value s_{ik} compared to every other sensor value $s_{im}; m \neq k$, the other agents' predictions P_{ji}, the differences in the other agents' predictions denoted as ΔP_{ji} and the existence of the other agents' predictions denoted as P^* as shown in Eq. (4).

$$x = \{u_i, S_i, \Delta S, P_{ji}, \Delta P_{ji}, P^*\} \quad (4)$$

The associated outputs y are the differences of the sensor values at times t and $t + \Delta t$, as shown in Eq. (5).

$$y = \{S(t + \Delta t) - S(t)\} \quad (5)$$

When calculating predictions for more than one step in advance, predictions of all agent's sensors for each following timestep (within the observed agent) are calculated prior to calculating predictions for the next timestep.

The next action is selected using the mean-square-error function for evaluating the predictions against the goal values for all sensors. The function is shown in Eq. (6), where S_i is a set of agent's sensors, H is the number of future times for which the effects of the action $u \in U_i$ are assessed, called the prediction horizon, and p_{ik} and $g_{ik}, k \in S_i$, are predictions and goal values for the k-th sensor, respectively.

$$f_A(u) = \frac{1}{|S_i|}\sum_{k=1}^{|S_i|}\left(\frac{1}{H}\sum_{h=t+\Delta t}^{H}(p_{ik}(h) - g_{ik})^2\right) \quad (6)$$

In 5 scenarios, the influence of different communication connection schemes, presented in Figure 3, was tested. In scenario A the agents don't communicate among themselves, in scenario B the compressor agents communicate with each other and the smart valves agents communicate with neighbouring smart valves agents. Communication connections in scenario C follow the flow of the air in the system. In scenario D each compressor agent communicates with the nearest two smart valves and the smart valves are also connected to their neighbours. The last scenario E represents full communication scheme in which all agents communicate with all other agents in the system.

The total time of one simulation is set to 4000 s. Each scenario was simulated 150 times. The consumers' activity is random, each consumer is alternately set to on for 1 – 4 s and off for 30 – 40 s. The goal values of the compressor and the smart valves agents are set to 6.3 and 5.0 bar respectively and the safety valve has a set pressure of 10 bar for all scenarios.

The time step Δt is set to 1 s. The predictions are calculated for the next 1-3 s. The duration of current prediction horizon determines the duration of the corresponding reasoning cycle. The environmental model is updated every 500s.

In all scenarios a disturbance was simulated. The disturbance is represented by compressor 1 failure and opening of the corresponding tank's release valve, it is repeated 5 times and lasts for 200 s. Time of start of disturbance in simulation is noted $t_d{}^1$ = 1700 s, $t_d{}^2$ = 2200 s, $t_d{}^3$ = 2700 s, $t_d{}^4$ = 3200 s and $t_d{}^5$ = 3700 s.

The results of the simulations were used to evaluate the ability of the system to withstand adverse conditions, called robustness, and its ability to recover from disturbance, called resilience. In the context of this paper, robustness c_{rob}, shown in Eq. (7), is defined as the ratio between the pressure drop in the system during disturbance Δp_d and the pressure p_s subtracted from the whole.

$$c_{rob} = 1 - \frac{\Delta p_d}{p_s} \quad (7)$$

The moment t_{bb} when the rise of the pressure in the system after the end of the disturbance reaches 1-1/e = 63,2% of the pressure drop is observed. The difference between this moment and the time of the end of the disturbance is defined as bounce-back time Δt_{bb}. Resilience c_{res}, for the context of this paper, is defined as the inverse of the bounce-back time as shown in Eq. (8).

$$c_{res} = \frac{1}{\Delta t_{bb}} \quad (8)$$

6. RESULTS

Simulation results for average sensor values measured by sensor S_{C1} for all five scenarios are shown in Figure 4. Due to learning, the pressure drop in the time of disturbance is lower after every repetition of the disturbance in the first four occurrences of the event.

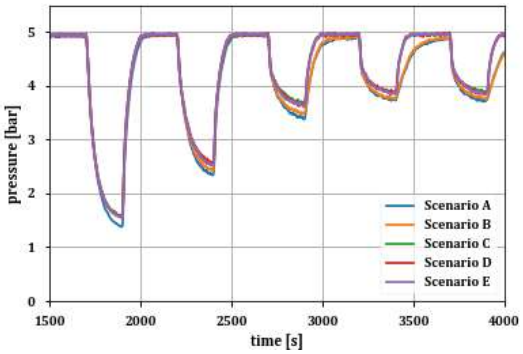


Figure 4. Comparison of average pressure on sensor S_{C1} for 5 scenarios

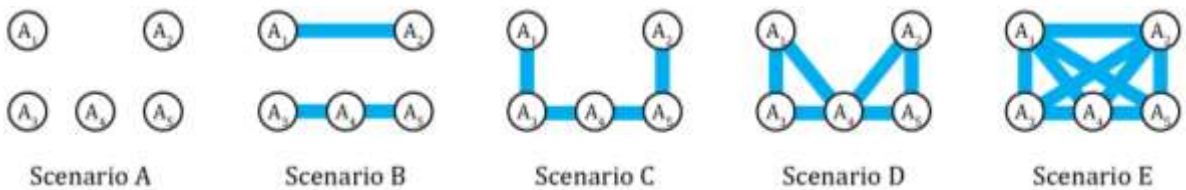


Figure 5 shows the pressure values 50 seconds after the end of the first and the fourth disturbance, effectively demonstrating the effect of learning in the considered scenarios. The first disturbance (Figure 5a) has approximately the same effect in all scenarios because the agents have not yet learned how to mitigate its effects. However, the bounce back from the fourth disturbance (Figure 5b) differs significantly depending on the scenario. In scenarios A and B in which the supply (compressors) and the demand (consumers) are not connected, the bounce back is slower and more scattered than in scenarios C, D, and E.

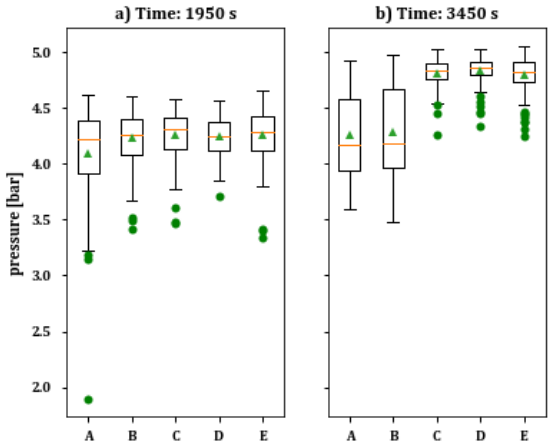


Figure 5. Pressure values measured by sensor S_{C1} after disturbances, green triangles show the average value and green dots represent the outliers

The results for evaluation of robustness and resilience from data measured by sensor S_{C1} are shown in Figure 6.

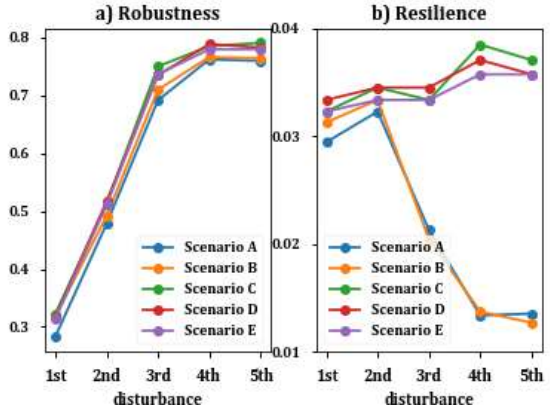


Figure 6. Robustness c_{rob} and resilience c_{res} during disturbances

As seen in Figure 6a, robustness improves over time in all scenarios. As agents learn to create better predictions of the effects of their actions, their responses improve. This is similar in all simulated scenarios, suggesting that it is not influenced by the communication structure.

Figure 6b shows that resilience improves over time in scenarios C, D, and E, in which the supply and the demand are connected. The resilience decreases in scenarios A and B where there is no communication between compressors and smart valves. In scenarios A and B, the agents on the demand side cannot directly detect the pressure drop which happens due to the disturbance. Their response is based solely on the observation of the pressures on the demand side. In time, they learn to prefer to keep the valves closed in order to maintain the pressure on the consumers during the disturbance. However, since they cannot detect the end of the disturbance directly, they have a delayed response when conditions normalize which lowers the overall resilience.

7. CONCLUSION

The paper argues that the current understanding of the role of IIoT, which is mostly related to monitoring and data analytics, should be extended to the domain of distributed control. A distributed, agent-based control model is presented. The model assumes that no agent has an overview of the whole system state, but rather only has a partial view of its neighbouring sensors, actuators, and other agents.

The paper builds on a previously conceived agent model [5] and explores the effects of the intra-agent communication structure for a simulated case of an industrial compressed air system. The results show that the communication structure influences resilience but not robustness. Robustness is improved through the learning mechanism in which the agents learn to predict the effects of their actions on the behaviour of the nearby system constituents. The presented agent model enables a smart controller to operate in a system without prior knowledge of the effects of its actions on the controlled variable. However, this paper shows that to achieve both robustness and resilience of the multi-agent control system, appropriate communication structure of the network must be implemented.

Future work will focus on the development of a real demonstrator and transfer of the learned policies from simulation to the demonstrator.

8. REFERENCES

[1] Monostori, L., Kádár B, Bauernhansl, D., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K. 2016. Cyber-Physical Systems in Manufacturing. CIRP Annals – Manufacturing Technology 65(2):621–641.

[2] Russell, S. J., and Norvig, P. 2010. Artificial intelligence: A modern approach (3rd ed.). Upper Saddle River, N.J. ; New Delhi: Prentice Hall/Pearson Education.

[3] Bruzzone, A.A.G., and D'Addona, D.M. 2018. New Perspectives in Manufacturing: An Assessment for an Advanced Reconfigurable Machining System. *Procedia CIRP* 67: 552-557

[4] Shen, W., Hao, Q., Yoon, H.J., and Norrie, D.H. 2006. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics* 20: 415 – 431.

[5] Vrabič, R., Kozjek, D., Malus, A., Zaletelj, V., and Butala, P.: Distributed control with rationally bounded agents in cyber-physical production systems. *CIRP Annals – Manufacturing Technology* 67(1): 507 – 510, 2018.

[6] Rubinstein, A. 1998. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. MIT Press.

[7] Simon, H.A. 1972. Theories of Bounded Rationality. *Decision and Organization* 1(1):161–176.

[8] Breiman, L. 2001. Random Forests. *Machine Learning*, 45: 5–32.

[9] Radgen, P. 2004. Compressed Air System Audits and Benchmarking. Results from the German Compressed Air Campaign "Druckluft effizient". In: *Proceedings of ECEMEI, Third European Congress on the Economics and Management of Energy in Industry*, Rio Tinto, Portugal, April 6-9.

[10] Saidur, R., Rahim, N.A., Hasanuzzaman, M.: A Review of Compressed-Air Energy Use and Energy Savings. *Renewable and Sustainable Energy Reviews* 14(4):1135–1153, 2010.

Complex Decision Rules in DEX Methodology: jRule Algorithm and Performance Analysis

Adem Kikaj
Jožef Stefan International Postgraduate School
Jožef Stefan Institute, Department of Knowledge Technologies
Jamova 39, 1000 Ljubljana, Slovenia
adem.kikaj@ijs.si

Marko Bohanec
Jožef Stefan Institute, Department of Knowledge Technologies
Jamova 39, 1000 Ljubljana, Slovenia
marko.bohanec@ijs.si

ABSTRACT
DEX (Decision EXpert) is a qualitative multi-criteria decision-modeling methodology. DEX models are used to evaluate and analyze decision alternatives. An essential component of DEX models are decision rules, represented in terms of decision tables. Decision tables may contain many elementary decision rules and may be difficult to be understood by the decision maker. A more compact and comprehensible representation is obtained by converting elementary decision rules to complex rules. The DEX-Rule algorithm, which is currently implemented in software DEXi, has been found inefficient with large decision tables. This research is aimed at improving the efficiency of the DEX-Rule algorithm. We propose a novel algorithm, called jRule, which generates complex rules by specialization. According to performance analysis, jRule is indeed more efficient than DEX-Rule. The compactness of complex rules produced by both algorithms varies and there is no clear winner.

Categories and Subject Descriptors
H.4.2 [**Types of Systems**]: Decision Support

F.2.0 [**General**]

General Terms
Algorithms, Performance, Experimentation

Keywords
DEX methodology, decision rules, complex decision rules, algorithm analysis

1. INTRODUCTION
Decision-making is a difficult and complex process. During this process, a decision maker (DM) faces several decision alternatives. To choose a particular alternative from the set of possible alternatives, a decision-analysis approach [3, 4] can help to satisfy the aims or goals of a decision maker. Decision analysis [3, 4] is the discipline used to help a decision maker to deal with uncertainty, complexity, risk, and trade-offs of the decision. The idea of decision analysis is to develop a decision model, which can help decision makers to evaluate alternatives and to choose the best action.

The decision maker in a decision problem have to deal with multiple and possibly conflicting criteria. Multiple Criteria Decision Analysis (MCDA) or Multiple Criteria Decision Making (MCDM) [3] provides methods for structuring, planning and solving such decision problems. DEX methodology is one of the MCDM methods. DEX is a qualitative multi-criteria decision-making methodology [1, 2, 5] aimed at the assessment and analysis of decision alternatives. DEX is supported by software DEXi (http://kt.ijs.si/MarkoBohanec/dexi.html).

DEX models have a hierarchical structure, which represents a decomposition of some decision problem into smaller, less complex sub-problems. DEX models are developed by defining (i) attributes, (ii) scales, (iii) hierarchically structured attributes (the tree of attributes), and (iv) decision rules. In DEX models, attributes are variables that represent properties of decision alternatives. Attributes can be either basic or aggregated. Aggregated attributes have subordinate attributes, while basic attributes do not. Basic attributes represent inputs and aggregate attributes represent outputs (results). A scale represents a set of values that can be assigned to an attribute. Scales are qualitative and can take discrete values like ‘excellent’, ‘acceptable’, ‘inappropriate’, etc. Decision rules represent the mapping of subordinate attributes to an aggregated attribute (see section 2 on more details about decision rules in DEX).

In a DEX model, an aggregated attribute may involve many subordinate attributes (e.g., more than five) in which case the decision table will contain many elementary decision rules and may be difficult to be understood. In order to obtain a more comprehensible representation, the DEXi software implements DEX-Rule, an algorithm that converts elementary decision rules to more compact complex rules. DEX-Rule has been found inefficient in decision tables with many subordinate attributes and many elementary decision rules that map to a single decision value.

This research is aimed at improving the efficiency of the DEX-Rule algorithm. We propose a novel algorithm, called jRule, which finds complex rules by specialization, i.e., by narrowing down too general rules that are constructed initially. The jRule algorithm performed better regarding the running time. The results generated by both algorithms are guaranteed to cover the whole decision table.

This paper is structured as follows: Section 2 formulates the Decision Rules in DEX, Section 3 presents the DEX-Rule algorithm, Section 4 presents the jRule algorithm, Section 5 presents the comparison of the two algorithms regarding the algorithm complexity and the number and form of complex decision rules that they generate. Section 6 summarizes and concludes the paper.

2. DECISION RULES IN DEX
In DEX models, attributes can be either basic or aggregated. Aggregated attributes are attributes which depend on their descendants, known as subordinate attributes. Decision rules in DEX define the bottom-up mapping of the scale values of

subordinate attributes to the values of the aggregated attribute. An example of such mapping, represented in terms of a decision table, is shown in Table 1. The example is taken from a well-known model for evaluating cars based on attributes such as buying price, maintaining price, safety, and comfort [1]. The example occurs at the top level (root) of the model and maps the subordinate attributes *PRICE* and *TECH.CHAR* (technical characteristics) to the overall evaluation of a *CAR*. The value scale of the involved attributes are ordered values as follows:

- *PRICE = {high, medium, low}*,
- *TECH.CHAR = {bad, acc, good, exc}*, and
- *CAR = {unacc, acc, good, exc}*.

Each row in Table 1 defines the value of the aggregated attribute *CAR* for each combination of subordinate attributes' values. Therefore, the decision table maps all the combination of *PRICE* and *TECH.CHAR* scale values into the value of *CAR*.

Table 1. Decision table with elementary decision rules of DEX model known as CAR Evaluation Model [1].

	PRICE	*TECH.CHAR*	***CAR***
1	high	bad	**unacc**
2	high	acc	**unacc**
3	high	good	**unacc**
4	high	exc	**unacc**
5	medium	bad	**unacc**
6	medium	acc	**acc**
7	medium	good	**good**
8	medium	exc	**exc**
9	low	bad	**unacc**
10	low	acc	**good**
11	low	good	**exc**
12	low	exc	**exc**

A decision rule consists of the condition and decision part:

if $subAttr_1$ = $value_1$
and $subAttr_2$ = $value_2$
...
and $subAttr_n$ = $value_n$
then *aggAttr* = *value* (or interval of values)

The condition part is the Cartesian product of the scale values of the subordinate attributes of an aggregated attribute ($subAttr_1$, $subAttr_2$, …, $subAttr_n$). The decision-maker defines the *value* of each decision rule, which might be a single value or an interval of values of the aggregated attribute. Such decision rules are also called *elementary decision rules*, since each rule defines the value for exactly one combination of subordinate attributes' values.

In this way, the first row in Table 1 represents the following elementary rule:

if *PRICE = high* **and** *TECH.CHAR = bad* **then** *CAR = unacc*

An alternative representation of the decision rules can be by an *n*-dimensional matrix, depending on the number of subordinate attributes. Figure 1 shows such a representation of Table 1. Here, each cell of the matrix represents one elementary decision rule from the decision table.

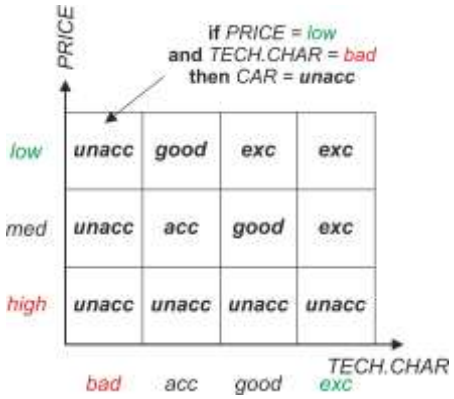


Figure 1. Elementary decision rules represented in a matrix.

In order to represent the decision table in a more compact and possibly comprehensible way, DEX uses *complex decision rules*. A complex decision rule consists of the condition and decision value part. In contrast with elementary rules, each clause in the condition part can represent an interval. The decision value is always a single value. Thus, a complex rule generally takes the form:

if $subAttr_1 \in [low\_value_1 , high\_value_1]$
and $subAttr_2 \in [low\_value_2 , high\_value_2]$
...
and $subAttr_n \in [low\_value_n , high\_value_n]$
then *aggAttr = value*

For comprehensibility, DEXi software traditionally represents intervals as follows:

- '*': the asterisk include all possible scale values of a specific subordinate attribute;
- '>=*w*': stands for *better than or equal to value*;
- '<=*w*': stands *for worse than or equal to value*;
- 'w_1:w_2': interval between value w_1 and value w_2, including the two values.

Figure 2 shows several complex decision rules on the matrix from Figure 1. It is important to notice that each complex decision rule covers an area that corresponds to one or more elementary decision rules. In this way, the number of complex rules that completely cover the matrix is generally lower than the number of elementary rules, and the resulting representation is more compact.

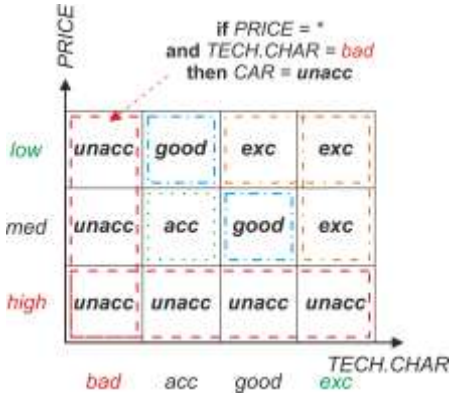


Figure 2. Complex decision rules represented in a matrix through different dotted rectangles for each decision value.

3. DEX-RULE ALGORITHM

DEX-Rule is an algorithm currently implemented in DEXi [1] that converts elementary decision rules into more compact complex decision rules. The DEX-Rule generates complex decision rules by finding areas limited by bounds, which may cover more than one elementary decision rule. An area is represented by two bounds: a low and a high bound. Both are vectors of scale values of the subordinate attributes.

The input to the DEX-Rule algorithm is a decision table, represented in a form of a decision matrix, such as in Figure 1. All the rules are marked as uncovered. The low and high bound (*l* and *h*) are vectors (coordinates) that define an area of decision rules with the target value *t*. Initially, *l* = *h*, which means that they define a single elementary decision rule. Later, with recursive invocation of the algorithm, these boundaries are gradually extended to cover larger areas with the target value *t*. On the output, DEX-Rule generates a set of decision rules, such as in the example shown in Table 4. DEX-Rule proceeds by considering all target decision values, *t*, in succession. For each *t*, DEX-Rule proceeds by generalization, as shown in Algorithm 1.

Algorithm 1. Pseudo-code of the DEX-Rule Algorithm.

```
Inputs:
   l := low bound.
   h := high bound.
   t := target decision value.
   m := last elementary decision rule from decision table
        (representing the highest current bound).
Outputs:
   p := complex decision rules
begin
   cover := ValidateBounds(l, h, t)
   if cover then
      for i = 0 to |h| do
         if h[i] < m[i] then
            DEXRule(l, Increase(h), t, m)
         end if
      end for
      for i = 0 to |l| do
         if l[i] > 0 then
            DEXRule(Decrease(l), h, t, m)
         end if
      end for
      p.add(l + h)
   end if
end
```

For each decision value *t* and each elementary decision rule that has not been covered so far (represented by *l* and *h*, *l* = *h*), DEX-Rule tries to extend the boundaries *l* and *h* in different directions. When the area cannot be extended any more, a complex decision rule is created. More precisely, a complex decision rule is generated in two cases:

- when the algorithm reaches the highest or lowest scale value for the specific subordinate attribute, see Figure 3.a, or
- when an extension would cover an elementary decision rules with a different target value, see Figure 3.b.

The process continues until the matrix has been completely covered by complex rules.

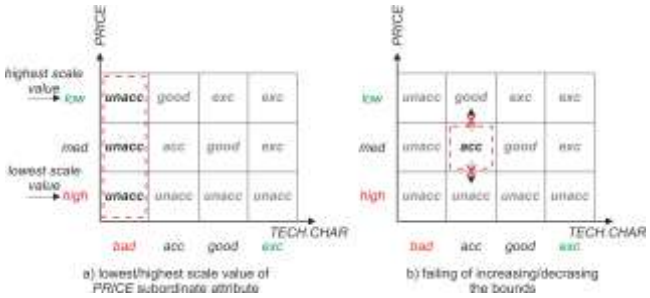


Figure 3. Two cases of generating complex decision rules with DEX-Rule algorithm.

4. JRULE ALGORITHM

The aim of this research was to improve the efficiency of the DEX-Rule algorithm. We propose a novel algorithm, called jRule. While the main idea behind DEX-Rule is to find areas by generalization (extending the area bounds), the main idea of the jRule is to reverse this method and use specialization. jRule proceeds by finding largest areas covering yet uncovered rules for *t* and gradually reducing them.

Algorithm 2. Pseudo-code of the jRule Algorithm.

```
Inputs:
   t := target decision value.
   ger := elementary decision rules for target value t,
          lexicographically sorted by subordinate attribute values.
Outputs:
   p := complex decision rules
begin
   l := lowest subordinate attributes' values from ger
   for i = |ger| to 0
      if !ger[i].isCoveredBy(p) then
         lb = l
         hb = ger[i]
         while !ValidateBounds(lb, hb, t) do
            lb = Increase(lb) // reduce the area by increasing the lb
         end while
         p.add(lb + hb)
      end if
   end for
end
```

The pseudo-code of the jRule algorithm is shown in Algorithm 2. First, the algorithm finds *l*, the lowest bounds for each subordinate attribute of elementary rules for the target value *t*. Then, it locates the last (i.e., highest) currently uncovered elementary rule. This gives the high bound of the area. If the area with bounds *lb* and *hb* is valid, meaning that covers only rules for *t*, a new complex rule is generated. Otherwise, this area is reduced by increasing the low bound *lb*. This process is repeated until all elementary rules for *t* have been covered by complex rules. Notice that, unlike DEX-Rule, areas in jRule are gradually reduced by increasing only the low bound *lb*. Figure 4 illustrates this process for elementary rules shown in Table 1 and the target value *t* = *unacc*. In this case, *ger* is composed of rules 1, 2, 3, 4, 5 and 9 (in this order) from Table 1. The low bound is *lb* = <high, bad>. jRule makes two iterations, first finding the high bound from rule 9 (*hb* = <low, bad>) and then from rule 4 (*hb* = <high, exc>). In both cases, the areas cover *t* and no reduction is necessary.

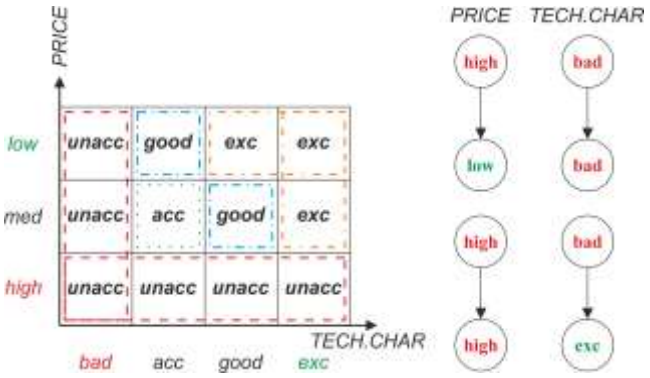


Figure 4. jRule Algorithm identifying the lowest and highest bound for elementary decision rules with decision value *t* = *unacc*.

5. PERFORMANCE ANALYSIS

The comparison between the DEX-Rule and jRule algorithms is made with respect to (i) time complexity, (ii) the running time and (iii) the number of complex rules that these two algorithms generate.

Regarding the time complexity, the DEX-Rule algorithm is $O(m^n)$ because of its recursive nature, where m is the number of subordinate attributes and n is the number of the elementary decision rules. On the other hand, the time complexity of the jRule algorithm is $O(n^2m)$.

The experimental comparison of the algorithms is based in different DEX models for different aggregated attributes. Both algorithms are implemented in JDEXi (http://kt.ijs.si/MarkoBohanec/jdexi.html) and DEX.NET2 (http://kt.ijs.si/MarkoBohanec/dexinet.html). The algorithms were compiled with the same compiler and run on the same computing environment. Table 4 shows running times of the algorithms on three selected DEX models. Generally, jRule is more efficient, and a major difference occurs with Model 2, which is a large decision table having five subordinate attributes and 1728 elementary decision rules.

Table 4. Difference between two algorithms based on running time and number of generated complex rules.

#	Running time [s]		# of complex decision rules	
Models	DEX-Rule	jRule	DEX-Rule	jRule
1	0.75	0.200	30	18
2	1280.00	0.395	121	64
3	1.94	0.980	11	26

Table 5. Complex decision rules generated by DEX-Rule for *CAR* aggregated attribute of CAR Evaluation model.

#	*PRICE*	*TECH.CHAR*	***CAR***
1	**high**	*	**unacc**
2	*	**bad**	**unacc**
3	medium	acc	acc
4	medium	good	good
5	***low***	acc	good
6	>=medium	***exc***	***exc***
7	***low***	>=good	***exc***

The two algorithms, in general, produce different complex decision rules. For example, Tables 5 and 6 show the respective complex rules for the *CAR* evaluation model. The rules are very similar, there is only a small difference in rule 7. In some other cases (Table 4), the differences between the algorithms are more pronounced: jRule produces more compact representations for Models 1 and 2, but less compact for Model 3. More research is needed to establish which algorithm is better and under which circumstances.

Table 6. Complex decision rules generated by jRule for *CAR* aggregated attribute of CAR Evaluation model.

#	*PRICE*	*TECH.CHAR*	***CAR***
1	**high**	*	**unacc**
2	*	**bad**	**unacc**
3	medium	acc	acc
4	medium	good	good
5	***low***	acc	good
6	>=medium	***exc***	***exc***
7	***low***	good	***exc***

6. CONCLUSIONS

In this work, we proposed a novel algorithm jRule for converting elementary decision rules to complex decision rules in the DEX methodology. In contrast with the current DEX-Rule algorithm, which employs generalization, jRule uses the principle of specialization.

Regarding the time complexity and running time, jRule algorithm perform better than DEX-Rule in all experiments performed for different DEX models. On the other hand, none of the algorithm was clearly better with respect to the number of generated complex decision rules. As part of this work, both algorithms were implemented in two open source libraries, JDEXi V4 and DEXi.NET2.

7. ACKNOWLEDGMENTS

The Public Scholarship, Development, Disability and Maintenance Fund of the Republic of Slovenia (Contract no. 11011-44/2017-14) financially supported this research.

8. REFERENCES

[1] Bohanec, M. (2015). DEXi: Program for Multi-Attribute Decision Making User's Manual. *Ljubljana, Slovenia: Institut Jozef Stefan*.

[2] Bohanec, M., Žnidaršič, M., Rajkovič, V., Bratko, I., & Zupan, B. (2013). DEX methodology: three decades of qualitative multi-attribute modeling. *Informatica*, *37*(1).

[3] Bouyssou, D., Marchant, T., Pirlot, M., Tsoukias, A., & Vincke, P. (2006). *Evaluation and decision models with multiple criteria: Stepping stones for the analyst* (Vol. 86). Springer Science & Business Media.

[4] Greco, S., Figueira, J., & Ehrgott, M. (2016). *Multiple criteria decision analysis*. New York: Springer.

[5] Trdin, N., & Bohanec, M. (2018). Extending the multi-criteria decision making method DEX with numeric attributes, value distributions and relational models. *Central European Journal of Operations Research*, 26(1), 1-41.

Sensitivity Analysis of Computational Models that Dissolve the Fermi Paradox

Jurij Nastran
"Jožef Stefan" Institute
Jamova cesta 39
Ljubljana, Slovenia
jurij.nastran@gmail.com

Beno Šircelj
"Jožef Stefan" Institute
Jamova cesta 39
Ljubljana, Slovenia
beno.sircelj@gmail.com

Drago Bokal
University of Maribor
Koroska cesta 46
Maribor, Slovenia
drago.bokal@um.si

Matjaž Gams
"Jožef Stefan" Institute
Jamova cesta 39
Ljubljana, Slovenia
matjaz.gams@ijs.si

ABSTRACT

Given the estimated number of stars and planets in our galaxy, the probability of existence of intelligent civilizations seems high. The first to indicate this was the Drake equation with the assumed default parameters. Yet, the actual observations have yet to reflect those expectations. This discrepancy corresponds to the so-called Fermi Paradox. Although many key factors about the likelihood of alien civilizations still remain largely unknown, new methods of estimating the probability are being proposed. Some of them use probability distributions and the Monte Carlo methods. In this paper we recalculate one of those – the Sandberg interpretation of the Drake equation, analyze the difference between the methods, their strengths and weaknesses. In the conclusion, we find that the probability distribution better reflects our ignorance about the properties of alien environments than the dot-product method.

In our opinion, there are several ways to further improve the computational model based on the Drake equation using the AI methods, thus eliminating the problem with too small probabilities and introducing 3D views.

What makes these analysis relevant, is not only the number of estimated civilizations in our galaxy and a probability that we encounter them in the near future. More important, these models enable estimation of the life-span of the human civilization. Unfortunately, there is a considerably high probability that it will be quite short.

Keywords: Drake equation, Fermi paradox, Monte Carlo, Probability distribution, Extraterrestrial intelligence

1. INTRODUCTION

There are billions of stars in the observable universe. Hundreds of millions are broadly estimated to be in our galaxy alone. If there is at least a modest chance of intelligent life emerging on a given planet, surely there should be at least some number of alien life in the relative vicinity, yet we see none. We are apparently the only one in our part of the universe even though we were able to expand our search quite successfully in recent decades [5].

Why is there no evidence of other civilizations in our galaxy despite the sheer number of planets we now know of? This is the question physicist Enrico Fermi first asked in 1950 and is known as the Fermi paradox. Fermi was particularly effective when dealing with estimates of ill-defined physical phenomena. However, he did not coin the first universally accepted equation for an estimate of the number of civilizations.

1.1 Drake equation

Probably the best known equation for an estimate of the number of detectable civilizations in the Milky Way was composed by Frank Drake [3], phrased as a product of seven factors:

$$N = R_* f_p n_e f_l f_i f_c L$$

The parameters are: R_* which is the rate of star formation per year, f_p is the fraction of stars with planets, n_e is the number of Earth-like (or otherwise habitable) planets per a star that has planets, f_l is the fraction of habitable planets with actual life, f_i is the fraction of life-bearing planets that develop intelligence, f_c is the fraction of intelligent civilizations that are detectable and L is the average longevity of such civilizations. finally N is the number of detectable civilizations.

The Drake equation is used to directly estimate the number of civilizations or as an analysis tool for various components in our galaxy. Most importantly, it can also be used to estimate the lifespan of our civilization (L). The Drake equation can provide exact numbers given proper parameters (i.e. factors), but the problem is that several factors in the equation are not well determined either by observations or with laboratory models. By assuming different values for them, say f_l – probability of life on a habitable planet, results vary a lot [1].

1.2 Point estimates

When trying to estimate the number of civilizations in the galaxy, a point estimate is often used for each of the seven parameters of the Drake equation. This provides an exact

numerical value. If we take estimates based on the distributions in Sandberg's paper [6], we get: $R_* = 10$, $f_p = 0.3$, $n_e = 0.3$, $f_l = 0.5$, $f_i = 0.03$, $f_c = 0.1$, $L = 10^6$ which gives us N around 270. Drake with colleagues originally estimated that $L = N$ and probably between 1000 and 100.000.000. Current best estimates[4] differ from several hundreds to several millions for the civilization's lifespan, and from being alone to several millions civilizations in our galaxy.
Based on actual observations, it seems quite likely that the optimists overestimate their factors: If an advanced civilization appeared somewhere in the galaxy before, moving with the speed of one percent of light speed, it would reach all parts of our galaxy in less than 2o million years. This is a tiny fraction of the lifespan of our galaxy, which is as old as our universe, i.e. around 13.5 billion years, with the perimeter around 100 - 200.000 light years.

2. SETTING THE PARAMETERS AND COMPUTATIONAL MODELS

Sandberg and colleagues [6] suggested that using point estimates to solve the Drake equation is too wild a guess, providing only one number. In their paper they suggest an approach that models each parameter by it's distribution, thus the computation results in a probability distribution. They used a Monte Carlo method for calculating the distribution of the final result. During each iteration, they sampled from distributions to obtain point estimates to use in Drake equation, which results can be used to generate final distribution of N.

2.1 A toy model

To show how using distributions for calculations leads to results that differ from those obtained by simply multiplying point estimates, Sandberg and colleagues introduced a simplified toy model. In their toy model to demonstrate the differences, there are 9 parameters (f_1,f_2,...), which if multiplied together determine probability of ETI (extraterrestrial intelligence) on a single star. Each of those parameters can obtain values from an interval [0, 0.2], with an average of 0.1. In their case-example, the point estimate of each factor is set to 0.1, the same as the average from the interval. For 100 billion stars (as in our galaxy) the computation gives 100 intelligent civilizations. But if instead of using a point estimate we sample from a uniform distribution with an average of 0.1, we get the results indicating that there is 21.45% chance that we are alone in the galaxy.

2.2 Recomputing the Sandberg interpretation of Drake equation

Sandberg et al. applied probability distribution as a way of recalculating the Drake equation, but instead of a uniform distribution presented in Subsection 2.1, they used the probability distribution for each factor obtained from scientific literature – a range and the type of the distribution. They defined a parameter "log-uncertainty" of a parameter X (LU[X]= $\log(\frac{\max(X)}{\min(X)})$) as an estimate of the number of orders of magnitude of the current uncertainty of parameter X. Consequently, the factors were defined in the following way:
– Star formation rate R_* is fairly well constrained by astronomical data and ranges over a maximum of 5 orders of magnitude given other galaxies. Our uncertainty about this parameter is from 2 to 16 solar masses, LU[R_*] = 0.9.
– Fraction of stars with planets f_p is also pretty well known and is about 1 with LU[f_p] = 1.
– The estimates for n_e, which is number of habitable plants, range from $<10^{-12}$ in rare earth arguments to >1 when taking non terrestrials like icy moons into account. Sandberg proposed LU[n_e]=12. Post-2000 literature estimates cover smaller orders of magnitude so they postulated earth-like planet as rocky planet within habitable zone and assumed LU[n_e]≈2.
– The parameters with the most uncertainty are f_l and f_i. f_l (probability of life) is modeled as a physical transition that occurs at some rate per unit time per unit volume of a suitable prebiotic substrate. The probability on a habitable planet with volume V, time period t and abiogenesis rate λ is $f_l = 1 - e^{-\lambda V t}$. They take log-uniform distributions of t with LU[t]=3 and range from 10^7 to 10^{10}, V with LU[V] $\geq$ 20 range from 10^{-35} to 10^{15}. They use log-normal distribution for λ with a mean of 1 and σ of several orders of magnitude. In the paper they do not specify the exact number used for σ, just that its very big. When we recreated their experiment we tested several values for sigma and 200 gave the most similar results.
– Based on the literature, the parameter f_c, which is the fraction of planets which develop civilization, is between 0.001 and 1.
– The final factor L, is longevity of a space-communicating civilization in years and is in the range from 100 to 10^{10}, which is the current estimate of the age of the universe.

All parameter(factor) distributions are listed in Table 1.

With this set of parameters we managed to obtain a distribution for N, displayed in Figure 1, quite similar to the one by Sandberg and colleagues. The two axes in Figure 1 represent N and its corresponding frequency with blue color. A vertical black line marks N equal to 1. The blue graph is therefore PDF (probability density function). It is scaled so that the highest value is 1 to fit on the same graph as the red line which is CDF (cumulative density function). From Figure 1 it can be observed that the probability of us being alone in our galaxy is about one half, e.g. since the red graph reaches 0.5 around N equal to 1.

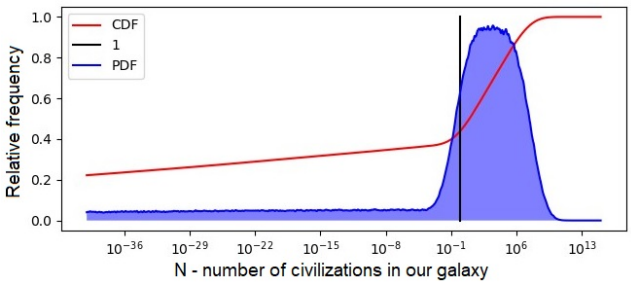


Figure 1: Recomputation of the Sandberg approach: probability density function and cumulative density function of N computed with Monte Carlo method.

Parameter	Distribution
R_*	log-uniform from 1 to 100
f_p	log-uniform from 0.1 to 1
n_e	log-uniform from 0.1 to 1
f_l	log-normal rate, described above
f_i	log-uniform from 0.001 to 1
f_c	log-uniform from 0.01 to 1
L	log-uniform from 100 to 10,000,000,000

Table 1: Summary of the current knowledge about the parameters of the Drake equation.

3. ANALYSIS OF THE SANDBERG'S COMPUTATIONAL MODEL

In this section, we present our initial analysis of the Sandberg et al. computational approach. We tested various issues only on the toy model as it is simpler and better reflects the issues with the computational model. We hypothesized that if one parameter is assigned a number close to zero, the whole product will be a very low number no matter what the other parameters are. For example, the blue graph in Figure 1 might give a misleading impression that the majority of the possibilities is on the right side of the black line indicating 1. However, due to the logarithmic x scale, it is about the same possible that there is only 1 civilization in our galaxy compared to 2 or more, as the red graph shows. The probabilities on the left of the black line indicating 1 are therefore significantly higher. The closer to 0, the higher, meaning lots of Ns are strangely close to 0. From the AI field we are familiar with this problem from the analysis of the naive Bayer theorem: if one of the factors is 0 or close to 0, it should better be modified. We tried to analyze this hypothesis by calculating the same results by using a different number of parameters and by using ranges with small offset ϵ to move away from zero. Namely, if any of the parameters is zero, then there is no intelligent civilization in that galaxy.

3.1 Effect of small values with multiple parameters

Tu study the effect of small values of parameters/factors in the product, we varied the number of parameters. To normalize the result, we adjusted the number of stars so that the average value of N according to the distribution obtained by the Monte Carlo analysis is still 100 as it is in the original case of 9 parameters and 10^{11} stars. So the number of stars is 10^{2+Np} where Np is the number of parameters. All the parameters are in the range [0, 0.2]. The graph showing 3 parameters and 9 parameters can be seen in Figure 2. The solid lines are PDFs and the dotted lines are CDFs. The PDF is scaled like before so that the maximum value is 1. In both cases the average value is still 100. With green dot we mark the probability that there is no civilizations in the toy galaxy. Note that the size of our toy galaxy also varies with the number of parameters to enable comparison. The dot is on the CDF at value of one civilization in the galaxy. It can be seen that the value at 9 parameters is higher than the value at 3 parameters. Therefore, the more parameters, the more probability of being alone. Or in other words: the more uniformly distributed parameters one introduces in a product, the more likely small values of N if parameter values are in [0,1].

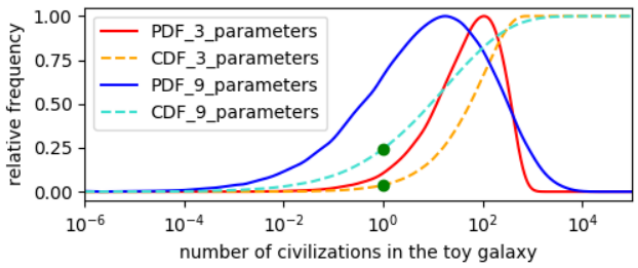


Figure 2: Graphs for Toy model with 3 parameters and 9 parameters.

We tested this phenomenon on multiple number of parameters ranging from 1 to 9 and the result can be seen in Figure 3. We can see that by increasing the number of parameters the PDF widens, thus increasing the possibility of us being alone.

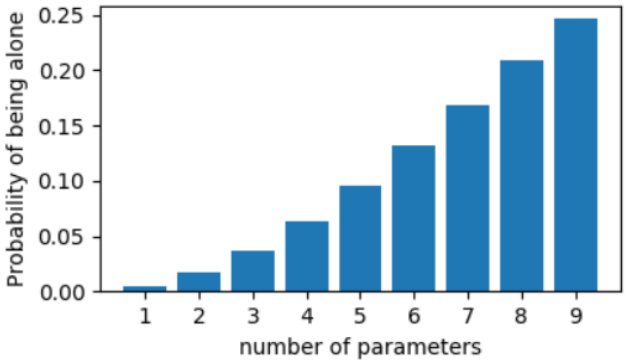


Figure 3: Probability of there being no civilizations depending on the number of parameters (1-9)

3.2 Modifying small values with epsilon

We tried to minimize the effect of parameter/factor values being too close to zero, which seemingly spoils the calculation, by introducing an offset ϵ. Instead of the range [0, 0.2] we now try the same experiments with range $[0+\epsilon, 0.2-\epsilon]$. The offset on the left side ensures that the minimum value is increased and the offset on the right side is there to keep the same average. As we can see in Figure 4 for [0.02, 0.18], the computed N changes significantly. In Figure 5, the same effect is demonstrated for values from 0.00 to 0.040 by a blue line. By increasing the ϵ, the curve becomes narrower thus chances of us being alone decrease.

3.3 Effects of log-uniform distributions

The toy model had all its parameters distributed uniformly. We recreated the same experiments, but with log uniform distribution of parameters, which is the same type of distribution as in the Sandberg paper. In Figure 5 one can see that the effect of adding epsilon in the log distribution modifies the graph even more. Figure 6 represent another analysis with the toy problem and various epsilons, using the log uniform distribution. This is another indication that the small values of parameters introduced by probability distribution by Sandberg strongly influence the final result.

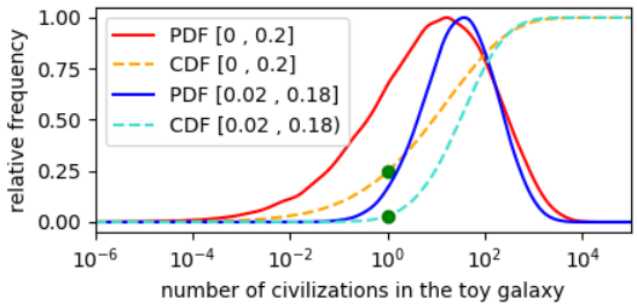


Figure 4: The effect of ϵ on the probability distribution

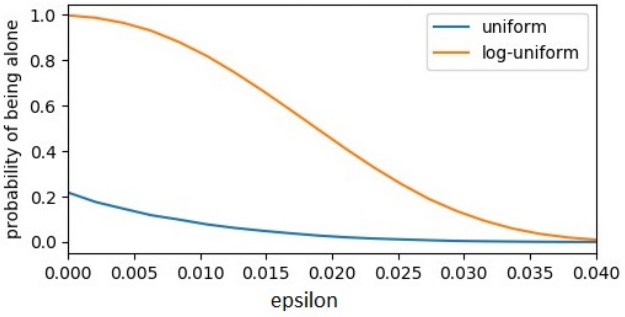


Figure 5: Effect of ϵ is enhanced with log-uniform distributions.

4. DISCUSSION: AI-BASED MODIFICATIONS

The proposed and to a certain extend already tested AI-based modifications are to be fully described in the submission of an SCI paper. Some of the ideas will be presented live at the paper presentation. Here, a couple of hints are presented here.

The first observation when reanalyzing the Sandberg approach is that the it is significantly more informative than point estimates. Unlike providing just one number based on assumptions, it shows the whole probability distribution, i.e. all possible combinations of values of parameters. On the second thought, the computational model has a weakness - when multiplying with zero or very small values. In the Drake equation, several factors are multiplied together, but originally, none of them was very close to 0. However, when dealing with probability distributions, numbers close to 0 can appear and as a result N becomes very small. We presented this effect in the Toy problem with uniform distributions, and later with log uniform distributions where the effect intensified.

In AI, there have been similar problems when multiplying with small values and solutions. For example, several analyses of the Naive Bayes were performed for the case when one of the factors was zero. One of the first analysis was by Cestnik [2].

By modifying the Cestnik or Laplace approach for the Sandberg method, we managed to introduce important modifications.

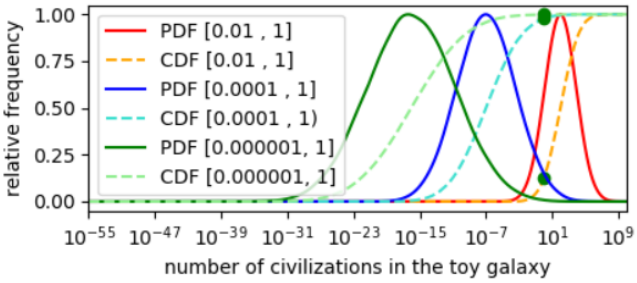


Figure 6: How ϵ affects distributions in log-space.

5. CONCLUSION AND DISCUSSION

We first showed how the number of ETIs was estimated using the original Drake equation, and then with the advanced approach by Sandberg and colleagues. Our re-computation of the Sandberg method yielded nearly the same results.

When reanalyzing the Sandberg approach, one issue emerged - multiplying with small values. We analyzed this phenomena and found out that with increasing number of parameters the probability gets closer to zero. We also introduced some corrections to the ranges of parameters.

While it is unlikely that the computing mechanisms will solve the Fermi problem on its own, they can provide better understanding of the current and future observations.

The computational models and universe observations progress, increasing our knowledge and narrowing the unknowns in the estimation of the number of civilizations and the lifespan of our civilization. However, until we meet another civilization or none for a long time or, as the third and most tragic option - our civilization decays, there is still a long way to go.

6. ACKNOWLEDGMENTS

We would like to thank Filip Talimdzioski for his advice and cooperation.

7. REFERENCES

[1] J. Achenbach and P. Essick. Life Beyond Earth. *National Geographic*, 197(1):24–51, 2000.

[2] B. Cestnik et al. Estimating probabilities: a crucial task in machine learning. In *ECAI*, volume 90, pages 147–149, 1990.

[3] F. Drake. *The Drake Equation: Estimating the Prevalence of Extraterrestrial Life Through the Ages.* Cambridge University Press, 2015.

[4] F. Drake and D. Sobel. *Is anyone out there?: The scientific search for extraterrestrial intelligence.* Delacorte Press, 1992.

[5] J. Gribbin. Are Humans Alone in the Milky Way? *Scientific American*, September 2018.

[6] A. Sandberg, E. Drexler, and T. Ord. Dissolving the Fermi paradox. *arXiv preprint arXiv:1806.02404*, 2018.

Context-Aware Stress Detection in the AWARE Framework*

Marija Trajanoska
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
1000 Skopje, Macedonia
marijatrajanoska@gmail.com

Marko Katrašnik
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
marko.katrasnik@gmail.com

Junoš Lukan
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
junos.lukan@ijs.si

Martin Gjoreski
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
martin.gjoreski@ijs.si

Hristijan Gjoreski
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
1000 Skopje, Macedonia
hristijang@feit.ukim.edu.mk

Mitja Luštrek
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT

Physiological signals are good predictors of stress, which can be thought of as part of a user's context. In this work, an option to combine the user's stress level with other contextual factors is presented. This is done in the form of two AWARE plugins – Android applications that can be incorporated into a smartphone monitoring setup. In the first part, the stress detection method is described, which consists of a lab stress detector, an activity classifier, and a context-aware stress model. In the second part, two plugins are described. One streams the data from the Empatica E4 wristband and the other one uses this physiological data to predict stress. Finally, some possibilities to improve this work are presented.

Keywords

AWARE, plugin, stress detection, Empatica E4, physiology

1. INTRODUCTION

Mental stress is most often researched because of its negative health consequences when it is chronic. The ability to detect stress from physiological signals collected with a wearable device is thus valuable for research in situations when stress occurs, as well as to trigger stress-relief interventions. In addition, stress affects one's short-term psychological state and behaviour, which makes it a part of the user's context as understood in ambient intelligence. Detecting stress is therefore also valuable for adapting intelligent services to the user (e.g., a mobile application may postpone non-essential notifications when the user is stressed). In this paper we present a stress-detection plugin (and its prerequisite – the plugin for the Empatica sensing wristband) for the AWARE framework [3]. This makes stress detection easily accessible to researchers and other interested parties.

AWARE is an Android framework, used to capture the phone sensors' data to infer context. Its modular nature enables it to be extended by plugins. There are already several plugins available such as a Google activity recognition plugin, which captures the users' mode of transportation, and a Fitbit plugin, which enables collecting data such as heart rate and sleep duration from a Fitbit device.

In this work, a state-of-the-art stress detection method is implemented as an AWARE plugin. To make this possible, the method was adapted to real-time operation, being previously only used offline. The plugin classifies the user's physiological data as representing a stressful or a non-stressful condition, after receiving the data from the Empatica wristband via another plugin. Both plugins are planned to be released publicly, so that other researchers will be presented with a ready-made solution for the first time.

In Section 2, we first present the stress-detection method and what data is needed for it. In Section 3 our implementation in the AWARE framework is presented: a plugin for the stress detection model itself (Section 3.2) and a plugin for data collection (Section 3.1). Finally, some possible improvements are outlined in Section 4.

2. CONTEXT-AWARE STRESS DETECTION METHOD

The stress-detection AWARE plugin is based on a real-life stress detection method as described by Gjoreski et al. [7], and the more general context-based reasoning framework introduced by Gjoreski et al. [5]. It consists of three separate machine learning components: a laboratory stress detector, an activity recognition classifier, and a context-based (real-life) stress detector. Each is presented in its own subsection.

2.1 Lab Stress Detection

The lab stress-detection model was trained using data obtained in a laboratory experimental setup during a standardized stress-inducing experiment [7]. The main stressor in this experiment was solving a mental arithmetic task under time and evaluation pressure. The laboratory data was then labelled taking into account both the difficulty of an

*The work presented in this paper was carried out as a part of a project funded by the Slovenian Research Agency (ARRS, project ref. N2-0081).

equation-solving session (easy, medium or hard) and short STAI-Y anxiety questionnaires [8] filled out by the participants. According to this information, the data was classified into three degrees of stress: no stress, low and high. Additionally, baseline no-stress data was recorded on a separate day when the participants were relaxed.

For the creation of the laboratory stress-detection classifier, the machine-learning pipeline involves segmentation, signal filtering, feature extraction, model learning, and evaluation of the models.

Segmentation refers to the partitioning of the data into windows for the purposes of feature extraction. According to the windowing experiments, which provide a performance comparison between models for varying data window sizes, the optimal data window size was 5 min with 2.5 min overlap.

The signals obtained from the Empatica sensors are: blood volume pulse (BVP), interbeat intervals (IBI), heart rate (HR), electrodermal activity (EDA) and skin temperature (TEMP). After filtering the signals to reduce noise, numerical features are extracted from each data window using statistical functions, regression analysis, and frequency and time analysis, depending on the type of signal. A total of 70 features are extracted from these signals.

The best performing classifier on this dataset proved to be the WEKA implementation of the support vector machine algorithm. Its final output is a stress level prediction of 0 (no stress), 1 (low stress) or 2 (high stress), which is then used as input to the context-based stress detector.

2.2 Activity Recognition

It is important for a stress-detection system to be aware of the user's physical activity, since physical activity elicits physiological arousal similar to psychological stress. For this reason, we used the 3-axis accelerometer provided by the Empatica wristband, which has proven to be successful in recognising activities, according to Gjoreski et al. [6]. The activity recognition model was trained on 60 minutes of real-life Empatica data from one person, with nearly 10 minutes of labelled data per class.

The machine-learning pipeline for the acceleration data is similar to the one used in the lab stress detector. Here, data segmentation involves an overlapping sliding-window technique, which divides the continuous stream into 4 s windows with a 2 s overlap.

Feature extraction produces 52 features: seven represent body posture, while the remaining represent body motion. The extracted feature vectors are fed into a machine-learning algorithm to build an activity-recognition classifier.

The best performing algorithm on multiple acceleration datasets was Random Forest [6], so this is the final algorithm used to build the activity-recognition model. The final output of the activity recognition model is a numeric activity level on a scale from 1 to 5, where each number corresponds to an everyday activity as follows: 1 = lying, 2 = sitting, 4 = walking or standing, 5 = cycling or running. Finally, the activity recognition classifier's output is input into the context-based stress detection model.

2.3 Context-Based Stress Detection

The context-based stress detection classifier was trained using the data obtained as part of the real-life experimental setup described in Gjoreski at al. [7] The data duration totalled to 1327 h and involved 5 participants who wore Empatica E4. The labelling process involved a combination of a stress log and Ecological Momentary Assessment (EMA) prompts implemented on a smartphone. For the stress log, the participants logged the start, duration, and intensity (on a scale from 1 to 5) of everyday stressful situations. The EMA prompts were additionally displayed randomly 4 to 6 times throughout the day, with at least 2 hours between consecutive prompts.

The labelled data was then windowed using non-overlapping windows lasting 10 min, since aggregation experiments showed that most algorithms perform better for smaller aggregation windows (10 min to 17.5 min) as compared to larger ones.

The context-based stress detector's input is four-fold:

- context features,
- features extracted from the output of the activity-recognition model,
- features extracted from the output of the lab stress-detection model, and
- a subset of the lab stress detector features.

The whole stress-detection method, including lab stress detection and activity recognition, is illustrated in Figure 1.

The context features refer to the hour of the day (1 to 24) and the type of day (a weekday or weekend).

The output from the activity recognition model gives an estimate of the reliability of the lab model's prediction. Aside from features extracted from the activity level predictions themselves, the activity level is also taken into account as a modifier to the lab stress predictions prior to performing feature extraction on them. The lab model's prediction is discarded if it is made in an unsteady environment, which is defined as the occurrence of an average activity level above 4 (high) in one of the (5-minute long) instances within the last 30 min. Additionally, the stress prediction is decreased (its class is changed to a lower one, or left unmodified if its already zero) if the subject exhibited an average activity level between 2 and 4 (moderate) within the last 20 min.

There are a total of 15 features extracted from the modified lab stress prediction. A subset of lab stress detector features is also used as input to the context-based stress detector.

The best performing algorithm for making a binary ("stress" or "no stress") prediction using the outlined windowing parameters on the labelled real-life data was a Decision Tree [7], so this algorithm was used to build the final context-based real-life model. Using event-based windowing, this algorithm achieved an F-score of 0.9 using leave-one-subject-out cross-validation.

3. AWARE IMPLEMENTATION

AWARE is a mobile instrumentation toolkit which had the initial purpose of inferring users' context [4]. Extensibility, however, was a primary requirement when developing the framework. Specifically, extending the context by using external sensors was explicitly envisioned, as was using the gathered data and machine-learning techniques for "creat[ing] new higher-level context"[4, p. 4].

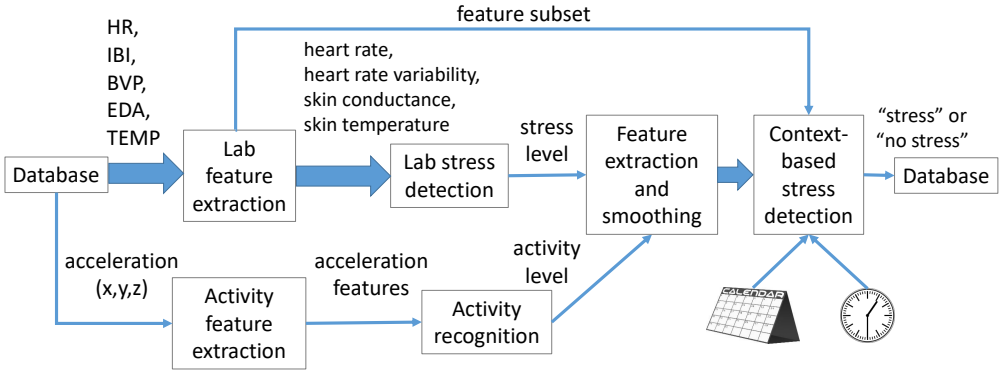


Figure 1: An overview of the context-based stress detection model. Features are extracted from physiological signals from Empatica and input to a lab stress detector. Its predictions are used in the context-based stress detector, which in addition takes activity features from acceleration data into account and considers the time and the type of day.

In the next two sections, two such extensions are presented. The first one gathers physiological data from Empatica, saving it in the standard AWARE format. The second implements the stress detection method presented in Section 2 as an AWARE plugin.

3.1 Empatica Data Streaming Plugin

The AWARE framework already offers plugins for acquiring data from Fitbit and Android Wear wristbands, but it does not have one for the more research-oriented Empatica E4. Our goal was to create an AWARE plugin that enables users to easily connect the Empatica E4 wristband to an Android smartphone.

Figure 2 shows the overview of the processes implemented in this plugin. The physiological data is first transmitted over a Bluetooth connection. It is then available to other plugins via broadcasts and written to a database for later use.

The data from Empatica is transmitted over a low-energy Bluetooth connection, so that the impact on the smartphone's battery is minimal. This enables Empatica E4 to stream up to 24 hours on a single charge [1]. When a sensor reading is successfully transmitted from the wristband to the smartphone, specific functions (`didReceiveAcceleration`, `didReceiveBVP` etc.) are called automatically.

In our case, these functions were expanded to include code to send broadcasts and to save the reading to the database. In this we follow the logic of other AWARE sensors or plugins, in which received data is accessed via broadcasts to display and handle in real time, via content providers (through database) for more complex analysis (the middle part of Figure 2).

Broadcasts are inter-app messages that are sent when a specific event happens, in our case triggered by the data transmission. These messages can be read with observers (broadcast receivers) from any app or plugin that is installed on the phone. Since broadcasting happens in real time, other plugins can use the data from Empatica without including code for communication with the wristband (an example of such a plugin is described in the following section). The data can also be displayed in real time in the native AWARE application.

The collected physiological data is also written to a database (the lower part of Figure 2) by using content providers. Because of different sampling rates, each Empatica sensor has its own content provider. A separate SQL table for each sensor contains columns for `id`, `timestamp`, `device id` and a `value` from the sensor. Columns are defined in this way, so that the database tables maintain the standard AWARE format. Again, each content provider has its own "content URI", a unique address used by other plugins to identify providers for specific sensors.

When broadcasts are sent and the data is written to a database, it is up to other plugins to use it. Even though the main purpose of this plugin is to provide the data that will be handled by other plugins, it also offers a basic user interface. It has its own activity (a user interface that most Android apps have) from which users can export and clear the data in the database.

Contrary to most other Empatica data acquisition applications, our plugin is meant to run in the background. Therefore, it was anticipated that the connection is lost without users noticing. In an effort to solve this problem, our plugin uses a notification to inform the user about the current state of connection.

3.2 Stress Detection Plugin

The goal of the stress detection plugin is to provide real-time stress predictions based on the context-based stress detection method described in Section 2.

In the original stress-detection study [7], the sensor data was recorded in the Empatica E4 wristband's internal memory and later transferred to a computer for further processing. The novelty in our AWARE plugin is that the data is streamed to an Android device via Bluetooth in real time and stored in the phone's database. This allows for real-time processing and classification.

The machine-learning pipeline for the stress detection plugin mirrors the original pipeline used to train and test the context-based stress detector. The three models (lab stress

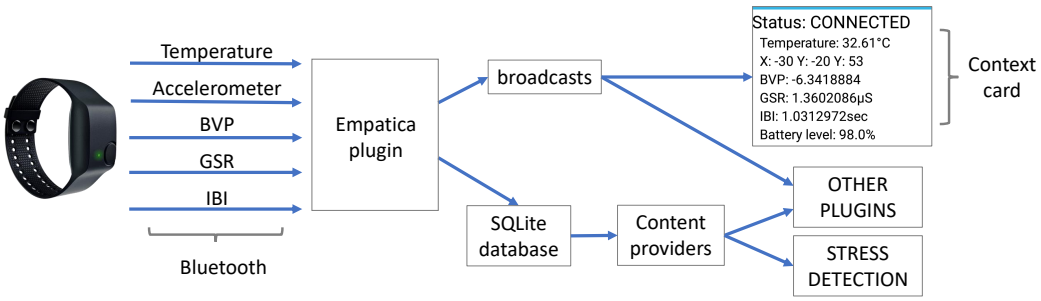


Figure 2: An overview of the Empatica data streaming plugin. The data is first received via a Bluetooth connection and can then be broadcast to other plugins or written to a database.

detector, activity recognition classifier and context-based stress detector) are independent and saved locally in the plugin's assets. The models are triggered periodically using the optimal time intervals discussed in Section 2.

As discussed in Section 3.1, the Empatica data streaming plugin writes the raw data from the E4 in SQL tables in real time. The stress detection plugin then has access to this data through the former's content providers. The plugin reads the last 5 min of raw Empatica data every 2.5 min and provides this data to the models for processing. The context-based model gets its context features using the phone's current date and time.

The features from the lab stress detector, the activity recognition classifier and the context-based stress detector are, both broadcasted and saved in the phone's database. The same is true for each lab stress prediction, activity level prediction and context-based prediction. The stored data is further accessible through content providers for any other application to use, as is the case with other AWARE plugins. In this way, both the Empatica data and the stress prediction method are easily available for other researchers to use.

4. FURTHER WORK AND CONCLUSIONS

The plugins described in the previous section offer a ready-made solution which researchers could use to add a stress level to the user's context. There are some limitations in their current implementation which we aim to amend.

Currently, the standardization (normalization) of some of the features is arbitrary. It is done by subtracting a "typical" value of a given signal and divided by a "typical" standard deviation. To account for inter-individual physiological differences, means and variability could be calculated on a person-specific basis. This would, at the very least, require keeping track of a user ID and then calculating signal mean and variability over a longer time-period when a new user would start using the application. If baseline values would be needed, this would also require the user to indicate when they are not under stress and calculate their specific physiological values in that time-window. This type of user interaction has not been accounted for in the current implementation.

An evaluation of this method is also planned. The models described in Section 2 have been evaluated as outlined in related work, but they have been used in different experimental scenarios. Online real-life use of the method would merit its own evaluation.

To understand causes of stress and the situation where a physiological stress response arises, it is helpful to know as much as possible about a user's context. The plugins described in this work will simplify combining physiological data with other contextual data the AWARE framework already provides. Additionally, stress predictions can be used as context themselves and inform other interactions with users, such as offering them prompts at certain stress levels by using the AWARE Scheduler [2] and using stress predictions as broadcast triggers.

5. REFERENCES

[1] E4 wristband for developers. `http://developer.empatica.com/`, 2018. Accessed: 2018-08-29.

[2] D. Ferreira. AWARE: Scheduler. `http://www.awareframework.com/scheduler/`, 2015. Accessed: 2018-08-29.

[3] D. Ferreira and V. Kostakos. AWARE: Open-source context instrumentation framework for everyone. `http://www.awareframework.com/what-is-aware/`, 2018. Accessed: 2018-08-29.

[4] D. Ferreira, V. Kostakos, and A. K. Dey. AWARE: Mobile context instrumentation framework. *Frontiers in ICT*, 2(6):1–9, 2015.

[5] H. Gjoreski, B. Kaluža, M. Gams, R. Mili, and M. Luštrek. Context-based ensemble method for human energy expenditure estimation. *Applied Soft Computing*, 37:960970, 2015.

[6] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams. How accurately can your wrist device recognize daily activities and detect falls? *Sensors*, 16(6), 2016.

[7] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski. Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73:159–170, 2017.

[8] C. D. Spielberger and S. J. Sydeman. *State-Trait Anxiety Inventory and State-Trait Anger Expression Inventory*. Lawrence Erlbaum Associates, Inc, 1994.

BRISCOLA: Being Resourceful In Stacking Cards - Opponent, Lament Away!

Vito Janko
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
vito.janko@ijs.si

Nejc Mlakar
Faculty of Computer and Information Science
Večna pot 113
Ljubljana, Slovenia
nejcmlakar37@gmail.com

Jani Bizjak
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
jani.bizjak@ijs.si

ABSTRACT

This paper describes a robot system that can play the popular Mediterranean card game called Briscola. It elaborates on the three main components needed for an operational platform. First, it describes several artificial intelligence agents that can play the game using a combination of probabilities, heuristics and the min-max algorithm. Second, it describes the computer vision component for card detection using both classical and deep learning approaches and finally, it proposes a scheme for a robotic arm that can move the cards on the table.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

Intelligent Society, Computer Vision, Game Theory, Robot Arm

1. INTRODUCTION

Briscula is one of Italy's most popular card games, played all around the Mediterranean (Italy, Spain, France, Greece, Slovenia, Croatia). Despite that, the possibility of making an artificial intelligence player for the game is poorly researched, with no papers on the topic found. We propose a system that can not only play the game on a computer, but can actually play against human players in the real world using a robotic arm. To do so, we developed three separate modules. First, we used a combination of heuristics, card probabilities and min-max algorithm to plan agent's moves. Second, we used computer vision (CV) algorithms to determine which cards the AI agent has and which cards are being played by the (human) opponent. Finally, we proposed a robotic arm that is capable of picking up a card from a predetermined spot using suction at the arm's end.

1.1 Briscula - Rules Overview

The game is played with a special deck, containing 40 cards divided equally into 4 colors - Spade, Coppe, Bastoni, Denari [Figure 1]. One card is selected at random at the beginning of the game and is placed face up under the deck. Its color is called "Briscola", giving the game its name. In this paper we consider the 2-player variant of the game. Both players start with three cards. Every turn both players play a card and at the end of the turn draw a card from the deck. After both players play their cards, the second player wins if his card shares the color with the Briscola card or shares the color with the first card and has greater strength, which is based on the card number. The winning player gets points corresponding to the played cards value. After 20 rounds, whoever gets more than 60 points wins (a 60-60 score results in a draw).

Figure 1: The 4 aces of Briscola.

2. PLAYING THE GAME

First task was to develop an artificial agent that would be able to play the game in a virtual environment. Due to the lack of external opponents, we developed several progressively stronger AI agents and matched them against each other to determine their strengths. Final version was also matched against five human opponents to further evaluate its performance.

Mr. Random

The first agent plays cards completely at random. While this strategy seems inadequate, it both provides a basic baseline, and demonstrates an interesting property of the game: the game's variance is so high, that even this agent wins more than 5% of the games against the best agent (and significantly more against others), simply by having superior cards. High degree of chance, explains why progressively better agents have diminishing returns in their win rate.

Mr. Greedy

This agent tries to maximizes the score after the current round. It opens the round with the lowest card, while taking with the strongest card, if possible, when second. While this provides the biggest boost in performance, its liberal spending of strongest cards does not lead to optimal play.

Mr. Heuristics
Agent implements author's expert knowledge using if-else rules. The wide set of rules includes holding strong cards until a valuable card is played, trying to start each round second if possible (as seeing your opponent play lets you know how to best respond) and being careful in which situations to open the round with a valuable card.

Mr. Probable
The use of heuristics can be amplified by predicting the likelihood of cards the opponent might be holding. Predicting that the opponent has a strong card of one color, might lead the agent to open the round with another. This was done by weighting the influence of each conflicting if-else rule, by the probability that it applies. The play was then determined by the "strongest" rule.

The prediction of card probabilities however, is not trivial. By counting the cards already played, we can determine cards left in the deck and calculate the base probability that any of them is in the opponent's hand, given his hand size. This probability can be further modified by two factors. First, we can exclude some types of cards from their hand, given their past plays. This step assumes that the opponent has an elementary knowledge of the game, and will play the obviously good play, given opportunity. For each card in opponent's hand, we track when was it drawn, and what plays were made since then. This allows us to predict more precisely what kind of card it is. Second, cards of high strength or Briscula color tend to get "stuck" in player's hand, as players wait for a good opportunity to play them. This means that their likelihood of being in a hand is greater than the base probability would suggest, especially in the later game. Their probability was weighted with an empirically determined weight, that moved from 1 to 1.5 as the game progressed.

Mr. Calculator
To avoid the if-else behavior, an agent can try to calculate different game branches and then decide for one that most likely leads to the desired outcome. There are two popular frameworks for this task: variants of the min-max algorithm and the Monte-Carlo tree search. We decided to try the former and leave the latter for future work.

The base version of the min-max algorithm [8] works with perfect information and thus had to be adapted for this probabilistic case. Instead of using the probabilistic variant of min-max, that would have a huge branching factor, we tried to transform the problem into a perfect information one. Three cases were considered. 1.) In the last three rounds, all cards are drawn and thus we have the case with perfect information. 2.) When only a few cards remain in the deck, we can do an exhaustive search of all possible orders of cards in the deck and all subsets of cards that can be in the opponent's hand, and do a simple min-max search for each possibility, averaging the results. 3.) In remainder of the game we sampled 100 different hands the opponent could have each round, with regards to the probability described in the previous subsection. For each of the possibilities the min-max search is performed and the results were averaged. In all cases, the search depth was set to three rounds, as reliability of our information on the opponent decreases with time. The heuristic used at the end nodes was simply the number of points accumulated in those rounds.

This variant performed best of all described and matches expert human play. The contribution of each min-max use case was individually assessed by replacing it with heuristics for that part of the game, and it was determined that all three parts contribute to the game-play improvement.

3. RECOGNIZING CARDS
In order to be able to play the game in real-life, it is essential to detect cards on the table and cards that are picked up from the deck. In this image recognition problem we assume that the card's images are constant and that they are placed on a mostly uniform background - table. The problem gets complicated due to the fact, that the card's images can be very similar, they frequently overlap in practical play and they can be sometimes covered by the opposing player's hands. Here we present several attempted approaches, ranging from the simplest to the beyond state-of-the art deep learning.

Removing background
Since the cards are on the table and thus the surface color does not change much, we first tried to remove the background color from the image and detect cards left on the table. A predetermined threshold, based on the RGB values of pixels was used. This approach proved unreliable, as the subtle changes in lightning (lights in the room, clouds over the sun) could change the color scheme enough for the threshold to fail to remove enough of the background.

Comparing differences
Similarly, since most of the image is static (table, deck of cards) and the only changes are the two cards being placed on the table, we looked at the history of images and tracked changes between them. Ideally, when a new card is placed on the table it should be the only changed part of the image and could easily be detected. Once exact card position is known, any template matching technique could be used to identify the card. The same approach could solve the overlapping cards problem, as they could be identified one by one, as they are played. In reality however, this approach did not work either. Due to the camera noise, most of the image was constantly changing. Second and bigger problem was that when a card was placed on the table, hands and their shadows went over half of the table, changing pixel values in the image, complicating the use of this approach.

Edge detection
Another attempt was to detect the edges of the objects on the table using Canny Edge detector [2]. To detect a card from the edges we used Hough Line Transform [3] in order to detect straight lines that could later be combined into square shapes to form cards. This improved the results significantly, however due to the camera noise and wood pattern of the table, the edges were often miss-detected.

SIFT
SIFT [5] is a scale-and-rotation-invariant image-recognition algorithm, which means that the object in the image can be rotated or scaled and the method should still be able to

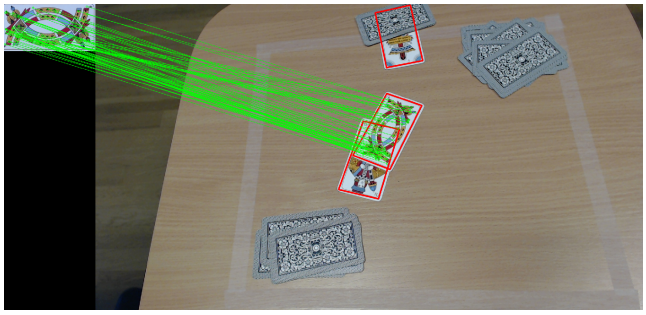


Figure 2: Detecting Spade 4 card. Matching features from template image to camera view.

detect it. SIFT finds so called interesting points (features) in the image, these are usually shapes of edges, stores all features from the sample images and then compares them with the features found in the new images, as seen on Figure 2. By comparing features position, the algorithm can also determine the image position, rotation and scale.

This algorithm proved to work much better than previously described methods. It's detection rate was high, detecting even partially obscured (overlapped) cards.However, it has two drawbacks. For each frame in the video (image) it has to compare image (table) with 40 template cards. Depending on the resolution this can be very slow, ranging from 1 to 15 seconds. Sadly the accuracy is correlated with the size of the image, which means that for good predictions high resolution images need to be used, which slows detection significantly. Second problem was distinguishing between lower "Denari" cards.

Deep learning

In the last decade deep learning has become the dominant ML approach for multiple domains, with different deep learning architectures achieving almost human level precision for problems regarding computer vision.

The standard approach is using several layers of convolution, which is similar to what SIFT does, and then combining several fully connected layers in order to classify the features. This works fine for image classification, but is unable to detect objects on the image. A naive approach would be to have a small sliding window that goes through the whole image and classifies every part of it. This would be accurate, but extremely slow. Several approaches have been developed in order to tackle such problems: YOLO [6], Faster R-CNN [7], SDD [4]. Mentioned papers all go through the image only once (working with 30-60 fps), but still achieve comparable results to slower window CNN approach.

To test how well deep-learning approach works on our problem we implemented the YOLO architecture. Architecture consists of roughly 120 layers of convolution, pooling, regularization and fully connected layers. The training started with pre-trained weights, obtained from VOC 2017 object detection. We manually labeled around 1000 card images, using the VOC format, and then trained the network for 3 days on NVIDIA's GeForce GTX 1080 graphic card. The trained network performed relatively fast, achieving around 15 fps, which is more than enough for real time detection. Detection accuracy was high when there were only 1 or 2 non-overlapping cards on the table [Figure 3], however it had problems with overlapped card. After investigating, we found out that since the network is trained with bounding boxes that are always aligned with the x and y axis, if the card is tilted at an angle, only half of it will be in the bounding box. Therefore the network is unable to learn to tightly detect a card and when they overlap the overall error is smaller if it just combines the two cards into one bounding box.

Figure 3: Detecting cards with YOLO is fast and reliable if objects do not overlap.

To solve this problem we started working on a modified architecture that in addition to bounding box also predicts the angle at which the bounding box is rotated. The initial results on generated data (photos of cards stitched on top of different backgrounds) show promising results, where for most of the single cards in the image the network correctly predicts the rotation of the bounding box. The network works a bit worse where there are two overlapping cards but still manages to recognize a large percentage of images. We believe that with some more time, larger set of training images, tweaks and optimization of the architecture we could achieve close to 100% accuracy for the detection using this new architecture.

3.1 Robotic Arm and Cameras

The last step in bringing the agent to the real world is the presence of sensors and actuators. This component is composed of two cameras and a simple robotic arm. The robotic arm has 4 Degrees of Freedom created by 4 servo motors, that are controlled by Wemos D1 mini board. The board acts as a web client, receiving the commands from the main server and executing movement (controlling servo motors) actions. At the end of the arm there is a suction pump for lifting and dropping the cards. All movements are predetermined and described with sets of motor's rotation degrees. Two movement patterns exist: drawing a card and placing it on one of the three predetermined spots and picking the card from one of the three spots and dropping it at the center of the table.

The system also has two cameras, first one to overlook the table – tasked with detecting the cards played by the opponent. The second one is behind the arm, turned from the floor up. Before dropping a card on the table, arm is rotated

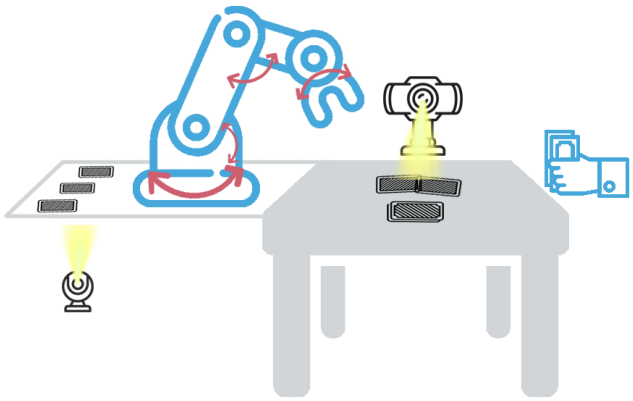


Figure 4: Server controls the robot arm. There are two cameras to oversee the table and picked up cards from the deck.

	R	G	H	P	C
R	-	26	13	14	6
G	74	-	21	18	15
H	87	79	-	42	34
P	86	82	58	-	40
C	94	85	66	60	-

Table 1: Win percentage (%) of row agent against the column agent. Agents: Mr. Random (R), Mr. Greedy (G), Mr. Heuristic (H), Mr. Probable (P), Mr. Calculator(C)

so the card is over the second camera and can be identified. The system is schematically presented in Figure 4 and its first prototype is recorded and can be seen on the web [1].

The main logic for controlling the arm and taking actions is on the server, coupled with CV model and AI in order to take appropriate actions.

4. RESULTS

We started by assessing the comparative strength of the different AI variants. Each played 1000 games against each other. Results are shown in Table 1 and show that the agent's skill increases with their increasing complexity. It also shows high degree of variance in the games, as even random agent got surprisingly many wins and the best agent - Mr. Calculator is achieving only 66% win rate against simple ones. In repeated experiments we noted, that win rate fluctuates ±2% between runs.

Next we compared the play strength against 10 human opponents of different skill levels. Each played 10 games against Mr. Calculator. Results are listed in Table 2 and show an average 69% win rate of the AI against the human opponents. Volunteers that played, commented that the skill level of the agent is quite high, with some room for improvement in regards to increasing the agent's risk aversion. While the sample size is too small for definitive conclusions, we can assume that the agent is at least on par with average human players of the game.

1	2	3	4	5	6	7	8	9	10	avg.
80	90	80	90	85	70	60	60	25	50	69

Table 2: Win percentage (%) of Mr. Calculator against 10 human opponents of roughly increasing strength.

To test the CV component we recorded several human games from the same angle as the final system uses. We then manually compared the cards predicted by the CV with the actual ones. The best two approaches were SIFT and YOLO algorithms. The first worked flawlessly in all cases, except differentiating some of the Denari cards. The second could flawlessly recognize all cards, when they were not overlapped. Overlapped cards had roughly 50% accuracy. In the end we decided to use the SIFT algorithm for our first system prototype, since second player usually overlaps the first card.

5. CONCLUSION

In this work we described three different components (from different computer science fields) of a system that is able to play the Briscola card game against the human opponent in a real-life setting. For each component we individually tried different approaches, creating a strong AI agent and a serviceable CV and robotic component. While the current version should be able to reliably play the game, all components still have room for improvement - we plan to test the Monte-Carlo search tree and improve the deep learning architecture. We hope that we will be able to successfully present a live demonstration at the paper's presentation.

6. REFERENCES

[1] Robot prototype. https://dis.ijs.si/wp-content/uploads/2018/10/briscola/briscola_AI.mp4, 2018.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[3] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[8] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

Emotion Recognition Using Audio Speech Signal

Maj Smerkol
Jožef Stefan Institute
Jadranska cesta 39
Ljubljana, Slovenia
maj.smerkol@ijs.si

Mitja Luštrek
Jožef Stefan Institute
Jadranska cesta 39
Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT
Emotion recognition is an important part of affective-aware applications. Specifically, using audio speech signal has the advantage of being compatible with applications using a natural language interface. There are multiple valid representations of emotions. We propose a new representation aimed at using differently labeled databases jointly. We include a short overview of some of the available databases and methods for feature extraction and selection. Both classification of emotions and regression in 2D emotional space are discussed. We concentrate on using neural networks for both tasks. Regression provides good results but is hard to interpret while classification is more robust.

Categories and Subject Descriptors
I.5 [**Pattern recognition**]: Neural nets; I.5.2 [**Design methodology**]: Classifier design and evaluation

Keywords
Emotion recognition, Neural networks, Affective computing

1. INTRODUCTION
Nowadays applications such as personal digital assistants are becoming more popular. Some also utilize natural language interfaces. Next step in this direction seems to be affective computing - applications that can detect human emotions. Such applications can enrich the user experience by responding according to the user's current mood and perhaps even detect when the user is not happy with the application's functioning. However, in order to implement such applications we need to first be able to understand the user's emotions. This, in conjunction with other knowledge (such as user's daily routines and other contextual information) makes it possible to detect certain mental health problems, such as depression or bipolar disorder, shown by Osmani et al. [8].

Models we are developing will be used in an emotionally-aware virtual assistant application. Our priority is to deliver information that can be acted upon in order to better the user experience. Application's target population are people from Italy, Spain and Denmark.

1.1 Representations of Emotions
When talking about emotions in the context of affective computing, we first need to consider how to represent human emotions. In psychology, there exist many different theories about human emotions. We can choose a discreet representation of emotions or a contiuous representation in some space of emotions. In first case, we define different categories that represent different emotional states. The most widely known categorization is Paul Ekman's basic emotions. Ekman studied facial expressions of emotions across different cultures and came to the conclusion that there are six basic emotions that are expressed equally across cultures. Those are sadness, happiness, anger, fear, disgust and surprise.

On the basis of Ekman's work others proposed different models. Some of them have a different set of categories. Others use a continuous representation in two, three or four dimensional spaces. There is Plutchik's wheel of emotions that represents emotions as four pairs of exclusive categories, that are treated as four axes along which emotions are spread. Emotions are represented as points in this space. J. Russel proposed a different model, a two dimensional space. Dimensions are arousal, which represents how active one feels, and valence, representing pleasurableness of the emotion.

For our purpose, we prefer classification robustness over precision. We don't need very fine-grained information to better the user experience of the application. Therefore in order to use as much training data as possible, we propose a four-class representation of emotions. The main idea of this representation is to be able to easily transform labels in other representations into a common one. Classes correspond to quadrants in space of arousal and valence, and to groups of Ekman's basic emotions: **Happy**: positive arousal and positive valence, includes basic emotion happiness. **Calm**: negative arousal and positive valence, there are no basic emotions in this quadrant. Instead we include neutral. **Sad**: negative arousal and negative valence, includes basic emotions sad and bored. **Upset**: positive arousal and negative valence, includes basic emotions disgust, anger and fear.

Therefore, we can jointly use databases that are labeled in space of arousal and valence (Recola, Semaine), as well as those labeled discretely (EmoDB, Ravdess).

1.2 Learning from Features or Raw Audio
Traditionally in machine learning we first extract features from audio. This can be done using specialized software, such as OpenSMILE [3], or libraries, such as LibROSA [6].

With deep learning, it is possible to learn from raw audio sig-

nal. This recent approach is interesting, as in the raw audio signal there is encoded certain information that is missing in extracted features. Deep learning has two problems: (1) larger databases are needed for training, and (2) training is very computationally expensive, both regarding computational power and large amounts of memory needed.

2. DATABASES

There are many public audio databases available for use in affective computing. Most of them are targeted towards speech recognition or a subset of emotions, specific for a given problem (such as detecting frustration in call centers). We describe the few of them that we have used. [1]

We chose those based on the way they were labeled, language and audio format used. Regarding labels we preferred labels in space of arousal and valence or basic emotions in order to be able to do both regression in some emotional space or classification of emotions. We decided to only use European languages, since it has been shown that model trained on language from similar cultural background to target population gives slightly better results [2]. Audio simply needs to be of high enough quality. Human speech ranges up to 5kHz so we need at least 10kHz sampling rate. To be on the safe side and not lose any non verbal information we decided to only use audio recorded at 16kHz or higher.

2.1 EmoDB

EmoDB [1] (Berlin Database of Emotional Speech) is an older database. It contains 535 utterances spoken by 10 different actors. Each actor expressed each of the Ekman's 6 basic emotions (and a neutral version) at least once for each of the ten different texts. Each file is labeled. Texts themselves are emotionally neutral. Utterances are quite short, recordings are between a couple of seconds long up to half a minute.

Problematic aspects of this database are:

- Utterances are very short. Often when classifying audio, recordings are cut into segments from 1 second up. If we do that with EmoDB, there are simply not enough instances to use deep learning techniques, in same cases there are even not enough for traditional ML.
- Expressed emotions are extreme to the point of overacting. This means that classifiers trained on this set may produce weak generalization, as most speech is closer to neutral as considered in this database.

2.2 Semaine

The SEMAINE [7] database is a multi-modal database that includes audio, video and transcripts of English texts. The database is labeled on a continuous scale along many dimensions. Not all sessions (couple of minutes long recordings) are labeled in all dimensions. Most are labeled along the arousal and valence dimensions, as well as intensity and power. Fewer are labeled along basic emotions (e.g. only 2 session labeled for fear). Each available dimension is labeled by at least 2 annotators. Differences among different annotators are quite noticeable which is to be expected in such a setting.

Problematic aspects of this database are:

- Very unbalanced due to the chosen labeling methodology. Counting each label sample, there are almost 4x as many examples of low arousal and high valence than examples of high arousal and low valence.
- For some dimensions label values span a very small interval, which may cause problems with regression along those dimensions.
- Differences between annotators are often quite big. Some files have inter annotator correlations below 0.2. While this is not unexpected - emotion expression and perception are inexact - it is problematic for training and testing.
- Expressed emotions are very mild and often noticeably acted. There are examples in which we can hear the actor, supposedly gloomy and depressive, express amusement by laughing. While extreme emotions are problematic so are very mild emotions - ML algorithms often overfit to find other characteristics in the data.

2.3 Recola

Recola Database is a French multimodal datset of emotional speech. It includes audio, video, biosignals, labels (annotations) and metadata. It is similar to Semaine in that it is also labeled continuously. It is only labeled along arousal and valence dimensions, but labels are of higher quality. Each recording is labeled by 6 different annotators, 3 male and 3 female.

Problematic aspects of this database are:

- Each file is exactly 5 minutes long, but some of the labels are missing a few samples. We have cut the audio files to match the label lengths.
- There are only 23 recordings. Since each is 5 minutes long it is still quite large.
- It is quite unbalanced. Counted by each label sample, there are more than 8x as many examples of high arousal and high valence than examples of low arousal and low valence.

2.4 Ravdess

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains video and audio of 24 actors. Emotions expressed include the 6 Ekman's basic emotions, neutral and calm. Each utterance corresponds to one emotion. There are 1440 files in the speech section and 1012 in the song section.

Problematic aspects of the speech section of this database are:

- All utterances contain one of the two texts: (1) "Kids are talking by the door" or (2) "Dogs are sitting by the door". This may represent a problem as all files are in a way very similar. On the other hand, this is good as it helps prevent overfit as the algorithm can't learn to differentiate utterances based on text contained.

[1] Some of reportedly high quality databases such as the Humaine database and Vera am Mittag (eng. *Vera at noon*, a database of German emotional speech taken from reality TV and talkshows) are not available anymore.

- Utterances are very short, similar to EmoDB.

3. FEATURES

Among features used for classification of audio of human speech are (1) simple features such as loudness, signal energy and pitch, (2) Mel spectrum: similar to frequency spectrum, transformed to Mel scale which corresponds roughly to human perception of pitch, (3) Mel frequency cepstrum coefficients: inverse Fourier transform of log-scaled Mel spectrum, (4) Jitter, shimmer: frequency noise instability and amplitude instability, (5) Formants: most present harmonic frequencies, (6) Spectral features: describe the shape of the frequency spectrum and (7) Chroma features: describe tonal properties, such as melody.

3.1 Tools

Some of the commonly used tools for audio feature extraction are OpenSMILE and LibROSA.

OpenSMILE is a standalone program with a very step learning curve. Writing custom configuration files which is needed for extracting custom features as opposed to using one of the predefined features sets is quite complicated. Most users use predefined configurations, which can also be found online.

LibROSA library is an easy to use alternative that works with Python and offers similar functionality. It also offers some utility functions for reading and storing audio files, filters etc.

3.2 Feature Selections and Analysis

Feature selection is an important step in the ML pipeline as having fewer features is beneficial for reducing training time as well as reducing the possibility of overfitting.

We have performed feature selection using each of described databases, using features calculated by OpenSMILE (using a slightly modified ComParE13_LLD configuration) and separately using features calculated using the LibROSA library.

1. Remove features with variance below 0.2, as they hold little information.

2. Sort by correlation with labels and remove those with absolute correlation below 0.1 as they mostly contribute noise or bias towards groups with certain vocal qualities.

3. Greedy feature selection: take the feature with the highest correlation, add it to the feature set and test on a surrogate model (logistic regression or random forest classification). We use surrogates to reduce the computation time. Keep feature if it improves performance of the surrogate.

Using this method the number of features was reduced from 132 to 60 (feature set ComParE_lld extracted using OpenSMILE), and from 167 to 110 (custom feature set extracted using LibROSA). We achieved the same performance on the models while reducing the training time compared to no feature selection.

4. EXPERIMENTS

We have tested regression in the space of arousal and valence and classification of basic emotions in order to compare two very different approaches and decide which is preferable for our use-case.

4.1 Regression in Arousal and Valence Space

While deep learning on raw audio signal is slower and more computationally expensive, it may produce better results as raw signal contains more information. We have replicated the experiment done by Trigeorgis et al. [9]. Due to hardware constraints we had to introduce certain modifications: (1) we had to use 3 second segments instead of 6 second segments and (2) we used a mono-directional LSTM layers instead of bi-directional as in the paper. Network topology is otherwise same.

Training and testing was done on the RECOLA database. Data was split into train and test sets by actors - 80% of actors in the train set and 20% in the test set. Our results were very similar to those reported in the paper. Measurements shown in Table 1 are Concordance correlation coefficients (CCC)[2] between predictions and ground truth, obtained as averaged labels. Predictions are scaled to have the same standard deviation as the ground truth and time-shifted in order to remove any delays that a human annotator may produce. Thus we can confirm that deep learning from raw audio data is feasible.

	Arousal CCC	Valence CCC
Raw audio	0.641	0.250
Features	0.574	0.187
Trigeorgis et al. [9]	0.684	0.249

Table 1: Valence and arousal regression results

The network is made of two distinct functional units. First are the convolutional layers that learn to perform feature extraction. It has been shown [9] that certain neurons are highly correlated to some of the known good features. The second part is made of two LSTM layers. These learn to regress arousal and valence from extracted features.

The same experiment was repeated using only the second part of the neural network, trained using extracted features (feature set ComParE_lld). Results were somewhat worse, which indicates that the convolutional part of the full neural network learns to extract a better set of features than we get using simple feature selection (as described above). Unfortunately predictions in the space of arousal and valence are hard to interpret and there is no direct way to convert them to basic emotions.

4.2 Classification of Emotions from Features

We have used EmoDB for initial experiments. All reported results are averaged over leave-one-person-out cross validation. Simple fully connected feedforward neural networks tend to overfit. This can be reduced with hyperparameter adjustment (learning rate and algorithm, mini-batch size, early stopping etc).

[2]Concordance correlation coefficient is a measure of agreement, often used to measure inter-rater reliability.

This was tested on the database split into 3 sets, train, test and evaluation. Using 3 sets show that overfit is still there, but the difference in performance was small between train set and test set probably due early stopping based on test set loss. Performance on evaluation set is still much lower.

We used all features from the ComParE_lld feature set. Input layer therefore has 130 units, fist hidden layer 70, second hidden layer 30 and output layer 7 (6 Ekman's basic emotions + neutral). For the experiment, MSE was used as loss function, and Adam as optimizer. Without using regularization, we achieve very poor performance. As the model starts to overfit we stop training it, which is before it achieves good performance. Without regularization accuracy is therefore very low on all sets. Even with strong regularization, using both added Gaussian noise ($std = 1/2std(features)$) to input layer and dropout ($p = 0.5$), large differences on train set and evaluation set can be seen.

We compare out results to state of the art as achieved by Yenigalla et al. [10] in 2018 and Gjoreski et al. [4] from 2014. Yenigalla et al. achieved high performance using convolutional neural networks, trained using extracted features and phonemes. IEMOCAP dataset was used. Gjoreski used Auto-WEKA, a machine learning tool that automatically chooses best classical-ML algorithm. They trained and tested using EmoDB.

	Accuracy (test)	Accuracy (eval)
No regularization	0.54	0.48
Noise, dropout	0.82	0.65
Gjoreski et al.	/	0.77
Yenigalla et al.	/	0.73

Table 2: Classification results for test set and evaluation set, compared to state of the art [10].

We have also performed some preliminary experiments using Optimal Brain Damage (OBD) algorithm[5] to prune the network. Results are not yet conclusive but seem promising. We did not achieve better performance, but did achieve same performance while pruning up to 60% of all units.

5. CONCLUSION

We have experimented with regression in space of arousal and valence. Results confirm that a combined convolutional and recursive neural network can effectively learn on raw audio signal. Since the authors who propose this approach state that the convolutional part of the network learns to perform feature extraction we tested only the recursive part of the neural network, trained on pre-extracted features. Results were somewhat worse, which can be interpreted as the convolutional part of the network learns to extract better features. Aditional experiments, such as classification using a similar neural network are needed in the future.

We have also tried using a fully connected artificial neural network (FNN) to classify emotional speech. FNN is extremely prone to overfit. Even using very aggressive regularization techniques show some overfit. It seems that either (1) FNNs need a larger amount of labeled training data or (2) are not well suited for this problem. Related future work is performing experiments using OBD to prevent overfit.

A new categorization of emotions was proposed with the aim of using multiple databases jointly. Preliminary experiments show that we can use it for machine learning on multiple databases. Whether models trained in such way will perform better is yet to be seen.

In conclusion, emotion recognition using audio signal is a complex and difficult task. Some of our experiments come close to state of the art, but still not very good. We believe we can improve our work further in the future.

6. REFERENCES

[1] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss. A database of german emotional speech. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[2] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[3] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM, 2010.

[4] Martin Gjoreski, Hristijan Gjoreski, and Andrea Kulakov. Machine learning approach for emotion recognition in speech. *Informatica*, 38(4), 2014.

[5] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[6] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, pages 18–25, 2015.

[7] Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schroder. The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Trans. Affect. Comput.*, 3(1):5–17, January 2012.

[8] Venet Osmani, Agnes Gruenerbl, Gernot Bahle, Christian Haring, Paul Lukowicz, and Oscar Mayora. Smartphones in mental health: detecting depressive and manic episodes. *arXiv preprint arXiv:1510.01665*, 2015.

[9] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE, 2016.

[10] Promod Yenigalla, Abhay Kumar, Suraj Tripathi, Chirag Singh, Sibsambhu Kar, and Jithendra Vepa. Speech emotion recognition using spectrogram & phoneme embedding. *Proc. Interspeech 2018*, pages 3688–3692, 2018.

Improvement of AI through Deep Understanding

Jani Bizjak
Department of Intelligent Systems
Jamova cesta 39
1000 Ljubljana
jani.bizjak@ijs.si

Matjaž Gams
Department of Intelligent Systems
Jamova cesta 39
1000 Ljubljana
matjaz.gams@ijs.si

ABSTRACT
Humans have concerned with semantics behind words, objects and facts for thousands of years. Yet, when a computer uses a model and learns to recognize it from a video, does it really understand what it is or does it only map a number of specially colored pixels and shapes to a term "object"? In addition, humans use semantics to improve performance; therefore, it seems reasonable to assume that computers would as well. This paper overviews latest state of the art papers that learn and use semantics in order to improve the results of established methods in different areas. Two branches of AI: natural language processing and computer vision seem to be especially active in this area.

Categories and Subject Descriptors
I.2.m [Artificial Intelligence]: Semantic analysis

General Terms
Algorithms, Theory.

Keywords
Review paper, semantics, ML

1. INTRODUCTION
Machine learning and artificial intelligence are as old as modern computer science. However, there is an essential difference. Even though we have thought the computer for example to recognize the images of a cat, be able to predict the future of a stock or play a game of chess, in reality we just thought computers a good enough mathematical model that approximates the real world application. The computer methods seem to be qualitatively much different than the way humans use intelligence and learn.

In early 2010, deep learning started gaining popularity. Google was the first one to successfully use deep neural networks (DNN) when trying to classify cats in photos on the internet. They used Convolutional Neural Networks (CNN) architecture and were incredibly successful. No method before achieved such classification accuracy for image recognition, not yet similar to humans, but splitting the difference in two. A couple of years later, the DNNs achieved the human level accuracy and from that threshold on, each year further improvement in their performance is made.

Because deep networks work like a black box, we do not know exactly why something is classified as is. People started to wonder, has Google made a first step to superintelligence? Has their method actually learned what a cat is, learned the meaning, semantics and everything that goes with it? With further experiments the question deepened. For example, the network "knew" that a cat has four legs, a tail and so on. It knew how to distinguish it from a dog, who also has a tail and four legs. In the end however, it tuned out the method was practically as "shallow" as everything that came before. The legs of the animals were collection of pixels statistically grouped around the body, the color was a number and classification was a calculation pointing towards one possible class. The class itself was a number and was linked to a term "cat" by humans.

In the past and nowadays alike, semantics are often treated just as another feature, another numerical input to the computer system. The computer uses it to improve performance, but in a similar way as any other numerical feature, without additional semantics, meaning or procedures attached to it. It can and usually does increase classification accuracy, but the model does not understand what it means. For computers to "understand" does not necessarily mean to be very similar to the way humans use semantics, in particular regarding the way the understanding is coded in the computer model, which is likely to be different in humans. Rather, understanding in computers should be functionally somehow similar, i.e., enabling solving tasks in somehow similar way.

In this paper we provide a brief overview of the latest state-of-the-art papers that learn or use semantics to further improve their models.

2. SEMANTICS IN NATURAL LANGUAGE PROCESSING

2.1 Approximating Word Ranking and Negative Sampling for Word Embedding [8]
Word embedding is a technique to present each word by a dense vector, aiming to capture the word semantics in a low rank latent space, e.g. each word is translated into a vector of 0 and 1s in such a way, that words that are semantically closer differ in less bits than semantically different words. It is widely adopted in Natural Language Processing (NLP) tasks. Variants can also be used in almost any domain where semantics play a role, such as computer vision. One of the latest approaches in implementing word embedding is Continuous Bag-of-Words (CBOW) [9]. CBOW predicts a target word given a set of contextual words, where the target word is labeled as positive and the others are classified as negative, e.g. if we have a sentence and want to predict word w_i (positive) we take a look at the neighboring words w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2} (negative). However, the method treats all words equally based on frequency in text instead of favoring the positive ones.

Authors of this paper [9] develop a new approach to word embedding, based on CBOW that favors positively ranked words. They do so by selecting negative words that tend to decrease overall performance. The method works in two steps: it first increases the score of the positive words, and in the next step decreases the score of the negative words. With this approach they improve the overall performance compared to other BOW approaches.

This approach works relatively well for larger texts but still struggles with short texts. Authors in [7] propose additional steps

that can be taken in order for the approach to also work with short texts. The words are usually represented as vectors, and based on the Hamming distance of the embedded vectors, one can notice that semantically closer words are also grouped closer (have shorter Hamming distance) as seen in Figure 1.

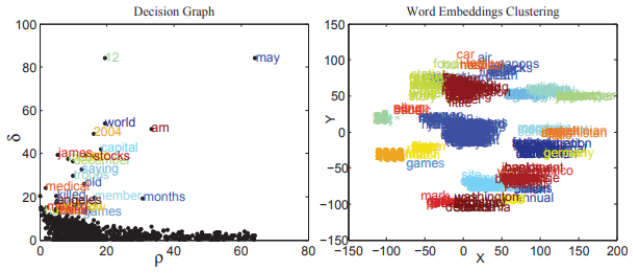


Figure 1. Word clustering based on their semantical meaning. In the left image the words are randomly located around 0,0, while on the right side image we can see that clusters form.

2.2 Task-Guided and Semantic-Aware Ranking for Academic Author-Paper Correlation Inference [6]

In this paper [6], the authors consider author-paper correlation inference in big scholar data, such as Google Scholar, Microsoft Academic or aMiner. In other words, they would like to provide an author relevant and related publications based on the author's previous papers and citations.

To solve the problem, the authors propose a model by joint content semantic encoding and heterogeneous relations augmented ranking, and design the corresponding learning algorithm. In the first step they use Gated Recurrent Neural Networks (GRU) in order to obtain latent features for authors and semantic embedding for each paper. To further improve the results, authors also include citations and transitive citations (multiple papers deep) of author and his/hers papers using a heterogeneous network (HetNet). The architecture is presented in Figure 2.

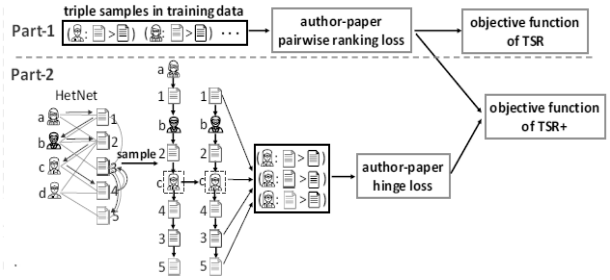


Figure 2. Framework is composed of GRU for direct author paper relation and HetNet for indirect author paper correlations.

The paper learns semantic representation of each paper and then compares it with related papers allowing authors to better find related work based on their paper history.

3. SEMANTICS IN COMPUTER VISION

3.1 Semantic Locality – Aware Deformable Network for Clothing Segmentation [1]

This paper [1] tries to solve the problem of clothing segmentation and identification from photos of people wearing them. While the network doesn't predict or use semantics as an output or input for predictions, semantics proves to be essential in training phase.

In order to learn to recognize different pieces of clothing, the authors proposed a twofold deep learning architecture [Figure 3]. The first part is standard CNN architecture while the second is only the feature extraction part. Both networks are then forced to produce features for different (pairs) of images. If the images are semantically similar, the output features should be as close as possible.

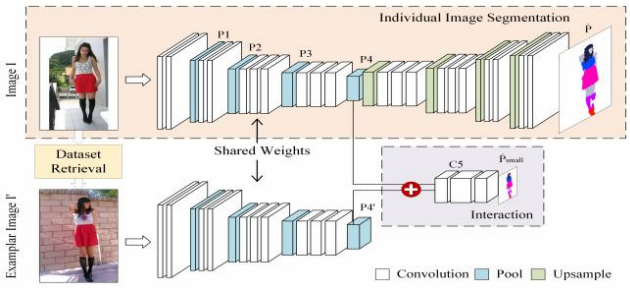


Figure 3. Semantically similar images should have similar weights.

In this paper the contextual knowledge of clothing images is manually defined as finding neighboring images with similar appearances or poses. This is because such images also have similar high-level features. The authors decipher the pose and appearance from the image using OpenPose [2] tool by extracting it from the convolutional layers. To then find the two closest images they use Euclidian distance between the extracted features.

The semantics in this paper are manually defined as a pose and appearance of the model in the image. The method in the end is still incapable of understand semantical meaning of the clothing, but can use it as an additional feature in order to help with training and in the end increases overall accuracy of the model.

3.2 Deep Joint Semantic-Embedding Hashing (DSHE) [3]

The days when searching for similar images meant comparing histograms, colors or metadata attached to the image are long gone. Nowadays the labels for the images are automatically created, for example if the image contains a cat it will be labeled as a cat and so on for every object it contains. This works well for reasonable numbers of pictures, but because of the sheer number of images on the internet it would take too long to compare labels for each image that exists. To solve this problem, a special hash is used that transforms the text label into a vector of 1s and 0s. Ideally the visually (contextually) similar images should have this vector very close to each other when using the Hamming distance. This means that a vector of a dog should be closer to the vector of a cat compared to a vector of a dinosaur, which should still be closer to the pair then to a vector of a truck. The approach is similar to the vectors gained from word embedding described in the previous sections.

Authors in this paper [3] present a new approach to hashing. They use twofold deep architecture [Figure 4], one part of which is tasked with feature extraction (CNN), while the other embeds labels to vectors. The features extracted are then joined in common semantic space, where dependencies between image and labels are learned. By doing so the features extracted from the image are forced to be similar for images with similar semantics.

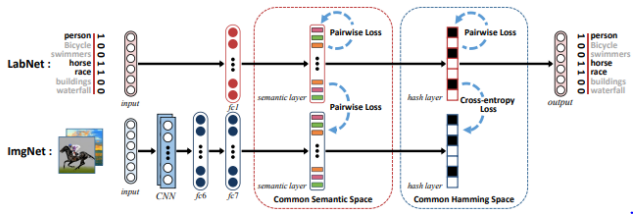


Figure 4. Features extracted from labels and images are joined in second half of the architecture where semantic dependencies are learned.

The semantics used in this paper are hidden and calculated inside the network. The network actually learns deeper connections between labels and the images. The network now not only knows how a cat looks like but also knows that a cat is closer to a dog than it is to a car.

3.3 Semantic Structure-based Unsupervised Deep Hashing [4]

Hashing is becoming increasingly popular for approximate nearest neighbor searching in massive databases due to its storage and search efficiency. Related work shows promising results when learning from labels, however it is significantly more difficult to do the same in an unsupervised setting.

The paper [5] shows that a lot of semantic information can be extracted from features obtain from CNN. Authors first analyze statistical properties from the obtained features. With this information they are able to construct a semantic structure that explicitly captures the semantic relationship across different data points. In the following step they calculate semantical distance between two points using cosine distance. The experiments show that semantically closer features have lower distance - as expected. In the last step they use special loss function that calculates inner product between significantly similar or dissimilar points and use it to train hash codes using deep learning. Network schematics can be seen in Figure 5.

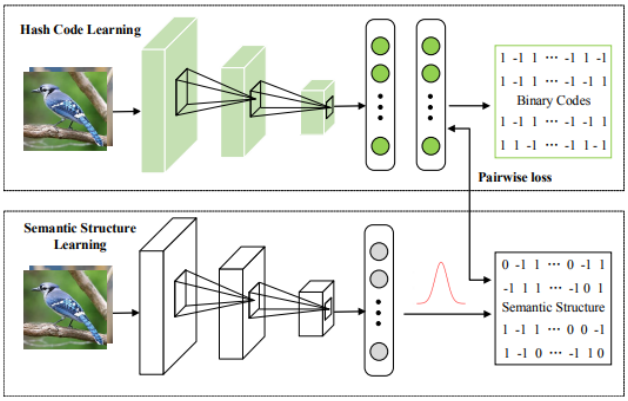


Figure 5. Network architecture. Network first discovers statistical properties from CNN features. Then it learns their hash functions using special loss.

This paper [4] takes the next step and removes the need of semantically labeling the images. It forces the architecture to learn semantic relations without telling it what the main context / feature of detected object is.

3.4 Adversarial Attribute-Image Person Re-identification [6]

In previous sections we described a different approaches in extracting and storing semantical meaning from different types of images. In this section, one possible use case is presented where semantics are used to find or identify a person in an image.

In the previous section [3.2] it was presented how the computer method finds similar images using semantical labels and hash codes. Those approaches work fine; however, they have one major drawback - they require an input image from which they can calculate those features and then search for similar image. However, when humans want to look up for someone or describe it to someone, they usually describe that person's features [Figure 6], for example: Caucasian, male, 1.8m tall, blue eyes, wearing a hat and a blue backpack. If a human were given these instructions it would be easily for them to find a person in a set of images, but for computer methods, on the other hand, this present a major problem.

Figure 6. Person identification using high level descriptors.

The framework described in the paper learns semantically discriminative structure of low-level person images, and generates a correspondingly aligned image-analogous concept for high-level attribute toward image concept. This averts direct estimation of the attributes in a person image and solves the problem of imperfect prediction and low semantic discriminability.

The framework [Figure 7] is trained using adversarial learning approach. This means that part of the network is trying to generate concepts while the other part is trying to distinguish if they are good or not. Both parts of the network are trained simultaneously, learning to be better at their task and thus competing against each other while at the same time improving.

The network again consists of two parts. The first part is tasked with concept extraction from the images, while the other branch is tasked with concept generation from the labels. Both branches are then joined in semantic classification.

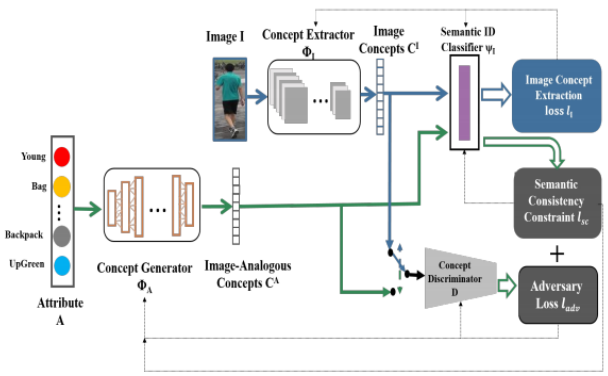


Figure 7. Network architecture has two parts, concept extraction from image and concept generation based from the description.

4. CONCLUSION

In this paper we presented a short overview of latest state of the art methods and approaches that use semantics in several different ways. We first looked at semantic extraction in NLP tasks, more precisely with word embedding, where semantically closer words also have smaller Hamming distance. In later sections we looked at semantics being used primarily as an additional attribute for object recognition in images.

The architectures used are similar across all papers. They consist of two part deep networks, one part is usually tasked with extracting features from the image, while the other part is tasked with extracting semantical meaning, either from labels or learning it on its own. In the next steps both parts are merged, which forces the network to incorporate semantical knowledge into the features extracted from the images.

One can see that semantics are more often than not used as an additional parameter, i.e. feature, which helps the established model achieve better accuracy. The question here is if true understanding of semantics behind could further increase the model's performance and bring us one step closer to true intelligence or even superintelligence. With enough training examples, can an architecture learn the deeper meaning behind the images, words and sentences and use it to better model the real world?

In summary, one of the central questions can be presented as follows: *Can deeper understanding through automated semantic extraction increase the AI performance independently of domain or task?*

5. REFERENCES

[1] Ji, W., Li, X., Zhuang, Y., Bourahla, O.E.F., Ji, Y., Li, S. and Cui, J., 2018. Semantic Locality-Aware Deformable Network for Clothing Segmentation. In *IJCAI* (pp. 764-770).

[2] Cao, Z., Simon, T., Wei, S.E. and Sheikh, Y., 2016. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*.

[3] Li, N., Li, C., Deng, C., Liu, X. and Gao, X., 2018. Deep Joint Semantic-Embedding Hashing. In *IJCAI* (pp. 2397-2403).

[4] Yang, E., Deng, C., Liu, T., Liu, W. and Tao, D., 2018. Semantic Structure-based Unsupervised Deep Hashing. In *IJCAI* (pp. 1064-1070).

[5] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[6] Zhang, C., Yu, L., Zhang, X. and Chawla, N.V., 2018. Task-Guided and Semantic-Aware Ranking for Academic Author-Paper Correlation Inference. In *IJCAI* (pp. 3641-3647).

[7] Wang, P., Xu, J., Xu, B., Liu, C., Zhang, H., Wang, F. and Hao, H., 2015. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (Vol. 2, pp. 352-357).

[8] Guo, G., Ouyang, S.C., Yuan, F. and Wang, X., 2018. Approximating word ranking and negative sampling for word embedding. IJCAI.

[9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Assessment and Prediction of Auxiliary Carabid Species in Agricultural Fields

Marko Debeljak
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova 39, Ljubljana, Slovenia
+386 1 477 3124
marko.debeljak@ijs.si

Vladimir Kuzmanovski
Jožef Stefan Institute and Jožef Stefan Int. Postgraduate School
Jamova 39, Ljubljana, Slovenia
+386 1 477 3143
vladimir.kuzmanovski@ijs.si

Sašo Džeroski
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova 39, Ljubljana, Slovenia
+386 1 477 3127
saso.dzeroski@ijs.si

Veronique Tosser
ARVALIS
91720 Boigneville, France
+33 1 64 99 23 15
v.tosser@arvalis.fr

Aneta Trajanov
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova 39, Ljubljana, Slovenia
+386 1 477 3662
aneta.trajanov@ijs.si

ABSTRACT
Biological pest control depends on the abundance and richness of beneficial species. Development of efficient pest management plans requires new knowledge about complex interactions between the elements of agricultural ecosystems and their natural and management environment. Empirical ecological data represent a big obstacle in the acquisition of this specific knowledge as they are most often, incomplete, inconsistent and imbalanced. In addition, they require a lot of pre-processing for their use in analyses and modelling. In this study, we are dealing with carabid beneficial species that could provide biological pest control in agricultural fields. In particular, our goal is to describe taxonomical and functional diversity of carabid species to assess the potential performance of biological pest control in the studied area and to develop predictive models for the most abundant carabid species and their predator functional group. The results show high potential of carabids to provide biological pest control in the studied area, but the predictive models achieved relatively low predictive performance. They could be improved by an additional set of attributes describing specific habitat requirements of carabid species.

Keywords
Carabidae, beneficial species, data pre-processing, taxonomic and functional diversity, data mining, predictive models

1. INTRODUCTION
Sustainable agriculture enhances biological pest control in order to reduce the use of pesticides and to foster natural biodiversity and improve the quality of the environment.

In this context, the control of pests provided by the natural enemy populations reduces the dependency on plant protection products. Predatory (beneficial) species such as *Syrphidae* (hoverflies), which control aphids, and *Carabidae* (ground beetles), which feed on slugs, are the main natural enemies of these crop pests [1]. To improve the regulation of pests by predatory species, we need knowledge about the effects of landscape, soil and crop management on these beneficial species. This is a demanding research challenge because of two main reasons. The first reason is the complexity of interdependencies among elements of agricultural ecosystems and their interactions with the environment (e.g., climate, soil humidity). The second reason are the empirical ecological data that are many times incomplete, inconsistent, containing out-of-range values, are collected at different temporal and spatial scales, dispersed in different databases, noisy and imbalanced [2]. To obtain new information and knowledge about biological pest control from empirical data, an extensive data pre-processing is needed [3].

The goals of this study was to assess and predict the abundance of predatory species that could perform biological pest control in agricultural fields. In particular, we are exploring the taxonomic and functional biodiversity of ground beetles of the family *Carabidae*, which prey on slugs, that are the most damaging pests to cereal crops.

In this paper, we first present the comprehensive data pre-processing that was carried out in order to obtain high quality datasets. These were later used for assessment of the abundance and taxonomic and functional diversity of carabid species. In the part that follows, we present the predictive models developed by data mining. The paper ends with conclusions and directions for further work.

2. MATERIALS AND METHODS
2.1 Data
Abundance and biodiversity assessment is based on data from field surveys. For the development of the predictive models, we included also data describing the environmental conditions and the applied agricultural practices. Therefore, we used data from several different datasets and data providers.

Specimen data from field surveys were provided by ARVALIS, Institut du vegetal, France. Field surveys of carabid species were conducted in Boigneville (central France) in years 2009 to 2011 and 2013 to 2015. Carabid species were collected in pitfall traps (Figure 1) using a standardized sampling scheme. There were 21 pitfall traps in the period from 2009 to 2011 and 15 for the period 2013-2015. They were placed in four fields and their locations were permanent throughout the study period. Specimens were taken from pitfall traps on a weekly basis between April and July and September and November. The caught specimens were determined to the species level and the number of caught specimens per species was counted. The total number of pitfall samples included in our research was 2873.

To describe the ecological functional traits of carabid species, we used an additional and extensive database "Functional traits of *Carabidae* species" [4], compiled, maintained and provided by ARVALIS. It includes information on species' size, diet (larvae and adults), humidity preferences, wintering habitat, reproduction period and ability to fly. The database contains traits information for 171 carabid species.

Figure 1: Pitfall trap.

Data about the landscape structure and crop properties were obtained for an area within a 500 m radius around each pitfall trap (Figure 2). In the delineated area, the absolute and relative surface of different crops and natural vegetation types were measured, as well as the length of linear corridors (tree lines, grass strips, grass pathways, hedges, roads). Landscape data were obtained from digital maps using GIS software tools. In addition, crop development stages were estimated for each crop in the studied area and crops were grouped into several categories according to the habitat preferences of carabid species.

Soil data were obtained from the ARVALIS soil database, which contains information on chemical, physical and biological soil properties (e.g., soil texture, available water holding capacity, bulk density, etc.).

Climate data were obtained from the French national meteorological station located in Boigneville. Data about maximum, minimum and average temperature, and cumulative rainfall have been collected at daily bases for the period from 1.1.2009 to 31.12.2019.

Figure 2: An area within a radius of 500 m around a sampling point for which data about landscape, soil and crop properties was obtained.

2.2 Data pre-processing

The collected data were very heterogeneous and as such their harmonization, normalization and aggregation for the purpose of the analysis and modelling was required. To overcome these problems, we used a lot of modelling and ecological knowledge background. In addition, we followed the standard data pre-processing procedure to ensure high quality of the input data, such as data cleaning, outlier detection, missing value treatment, etc. The data describing the abundance of carabid species were highly imbalanced. Therefore, we used the Inverse Hyperbolic Sine transformation of the abundance data that were later used for development of the data mining predictive models.

In addition, several other attributes describing the taxonomic and functional diversity of carabid species were calculated from the available data. The richness of carabid species was described by the Shannon's and Simpson's diversity indices [2]. To measure the distribution of abundances between species, evenness was calculated [3]. Since the interpretability of individual diversity is hard we combined indices of species richness into Hill numbers (N0, N1, N2), which are very suitable for ecological interpretation [3]. In particular Hill numbers (H0-number of all species, H1-numer of abundant species and H2-number of very abundant species) are calculated from the three most important and known measures of diversity: S-number of species, H'-Shannon's index and λ-Simpson's index [2]. A more detailed explanation of diversity indices is given in the conference paper explaining the diversity of syrphid species that provide biological control of aphid pest species [5].

To describe the habitat diversity in a radius of 500 m around the sampling point, we applied Shannon's and Simpson's indices and evenness indices for landscape diversity as well. The former two describe the habitat richness and diversity and the latter emphasizes the evenness of the landscape categories [2]. Despite the fact that the used indices were initially developed for description of species diversity, they can be used for description of landscape diversity as well. In our case, we used the types of landscape instead of species and instead of abundance, we used the land cover area (m^2) of a particular landscape type.

The presence and activities of carabid species depend on their development stage (e.g., larva, pupa, adult) that is primarily driven by daily temperatures that are expressed in degree-days. Degree-days are the most common phenological indicator in entomological research. Degree-days provide information about the cumulative value of average daily temperature for a time period when the average daily temperature is above the selected minimum threshold. We used the most recommended simple logistic equation for calculation of the degree-days with the minimum temperature threshold of 5^0C. Using degree-days, we can compare the abundance and diversity dynamics of carabid species between years and locations. This could not be done by using calendar dates because the climatic and environmental conditions of the selected dates are different in different years.

The pre-processing of environmental, agricultural, taxonomic and functional data resulted in a dataset containing groups of attributes describing the taxonomic and functional structure of carabid species, soil properties, climatic conditions, landscape and crop properties. The total number of obtained/calculated attributes used was 95 (Table 1).

Table 1: Groups and number of attributes in the final dataset.

Group of attributes	*Number of attributes*
Field description	13
Species description	7
Soil description	7
Landscape description	48
Climatic conditions	7
Temporal component	4
Functional aspect of species	9

2.3 Data mining

To discover the interactions between the attributes describing geographical, environmental and management parameters and abundance of the most abundant carabids, we choose data mining methods for induction of decision trees. They are ideally suited for discovery of relations between attributes in complex ecological data, because they are interpretable and can provide meaningful explanations of the relationships and causalities among attributes. In our case, the dependent variables are the abundance of the selected carabid species and the abundance of the predator functional category, which comprises of all predatory carabid species that have been caught in the fields by pitfall traps.

To develop the predictive models, we applied the M5 algorithm to induce regression trees using the WEKA data mining software. To evaluate the induced data mining models, we used 10-fold cross-validation as the most common and standard way of estimating the performance of a model on unseen cases [6].

3. RESULTS

For the purpose of biological pest control, the abundance of predatory and parasitic individuals is as important as diversity. Analysis of the abundance of all caught carabid species in the studied area shows that at yearly basis three species significantly prevail over the others (Table 2) with *Poecilus cupreus* being the most abundant one (Figure 3). Three species appear as the most abundant in all sampled years and in total as well (Table 2).

Table 2: Relative abundance of carabid species for all years (2009-2011 and 2013-2015).

Carabid species	2009-2011 and 2013-2015 (%)
Poecilus cupreus	53
Pterostichus melanarius	13
Anchomenus dorsalis	7
All other carabid species (98 carabid species)	27

Figure 3: *Poecilus cupreus*, the most abundant carabid species in all sampling years (2009-2011, 2013-2015).

The richness metrics of carabid species are presented in Table 3. The values of the Hill number N2 (N2 – number of very abundant species), which varies from 2.8 to 3.5, and the evenness (values from 0.439 to 0.556) are consistent with the results about the rank abundance of carabid species presented in Table 1. A highly uneven distribution of the abundance is indicated when the evenness index has values far from 0 (i.e., equal distribution of the abundances of all species gives values of evenness index close to 0).

Table 3: Richness indices, evenness and Hill numbers for the carabid species caught at the experimental sites in different years.

Year	Richness metrics			Hill numbers		
	Shannon	Simpson	Evenness	N0	N1	N2
2009	1.521	0.335	0.556	58	4.6	3.0
2010	1.542	0.354	0.497	68	4.7	2.8
2011	1.905	0.285	0.439	42	6.7	3.5
2013	1.550	0.327	0.555	54	4.7	3.1
2014	1.729	0.316	0.467	60	5.6	3.2
2015	1.733	0.322	0.452	57	5.7	3.1
All	1.680	0.326	0.473	101	5.4	3.1

The functional characteristics of the three most abundant species (*Poecilus cupreus*, *Pterostichus melanarius*, *Anchomenus dorsalis*) show that they are all predators throughout all of their life stages (larvae and adults). They avoid dry habitat conditions, they are wintering as adults and they reproduce in spring time.

According to the selection of the independent attributes, we constructed two types of models to predict the abundance of carabid species. The first group of models included the attributes describing the current structure of the carabid species in the sampled fields, such as Hill numbers, evenness, Shannon's and Simpson's indexes. This type of prediction models gave us insight into the interspecies interactions and because of that, we named them "ecological models". However, to use these models for predictions in reality is very demanding because they require very specific data describing the carabid community, which are very hard to obtain. To overcome this practical problem with data, we created a second type of predictive models, where the attributes describing the community structure were not included as independent attributes. The data required to populate this type of models can be easily obtained. This makes the application of the models in reality easier and therefore we named them "management models".

Table 4: Validation performances of the regression trees for predicting the abundances of the most dominant carabid species and the two relevant carabid functional groups for biological pest control.

Predictive models	***Correlation coefficient***		***Mean absolute error***	
	Ecolog. model	***Manage. model***	***Ecolog. model***	***Manage. model***
Poecilus cupreus	0.694	0.628	1.42	1.55
Pterostichus melanarius	0.614	0.542	1.20	1.21
Anchomenus dorsalis	0.318	0.287	1.10	1.15
Predator species – larvae	0.253	0.211	1.48	1.64
Predator species – adults	0.374	0.328	1.24	1.42

We obtained predictive models (regression trees) for the three most abundant carabid species (Table 1) and for two functional groups of predator carabid species, where we made a distinction between larval and adult life cycle development stages. The

predictive performances obtained using 10-fold cross validation for all induced predictive models are given in Table 4. The total number of instances was 2873. The abundance is predicted in a weekly time step.

Figure 4: Ecological predictive model for abundance of *Poecilus cupreus.*

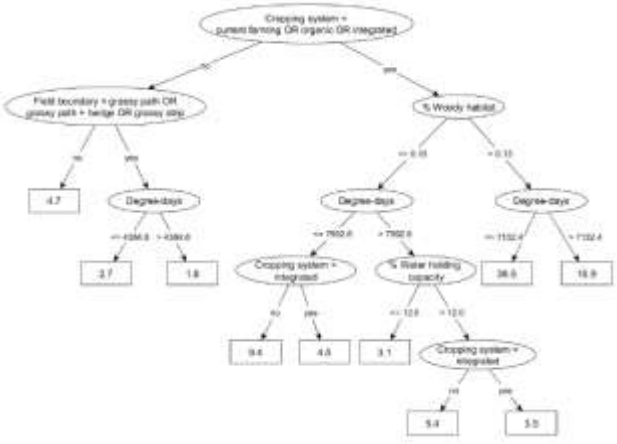


Figure 5: Management predictive model for abundance of *Poecilus cupreus.*

The structure of the ecological model for *Poecilus cupreus* (Figure 4) shows the sensitivity of this carabid species on the presence and abundance of other carabid species and on the habitat type. Its highest weekly abundance appears in conditions of low number of other carabid species and presence of forest habitat. Under such conditions, this species gets very abundant, surpasses other species and it becomes the most abundant one. In the case of absence of attributes describing the carabid community (management model), the abundance of *Poecilus cupreus* depends mostly on the quality of the habitat they are occupying (Figure 5). In particular, the presence of a woody habitat nearby stimulates its abundance. Both models gave consistent and complementary interpretations, which show that *Poecilus cupreus* can be potentially efficient predator of slugs in well preserved agricultural environments and in conditions of low diversity of the carabid community. This is consistent with observations where high abundance of *Poecilus cupreus* in fields is detected in early spring time, particularly if semi-natural habitats, like woodland, are in the vicinity of the fields.

4. CONCLUSIONS

This study has confirmed the complexity of using empirical ecological data. The data pre-processing was the most demanding and time consuming step in the analysis of the taxonomic and functional diversity of carabid species for the construction of predictive models using data mining methodologies.

The results about the taxonomic and functional diversity of carabid species show a great potential for biological pest control in the studied area, as the identified most abundant species of ground beetles are predators of slugs. In addition, the high abundance of the predatory carabids provides an additional quantitative support to the biological control of slugs.

However, the predictive performances of the model are not that promising. Despite the large amount of data and the long list of attributes, we were not able to produce trustable predictive models. The main reason could be having a non-optimal selection of the attributes that ARVALIS used for describing the sampling locations and habitats. The structures of the predictive models indicate that both carabid community and habitat properties influence the abundance of the predicted species. In addition, the attributes describing the temporal variation of environmental conditions appear in the models as well (e.g., degree days, soil humidity). So the models contain all major ecological components that direct the abundance of carabid species.

We can conclude that both the abundance and the diversity (taxonomic and functional) of carabid species in the studied area has high potential to provide efficient pest regulation. Based on the structure of the predictive models, simple guidelines for crop management for enhancement of the biological control of slugs can be proposed (e.g., enlarge woody area, introduce organic or integrated crop production). In addition, we suggest to include additional attributes in the monitoring schema describing habitat conditions that are specific for ground dwelling carabid species. Such additional data would enable us to employ several other data mining methodologies in order to provide significant contribution to the development of efficient biological pest control strategies.

5. ACKNOWLEDGMENTS

This research is supported by the applied project “Structured output prediction with application in sustainable agricultural production” financed by ARRS and co-financed by ARVALIS.

6. REFERENCES

[1] Chaplinin-Kramer, R., Valpine, P., Mills, N.J., Kremen C., 2013.Detecting Pest Control Services across Spatial and Temporal Scales. Agriculture, Eco. and Environment 181 (2013) 206–212.

[2] Legendre P., Legendre L. 2012. Numerical ecology. Elsevier, Amsterdam, Netherlands.

[3] Ludwig, J.A., Reynolds, J.F. 1988. Statistical Ecology. New York, Chichester, Brisbane, Toronto, Singapore, John Wiley & Sons.

[4] ARVALIS 2015. Functional traits of *Carabidae* species. Technical document and internal dataset. Boigneville, France.

[5] Debeljak, M., Kuzmanovski, V., Tosser, V., Trajanov, A. 2017. Knowledge discovery from complex ecological data: exploring Syrphidae species in agricultural landscapes. In: Luštrek,M et al. (Eds) Proceedings of the 20th International Multiconference Information Society, volume A. Ljubljana: Institut Jožef Stefan, pp. 55-58.

[6] Witten, I.H., Frank, E. 2011. Data Mining: Practical Machine Learning Tools and Techniques - 3rd edition. Morgan Kaufmann.

Taxonomies for Knowledge Representation of Sustainable Food Systems in Europe

Aneta Trajanov
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39
1000 Ljubljana, Slovenia
aneta.trajanov@ijs.si

Tanja Dergan
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
tanja.dergan@ijs.si

Marko Debeljak
Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39
1000 Ljubljana, Slovenia
marko.debeljak@ijs.si

ABSTRACT
Sustainability is becoming a core concept in every area (scientific, social, environmental and economic) of human life. Sustainability acknowledges that human civilization takes resources to sustain our modern way of life and strives towards balancing between our competing needs – our need to continue developing technologically and economically, and the need to protect the environment in which we live. However, sustainability is a very complex concept that incorporates social, environmental and economic aspects and interactions between them and can be described by a number of different sustainability indicators. Therefore, assessing the sustainability of a system is a demanding task and requires gathering and structuring of knowledge from experts, literature surveys and other sources. In this paper, we present the use of taxonomies to represent the complex concept of sustainability of European food systems. Structuring the knowledge on sustainable food systems in Europe is a first step in assessing their level of sustainability. The goal of this study is to use the developed taxonomies as basis for the development of a complex DSS system for assessment of the sustainability of legume food systems across the whole quality chain.

Keywords
Taxonomies, knowledge representation, sustainability, food systems.

1. INTRODUCTION
The world's population is increasing at a speeding rate and with that the production and consumption of food as well. All this comes at an enormous environmental cost. Each year, more than 10 million hectares of arable land are lost to degradation, plant-protection products pollute the rivers and aquifers and one third of all greenhouse gas emissions are due to agriculture [4]. Therefore, a shift to more sustainable agri-food systems is needed in order to address these problems. A formal definition of sustainable food systems given by the Food and Agriculture Organization (FAO) states the following: "A sustainable food system is a food system that ensures food security and nutrition for all in such a way that the economic, social and environmental bases to generate food security and nutrition of future generations are not compromised." [2].

Assessment of the sustainability of a food system is not an easy task, as there is not a simple and measurable indicator to assess it. Instead there exists a set of interrelated concepts and indicators that describe the sustainability from different aspects. The sustainability is defined through three main pillars/aspects: economic, social and environmental pillar/aspect (Figure 1) [7].

The ***economic pillar*** of sustainability represents the economic functions of the food systems, which should provide prosperity (wealth) to the (farming) community and thus refers to the economic viability of the food system. The ***social pillar*** represents several social functions, both at the level of the community, as well as at the level of society (e.g., awareness and legislation protection of the health of people from pollution, or access to basic resources without compromising the quality of life). The ***environmental pillar*** represents environmental functions that are connected to the management and conservation of natural resources (water, air, soil, energy and biodiversity) and fluxes within and between these resources [13].

Figure 1. The three pillars of sustainability and their intersections describing partial (bearability, viability, equitability) and overall sustainability.

In order to assess the sustainability of a food system, one needs to understand and take into account all these different aspects of sustainability, which is a demanding task. This paper describes the first step towards modelling the transition towards sustainable food systems, which is done within the H2020 project TRUE

(TRansition paths towards sUstainable legume based systems in Europe) [12]. In order to set the foundations for the development of a Decision Support System (DSS) for sustainability assessment of legume systems, we carried out an extensive literature survey in order to capture as extensive knowledge as possible on sustainable food systems and all the concepts and indicators connected to that. The knowledge and concepts were organized in a hierarchical structure using taxonomies. This kind on knowledge has not been represented in an organized, systematic and formal way so far. Using these taxonomies, we wrote a glossary of terms, which will serve as a knowledge library when constructing the DSS system.

2. MATERIALS AND METHODS

In order to produce a working protocol for harmonization of data and knowledge to develop the future DSS within the TRUE project, we had to derive definitions of sustainability terms and concepts, and review as much sustainability indicators as possible, which are non-deterministic and ambiguous. For that purpose, we reviewed more than 24 papers and 7 books dealing with different aspects of sustainability of (legume) food systems.

The obtained knowledge from the extensive literature survey was structured using taxonomies. Taxonomies, like ontologies, provide ordered/structured representation of concepts and terms in a form of a hierarchy. They are semantic classification schemes and represent a knowledge map [6]. They are *classification schemes*, because they group related things together, so that if you search one thing within a category, it is easy to find other related things in that category. They are *semantic* because they provide a vocabulary to describe the knowledge in them. Finally, if the taxonomy is complete, it should provide an immediate grasp of the overall structure of the knowledge domain it covers.

Many of the taxonomies have hierarchical tree structures. The tree structure is the most intuitive representation, because it provides a visual representation of the relationships between categories and sub-categories, enabling navigation between categories. However, they can be represented in other forms, such as:

- Lists
- Trees
- Hierarchies
- Polyhierarchies
- Matrices
- Facets
- System maps.

The taxonomies presented in this paper are represented in a tree structure and discussed in the Results section.

3. TAXONOMIES FOR SUSTAINABLE FOOD SYSTEMS: RESULTS AND DISCUSSION

The taxonomy describing the knowledge on sustainability of food systems in Europe starts by the general sustainability aspect of European food systems (Figure 2). It incorporates sustainability in its general form, sustainability level, sustainability indicators and sustainability assessment.

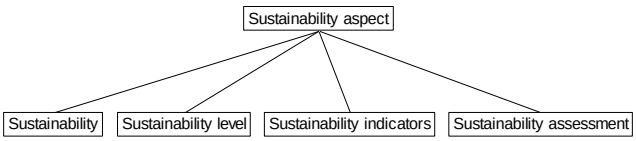


Figure 2. Top level of the taxonomy, decomposing the sustainability aspect into four sublevels: sustainability, sustainability level, sustainability indicators and sustainability assessment.

3.1 Sustainability

The general sustainability as described in the Introduction section, consists of three sustainability pillars: *environmental*, *social* and *economic* [5, 11]. True sustainability requires a balance between the environmental, social and economic aspects describing it. Besides these, the intersections between them (*bearability*, *viability* and *eqitability*) are also an important partial aspect of the sustainability as a whole (Figure 3) [5].

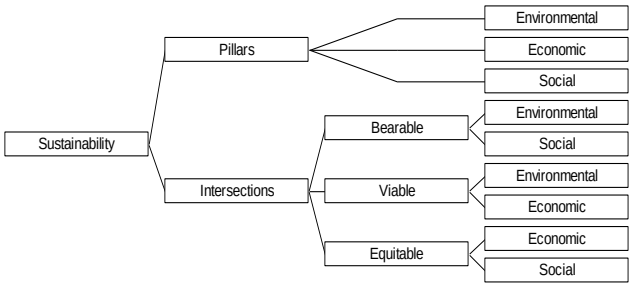


Figure 3. Decomposition of the "Sustainability" part of the taxonomy.

Bearability is the intersection between the environmental and social sustainability pillars. It represents a system that is both environmentally and socially sustainable, but lacks an economic sustainability [11]. *Viability* is the intersection between the environmental and economic sustainability pillars, and e*quitability* is the intersection between the economic and social sustainability pillars.

3.2 Sustainability Level

The next part of the taxonomy represents the concepts connected to the sustainability levels, with respect to the different spatial and temporal scales of sustainability. The sustainability levels can be observed and defined through different aspects (Figure 4):

- Normative
- Spatial
- Temporal
- Systems

Normative level refers to the building blocks (aspects) of sustainability (environment, economic, social), which we described earlier, as well as their interactions (equitability, viability, bearability) [3].

The *spatial level* of sustainability refers to the spatial specifics of sustainability. Sustainability of a food system can be considered/assessed on a local, regional, national or international level [3].

The *temporal* aspects of sustainability refers to the time horizon of the sustainability assessment.

Finally, the boundary of the (food) system under consideration should be defined, the hierarchy of aggregation levels and their interactions for descriptive, assessment and management purposes in relation to sustainable development, which gives us the *system level* of sustainability [3].

3.3 Sustainability Indicators

Indicators are quantified information, which explain how things are changing over time. The sustainability indicators measure the sustainable development and its progress. They have to reflect the definition of sustainability and be able to connect partial conditions to policies for sustainable development and monitoring its progress [9]. Indicators are used to compare the actual state of the system with reference values for sustainability (sustainability assessment), or with the state of the system in the past and in the future (sustainability monitoring) [13]. The part of the taxonomy addressing the Sustainability indicators is given in Figure 5.

Sustainability indicators should satisfy certain *criteria*, which represent specific objectives relating to a state of the system. The criteria should consider the environmental, economic and social characteristics of the system. They must provide specific conditions for the development of sustainability indicators that will have analytical soundness and will be measurable and suitable for application at different scales (e.g., farm, district, country, etc.).

The actual *indicators* are variables of any type that can be induced from the sustainability criteria and can provide information about the potential or realized effects of human activities on the sustainability of the food system. These are variables that can be used to assess both the socio-economic and environmental conditions of the food system, to monitor trends and conditions over time, to provide early warning signal of change and a solid basis for decision making processes, consistent with sustainable development principles at all levels [1, 10]. The indicators can be also used to reduce the complexity of the system description and integrate information about processes, trends or states into a more readily understandable form at local, regional and global levels.

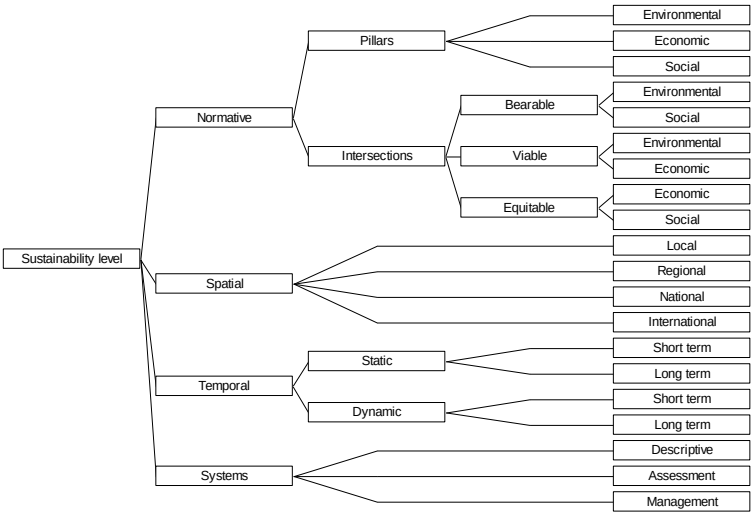


Figure 4. Decomposition of the "Sustainability level" part of the taxonomy.

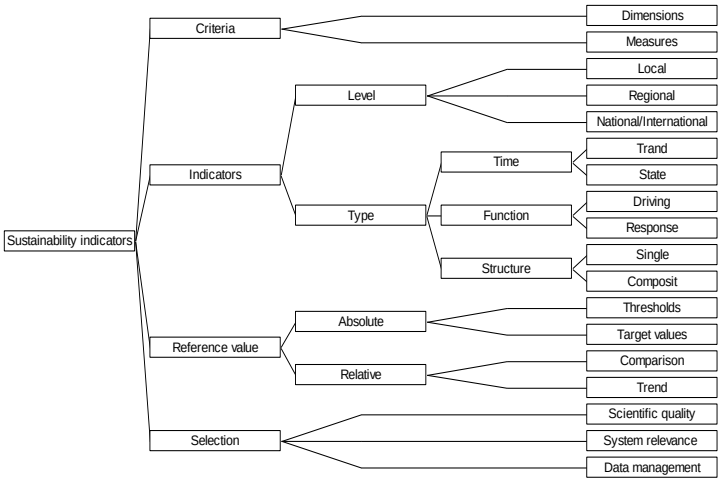


Figure 5. Decomposition of the "Sustainability indicators" part of the taxonomy.

The indicators can have different *levels* [3, 8, 10]:

- *Local* – measure the state of the system regarding sustainability
- *Regional* – compare the system's performance from an economic, social and environmental aspect
- *National/international level* – inform policy makers about the current state and trends in sector performance and facilitate public participation in sustainability discussions.

The *type* of indicators refers to their functional category according to their purpose of use. According to the type, the indicators can describe [1]:

- *Time-related issues*:
    - Trend indicators – describing temporal dynamic aspects of sustainability over time
    - State indicators – describing the condition of the assessed system at a certain time point
- *Function-related issues*:
    - Driving indicators – measure the changes of the assessed food system due to management practices and other farming inputs
    - Response indicators – measure the response of a system to the induced management changes or inputs
- *Structure-related issues*:
    - Single indicators – characterizing single parts of the food system
    - Composite indicators – aggregate environmental, social and economic indicators into a unique measure describing complex functions and processes of the system.

The *reference values* of the sustainability indicators refer to the desired level of sustainability for each indicator. They are established on scientific or empirical basis and can be presented as absolute (fixed) values, as threshold values, or as relative reference values according to a selected baseline [13].

Finally, the *selection* of indicators should be made according to their scientific quality, system relevance and data management principles.

4. CONCLUSIONS

Structuring and organizing domain knowledge in a structured and formalized way, using taxonomies, is a crucial first step towards understanding complex problems and concepts. It also represents a crucial pre-processing step, which sets the basis for data mining analyses or development of Decision Support Systems.

Within the TRUE project, the taxonomies for knowledge representation of sustainable food systems in Europe were "translated" into a glossary of terms, which will be evaluated by a group of domain experts. In that way, they will validate and confirm the structure of the taxonomy.

The obtained knowledge on sustainability of European food systems, represented using taxonomies will represent the basis for the development of a complex Decision Support System for finding transition pathways towards sustainable legume-based food systems.

5. ACKNOWLEDGMENTS

This research is supported by the TRUE project, funded by the EU Horizon2020 Research and Innovation Programme, Grant Agreement number 727973.

6. REFERENCES

[1] Dale, V.H., Efroymson, R.A., Kline, K.L., Langholtz, M.H., Leiby, P.N., Oladosu, G.A., Davis, M.R., Downing, M.E., Hilliard, M.R. 2013. Indicators for Assessing Socioeconomic Sustainability of Bioenergy Systems: A Short List of Practical Measures. Ecological Indicators, 26: 87–102.

[2] FAO, 1984. Potential population supporting capacities of lands in the developing world. FAO, (Food and Agriculture Organization of the United Nations), Rome.

[3] Hayati, D., Ranjbar, Z., Karami, E. 2010. Measuring Agricultural Sustainability. In: Biodiversity, Biofuels, Agroforestry and Conservation Agriculture, Eds. Eric Lichtfouse, 5:73–100. Dordrecht: Springer Netherlands.

[4] HLPE, 2014. Food losses and waste in the context of sustainable food systems. A report by the High Level Panel of Experts on Food Security and Nutrition of the Committee on World Food Security, Rome 2014.

[5] Kwami, H.I, Che-Ani, A.I., Tawil, N.M., Tahir, M.M., Basri, H. 2014. Approach to Campus Sustainability at Universiti Kebangsaan Malaysia (UKM): A Review. Eds M.A. Othuman Mydin and A.I. Che Ani. E3S Web of Conferences 3: 01011.

[6] Lambe, P. 2007. Organising knowledge: Taxonomies, Knowledge and Organisational Effectiveness. Chandos Publishing, Oxford, England.

[7] Moldan, B., Janouskova, S., Hak, T., 2012. How to understand and measure environmental sustainability: indicators and targets. Ecological Indicators, 17: 4–13.

[8] Muñoz, Lucio. 2016. Linking Sustainable Development Indicators by Means of Present/Absent Sustainability Theory and Indices: The Case of Agenda 21, GDS, IIG, Spain.

[9] Pollesch, N., Dale, V.H. 2015. Applications of Aggregation Theory to Sustainability Assessment. Ecological Economics, 114: 117–27.

[10] Samuel, V.B., Agamuthu, P., Hashim, M.A. 2013. Indicators for Assessment of Sustainable Production: A Case Study of the Petrochemical Industry in Malaysia. Ecological Indicators, 24: 392–402.

[11] Slocum, S.L. 2015. The Viable, Equitable and Bearable in Tanzania. Tourism Management Perspectives, 16: 92–99.

[12] TRUE (TRansition paths towards sUstainable legume based systems in Europe), 2018. H2020 project, https://www.true-project.eu (11 September 2018).

[13] Van Cauwenbergh, N., Biala, K., Bielders, C., Brouckaert, V., Franchois, L., Cidad, V.G., Hermy, M., Mathijs, E., Muys, B., Reijnders, J., Sauvenier, X., Valckx, J., Vanclooster, M., der Veken, B.V., Wauters, E., Peeters, A. 2007. SAFE – a hierarchical framework for assessing the sustainability of agricultural systems. Agric Ecosyst Environ, 120: 229–2

Uporaba povezave kalkulacijskega simulacijskega modela z analizo tveganja pri podpori odločanja v kmetijstvu

Tanja Dergan
Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
Slovenija
tanja.dergan@ijs.si

Aneta Trajanov
Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
Slovenija
aneta.trajanov@ijs.si

Marko Debeljak
Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
Slovenija
marko.debeljak@ijs.si

POVZETEK

Kmetijska gospodarstva se pri načrtovanju pridelave neprestano odločajo o načinu proizvodnje, izbiri poljščin in količini pridelave. Pri tem se soočajo z vprašanjem kako ob čim manjšem ekonomskem vložku ter obvladovanju pridelovalnih in okoljskih tveganj zagotoviti optimalen ekonomski rezultat. Raziskava obravnava sistem za podporo odločanja, ki temelji na integraciji simulacijskega modela ekonomske kalkulacije in sistema analize ekonomskega tveganja na primeru pridelave ječmena, fižola in koruze v premenah hmelja. Rezultati simulacijskih modelov in analiz tveganja so pokazali, da je v premenah hmelja najboljše ekonomske kriterije dosegla pridelava fižola. Zaključki naše raziskave govorijo v prid povečanju pridelave stročnic, ki predstavljajo trajnostno možnost za povečanje samooskrbe z rastlinskimi beljakovinami in zmanjšanju obremenitve okolja z negativnimi vplivi kmetijske pridelave.

Ključne besede

Sistem za podporo odločanja, kmetijska pridelava, ekonomske kalkulacije, ekonomsko tveganje, hmeljarstvo, trajnostno kmetijstvo

1. UVOD

V kmetijstvu ekonomsko upravičenost skupnih stroškov pridelave ugotavljamo s pomočjo ekonomskih kalkulacij, ki so eno od osnovnih orodji za planiranje in podporo odločanja v kmetijskem menedžmentu. Na podlagi kalkulacij so ocenjeni skupni stroški pridelave in izračunani indikatorji ekonomske uspešnosti, kar predstavlja osnovo za nadaljnje načrtovanje kmetijske pridelave [7].

Tveganje je pomemben poslovni vidik v kmetijstvu. Visoka proizvodnja in cenovna tveganja so postala stalnica kmetijske proizvodnje in so v večji meri odraz nepredvidljivega obnašanja naravnih dejavnikov (vreme, škodljivci, bolezni, itd.). Uspešnost obvladovanja teh tveganj vpliva na ekonomsko uspešnost kmetijskih gospodarstev, saj napačne odločitve hitro vodijo v zmanjšanje dohodka. Kmetijski pridelovalci so pri tem soočeni s kompleksnostjo odločevalskega problema saj so primorani sprejemati številne odločitve tako na vsakdanjem nivoju, kot tudi na nivoju dolgoročnejših investicij [6].

Doseganje boljšega in predvsem stabilnejšega rezultata kmetijske pridelave je zato zelo odvisno od obvladovanja tveganja v procesu odločanja. V načrtovanju kmetijske proizvodnje zato spremljanje in ocenjevanje tveganja predstavlja zelo pomemben segment v procesu spremljanja in načrtovanje kmetijske pridelave [4].

Zaradi velike odvisnosti od uvoza hrane in krme se v Sloveniji premalo zavedamo kako pomembno je varovanje in ohranjanje kmetijskih pridelovalnih zemljišč za samooskrbo pridelave hrane [2]. Učinkoviti sistemi ekonomskega upravljanja in obvladovanja tveganja bodo omogočili učinkovitejše upravljanje s tem naravnim virom ob hkratnem zagotavljanju kvalitetnega življenjskega standarda prebivalstva, ki mu kmetijstvo predstavlja osnovno ekonomsko dejavnost. Od tega je odvisna tudi sposobnost ohranjanja kmetij in agrarne krajine.

Kmetijska gospodarstva se morajo neprestano prilagajati spremembam lokalnih in globalnih družbeno ekonomskih dejavnikov. Okoljska in gospodarska razvojna politika od pridelovalcev zahteva povečevaje pridelave in izboljševanje njene kakovosti ob hkratnem upoštevanju vse bolj zahtevnih kriterijev trajnostnega kmetijstva.

Pridelava hmelja predstavlja eno od zelo potencialnih kmetijskih dejavnosti glede možnosti izpolnjevanja kriterijev in zahtev sodobne družbe glede trajnostne pridelave hrane [5]. Hmelj je trajnica in kot večletna monokultura negativno vpliva na kvaliteto tal. Povečuje zbitost tal in zmanjšuje količine aktivnega humusa v tleh [9]. Da bi odpravili tovrstne negative vplive na tla in da bi nasade zavarovali pred povzročitelji bolezni in škodljivci, se na površinah za pridelavo hmelja, hmelj za krajše obdobje nadomesti z drugimi poljščinami, kar imenujemo premena [3].

Osrednji cilj raziskave je izgradnja sistema za podporo odločanja, ki temelji na integraciji simulacijskega modela ekonomske kalkulacije in sistema analize ekonomskega tveganja na primeru pridelave ječmena, fižola in koruze v premenah hmelja. S pomočjo izbranih kriterijev tveganja ocenimo posamezne alternative, ter izberemo tisto, ki je glede na specifiko konkretnega primera premene hmelja najustreznejša. Izračuni, ocenjeni na osnovi uporabljene metodologije tveganja in predpostavljenih ekonomskih parametrov pridelave na modelni kmetiji, predstavljajo pomembno podporno orodje za nosilčeve nadaljnje odločitve.

Preostali del prispevka je strukturiran kot sledi. Opis podatkov in zasnova so predstavljeni v poglavju 2. Sledi opis obravnavanih metod v poglavju 3. Rezultate razprave predstavimo v poglavju 4 in zaključimo v poglavju 5.

2. PODATKI

Raziskava analizira podatke pridobljene iz referenčne kmetije izbrane v Žalcu v spodnji Savinjski dolini. Kmetija je poljedelsko-živinorejska, integrirano usmerjena, s 25 ha pridelovalne površine hmelja. V premenah hmelja trenutno pridelujejo izmenično koruzo, fižol in ječmen (Slika 1). V prihodnosti želijo kmetijo preusmeriti zgolj v poljedelsko dejavnost. Podatke, ki opisujejo trenutno stanje na kmetiji smo pridobili z osebnim intervjujem lastnika kmetije in sodijo v obračunsko leto 2017.

Slika 1: Premena hmelja z visokim fižolom

3. METODE

Za reševanje problemov ekonomske upravičenosti pridelave in tveganja smo uporabili izviren pristop integracije dveh sicer pogosto posamično uporabljenih metod, ki jih uporabljamo v načrtovanju pridelave na nivoju kmetije. Z njuno uporabo lahko z različnih zornih kotov ovrednotimo razvoj kmetijskih gospodarstev in pokažemo možnost povečanja dodane vrednosti v kmetijstvu.

3.1 Kalkulacijski modeli

Ekonomske kalkulacije v kmetijstvu zahtevajo uporabo kvalitetnih vhodnih podatkov [10], ki smo jih v našem konkretnem primeru zagotovili.

Kalkulacije so samostojni simulacijski modeli, ki na podlagi opredeljenih (izbranih) vhodnih atributov omogočajo oceno porabe vnosov v kmetijsko proizvodnjo (npr. semena, gnojila, krmila, škropiva, gorivo, najem strojev…) in s tem oceno skupnih stroškov pridelave kmetijskih pridelkov oz. proizvodov [7]. Poraba vnosov je odvisna od intenzivnosti pridelave, pridelovalne površine, oddaljenosti od kmetije, nagiba terena itd. Za razliko od t.i. kalkulacij pokritja, modelne kalkulacije pri posameznem pridelku neposredno vključujejo vse skupne stroške pridelave, ki so povezani s proizvodnjo in ne samo spremenljive stroške, kar omogoča tudi neposredno primerjavo skupnih stroškov pridelave s skupnim prihodkom ter izračunom različnih ekonomskih kazalcev. Za potrebo naše raziskave smo uporabili v nadaljevanju na kratko opisane kazalce:

$$SS=FS+ VS$$

Skupni stroški pridelave (SS) predstavljajo seštevek fiksnih stroškov (FS) (zavarovanja, obresti kreditov, plače delavcev) in variabilnih stroškov (VS) (stroški ki so odvisni od obsega proizvodnje, npr. stroški amortizacije…) [11].

$$FR=VP-SS$$

Finančni rezultat (FR) predstavlja razliko med skupnim pridelkom proizvodnje (VP) in skupnimi stroški pridelave (SS) [11].

$$VP = Y \times C_y \times PR + Y_1 \times C_{y1}$$

Vrednost pridelave (VP) predstavlja zmnožek količine pridelka (Y) in njegove cene (C_y), pomnoženega s površino pridelka (PR), k temu pa prištejemo še količino stranskega pridelka (Y1), (npr: pri koruzi je stranski pridelek slama), pomnoženega s ceno stranskega pridelka (C_{y1}) [11].

$$LC = SS/Y$$

Lastna cena (LC) predstavlja višino skupnih stroškov pridelave za proizvodnjo enote izbranega pridelka (proizvodnja enota pridelka je definirana kot vrednost pridelave za 1 kg pridelka). Izračunan je kot koeficient med skupnimi stroški pridelave (SS) in količino pridelka (Y) in je ekvivalent prelomni ceni proizvoda [11].

$$KE = SP/SS$$

Koeficient ekonomičnosti (KE) predstavlja razmerje med skupnimi prihodki (SP) in skupnimi stroški pridelave (SS). Z njim ugotavljamo gospodarnost poslovanja. Če je koeficient ekonomičnosti večji od ena pomeni, da je poslovanje ekonomično in obratno [11].

Glavni namen kalkulacij je spremljanje skupnih stroškov pridelave. S tem pridobljene informacije predstavljajo kmetu osnovno informacijsko podporo za odločanje v načrtovanju proizvodnje, načrtovanju nadaljnjih investicij in ostalih aktivnosti na kmetiji.

3.2 Analiza tveganja v kmetijstvu

Deregulacija trgov, spremembe glede potreb po hrani in druge družbene zahteve (npr. trajnost), ter vplivi podnebnih klimatskih sprememb vodijo v vedno večja cenovna, pridelovalna in dohodkovna nihanja in posledično s tem tudi v povečevanje tveganja v kmetijstvu [1]. Tveganje na kmetijskih gospodarstvih ne smemo zanemariti, saj poskušajo nosilci odločanja v kmetijstvu tveganja obvladati in predvsem tudi zmanjševati [6].

Med kmetijskimi gospodarstvi obstajajo pomembe razlike v izpostavljenosti, zaznavanju in upravljanju tveganja. Učinkovito obvladovanje tveganja je eden izmed ključnih dejavnikov uspešnega poslovanja. Nosilci kmetijskih gospodarstev lahko pri upravljanju s tveganjem izbirajo med različnimi ukrepi in orodji. Kaj bo posameznik izbral, pa je odvisno od njegove naklonjenosti tveganju in okoliščinam v katerih kmetuje [12] .

Za analizo tveganja je potrebno zapisati tako imenovano tabelo odločanja (Tabela 1), ki se uporablja pri vseh kriterijih tveganja. V tabeli prikažemo vse alternative (A) kot vrstice (kjer i= 1, 2,…m) in stanje (S) kot stolpce (kjer j=1, 2,…n). R_{ij} nam ponazarja ekonomski donos za izbrano alternativo A_i, če pride do stanja S_j in p nam ponazarja porazdelitev verjetnosti, ki velja za S (niz verjetnosti p_j, ki opisuje verjetnost, da se bo stanje S_j zgodilo).

Tabela 1: Tabela odločanja

A	**S**			
	S_1	S_2	…	S_n
	p_1	p_2	…	p_n
A_1	R_{11}	R_{12}	…	R_{1n}
A_2	R_{21}	R_{22}	…	R_{2n}
…	…	…		…
A_m	R_{m1}	R_{m2}	…	R_{mn}

V naši raziskavi smo kot modelno poljščino obravnavali pridelavo hmelja, kjer se v premenah na isti pridelovalni površini pridelujejo tudi druge poljščine. Pridelavo poljščin smo analizirali s pomočjo petih kriterijev tveganja, ki so v naslednjih podpoglavjih na kratko opisani. Pri tem je potrebno poudariti da uporaba različnih kriterijev, lahko pripelje do izbere različnih alternativ.

3.2.1 Waldov kriterij (MaxMin)

Waldov kriterij ali *MaxMin* kriterij je kriterij pesimizma, kjer odločevalec upošteva le najmanjše vrednosti alternativ in izbere alternativo ki ima izmed najnižjih najvišje rezultate. *MaxMin* kriterij vpliva na odločevalčevo zavest, kateri si prizadeva zagotoviti, da v primeru negativnega izida, obstaja vsaj minimalno izplačilo[8].

3.2.2 MaxMax kriterij

Kriterij imenujemo tudi optimističen kriterij in je ravno nasprotje MaxMin metode. Je optimističen in agresiven pristop k odločitvi pod pogojem negotovosti. Z MaxMax kriterijem bo odločevalec vedno računal na najboljši izid pri vsaki alternativi. MaxMax pravilo je primerno za ekstremne optimiste, kateri pričakujejo najbolj udoben položaj [8].

3.2.3 Hurwizcev kriterij (H)

Hurwizc pristop poskuša vzpostaviti ravnovesje (sredino) med *MaxMax* in *MaxMin* kriteriji. Namesto ob predpostavki popolnega optimizma ali pesimizma, Hurwicz uporabi parameter (α), ki je na intervalu med 0 in 1 in jo odločevalec izbere subjektivno [8].

Če je vrednost α bližje 1, pomeni absolutni optimizem in velja *MaxMax kriterij* (maksimalna vrednost maksimalne vrednosti prihodka na letni ravni)

Če je vrednost α bližje 0, pomeni daje absolutni pesimizem in velja *Waldov MaxMin kriterij* (maksimalna vrednost minimalne vrednosti prihodka na letni ravni)

Vrednost α nam hkrati poda koeficient pesimizma 1-α, ki odraža odločitev odločevalca na tveganje. Hurwiczovo tehtno povprečje se sedaj lahko izračuna za vsako alternativo A_i [8].

$H(A_i) = \alpha$ (vrstica alternative z maximum vrednostjo) + (1-α)(vrstica alternative z minimum vrednostjo) je pozitiva (dobiček, prihodek)

$H(A_i) = \alpha$ (vrstica alternative z minimum vrednostjo) + (1-α)(vrstica alternative z maximum vrednostjo) je negativna (stroški, prihodek)

3.2.4 Savageov kriterij (MinMax)

Z drugim imenom poznan tudi kot *MinMax* kriterij obžalovanja. Je pesimistični pristop, ki proučuje obžalovanje, izgubo [8]. Ta kriterij se povsem osredotoča na izogibanje najhujših možnih posledic, ki lahko nastanejo pri odločanju [8].

Izguba priložnosti (OL) se definira kot razlika izplačil med najboljšim možnim izidom pod pogojem S_j in dejanskim rezultatom iz izbire A_i, če se pojavi S_j. To pomeni, da če izbrana alternativa poda najvišjo možno vrednost, potem izguba ni prisotna in je definirana kot vrednost nič.

Savageov kriterij je definiran kot:

OL_{ij} = (stolpec stanja j maximum izplačil) - R_{ij}, je pozitivna (upoštevamo dobiček, prihodek)

$OL_{ij} = R_{ij}$ - (stolpec stanja j minimum izplačil) je negativna (upoštevamo vrednost stroškov)

Rij ponazarja vsoto vrstic i in stolpcev j v tabeli odločanja (poglavje 3.2). Najboljši možen rezultat je 0 – kar pomeni, da ni obžalovanja. Višja kot je vrednost rezultata, večje je obžalovanje za odločitev.

3.2.5 Laplaceov kriterij

Ta kriterij je realističen, ter izhaja iz principa maksimalnega neznanja. Po Laplaceovem kriteriju predpostavljamo, da so vsi rezultati enako verjetni, kjer vrednosti med seboj seštejemo ter določimo alternative z najvišjo dano vrednostjo. Odločevalec lahko izračuna iz vsake vrstice tabele odločanja (Tabela 1) povprečno vsoto, kot rezultat pa izberemo najvišjo povprečno vrednost vrstice [8]. Pravilo Laplaceove odločitve:

1. Dodajte $p_j = P(S_j) = 1 / n$ vsaki S_j v S, za j = 1, 2, ..., n.
2. Za vsako A_i (vrstica matrike izplačil) izračuna pričakovano vrednost (E): $E(A_i) = \Sigma_j p_j (R_{ij}) = p_j \Sigma_j R_{ij}$.
3. Kot optimalno odločitev izberemo najboljšo vrednost alternative $E(A_i)$, ki najboljše predstavlja dobiček in minimalno predstavlja stroške.

4. REZULTATI IN RAZPRAVA

Raziskava je zasnovana v dveh sklopih. V prvem sklopu smo razvili modelno matematično orodje za analizo ekonomske upravičenosti posamezne pridelave. Pridobljene rezultate smo nato v drugem sklopu prenesli in uporabili pri analizi tveganja v kmetijstvu. Z integracijo kalkulacije in tveganja smo odločevalcu omogočili dodatno natančnejšo vrednotenje alternativ ter s tem zanesljivejše odločanje.

4.1 Rezultati kalkulacij

S simulacijskim kalkulacijskim modelom smo za vsako od treh obravnavanih poljščin (fižol, ječmen in koruzo) ocenili njihove glavne ekonomske kazalce, predstavljene v poglavju 3.1 (Skupne stroške pridelave (SS), skupne prihodke (VP), lastno ceno (LC), koeficient ekonomičnosti (KE) in končni finančni rezultat (FR)), ki temeljijo na podatkih iz obravnavane kmetije. Izračuni so pokazali, da so največji skupni stroški pridelave nastali pri pridelavi fižola, medtem ko so najnižji skupni stroški pridelave bili pri ječmenu (Tabela 2). Analiza ekonomičnosti pridelave je pokazala, da je najvišjo vrednost pri glavnih ekonomskih kazalnikih dosegala pridelava fižola, najslabše rezultate pa je kljub boljši lastni ceni od koruze prejela pridelava ječmena (Tabela 2).

Tabela 2: Rezultati analize ekonomičnosti

	Skupni stroški pridelave (eur/ha)	**Finančni rezultat (eur/ha)**	**Lastna cena (eur/kg)**	**Vrednost proizvodnje (eur/ha)**	**KE**
Fižol	**4378**	**3032**	**0,89**	**7411**	**1,7**
Koruza	**2026**	**1303**	**0,10**	**3330**	**1,6**
Ječmen	**1505**	**457**	**0,28**	**1962**	**1,3**

Koeficient ekonomičnosti (KE) je vseh treh primerih večji od ena (Tabela 2), kar pomeni da je prodajna vrednost večja od skupnih stroškov pridelave (poglavje 3.1). Pridelava fižola je tako kljub najvišjim skupnim stroškom pridelave, iz ekonomskega vidika najustreznejša poljščina za pridelavo v premeni hmelja (Slika 2).

Slika 2: Finančni rezultat kalkulacij

4.2 Rezultati tveganja

V okviru ekonomske analize smo kot vhodne podatke za nadaljnjo analizo kriterijev tveganja uporabili finančne rezultate pridobljenih v prvem delu raziskave (poglavje 4.1).

Tabela 3 prikazuje tabelo odločanja (poglavje 3.2), ki prikazuje vrednosti prihodka na letni ravni in sicer glede na različne scenarije potencialne prodaje pridelka (p), (100%, 80% in 50% možnostjo prodaje).

Tabela 3: Tabela odločanja vrednosti prihodka različnih poljščin (alternativ) na letni ravni glede na različne scenarije prodaje

	Možni scenariji		
Možni scenariji	**Scenarij 1 (p_1 = 1.00): 100% prodaja (eur/ha)**	**Scenarij 2 (p_2 = 0.80): 80% prodaja (eur/ha)**	**Scenarij 3 (p_3 = 0.50): 50 % prodaja (eur/ha)**
A1 - fižol	3032	2426	1516
A2 - koruza	1304	1043	652
A3 - ječmen	457	366	228

Če primerjamo rezultate analize tveganja pridelave obravnavanih poljščin (Tabele 4) vidimo, da Waldov kriterij predlaga alternativo A1 - fižol z dobičkom 1516 EUR/ha. Najvišji donos za pridelovalca kažejo izračunane vrednosti za MaxMax kriterija, ki predlaga alternativo A1 - fižol, v vrednosti 3032 EUR/ha. Prav tako prikaže najboljši donos alternative A1 - fižol Hurwicz in Laplacov kriterij. Poudariti je potrebno da smo pri Hurwicz kriteriju uporabili koeficientom optimizma, ki je v našem primeru znašal (k=0,7), ter koeficientom pesimizma, ki smo ga izračunali kot: 1- koeficient optimizma (k=0,3). Po Savage kriteriju pa se odločimo za alternativo A2 - pridelava koruze, s čim bi dosegli 1728 EUR/ha.

Tabela 4: Skupni rezultati analiz tveganj

Kriterij tveganja	**Odločitev**
Waldov kriterij pesimizma - MAXMIN	A1 – fižol
Kriterij optimizma - MAXMAX	A1 – fižol
Hurwiczov kriterij za koef.optimizma = 0,7	A1 – fižol
Savageov kriterij - MINMAX	A2 - koruza
Laplaceov kriterij	A1 - fižol

Rezultati so pokazali, da je z ekonomskega vidika kot najbolj primerna odločitev izbira alternative A1 - fižol. Slednje pomeni, da je ekonomsko najučinkovitejše, da obravnavana kmetija v premeni hmelja prideluje fižol.

5. ZAKLJUČEK

Cilj raziskave je bil izgradnja sistema za podporo odločanja, ki temelji na kalkulacijskem modelu za oceno kmetijske pridelave ječmena, fižola in koruze v premenah hmelja, ter s pomočjo pomembnejših kriterijev tveganja opredeli katera izmed izbranih alternativ je za izbrano kmetijo ekonomsko najugodnejša. Predstavljeno metodo lahko zlahka prenesemo v proces izbire tudi drugih poljščin in v druge pridelovalne procese. Želeli smo oblikovati trajnostne smernice za razvoj kmetije in z izračuni ocenjenimi na osnovi uporabljene metodologije razviti podporno orodje za pridelovalčeve nadalje odločitve, ki bodo poleg ekonomskih kazalnikov upošteval tudi tveganje, ki je v kmetijstvu močno prisotno. Zavedati se moramo da je znižanje tveganja možno le do določene mere in da se bo tveganje v prihodnje, verjetno samo še povečalo. Ključno merilo tveganja je variabilnost obsega proizvodnje, ki je tesno povezana z tipom proizvodnje. Tako je npr., variabilnost rastlinske pridelave bolj povezana z vremenskimi vplivi kot živinorejska. Ob tem velja izpostaviti, da tveganje samo po sebi še ne pomeni škode. Ta nastane takrat, ko se predvideno tveganje tudi zgodi, kar povzroči negativne ekonomske posledice.

6. ZAHVALA

Raziskavo podpira TRUE projekt, ki ga financira program EU za raziskovanje in inovacije Horizon2020, sporazum o dodelitvi sredstev št. 727973.

7. LITERATURA

[1] Almin H. 2010. Uncertainty Aversion and Risk Management in Agriculture. Agriculture and agricultural Science Procedia 1. 152-156 str.

[2] Bavec F. 2001. Pridelovanje nekaterih poljščin v ekološkem kolobarju. Ekološko kmetovanje. 158 – 231 str.

[3] Džuban T. s sod. 2008. Tehnološka navodila za intergrirano pridelavo poljščin. Ministerstvo za kmetijstvo, gozdarstvo in prehrano, Ljubljana. Str: 50.

[4] Girdžiute L. 2012. Risk ain agriculture and opportunities of their integrated evaluation. Proc. Soc.Beh.Sci: str. 783-790.

[5] Hacin J. 1982. Podorine v hmeljarstvu. Hmeljar.4. Str: 3-4.

[6] Hardaker J.B., Huirne R.B. M., Anderson J.R., Lien G. 2007. Coping with Risk in Agriculture (2nd ed.). Oxfordshire: CABI Publishing

[7] Pažek K., 2003, Finančna analiza ocenjevanja investicij dopolnilnih dejavnosti na ekološki kmetiji: Mag. Maribor

[8] Pažek, K., Rozman Č. 2008. Decision Making under Conditions of Uncertainty in Agriculture: Case study of oil crops. Original scientific paper

[9] Plazovnik R. M. 2008. Krmne poljščine v slovenskih hmeljiščih v premeni. Diplomsko delo. Univerza v Ljubljani

[10] Radnak M. 1998. Splošna izhodišča in metodologija izdelave modelnih kalkulacij za potrebe kmetijske politike. Prikazi in informacije 189. Ljubljana: Kmetijski inštitut Slovenije

[11] Vinčec J. 2010. Analiza poslovanja na zelenjadarski kmetiji. Diplomsko delo. UM. Dislocirana enota Rakičan

[12] Žgajnar J. 2013. Možnosti upravljanja s tveganji v kmetijstvu.6. konferenca DAES. Orodje za podporo odločanju v kmetijstvu in razvoju podeželja. Krško.Str.15-32

Hierarchical Multi-label Classification for Activity Recognition

Nina Reščič
Jožef Stefan Institute
International Postgraduate School Jožef Stefan
Jamova cesta 39
1000 Ljubljana
nina.rescic@ijs.si

Mitja Luštrek
Jožef Stefan Institute
International Postgraduate School Jožef Stefan
Jamova cesta 39
1000 Ljubljana
mitja.lustrek@ijs.si

ABSTRACT
Activity recognition using wearable sensors is very important in many domains of health monitoring and is therefore well researched. Most commonly classification considers all activities to be 'equal' (we will use term flat classification). However, intuition suggest better results could be achieved using a hierarchical approach for classification. In this paper we compare three different approaches to classify activities: (i) *Flat classification* - classes are equal and we build one model to classify all of them; (ii) *Multi-model hierarchical classification* - classes are arranged in trees, we build different models to classify activities on different levels. We apply two different approaches; (iii) *Hierarchical classification using CLUS* software[1].

Keywords
Activity recognition, hierarchical multi-label classification, wearable sensors

1. INTRODUCTION
Activity recognition (AR) using wearable sensors has been addressed many times, some of the most important application being personalized health systems. Many of developed methods for recognizing different activities used triaxial accelerometers worn on different body parts. With development of wrist-worn devices in past several years and with their growing popularity in everyday life, methods for recognizing sports activities [2], daily activities [3] and hand-specific activities [4] using just wrist-worn sensors were proposed. Although the performance gets better with adding additional body sensors, as Attal and al. [1] proved in 2015 by reviewing the research done by then, we decided to focus our research on wrist-worn sensors due to before mentioned accessibility and popularity.

Vens and al. [6] defined hierarchical multi-label classification (HMC) as a variant of classification, that differs from normal classification in two ways: (1) a single example may belong to multiple classes simultaneously; and (2) the classes are organized in a hierarchy: an example that belongs to some class automatically belongs to all its superclasses, the so-called *hierarchy constraint*.

Although hierarchical approach might seem quite intuitive for AR, as certain activities are pretty obvious grouped together, the usage of hierarchical classification for AR has only been addressed a few times. None of the cases was specifically directed towards usage of wrist-worn device for recognizing different hierarchical activities (physical, daily, hand-movement activities). Khan and al. [8] proposed a hierarchical recognizer for recognition of limited amount of physical activities (static, transitions, dynamic) using a chest-worn sensor device. Zheng [9] explored human activity based on the hierarchical feature selection and classification framework. He explored 2D and 3D motion (jumping, running, walking forward/left/right, upstairs/downstairs, static activities).

2. DATASET
The dataset we are working with consists of data from seven people involved in different activities (sport, rest, handwork, eating chores...). We organize the activities in hierarchy as presented in Table 1. First we tried to create structure tree by using Orange[2] software for hierarchical clustering. We calculated features as will be explained later in paper and put them into Orange software. We were looking for some indications of the hierarchy for different groups of activity. However, there was no clear or extremely obvious structure visible. The final structure was designed using knowledge achieved from previous research on the same dataset where flat classification (for instance in research made by Cvetkovic et al. [4]) has been used for recognition of activities.

Table 1: Activity grouping

Group	Activity
Daily activities	chores eating handwork washing
Exercise	nordic running walking
Static	lying sitting standing

3. METHODS
In this paper we are comparing three different approaches for activity recognition. First we addressed flat classifica-

[1]https://dtai.cs.kuleuven.be/clus/index.html

[2]https://orange.biolab.si/

tion, which is commonly used in previous research. Next, we implemented two multi-model hierarchical algorithms, based on approach proposed by Paes et al.[11]. We use the term multi-model as different models were used for different levels of hierarchy. Finally we used Clus software, which has algorithms for hierarchical multi-classification (HMC) already implemented and is mostly used in the field of functional genomics and text classification as shown by Vens et al.[6].

The users were wearing a wearable device (wristband or smartwatch) on their non-dominant hand. For the purpose of this paper we only considered triaxial accelerometer data, however for further research other measurements are available as well (heart rate, galvanic skin response..)

From raw measurements we crated instances using 2 second sliding window and computed set of various features from accelerometer data that were shown to perform well in similar setting (mean, average, skewness, kurtosis, peak counts) [4]. Additionally we computed the Euler angles pitch and roll and calculated some extra features from them as well - for instance, pitch and roll manipulation, amount of roll motion, regularity of roll motion... Altogether we computed 105 features. Afterwards feature selection was applied and the best of them were used to build models.

3.1 Feature Selection

Feature selection was used only in the cases of flat classification and MM-HMC.For feature selection, we first ranked the features by gain ratio. After that, we used a wrapper approach. We started with an empty feature set and added features in the order of their rank. After each feature was added, we evaluated its contribution by building random-forest classifiers and internally cross-validating them on the training set. The feature was kept only and only if it increased the overall average accuracy. The ranking by gain ratio and the random forest algorithm were implemented in the Weka machine-learning suite and run with default parameter values.

3.2 Flat Classification

The most common approach for AR is the so-called flat classification. All classes are considered equal, hierarchy is not taken into account. Algorithms were implemented in java, using Weka[3] library.

3.3 Multi-Model Hierarchical Classification

We implemented two different approaches for hierarchical classification. The first one, traditional hierarchical strategy *Per Parent Top Down* (PPTD - Figure 1), based on "local per parent node" model, and the second one, named *Sum of weighted Votes* (SWV - Figure 2), "local per level" model, proposed by Paes et al. in [10]. On the upper level we built a model to distinguish between three groups - daily activities, exercise and static. This was done the same for both approaches. From here on, the approaches differ.

1. **PPTD** For this approach, we split instances into three different subsets regarding to the classified group. We then run feature selection for each of the subsets separately and built three different models - on for each group of activities. Features were different for each group.
2. **SVW** After the first level, the classified group has been added to instances as an additional feature. Feature selection has been done again - this time for the whole level, and one model has been built to distinguish between activities.

[3]https://www.cs.waikato.ac.nz/ml/weka/

Same authors have explored feature selection for both approaches in [11], where they have shown that the best results are obtained when using the *lazy* approach - this approach executes feature selection at the classification time of each instance. We have decided to use the *eager* approach, where feature selection is done prior to classification.

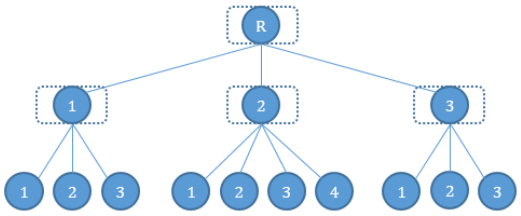


Figure 1: PPTD - local per parent node approach

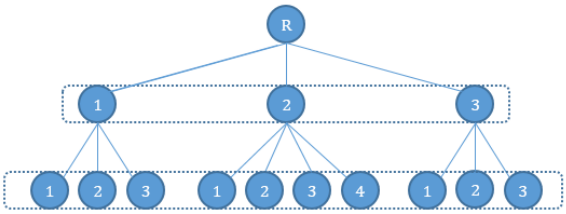


Figure 2: SVW - local per level approach

Figure 3: Hierarchical classifiers

3.4 CLUS-Classification

CLUS is a decision tree and rule learning system that works in the predictive clustering framework. One of its important functionalists is the CLUS-HMC algorithm for hierarchical multi-label classification. The software has been shown to work very well in the field of functional genomics [6], so the idea to use it in hierarchical classification for activity recognition seems reasonable. Clus-HMC algorithm is a variant of standard greedy top-down algorithm for decision tree induction. To achieve the task of predicting a set of classes instead of a single class, additional changes to the learning procedure are needed, as shown in [12].

```
[Hierarchical]
Type = Tree
HSeparator = /
WType = ExpMinParentWeight
WParam = 0.5
OptimizeErrorMeasure = AverageAUPRC
ClassificationThreshold = [0.5, 0.7, 0.9, 0.95]
MEstimate = Yes
SingleLabel = Yes

[Tree]
PruningMethod = M5
M5PruningMult = 2.0
FTest = 0.1

[Constraints]
MaxDepth = 20

[Ensemble]
EnsembleMethod = RForest
FeatureRanking = Genie3
PrintAllModelInfo = Yes
```

Figure 4: CLUS settings file example.

In our experiment we worked with random forest (to make it comparable with other two approaches), and we allowed the decision tree to go up to depth 20. We have shown experimentally that performance increases sharply up to decided depth, while afterwards the contribution has become negligible. The error we used for optimization was the average AUPRC (area under the precision-recall curve). We have tested the performance by changing the threshold determining when the probability output by the model is considered to predict a class. All of the above mentioned parameters are set in the settings file as seen in Figure 4.

4. EXPERIMENTAL SETUP AND RESULTS

In our case the hierarchy is very simple, reduced to two levels. For HMC problems Clus returns several error values. To get fair results for each person included in the dataset, leave-one-person-out approach has been used, as mentioned before. For evaluation of the results we decided to choose standard measurements - precision, recall and F-score. However, when it comes to the evaluation of highly skewed class distributions, similar as with our dataset where for instance daily activities have a much higher frequency than rest, precision-recall curves are the most suitable evaluation tool [7], so this was also added. Vens el al. [6] have addressed the problem of most eligible evaluation tools for hierarchical classification. From the proposed evaluation tools we used the area under the precision-recall curve.

To evaluate predictive models independently from the threshold, two types of evaluation are suitable: ROC analysis and analysis of precision-recall curves (PRC). ROC analysis is better known in machine learning, however for hierarchical multi-label classification PR is more suitable. [**?**] PR curve plots the precision of a model as a function of its recall, and although it helps understanding the predictions, single value is more appropriate for comparing quality of different models. A score often used to represent this is the so-called "area under the PR curve" (AUPRC). The closer the AUPRC is to 1.0, the better the model.

$$\overline{AUPRC}_{\mathrm{w}} = \sum_i \mathrm{w}_i \cdot AUPRC_i$$

If all the weights are set to $\mathrm{w}_i = 1/|C|$, where C is the set of classes, score is called average AUPRC, and is denoted as $\overline{AUPRC}$. If the weigths are set to $\mathrm{w}_i = v_i / \sum_j v_j$ where v_i is the frequency of class c_i in data, we call this weighted AUPRC and denote it as $\overline{AUPRC}_{\mathrm{w}}$. We have compared the performance of the proposed methods by comparing the precision, recall, F-score and $\overline{AUPRC}$ score by activity. Validation has been done using "leave-one-person-out" approach. We computed all of the mentioned measures for each person and averaged them to get the performance accuracy by method. Methods that we compared are flat classification, multi-model classification using SVW (local per level) approach and CLUS-classification using same approach. We decided to leave out the comparison of PPTD algorithm due to lack of data. Classes for *static* gorup were poorly represented from the beginning and after classification on the first level some were left with only few examples. To avoid loosing data we propose additional approach, which is roughly explained in the conclusion.

Using the same dataset Cvetkovic and al. [4] have reported on 70% accuracy for five different classes (sports, eating, chores, handwork, washing). We expected high confusion in group of daily activities (handwork, chores, eating, washing) and some confusion between other groups and within them as hand movements can be very similar in this group. Table 2 and Table 3 show the results of the experiments. We could not compare the $\overline{AUPRC}$ of flat classification when classifying groups, as we only get the values for classified activities on lower level. However, we could compare flat classification to other two approaches using other measures. As shown in Table 2 MM-HMC performs the best for AR on the upper level, but not much better than flat. On the lower level the results from flat classification and from MM-HMC were quite similar, with one approach performing better in some cases and worse in others. From the fact that direct classification on the upper level (MM-HMC) is not much better from the indirect, it is safe to conclude that this is the reason, that for similar results between the mentioned two approaches on the lower level. The achieved average accuracy for flat classification has been 70.5% and very similar for MM-HMC. Each works better in some cases. Results using CLUS are not the most promising. However, there are many possible combinations of settings available and the performance could be improved by choosing different set of parameters and their values. We tried many possible combinations and the presented results are the best so far.

Table 2: Results upper level (group)

	Flat	MM-HMC	CLUS
Fscore	82.05%	83.71%	74.36%
Precision	82.03%	83.73%	76.22%
Recall	82.12%	84.05%	73.10%
$\overline{AUPRC}$		89.61%	81.09%

Table 3: Results lower level (activity)

	Flat	MM-HMC	CLUS
Fscore	65.14%	66.79%	52.23%
Precision	68.29%	65.92%	58.31%
Recall	65.48%	67.69%	51.08%
$\overline{AUPRC}$	68.63%	66.67%	54.76%

5. CONCLUSION

In this work we compared three approaches to activity recognition. Our results show that for the purpose of activity recognition with 2 levels of activity (group and activity), flat classification performs as well as both types of hierarchical classification - or even better. In some other uses of HMC, for instance functional genomics, fast performance and correct classification of higher levels is of greater importance than correct classification of lower levels. Unfortunately in the case of activity recognition fast performance was the only upside.

There are some possible improvements for future work. The dataset we were working on, was not really extensive. There were many activities involved and not many instances of each. This could be solved with joining more similar datasets.

Some of the HMC-related papers mentioned different classifiers for classification. We used random forest, as it has performed the best in our previous research where we were only using flat classification, however some other classifiers may perform better on the hierarchical problem. Better accuracy could as well be achieved by adding measurements from some other sensors (heart rate sensor), as maybe there are some more distinctive differences between subsets of the proposed hierarchy.

A possibility to improve the performance of MM-HMC is to add additional activities to each of the groups. For instance, we add *exercise* and *static* as two new activities in group of *daily activities*. Similar would be done for other two groups of activities. After building models for the lower level, we would then build additional models for all "new activities" classified to wrong group. We will try this approach in our future work.

6. REFERENCES

[1] Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi. F., Oukhellou, L., Amirat, Y. Physical Human Activity Recognition Using Wearable Sensors. In: Sensors 15(12). Basel (2015). 31314-18

[2] Siirtola, P., Lurinen, P., Haapalainen, E., Röning, J., Kinnunen, H. Clustering-based activity classification with a wrist-worn accelerometer using basic features. In: 2009 IEEE Symposium on Computational Intalligence and Data Mining. CIMD 2009 - Proceedings. (2009) 95-100

[3] Chernbumroong, S., Atkins, A.S. Activity classification using a single wrist-worn accelerometer. In: 2011 5th International Conference on Software, Knowledge Information, Industrial manegment and Applications (SKIMA) Proceedings. (2011) 1-6

[4] Cvetkovic B., Drobnic V., Lustrek M.: Recognizing Hand-Specific Activities with a Smartwatch Placed on Dominant or Non-dominant Wrist. In: Information Society. Ljubljana (2017)

[5] Nweke, H.F., Teh, Y.M. Al-gardi, M.A., Alo, U.R. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challanges. In: Expert Systems With Applications. 105 (2018) 233-261

[6] Vens, C., Stryuf, J., Schietgat, L., Dzeroski, S., Blockeel H. Decision trees for hierarchical multi-label classsification. In: Machine Learning, 73(2): 185-214

[7] Davis, J., Goadrich, M. The relationship between precsion-recall and ROC curves. In: Proceedings of the 23rd Internation Conference on Machine Learning. (2006) 233-240

[8] Khan, A.M., Lee, Y.K., Lee, S.Y., Kim, T.S. A Triaxial Accelerometer-Based Physical-Activity recognition via Augmented-Signal Features and a Hierarchical Recognizer. In: IEEE Transactions on Information Technology in Biomedicine. 14(5). (2010) 1166-72

[9] Zheng, Y. Human Activity Recognition Based on the Hierarchical Feature Selection and Classification Framework. In: Journal of Electrical and Computer engineering. (2015) doi:10.1155/2015/140820

[10] Paes, B.C., Plastino, A., Freitas, A.A. Improving Local Per Level Hierarchical Classification. In: Journal of Information and Data Manegement 3 (3): (2012) 394-409

[11] Paes, B.C., Plastino, A., freitas, A.A. Exploring Attribute Selection in Hierarchical Classification. In: Journal of information and Data manegment- Vol. 5(1). (2014) 124-133

[12] Blockeel, H., Bruynooghe, M., Džeroski, S., Ramon, J., Struyf; J.: Hierarchical mulit-classification. In: Proceedings of the ACM SIGKDD 2002 Workshop on Mulit-Relational Data Mining (MRDM 2002). (2002) 21-35

[13] Cesa-Bianchi, N., Gentile, C., Zaniboni, L. Incremental algorithms for hierarchical classification. Journal of Machine Learning. (2006) 31-54

[14] Clus Homepage (last accessed 26 Jun 2018) https://dtai.cs.kuleuven.be/clus/index.html.

Aiding the Task of Process-Based Modeling with ProBMoTViz

Gjorgi Peev
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
gjorgi.peev@ijs.si

Nikola Simidjievski
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
nikola.simidjievski@ijs.si

Sašo Džeroski
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia
saso.dzeroski@ijs.si

ABSTRACT

Process-based modeling (PBM) is an equation discovery approach for automated modeling of dynamical systems, which takes at input substantial expert knowledge and measured data of the observed system. The resulting process-based models offer both high-level representation (in terms of building blocks / model components, i.e., entities and processes) and low-level representation (a set of ordinary differential equations). ProBMoT, a software platform for modeling, parameter estimation, and simulation of process-based models, is the latest implementation of the process-based modeling approach. While ProBMoT has been successfully applied to the task of modeling dynamical systems, compared to other modeling and simulation software, it is substantially behind in terms of user-friendliness. The goal of the present work is to overcome this limitation of ProBMoT. We design and implement an extension of ProBMoT, named ProBMoTViz, which is a platform consisting of a GUI (Graphical User Interface) for ProBMoT and a PBM Visualizer for process-based models. We evaluate the versatility of ProBMoTViz on example case studies.

1. INTRODUCTION

ProBMoT [4, 10] is the latest implementation of the process-based modeling approach [3, 6] for automated modeling of dynamic systems. Given a background knowledge, modeling constraints and measured data at input, ProBMoT constructs completely defined process-based models, represented with entities and processes.

ProBMoT has been successfully applied to a variety of modeling tasks in a number of real-world domains, such as aquatic ecosystems [4]; population dynamics [5]; biological systems [11]; oscillatory systems [8]; as well as predicting future behavior of the system at hand [9]. Unlike other modeling and simulation tools [7], ProBMoT is a domain-free tool and can be applied to any modeling task that involves model structure identification and/or parameter estimation. However, it still straggles behind these tools in terms of graphical/visual representation of the constructed models, comprehensibility of the output for a broader user-base as well as user-friendliness when it comes to preparing and running a PBM task. User feedback indicates that a GUI and a visual representation of process-based models can overcome these obstacles.

In this work, we set out to overcome the usability limitations of ProBMoT, i.e., to expand its user scope by developing and implementing an extension for it. In particular, we propose ProBMoTViz, a software platform which includes a Graphical User Interface (GUI) for ProBMoT and a PBM Visualizer for process-based models. On one hand, the GUI supports the basic operations of (automated) modeling dynamical systems in terms of providing appropriate input for the modeling and examining the outputs thereof. On the other hand, the PBM Visualizer illiterates the (currently textual) output models with a higher-level visual representation, that better communicates with the domain experts.

2. PROBLEM DEFINITION

ProBMoT addresses the task of automated modeling in terms of automated search of the appropriate model structure and estimating its parameter values. The input to the tool includes several input files: (1) a library of background knowledge (*.pbl* file specifying the domain); (2) a conceptual model (*.pbm* file specifying the problem); (3) a data file; and (4) a task specification *.xml* file, specifying the particular task.[1]

To this end, running ProBMoT require cumbersome and time-demanding procedures of preparing an appropriate input. For instance, the *.xml* task specification file defines all the hyper-parameters needed for ProBMoT to run properly, such as the paths of the input files, definition and mapping of variables and outputs to data sets, settings of the parameter fitter and the simulator, etc. All these components are represented with different XML tags. In response, the main contribution of ProBMoTViz is facilitating the task of process-based modeling with ProBMoT. In particular, the ProBMoT workflow will be encapsulated in a shell, where the *.xml* file is not written manually, but its representative settings are tuned interactively.

On the other hand, the constructed process-based models can be complex and difficult to understand[2]. The textual representation of process-based models can be improved, thus further enhancing their interpretability and communicability with domain experts. ProBMoTViz implements state-of-the-art visualization techniques able to overcome

[1]Tables 3-5 in the Appendix, present these inputs for a particular modeling task.
[2]Table 6 in the Appendix

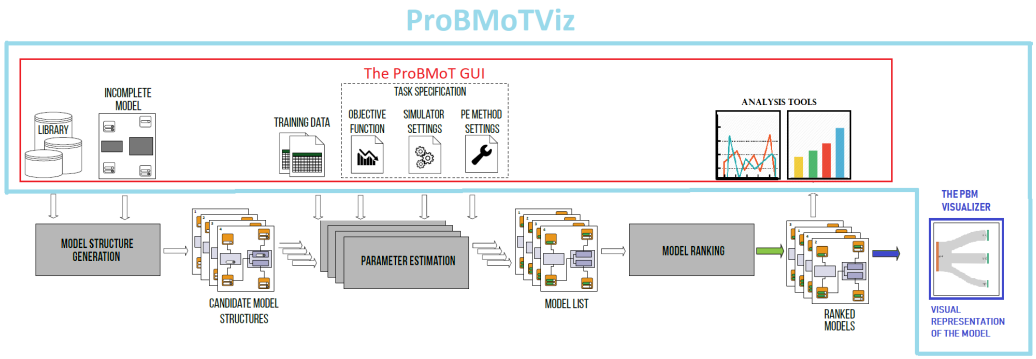


Figure 1. **ProBMoTViz, the proposed extension of ProBMoT, consists of a GUI for ProBMoT and a PBM Visualizer.**

the potential comprehensibility obstacles and usability limitations of the current textual representations.

3. PROBMOTVIZ

ProBMoTViz consists of two main components: the ProBMoT GUI and the PBM Visualizer (Figure 1). The former guides the user through the process of creating and defining a new PBM task step-by-step. It is a desktop-based application, developed in JavaFX [12], that facilitates the task of PBM, allowing for the settings, mappings, and all of the other customizable properties which must be specified in the *.xml* task settings file, to be now adjusted interactively in a workflow. The platform is divided into nine main scenes through which the user must progress in order to define the modeling task, monitor its progress, as well as to analyze the process-based models obtained at output. In particular, the scene sequence is as follows: (1) Library - the scene where the library file must be chosen and all the library components (template entities and processes) are shown; (2) Model - the scene where the (in)complete model file must be chosen and all the model components (instance entities and processes) are shown; (3) Data - the scene where the data files must be chosen, with the opportunity to inspect/ visualize the data; (4) Inputs - the scene where the input mappings must be specified, i.e., the mapping of the time dimension and the exogenous variables to a column in the data set; (5) Outputs - the scene where the outputs and their mappings to a specific column in the data set must be specified; (6) Overview scene; (7) Settings - the scene where all the task settings are specified, i.e., the evaluation, simulator, fitter, and other settings; (8) Run scene - where the particular task can be exported in an .xml format for later (re)use and (9) Results - the scene where the resulting process-based models can be inspected and analyzed.

The latter component, the PBM Visualizer, is a web application produced using the D^3.js (Data Driven Documents JavaScript library) [2]. It offers an interactive visual representation of the complex hierarchies of process-based models depicting both the high-level structure of the models as well as the interactions between its components. In particular, process-based models are depicted as a Sankey diagram [1], where the nodes denote the components of the process-based models. We define two main types of components that correspond to: *entities* and *processes*. Moreover, entities have one sub-type: *hierarchical entities* (representing the hierarchy that the entity comes from), while the processes have two sub-types: *hierarchical processes* (representing the hierarchy that the process comes from), and *children processes* (representing the nested processes in a process), as shown in Figure 2.

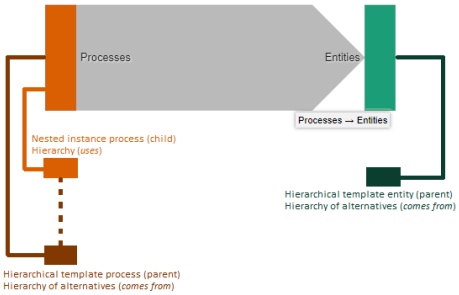


Figure 2. **Schematics of the visual representation of model components.**

To better illustrate how ProBMoTViz works, we present the important details of preparing a task for modeling a two cascaded water tanks system (Equation 1). The system consists of two cascaded water tanks with free outlets, placed one above the other, fed by a pump. In the governing equations for this system, the water levels of the tanks are denoted with *h1* and *h2*. *A1*, *A2*, *a1* and *a2* denote the areas of the tanks and their effluent areas, while the applied voltage-to-flow conversion constant is denoted with k. The task is to model the water level in the lower tank. The data is obtained by laboratory measurements [13] and it consists of 2500 samples (1500 train and 1000 test set) of the input voltage applied to the pump and the water levels in both tanks.

$$\begin{cases} \frac{dh_1}{dt} = -\frac{a_1\sqrt{2g}}{A_1}\sqrt{h_1} + \frac{k}{A_1}u(t) \\ \frac{dh_2}{dt} = -\frac{a_2\sqrt{2g}}{A_2}\sqrt{h_2} + \frac{a_1\sqrt{2g}}{A_2}\sqrt{h_1} \end{cases} \quad (1)$$

First, after loading the PBM library into ProBMoTViz (Figure 3), one can explore the encoded domain knowledge (for this example, for modeling fluid dynamics) in the traditional PBM formalism. In particular, the entity *Tank* encodes a variable that represents its water height level, and constants denoting the inflow and outflow areas. Analogously, the

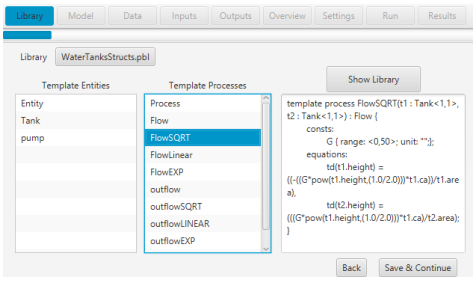


Figure 3. **The water tanks library of background knowledge.**

other entity *Pump* incorporates a variable denoting the input voltage in the system. Moreover, the library also encodes the different (plausible) interactions between the entities in terms of water transmissions between: two tanks, a tank and the environment as well as a tank and a pump. Note that, these interactions can also have different behaviour, therefore the library encodes different modeling alternatives for each of them in terms of a squared-root, linear or exponential dynamics.

In the next step, after loading the incomplete model (Figure 4), one can explore the specific components of the particular problem, i.e. two tanks and one pump. The task is defined as follows: in a two-tank system with an electric pump, identify the underlying dynamics of the three different interactions that describes the behaviour of the lower thank.

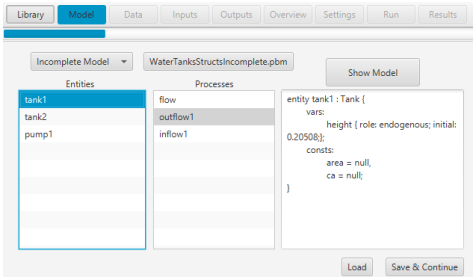


Figure 4. **The water tanks conceptual model.**

With loading the data, followed by specifying the mappings, defining the outputs and specifying the settings, our particular task is completely defined and ready for execution, as shown in Figure 5.

Finnaly, ProBMoTViz, lists the constructed models (Figure 6), and offers additional tools (error-plots and visual representation of the constructed process-based models) for further analyses.

4. CASE STUDIES

As a case study, we tackle the tasks of modeling two benchmark nonlinear dynamical systems: (1) Two cascaded water tanks system and (2) The SilverBox – an oscillatory system using ProBMoTViz. For evaluating the performance of our models, we measure the relative root mean squared error (RRMSE) of each model's output, shown in Equation 2. The number of samples in the test set is denoted with n; y_t is the measured and $\hat{y}_t$ is the predicted value (obtained

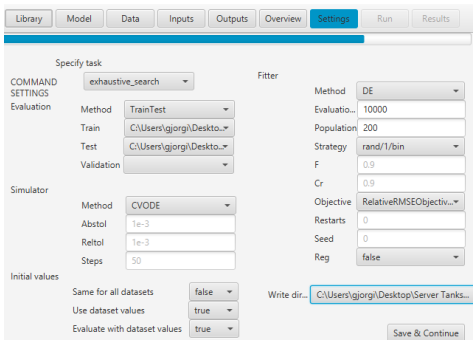


Figure 5. **The settings parameters for modeling the two water tanks system shown.**

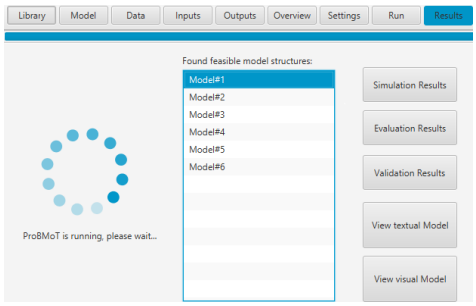


Figure 6. **The resulting models from modeling the two water tanks system.**

by simulating the model m on the test set) of the system variable y at time point t. The mean value of y in the test set is denoted with $\overline{y}$. This metric is relative to the standard deviation of the system variable in the test data, thus allowing us to compare the errors of models for different system variables with measured values on different scales.

$$RRMSE(m) = \sqrt{\frac{\sum_{t=0}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=0}^{n}(y_t - \overline{y})^2}} \quad (2)$$

4.1 Two cascaded water tanks system

For the previously defined water tanks system, given an input voltage, the output of interest in our model is the water height level of the lower tank. The particular process-based modeling task yields 9 feasible models, as shown in Table 1. The best obtained model (Figure 7) is contained of the processes *Inflow*, *FlowSQRT* and *OutflowSQRT*, which corresponds to the original system.

Table 1. **The results of modeling the two cascaded water tanks system.**

Model	Train RRMSE	Test RRMSE
ModelSqrtSqrt	**0.2208**	**0.2673**
ModelLinSqrt	0.2285	1.1837
ModelExpSqrt	0.2448	0.3101
ModelSqrtLin	0.2916	0.3200
ModelLinLin	0.3138	0.3323
ModelExpLin	0.3576	0.4249
ModelSqrtExp	0.7233	0.8507
ModelLinExp	0.7312	0.8585
ModelExpExp	0.7457	0.8835

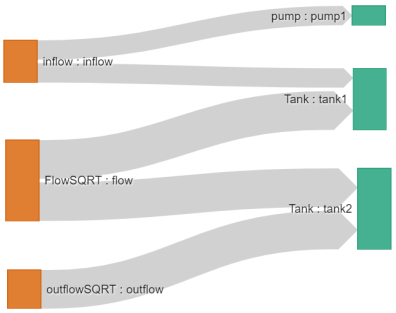


Figure 7. **The water tanks visual process-based model.**

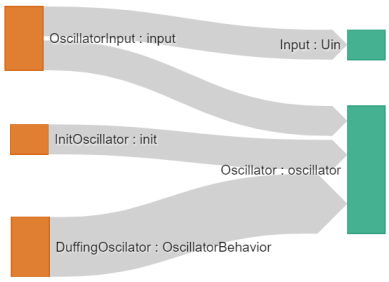


Figure 8. **The SilverBox visual process-based model.**

4.2 SilverBox Oscillator System

The second case study, addresses the task of reconstructing a nonlinear mechanical oscillating system, referred as the SilverBox system - an electronic implementation of the Duffing oscillator. The system's dynamics (Equation 3) relates to the displacement *y(t)* (the output) to the input voltage *u(t)*. The parameter m is a moving mass, d is viscous damping, and *k(y)* is a nonlinear progressive spring described by a static but position-dependent stiffness. The data is generated by an almost idealized representation of the oscillator [13]. It consists of 130000 samples (90000 train 40000 test set) of the input voltage and the output displacement of the oscillator.

$$\begin{cases} m\frac{d^2y(t)}{dt^2} + d\frac{dy(t)}{dt} + k(y(t))y(t) = u(t) \\ k(y(t)) = a + by^2(t) \end{cases} \quad (3)$$

The PBM library incorporates the domain knowledge where the entity *Oscillator* encodes the input voltage, output displacement and its mass. Moreover, behaviors of different oscillators: (1) Duffing, (2) Simple, (3) Harmonic and (4) Universal oscillator, all of which differently affect the output displacement of the oscillator are also encoded in the library. The incomplete model specifies the particular problem of one oscillator with unknown oscillatory behavior.

The process-based modeling task yields 4 feasible process-based models, as shown in Table 2. The best obtained model (Figure 8) contains the processes *InitOscillator*, *Oscillator-Input* and *DuffingOscillator*, which corresponds to the original system.

Table 2. **The results of modeling the SilverBox system.**

Model	Train RRMSE	Test RRMSE
ModelDuffing	**0.2353**	**0.2741**
ModelSimple	∞	∞
ModelUniversal	0.8803	0.8796
ModelHarmonic	0.9330	0.9327

5. CONCLUSIONS

In this work, we present a novel software tool, ProBMoTViz. It consists of two components: a GUI for ProBMoT and a PBM Visualizer for process-based models. The GUI supports the basic operations necessary for (automated) modeling of dynamical systems. Moreover, it allows for comprehensible and user-friendly preparation and analyses of modeling tasks. This enables the user to have better overview and control over the input parameters necessary when running ProBMoT, therefore saving time and computational resources when performing large amount of- and/or delicate experiments. The PBM Visualiser, on the other hand, aids in visualizing the complex (hierarchical) structure of process-based models, in turn allowing for better understanding and comprehensibility.

6. REFERENCES

[1] N. Atkinson. Bi-directional hierarchical sankey diagram [online: https://github.com/neilos/bihisankey], 2015.

[2] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309, 2011.

[3] W. Bridewell, P. Langley, L. Todorovski, and S. Džeroski. Inductive process modeling. *Machine learning*, 71(1):1–32, 2008.

[4] D. Čerepnalkoski. *Process-based Models of Dynamical Systems: Representation and Induction: Doctoral Dissertation*. PhD thesis, D. Čerepnalkoski, 2013.

[5] S. Džeroski and L. Todorovski. Encoding and using domain knowledge on population dynamics for equation discovery. In *Logical and computational aspects of model-based reasoning*, pages 227–247. Springer, 2002.

[6] P. Langley, J. N. Sanchez, L. Todorovski, and S. Džeroski. Inducing process models from continuous data. 2002.

[7] G. Peev, N. Simidjievski, and S. Džeroski. Modeling of dynamical systems : a survey of tools and a case study. In *20th International Multiconference Information Society - IS 2017*, volume A, pages 15–18, 2017.

[8] G. Peev, N. Simidjievski, and S. Džeroski. Identification of a nonlinear dynamical benchmark system using process-based modeling. In *10th JoÅ¿ef Stefan IPS Students' Conference*, page 36, 2018.

[9] N. Simidjievski, L. Todorovski, and S. Džeroski. Predicting long-term population dynamics with bagging and boosting of process-based models. *Expert Systems with Applications*, 42(22):8484–8496, 2015.

[10] J. Tanevski, N. Simidjievski, L. Todorovski, and S. Džeroski. Process-based modeling and design of dynamical systems. In *ECML PKDD*, pages 378–382. Springer, 2017.

[11] J. Tanevski, L. Todorovski, and S. Džeroski. Process-based design of dynamical biological systems. *Scientific reports*, 6:34107, 2016.

[12] K. Topley. *JavaFX Developer's Guide*. Pearson Education, 2010.

[13] T. Wigren and J. Schoukens. Three free data sets for development and benchmarking in nonlinear system identification. In *2013 European Control Conference (ECC)*, pages 2933–2938, July 2013.

Appendix: ProBMoT inputs

Tables 3-5 present the necessary inputs for ProBMoT for the tasks of automated modeling of a water tank dynamic system. Table 6 presents a resulting process-based model in the standard PBM formalism.

Table 3. Library of domain knowledge

```
library WaterTanksLibrary;
//ENTITIES
template entity Tank {
vars:
        height {aggregation:sum, range:<0,500>};
consts:
        area {range: <1.0E-3,30>},
        ca {range: <1.0E-3,30>}; }
template entity Pump {
vars:
        v{aggregation:sum, range:<-15,15>};
consts:
        k {range:<0.2,1E6>}; }

//PROCESSES
template process Flow (t1 : Tank, t2 : Tank) {
consts:
        G {range: <0,50>},
template process FlowSQRT : Flow {
equations:
        td(t1.height) = - (G * pow(t1.height,1/2) * t1.ca)/t1.area,
        td(t2.height) = (G * pow(t1.height,1/2) * t1.ca)/t2.area; }
template process FlowLINEAR: Flow {
equations:
        td(t1.height) = - (G * t1.height * t1.ca)/t1.area,
        td(t2.height) = (G * t1.height * t1.ca)/t2.area; }
template process FlowEXP : Flow {
equations:
        td(t1.height) = - (G * exp(t1.height) * t1.ca)/t1.area,
        td(t2.height) = (G * exp(t1.height) * t1.ca)/t2.area; }
template process Outflow (t:Tank) {
consts:
        G {range: <0,50>}; }
template process OutflowSQRT: Outflow {
equations:
        td(t.height) = - G * pow(t.height,1/2) * t.ca/t.area; }
template process OutflowLINEAR :Outflow {
equations:
        td(t.height) = - G * t.height * t.ca/t.area; }
template process OutflowEXP :Outflow {
equations:
        td(t.height) = - G * exp(t.height) * t.ca/t.area; }
template process Inflow (p: pump, t: Tank) {
equations:
        td(t.height) = p.k * p.v/t.area; }
```

Table 4. Incomplete model of the two cascaded water tanks system.

```
incomplete model WaterTanksIncomplete : WaterTanksLibrary;
//Entities
entity tank1 : Tank {
vars:
        height { role: endogenous; initial: 0.20508;};
consts:
        area = null,
        ca = null;
}
entity tank2 : Tank {
vars:
        height { role: endogenous; initial: 0.38086;};
consts:
        area = null,
        ca = null;
}
entity pump1 : Pump {
vars:
        v { role: exogenous;};
consts:
        k = null;
}

//Processes
process flow (tank1, tank2) : Flow {
consts:
        G = 4.429; }
process outflow1 (tank2) : Outflow {
consts:
        G = 4.429; }
process inflow1 (pump1, tank1): Inflow {
}
```

Table 5. An example task specification file in XML format.

```
<task>
        <library>C:/Users/WaterTanksLibrary.pbl</library>
        <incomplete>C:/Users/WaterTanksIncomplete.pbm</incomplete>
        <data>
                <d separator="," id="1">C:/Users/Data1.csv</d>
                <d separator="," id="2">C:/Users/Data2.csv</d>
        </data>
        <mappings>
                <dimensions>
                        <dim name="time" col="t"/>
                </dimensions>
                <exogenous>
                        <exo name="WaterTanksIncomplete.pump1.v" col="u"/>
                </exogenous>
                <endogenous>
                        <endo name="WaterTanksIncomplete.tank1.height" col="h1"/>
                        <endo name="WaterTanksIncomplete.tank2.height" col="h2"/>
                </endogenous>
                <outputs>
                        <out name="h2" col="h2"/>
                </outputs>
        </mappings>
        <output>
                <variables>
                        <var name="h2">WaterTanksIncomplete.tank2.height</var>
                </variables>
        </output>
        <writeDir>C:/Users/</writeDir>
        <command>exhaustive_search</command>
        <settings>
                <initialvalues>
                        <sameforalldatasets>false</sameforalldatasets>
                        <usedatasetvalues>true</usedatasetvalues>
                </initialvalues>
                <simulator method="CVODE">
                        <abstol>0.001</abstol>
                        <reltol>0.001</reltol>
                        <steps>1000</steps>
                </simulator>
                <fitter method="DE">
                        <restarts>0</restarts>
                        <evaluations>10000</evaluations>
                        <population>200</population>
                        <strategy>rand/1/bin</strategy>
                        <F>0.9</F>
                        <Cr>0.9</Cr>
                        <seed>0</seed>
                        <reg>false</reg>
                        <objectives>
                                <obj>RelativeRMSEObjectiveFunctionMultiDataset</obj>
                        </objectives>
                </fitter>
                <evaluation method="TrainTest">
                        <train>1</train>
                        <test>2</test>
                </evaluation>
        </settings>
</task>
```

Table 6. A process-based model of a two cascaded water tanks system.

```
model WaterTanksModel : WaterTanksLibrary;
entity tank1 : Tank {
vars:
        height { role: endogenous; initial: 0.20508;};
consts:
        area = 19.944,
        ca = 1.087;
}
entity tank2 : Tank {
vars:
        height { role: endogenous; initial: 0.38086;};
consts:
        area = 23.051,
        ca = 3.066;
}
entity pump1 : Pump {
vars:
        v { role: exogenous;};
consts:
        k = 20.305;
}

//Processes
process flow (tank1, tank2) : FlowSQRT{
consts:
        G = 4.429; }
process outflow1 (tank2) : OutflowSQRT {
consts:
        G = 4.429; }
process inflow1 (pump1, tank1): Inflow {
}
```

Evaluation and Prospects of Semi-Automatic Video Distance Measurement in Ski Jumping

Matjaž Kukar
University of Ljubljana
Faculty of Computer and Information Science
Večna pot 113
SI-1000 Ljubljana, Slovenia
matjaz.kukar@fri.uni-lj.si

ABSTRACT

Great competitive results of Slovenian ski jumpers in world cup and continental competitions have sparkled a lot of interest for active participation in this attractive sport. In junior levels, national competitions with considerably more than 100 jumpers are becoming the norm. However, due to lack of technologic aids for distance measurement, such competitions can last over half a day. Only at the top-level competitions (world cup, continental cup) expensive and logistically demanding commercial video distance measuring tools are used for this purpose. In a previous project we developed a video distance measuring system from low-cost commercial components, which was not suitable for real-time usage due to technological limitations, but worked great for offline measurement. We analyze the results of offline measurements for several competitions and show that measurement errors are often unacceptably high. This serves as a motivation for an ongoing project, where video measurement is performed in real time and supported by advanced computer vision and deep learning methods.

Keywords

ski jumping, video distance measurement, computer vision, machine learning, deep learning

1. INTRODUCTION

In recent years, we have witnessed a boom in Slovenian ski jumping, mainly as a consequence of excellent results of Slovenian competitors. There is a marked increase in interest at the primary level; the ski jumping clubs have reportedly doubled the number of younger, primary school competitors (7-10 years). This has considerably increased the burden on ski jumping coaches, as well as on organizers and professional staff in competitions, that are carried out even in the youngest categories (from 2018, up to 10 years on only as "animations").

The administrative support for small competitions is mostly covered by the information system "Spletni Smuško[1]", while the IT support is virtually nonexistent for style and distance umpires. Only at the highest competitive levels (world cup, continental cups), professional staff — delegates, style and distance umpires — are supported by expensive commercial solutions [6, 2]. In our project, we focus primarily on supporting distance umpires who have a demanding, exposed role, and their mistakes often lead to bad will among coaches, competitors' parents and spectators, as well as in public opinion.

The aim our previous work [1] was to develop a system for supporting video distance measuring on smaller hills, with accessible hardware requirements (a single video system and a laptop). In this paper we evaluate the system results from ski jumping competitions in younger categories on small hills in regional competitions (Cockta Cup), and provide some directions for future development.

1.1 Ski Jump Distance Measurement

In ski jumping the jump distance is defined as a distance between the edge of the jumping ramp and a point where both ski jumper's legs have touched the ground with full surface [4, article 432.1]. The middle point between both legs is used when the legs are apart (e.g., Telemark landing style). There are however three exceptions [3]:

1. In one-legged landings (i.e. the second ski is longer in the air than what is typical during the normal landing routine) the correct distance is measured where the first ski touches ground with full surface.
2. In a fall (where the landing does not result on the skis as is normal), the correct distance is measured at the location where the ski jumper contacts first the landing surface with a body part.
3. In arbitrarily delayed landings (i.e. the ski jumper is positioned extremely behind thus delaying the normal landing routine and the touch down of the ski tips to the landing surface) the correct distance is measured where both feet contact first the landing surface.

Even on the smallest competition hills ($HS \leq 15\ m$) it is difficult to measure the exact flying distance by eyes only, since landing speeds exceed 10 m/s (36 km/h), and the angle

[1] http://smusko.adamsoft.si

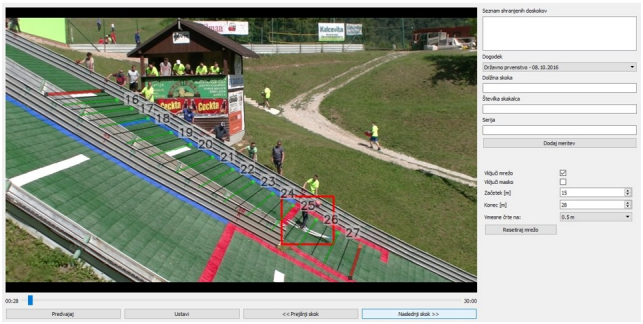


Figure 1: Offline video distance measuring system. A calibrated measuring grid is overlaid over video stream. Eight distance-measuring umpires can be seen standing along the landing slope.

between the landing slope and landing trajectories of ski jumpers is often very small [5].

Therefore, umpire tower is often built not far from the lower end of landing slope. It allows good view, but is utilized chiefly by style-measuring umpires. During ski jumping competitions, distance-measuring umpires are stationed a few meters apart along the landing slope (Figure 1). Usually they are volunteers from the organizing ski jumping club, and often have no training and very little experience with distance measuring. With speeds exceeding 10 m/s, the umpires have less than 0.05 second (with resolution of 0.5 m) to decide on a particular distance. Also, as declared by ski jumpers, they are almost never able to determine their flying distance with a reasonable accuracy. As a consequence, ski jumpers and their coaches are challenged when evaluating the progress in terms of jump/flight distance. A reasonably automated video distance measuring system therefore has the potential to become an important coaching aid in everyday practice.

2. OFFLINE VIDEO DISTANCE MEASURING

In a recent project cooperation with the Ski Association of Slovenia[2] (SAS) we developed a system for offline video measurement [1]. The aim of the project was to develop a reasonably priced system for video distance measurement based on commercially available components. It utilized a JVC GC-PX100 camcorder[3] that allows recording of up to 600 frames per second (FPS). While the camcorder was great for offline video measurement due to standalone video recording, it was impossible to use it in an online setting due to it's incapability of live video streaming to the computer.

In the offline setting we recorded several competitions on small hills (HS up to 25 m). Two professional ski-jumping coaches utilized specially developed software developed within the project (see [1] and Figure 1) to facilitate offline video measurement. In total, more than 200 ski jumps were video measured. For 86 we identified the jumpers and obtained officially measured distances, that were used for further evaluation. All jumps were either successful or with the ski jumper touching the landing slope with his/her hands. No spectacular falls were included.

[2]https://www.szs.si
[3]https://eu.jvc.com/microsite/eu/gc-px100/index.html

Table 1: Basic statistic of official and video measurements.

	Official distance	Video distance	Abs. diff.	Abs. diff. (centered)
count	86	86	86	86
mean	22.52	21.99	0.62	0.33
st. dev.	2.11	2.23	0.37	0.35
min	17.00	15.50	0.00	0.00
max	26.00	25.50	1.50	1.50

3. EVALUATION OF OFFICIAL DISTANCE MEASUREMENT RESULTS

For 86 jumps we compared the official results (measured by eyes only) and offline video measurements, performed by two professional ski jumping coaches. Figure 2 depicts a scatter plot of official measurements vs. video measurements. From the placement of measurement pairs (almost all are below the diagonal) it is obvious that manually measured distances are bigger than video measured ones. This bias is a result of different positioning of umpires and video camera, resulting in different parallax errors (the camera was mostly positioned slightly higher than umpires and more towards the outrun). In Figure 3 this bias can be clearly seen as nonnegative differences in distances for all but five jumps.

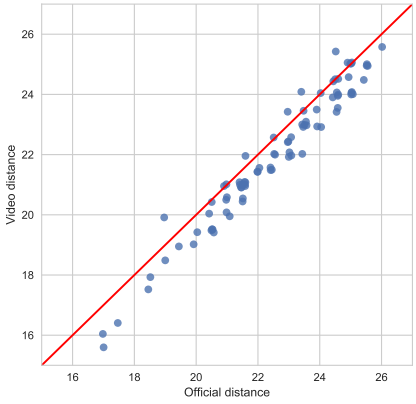


Figure 2: Scatter plot of official vs. video measurements.

Table 1 shows basic statistics of official and video measurements. The difference in means (0.53 m) indicates the need to account for different biases for each method. For this reason we compare the two distance measuring approaches with their values centered around their means (Eq. 1).

$$center_j^{(m)} = d_j^{(m)} - \bar{d}^{(m)} \quad (1)$$

where $\bar{d}^{(m)}$ is the mean value of all measured distance for a particular measuring method m, and $d_j^{(m)}$ is the measured distance for the jump j (again for a particular measuring method m). This allows us to contain the bias within the $\bar{d}^{(m)}$ and focus only on the differences dif_j (Eq. 2).

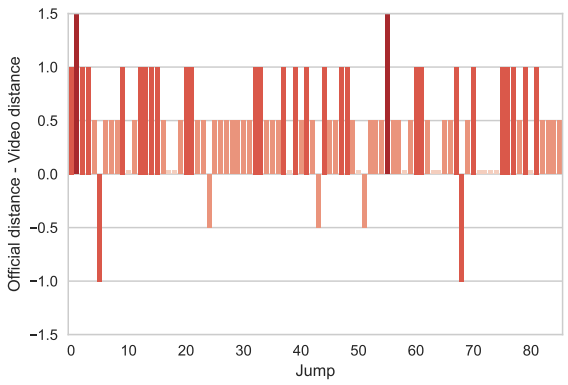


Figure 3: Differences between original official and video measurements.

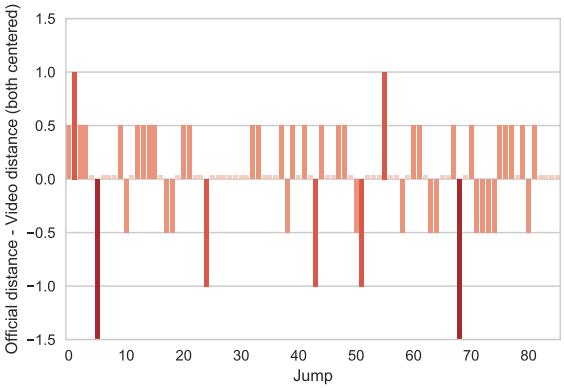


Figure 4: Differences between centered official and video measurements.

Table 2: Frequencies and percentages of absolute differences between centered official and video measurements.

Difference (centered)	Count	%
0.0	39	45
0.5	40	47
1.0	5	6
1.5	2	2

$$dif_j = \Big(\underbrace{d_j^{(manual)} - \bar{d}^{(manual)}}_{\text{centered manual distance}} \Big) - \Big(\underbrace{d_j^{(video)} - \bar{d}^{(video)}}_{\text{centered video distance}} \Big) \tag{2}$$

According to the involved ski jumping coaches, our video measurements are much more reliable than the official manual ones, and can be considered as correct. Figure 4 and Table 2 show a much clearer picture of official measurement errors. 45% of measurements are deemed to be exact (within 0.5 m), 47% are off by ±0.5 m, while additional 8% errors are in the range of 1-1.5 m. For distances around 15 m this means a whopping 10%! To put this in perspective, for a world record jump (253.5 m) this would translate to 25 m! On small hills, 1 m is worth 4.5-6 points, and such errors can easily influence podium places, especially in closely fought competitions. With introduction if video distance measuring system would therefore benefit both umpires (less demanding work) competitors and spectators (less distance measuring errors).

4. BEYOND OFFLINE MEASURING

In the ongoing ŠIPK project we are partnered with the Technix[4] d.o.o company, the biggest provider of traffic surveillance network cameras in Slovenia. They kindly provided various camera models, produced by Axis Communications. We settled for the high frame rate model Q1645[5], that connects to the computer via 100 Mbit Ethernet connection, allows frame rates up to 120 FPS with full HD resolution, and supports low light recording. At the time of paper submission the project is still in progress, therefore we are reporting only partial results. Figure 5 shows the video distance measuring system (camera and laptop) in action.

Figure 5: An online video distance measuring system consisting of a network camera (left) and a laptop computer.

One of the main drawbacks of our original system [1] was the lack of online distance measuring. This is now effectively solved by utilizing the network camera. The video processing pipeline consists of several steps:

- a frame is acquired from the camera (MJPEG or H.264 stream)
- Gaussian blur is used to get rid of noise
- background is subtracted by using the MOG2 algorithm [7], and the image is converted to black (background) and white (moving) pixels, based on the last five frames
- of all the moving contours, the largest is selected as the ski jumper, and the corresponding bounding box is superimposed to the frame (Figure 6).
- once detected, the ski jumper is tracked until he/she has left the camera view

According to [3] and [4, article 432.1], video distance measurement is performed in two steps:

1. determining the correct landing frame
2. determining the correct landing point corresponding to the ski jumper's foot positions.

[4]https://www.technix.si
[5]https://www.axis.com/products/axis-q1645

Figure 6: Trained operator determines the distance by using the calibrated measuring grid in 5-8 s.

A heuristic approach based on the flight curve derivatives is used to approximately detect the landing frame. It works with accuracy of approximately 1 m (on small hills). The human operator is still needed to determine the correct landing frame, and determine the distance based on the superimposed measuring grid (Figure 6).

4.1 Automatic Detection of Landing Point with Deep Learning

We are currently experimenting with two approaches to automate and speed up video distance measuring. The first approach is using a deep convolutional neural network with 10 hidden layers in order to automatically detect the correct landing frame. Its input is a framed ski jumper in resolution 150 × 150 color pixels. Each frame is classified either as "air" or "ground". Due to real time processing requirement (network executes on CPU only) the current topology it is relatively shallow. It consists of two 2D convolutional (C), two pooling (P), four dropout (D), one flattening (F), and three dense layers (De) as follows: C-P-D-C-P-D-F-De-D-De-D-De. A sequence of frames can be classified as shown in Figure 7. The sequence always starts with "air" and ends with "ground". When at least two subsequent "ground" frames are detected, the first is selected as the landing frame. This approach currently achieves 96% classification accuracy for determining the type of frame. However, as the errors always occur near the correct landing frame, human intervention is still necessary. The second approach utilizes classic computer vision image segmentation techniques to acquire positions of ski jumper's skis and legs in order to determine the correct landing point within the frame, and therefore the distance based on the measuring grid (currently with accuracy of 0.5-1 m). Regarding the processing speed, for small hills 30 FPS are sufficient to achieve 0.5 m accuracy, however the process works well even for 100 FPS video stream (tested in laboratory conditions). For deeper neural networks, a gaming laptop with a discrete GPU will be required.

5. CONCLUSIONS

Our evaluation has shown that there is great need for improvement in ski jumping distance measuring, especially for small hills. In order to achieve objective results and reduce errors, video distance measurement is highly advisable. There is considerable interest from ski jumping clubs and

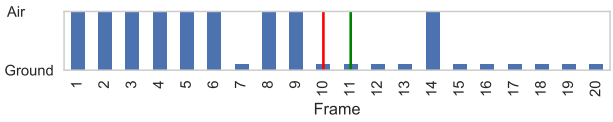


Figure 7: A sequence of frames classified as "air" and "ground". The green line marks the correct landing frame, and the red line the predicted landing frame. Frames 7, 10 and 14 are incorrectly classified.

SAS for widespread testing. For use on larger hills, slight modification of software will be needed in order to allow for two, three or four network cameras. The system still needs further testing (especially the automated components) under artificial lighting conditions (night competitions).

6. ACKNOWLEDGMENTS

Original software for video distance measurement was developed within the PKP project "Video meritve dolžin smučarskih skokov" by T. Ciglarič, K. Gostiša, T. Kovač, D. Peternel, M. Pograjc, N. Stoklas, B. Štempelj and G. Vodan [1]. Advanced automatization is developed within an ongoing ŠIPK project VIDEOMEN in the ski jumping center Mengeš, Slovenia.

7. REFERENCES

[1] T. Ciglarič, K. Gostiša, T. Kovač, D. Peternel, M. Pograjc, N. Stoklas, B. Štempelj, G. Vodan, R. Rozman, and M. Kukar. Video meritve dolžin smučarskih skokov. In *ERK'2017*, pages 337–340, 2017.

[2] ewoxx. Sports data service. *https://ewoxx.com/sports*. Accessed: August 2018.

[3] FIS Ski Jumping Committee Sub-Committee for Officials, Rules and Control. Guidelines to Video Distance Measurement of Ski Jumping 2011. *http://www.fis-ski.com/mm/Document/documentlibrary/Skijumping/03/20/28/Guidelines-VDM-2011-eng-deutsch_Neutral.pdf*. Accessed: August 2018.

[4] FIS Ski Jumping Committee Sub-Committee for Officials, Rules and Control. *The International Ski Competition Rules (ICR). Book III: Ski Jumping*. June 2018 edition. *http://www.fis-ski.com/mm/Document/documentlibrary/Skijumping/03/19/96/ICRSkiJumping2018_clean_English.pdf*. Accessed: August 2018.

[5] N. Sato, T. Takayama, and Y. Murata. Early evaluation of automatic flying distance measurement on ski jumper's motion monitoring system. In *Proc. 27th IEEE Int. Conf. on Advanced Information Networking and Applications*, pages 838–845, 2013.

[6] Swiss timing. Video distance measurement. *https://www.swisstiming.com/fileadmin/Resources/Data/Datasheets/DOCM_SJ_VDMS_1215_EN.pdf*. Accessed: July 2018.

[7] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.

Opis zmagovalne rešitve na mednarodnem tekmovanju o napovedovanju izida točk v tenisu

Miha Mlakar
Jozef Stefan Institute
1000-SI Ljubljana, Slovenia
miha.mlakar@ijs.si

Scott Sobel
Oliver Wyman
Columbia, South Carolina, USA
scott.sobel@oliverwyman.com

POVZETEK

Januarja 2018 je Avstralska teniška zveza v sodelovanju s Tennis Australia's Game Insight Group organizirala tekmovanje z naslovom *From AO to AI: Predicting How Points End in Tennis.* Cilj tekmovanja je bilo narediti model, ki bi na podlagi podatkov o udarcih in letu žogice, pridobljenih iz kamer, čim bolje klasificiral konec točke v enega od treh razredov; nepotrebna napaka, prisiljena napaka in neubranljiv (*winner*) udarec. Izmed 750 tekmovalcev, ki jih je sodelovalo na tekmovanju sva soavtorja osvojila prvo mesto. V referatu je prikazan postopek razvoja zmagovalne rešitve, ki vključuje generiranje in izbiro spremenljivk, izbiro ustreznega modela za strojno učenje, optimizacijo njegovih hiperparametrov, ter predstavitev rezultatov, dobljenih z zgrajenim modelom.

Ključne besede

Tenis; Tekmovanje; Strojno učenje; Umetna inteligenca; XGB

1. UVOD

Tenis je glede na gledanost eden izmed najpopularnejših športov na svetu. Da bi gledalcem in igralcem zagotovili zanimive statistike, se na vsaki tekmi meri veliko različnih parametrov, od uspešnosti prvega servisa, do števila dobljenih točk na nasprotnikov servis. Ena izmed pomembnejših statistik je tudi število napak in neubranljivih udarcev, ki jih igralec izvede tekom tekme. Te statistike trenutno beležijo ročno, strokovnjaki, ki ob igrišču gledajo tekmo in glede na svoje izkušnje označijo zaključek točke, kot nepotrebna napaka, prisiljena napaka ali pa kot neubranljiv udarec.

Vendar pa tak način zbiranja podatkov ni najbolj primeren zaradi možnosti napak, nekonsistentnih označb s strani različnih ljudi in pa tudi zaradi stroška ter dodatne logistike, saj na večjih turnirjih hkrati poteka veliko število tekem. Če bi hoteli dobiti te vrste podatkov za vse tekme bi morali imeti zaposlenih veliko ljudi. Zato je Avstralska teniška zveza v sodelovanju z Tennis Australia's Game Insight Group organizirala tekmovanje z naslovom *From AO to AI: Predicting How Points End in Tennis*, kjer so kot rešitev iskali algoritem, ki bi na osnovi podatkov pridobljenih iz kamer objektivno in čim bolj natančno lahko avtomatsko določil tip zadnjega udarca.

Tekmovanje je bilo organizirano na platformi CrowdANALYTIX (https://www.crowdanalytix.com/), ki omogoča organiziranje spletnih tekmovanj s področja umetne inteligence. Na tekmovanje se je registriralo 750 tekmovalcev oziroma tekmovalnih ekip.

Analiza in razvoj modelov je temeljila na podatkih pridobljenih iz sistema desetih visoko-resolucijskih kamer imenovanega HawkEye (https://www.hawkeyeinnovations.com/), ki se v tenisu primarno uporablja kot pripomoček za detekcijo oziroma določanja ali je določena žoga padla v avt ali ne. Poleg te primarne funkcije lahko s tem sistemom zelo natančno sledimo letu in hitrosti žogice, ter tako dobimo veliko podatkov, ki so bili nato na voljo za gradnjo modelov.

Po začetni analizi podatkov smo iz obstoječega nabora zgradili veliko dodatnih spremenljivk, ki smo jih nato uporabili za modeliranje. Sledila je optimizacija parametrov algoritmov strojnega učenja in predstavitev dobljenih rezultatov. Končna klasifikacijska točnost napovedi je bila 95% (95% za neubranljive udarce, 89% za prisiljene napake in 98% za neizsiljene napake).

2. OPIS IN ANALIZA PODATKOV

Analiza teniških udarcev se je izvajala na zaključnih udarcih točk pridobljenih iz Australian Open 2017 turnirja. Točke v naši bazi so bile odigrane tako v moški, kot tudi v ženski konkurenci, pogoj pa je bil, da je bila dolžina točke več kot dva (servis in return) udarca.

Podatki, ki so bili na voljo so bili razdeljeni ne učni in testni množici. Učni množici sta vsebovali podatke za 5000 točk odigranih v moški konkurenci in za 5000 točk odigranih v ženski konkurenci. Testni množici sta vsebovali podatke za 2000 točk v moški in 1000 točk v ženski konkurenci. Seveda testni podatki niso imeli določenega tipa zaključnega udarca, saj je bila to naloga našega algoritma.

Podatki so vsebovali 27 spremenljivk in pa tip zadnjega udarca. Imena parametrov, njihovimi opisi in možne vrednosti so predstavljeni v tabeli 1.

Za kreiranje dobrih napovednih modelov so ključni dobri podatki in odlično poznavanje njihovih lastnosti. Zatosmo najprej naredili podrobno analizo podatkov. Ta analiza nam pomaga razumeti podatke, odkrije povezave med spremenljivkami in razredi in identificira izvore šuma ter druge probleme, ki so vezani na kvaliteto podatkov.

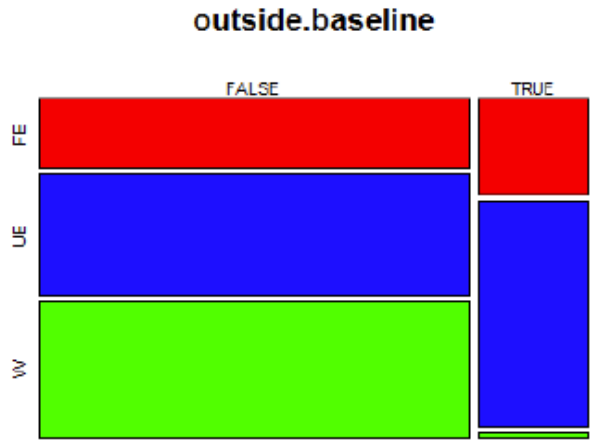
Začeli smo z izračunom distribucije tipov zadnjih udarcev čez celotno učno množico 10.000 točk, ki je sestavljena iz 3,352 (33.5%) neubranljivih udarcev, 2,272 (22.7%) prisiljenih napak in 4,376 (43.8%) nepotrebnih napak. Ta podatek nam pove, da so razredi delno neuravnoteženi, da pa ta neuravnoteženost ni takšna, da bi pri strojnem učenju bilo potrebno uporabiti prav posebne metode namenjene reševanju problemov z neuravnoteženimi podatki.

Tako učna kot testna množica nista vsebovali manjkajočih podatkov, tako da nam ni bilo potrebno uporabiti algoritmov za nadomeščanje praznih vrednosti. Se pa je med podatki pokazalo, da obstajajo netočne vrednosti. Primer je viden na sliki 1, kjer lahko vidimo, da so nekateri udarci, označeni, kot da je žoga padla izven dovoljenega območja a so še vedno označeni kot neubranljivi udarci in ne kot napaka. Razlogi za to so lahko trije:

Spremenljivka	Opis	Vrednosti
outcome	Razredna spremenljivka – tip zadnjega udarca	W=zaključni udarec, FE=prisiljena napaka, UE=nepotrebna napaka
speed	Hitrost zadnjega udarca	Zvezna vrednost (m/s)
previous.speed	Hitros predzadnjega udarca	Zvezna vrednost (m/s)
net.clearance	Razdalja leta žoge nad mrežo za zadnji udarec	Zvezna vrednost (cm). Negativna če je žogica letela nižje od mreže
previous.net.clearance	Razdalja leta žoge nad mrežo za predzadnji udarec	Zvezna vrednost (cm). Lahko negativna če je žogica letela nižje od mreže
distance.from.sideline	Razdalja odboja žoge zadnjega udarca od najbližje črte	Razdalja v metrih (pozitivna tudi če je odboj žogice izven igrišča)
depth	Razdalja odboja žoge od osnovne črte za zadnji udarec	Razdalja v metrih (pozitivna tudi če je odboj žogice izven igrišča)
player.distance.travelled	Razdalja, ki jo je igralec pretekel pred zadnjim udarcem	Euklidska razdalja v metrih
player.impact.depth	Oddaljenost igralca od mreže v trenutku, ko je udaril zadnji udarec	Razdalja od mreže v metrih
player.impact.distance. from.center	Oddaljenost igralca od sredine igrišča v trenutku, ko je udaril zadnji udarec	Razdalja od sredine igrišča v metrih
player.depth	Oddaljenost igralca od mreže v trenutku, ko je udaril predzadnji udarec	Razdalja od mreže v metrih
player.distance. from.center	Oddaljenost igralca od sredine igrišča v trenutku, ko je udaril predzadnji udarec	Razdalja od sredine igrišča v metrih
oponent.depth	Oddaljenost nasprotnega igralca od mreže v trenutku, ko je udaril predzadnji udarec	Razdalja od mreže v metrih
opponent.distance. from.center	Oddaljenost nasprotnega igralca od sredine igrišča v trenutku, ko je udaril predzadnji udarec	Razdalja od sredine igrišča v metrih
previous.distance. from.sideline	Razdalja odboja žoge predzadnjega udarca od najbližje črte	Razdalja v metrih (pozitivna tudi če je odboj žogice izven igrišča)
previous.depth	Razdalja odboja žoge od osnovne črte za predzadnji udarec	Razdalja v metrih (pozitivna tudi če je odboj žogice izven igrišča)
previous.time.to.net	Koliko časa je žoga pri predzadnjem udarcu potrebovala od udarca pa do mreže	Zvezna vrednost v sekundah
server.is.impact.player	Indikator ali je zadnji udarec v točki odigral server	TRUE = DA, FALSE = NE
same.side	Logični indikator, ki pove ali sta se igralca nahajala na isti strani igrišča v času predzadnjega udarca	TRUE = DA, FALSE = NE
outside.sideline	Logični indikator ali je žogica padla znotraj stranskih črt	TRUE = DA, FALSE = NE
outside.baseline	Logični indikator ali je žogica padla znotraj osnovne črte	TRUE = DA, FALSE = NE
train	Indikator ali je točka del učne ali testne množice	1 = Training, 0 = Test
serve	Ali je bila točka odigrana na prvi ali na drugi servis	1= Prvi servis, 2= Drugi servis
gender	Indikator, ki pove ali je bila spol igralcev	mens=moški, womens =ženske
previous.hitpoint	Kateri je bil predzadnji udarec v točki	F = forhend, B = bekend, V = volej, U = neznan tip udarca
hitpoint	Kateri je bil zadnji udarec v točki	F = forhend, B = bekend, V = volej, U = neznan tip udarca
id	10-črkovni unikatni identifikator točke	črkovni niz
rally	Število udarcev v točki	3, 4, 5, …

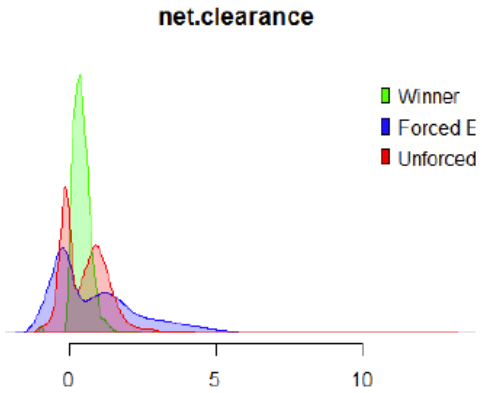
Tabela 1: Opis spremenljivk

(i) človeška napaka pri označevanju, (ii) napačna prepoznava mesta odboja žogice z strojnim vidom iz kamer ali pa (iii) žoga je padla v avt, vendar pa sodniki na tekmi tega niso dosodili, zato je bil udarec dosojen kot dober in zato je v podatkih označen kot neubranljiv udarec. Ker je razlog za take primere neznan in ker je takih primerov manj kot 1% smo jih pustili v testni množici.



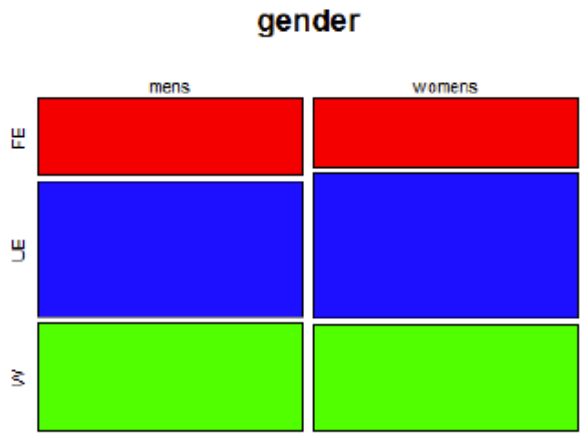
Slika 1: Porazdelitev tipa zadnjega udarca glede na to ali je žoga padla znotraj igrišča po dolžini ali ne.

Nadalje smo analizirali razdaljo leta žoge nam vrhom mreže (net.clearance). Distribucija podatkov je prikazana na sliki 2. Hitro lahko vidimo, kako imamo različne distribucije glede na končni tip udarca. Zanimivo je, da imamo tudi tu udarce, ki so klasificirani kot zaključni udarci in imajo negativno vrednost, kar naj bi pomenilo, da so pristali v mreži, kar je značilnost napak. Podobno kot pri pristanku žoge izven polja, je tudi tu lahko napaka v računalniškem vidu ali pa se lahko zgodi, da se je žogica zaletela v vrh mreže in se odbila čez mrežo v polje. Ne glede na razlog smo tudi tu se odločili, da teh točk ne bomo izločali ampak jih bomo obdržali.



Slika 2:Porazdelitev tipa zadnjega udarca glede na razdaljo leta žoge nad vrhom mreže v metrih

Nadalje smo analizirali kako se točke zaključujejo v moški in ženski konkurenci. Na sliki 3 lahko vidimo, da sta deleža ne glede na spol (gender) zelo podobna, kar je bilo morda malo presenečenje (predvsem) glede na različne hitrosti udarcev, med spoloma. To razumevanje podobne distribucije, nam je dalo zaupanje, da smo se odločili podatke o moških in ženskih točkah združiti. Na ta način smo nato lahko dobili višjo klasifikacijsko točnost, saj smo tako dobili na voljo več učnih podatkov za treniranje in pa predvsem optimizacijo parametrom modelov strojnega učenja.



Slika 3: Porazdelitev tipa zadnjega udarca glede na spol

3. KREIRANJE NOVIH SPREMENLJIVK

Gradnja novih spremenljivk je ključnega pomena pri izboljšanju napovedne točnosti prediktivnih modelov. V našem primeru smo naredili sedem sklopov kreiranja novih spremenljivk.

1. S kombinacijo spremenljivk *distance.from.sideline* in *outside.sideline* smo naredili novo zvezno spremenljivko, ki je bila pozitivna, če je žoga padla v polju in negativna, če je bila žoga v avtu. Na enak način smo združili tudi spremenljivki *depth* in *outside.baseline.*
2. Za spremenljivke, ki so bile vezane na razdaljo in so imele samo pozitivne vrednosti smo izračunali dodatno nove spremenljivke, ki so bile zvezne in so bile lahko tudi negativne. Na primer za *distance.from.center* smo izračunali kot 8,23m (širina igrišča) minus *distance.from.sideline* (popravljena razdalja vključno z negativnimi vrednostmi). Te spremenljivke same po sebi nimajo dodane prediktivne moči, vendar pa nam lahko pomagajo pri delitvi odločitvenega prostora, kar lahko v kombinaciji z drugimi spremenljivkami prinese izboljšanje.
3. Izračun novih spremenljivk glede na položaje igralcev in kote udarcev ter domensko znanje. Kreiranje teh spremenljivk je temeljilo na vednju kakšne lastnosti zadljega udarca gledajo ljudlje, ki notirajo te tipe točk. Tako smo izračunali 8 novih spremenljivk: pretečena razdalja igralca, oddaljenost udarca od odboja žoge, oddaljenost igralca od žoge v času udarca, hitrost žoge pred udarcem, čas, ki ga je imel igralec na voljo za udarec, hitrost teka pred udarcem, kakšen je bil kot udarca in pa pod kakšnim kotom je prišla žoga do igralca. Kot primer oddaljenost udarca od odboja žoge smo izračunali iz kombinacije spremenljivk *previous.distance.from.sideline, player.impact.depth, player.impact.distance.from.sideline* in *previous.depth*.
4. Izračun skoraj 1000 novih spremenljivk, ki smo jih dobili z izračunom vsote, razlike, zmnožka, delitve, povprečja in standardnih deviacij čez vse kombinacije dveh numeričnih spremenljivk. Ta pristop se izkaže za zelo uporabnega pri *boosted tree-based ensemble algorithms*, saj ta transformacija odločitvenega prostora omogoča algoritmu boljšo zaznamo intaktivnih in nelinearnih relacij med spremenljivkami.

5. Izračuna spremenljivk, ki nam je na podlagi sode ali lihe dolžine točke povedala ali je zadnji udarec izvedel server ali ne.
6. Vse kategorične spremenljivke smo paroma združili, da je algoritem lahko zaznal vsako možno interakcijo (zaporedje) med kategoričnimi spremenljivkami.
7. Odstranitev vseh spremenljivk, ki smo jih dobili po zgornjih postopkih in so imele nič ali pa zelo majhno varianco vrednosti, saj take spremenljivke niso uporabne pri strojnem učenju.

4. IZDELAVA KONČNEGA MODELA

Kot metrika za uspešnost predikcij se je na tekmovanju uporabilo »multi-class logloss« funkcijo, ki se izračuna kot vsota logaritma napake za vsak razred. To pomeni, da mora naš model vračati verjetnosti za vsakega od treh razredov in ne le podatka kateri razred je najbolj verjeten.

Končni model, ki smo ga uporabili za to tekmovanje je bil eXtreme Gradient Boosting algorithm (XGB) [1]. Algoritem je znan po svoji dokazani visoki učinkovitosti, hitrosti in fleksibilnosti zato se je tudi uporabil kar nekajkrat kot zmagovalni algoritem na Kaggle tekmovanjih [2, 3]. Glede na veliko število spremenljivk, ki smo jih imeli v učni in testni množici je bil velik poudarek namenjen optimizaciji hiperparametrov XGB algoritma.

Pri optimizaciji hiperparametrov je bil največji povdarek na optimizaciji parametrov *max_depth* in *min_child_weight.* Začeli smo z višjo vrednostjo prametra *learning_rate* in ko smo dobili približno optimalno vrednost izbranih parametrov smo znižali *learning_rate* in tako še dodatno izboljšali napovedno točnost. Na vsakem koraku optimizacije smo uporabili postopek prečnega preverjanja, da smo lahko dobljenim vrednostim parametrov lahko bolj zaupali. Postopno zmanjševanje parametra *learning_rate* je izboljšalo rezultate zaradi nelinearnosti vhodnjih spremenljivk in hkrati preprečilo, da bi se model preveč prilagodil (naučil) samo na učne podatke (*overfeeting)* ampak, da je ostal dovolj splošen, da se je odlično odnesel tudi na še nevidenih podatkih.

5. REZULTATI

Kot je običaj v spletnih tekmovanjih se rezultate napovedi objavi preko platforme (v našem primeru CrowdANALYTICS). Te napovedi se nato po skritem ključu razdeli in uporabi za prikaz na javni in privatni lestvici. V našem primeru se je 40% napovedi uporabilo za javno lestvico in 60% za privatno. Rezultat na javni lestvici je namenjen primerjavi kvalitete njegove rešitve napram rešitvam ostalih tekmovalcev. Rezultat na privatni lestvici pa se nato uporabi za določanje zmagovalca tekmovanja. Ker se rezultat na privatni lestvici vidi le enkrat ni mogoče, da bi se algoritem prilagodil tako, da bi imel čim boljše napovedi na privatni lestvici.

Z našim modelom smo dobili rezultat (»multi-class logloss«) na javni lestvici 0.179, kar je bilo dovolj za osmo mesto. Ja privatni lestvici pa smo dobili rezultat 0.188, kar je bilo dovolj za prvo mesto.

Za lažjo predstavo kvalitete napovedi smo izračunali tudi klasifikacijske točnosti. Klasifikacijska točnost modela preko vseh razredov je znašala 94.5%, kar je za tak problem visoka številka. Poleg skupne klasifikacijske točnosti pa smo izračunali tudi klasifikacijsko točnost za vsak razred posebej. Rezultati so predstavljeni v tabeli 2.

	Napovedani razredi		
Dejanski razredi	Zaključni udarec	Prisiljena napaka	Nepotrebna napaka
Zaključni udarec	**98.2%**	0.5%	1.3%
Prisiljena napaka	1.8%	**89.0%**	9.2%
Nepotrebna napaka	2.1%	3.2%	**94.6%**

Tabela 2: Klasifikacijske točnosti po razredih

Kot lahko vidimo model najslabše napoveduje prisiljene napake, kar je pričakovano, saj dejansko ne obstaja nekega (nepisanega) pravila kdaj je nek udarec prisiljena napaka, tako da je tu tudi največ šuma v podatkih.

6. ZAKLJUČEK

V referatu smo predstavi tekmovanje z naslovom *From AO to AI: Predicting How Points End in Tennis* na katerem sva avtorja dosegla prvo mesto. Opisali smo postopek priprave podatkov z generiranjem novih spremenljivk in uporabe modela z optimizacijo parametrov ter predstavili rezultate.

Izkušnje, ki smo jih pridobili tekom tekmovanja in bi morda bile uporabne tudi pri drugih podobnih tekmovanjih bi lahko strnili na sledeče točke:

- Če imamo dve učni množici, ki sta si zelo podobni, je smiselno množici združiti, saj tako dobimo več podatkov za učenje in posledično boljše rezultate.
- Če imamo klasifikacijski problem, ki ni primeren za reševanje z globokimi nevronskimi mrežami se za izhodišče lahko uporabi XGB saj se večinoma izkaže kot zelo dober algoritem.
- Optimizacija hiperparametrov XGB algoritma je ključnega pomena za izboljšanje rezultatov.
- Uporaba domenskega znanja za kreiranje novih spremenljivk izboljša rezultate, saj algoritem sam ne zna smiselno povezati spremenljivk. Kot primer lahko navedemo izračunano hitrost teka igralca, ki jo ljudje, ki označujejo točke, (podzavestno) uporabljajo pri ločevanju nepotrebne od izsiljene napake.

7. VIRI

[1] Chen, Tianqi, Tong He, and Michael Benesty. "Xgboost: extreme gradient boosting." R package version 0.4-2 (2015): 1-4.

[2] Mangal, Ankita, and Nishant Kumar. "Using big data to enhance the bosch production line performance: A Kaggle challenge." Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016.

[3] Sheridan, Robert P., et al. "Extreme gradient boosting as a method for quantitative structure–activity relationships." Journal of chemical information and modeling 56.12 (2016): 2353-2360.

Indeks avtorjev / Author index

Konferenca / Conference

Uredili / Edited by

Slovenska konferenca o umetni inteligenci / Slovenian Conference on Artificial Intelligence

Mitja Luštrek, Rok Piltaver, Matjaž Gams