**FURIA in Task Mining**

Petr Průcha 1 [0000-0003-2197-7825], Jan Skrbek1 [ORCID]

1 Faculty of Economics, Technical University of Liberec, Studentská 1402/2, 461 17 Liberec, Czech Republic; petr.prucha@tul.cz, jan.skrbek@tul.cz

**Abstract:**

Nullam felis quam, scelerisque ut consectetur sed, suscipit ac nisi. Praesent iaculis sollicitudin

mauris, id mollis velit. Phasellus rhoncus finibus sapien, id faucibus dolor finibus quis.

Integer tempus consequat arcu et gravida. Vivamus est felis, gravida malesuada auctor eget,

efficitur nec velit. Praesent egestas ullamcorper metus, id commodo nulla commodo vel. Nam

fringilla tortor sed pulvinar viverra. Aenean magna tortor, mattis nec gravida in, tristique vitae

enim. Vivamus dignissim mauris a lorem placerat porta. Praesent et lectus ut orci blandit

venenatis ut dignissim justo. Nullam pharetra id lacus eget imperdiet. Vivamus dapibus libero

sit amet tristique eleifend. Proin at accumsan erat. Ut nec sapien in risus tincidunt commodo

in nec sem.

**Keywords:** FURIA, Task-mining, RPA task-mining, RIPPER, Automatable routines

**Acknowledgement:** This paper was prepared thanks to Technical University of Liberec SGS grant with number: SGS-2021-1033

# Introduction

V dnešní době, kdy se mnoho organizací snaží minimalizovat náklady a zbavit zvyšujícího počtu administrativních úkonů, které je nutné plnit například z legislativních důvodů nebo protože jsou důležité pro chod organizace, je možné automatizovat mnoho rutinních a administativních úkolů pomocí současných technologií, jako je například propojení aplikací přes API nebo pomocí RPA (Robotic Process Automation) robotů. Tyto roboti jsou schopní jednoduše vykonávat rutinní činnosti stejně jako uživatel počítače. Tato technologie v poslední době získává mnoho pozornosti. Nicméně i tato technologie má svoje limity a jeden z problémů je výběr vhodné činnosti nebo procesu k automatizaci (Syed et al., 2020). Task-mining je podoblast process minigu, která se zaměřuje na nálezní vhodných procesů a úkolů k automatizaci. V oboru task-miningu drží současný stav poznání (Bosco et al., 2019; Leno, Augusto, et al., 2021). Jejich algoritmy jsou schopné automaticky identifikovat z UI logů činnosti, které by bylo možné automatizovat. Tento výzkum vychází z práce (Bosco et al., 2019), na které budou testovány další metody, které by mohly přinést lepší výsledky a tím vylepšit současný algoritmus.

Bosco et al., (2019) se snaží objevit deterministické procesy k automatizaci pomocí porovnání UI logů s předchozími a také pomocí algoritmů strojového učení jako je RIPPER od Cohen (1995) a Foofah od Jin & Anderson (2017). To, že pracují s deterministickými procesy, kde jsou stoprocentně schopní určit, jestli jsou činnosti automatizovatelné, velmi zužuje potenciální kandidáty k automatizaci. V tomto článku bude otestováno využití pravidel, které nejsou stoprocentně deterministická, ale splňují určitý práh, který lze považovat za dostatečný proto, aby se činnost pořád vyplatila automatizovat.

Tím, že nebudou využívána striktně deterministická pravidla rozšíříme okruh potencionálních kandidátů k automatizaci. Dále otestujeme přesnost pravidel získané rule learning algoritmem (RLA) FURIA oproti pravidlům získaných z RLA RIPPER. FURIA je novější algoritmus než RIPPER. Oba tyto algoritmy spadají do kategorie RLA algoritmů (Hühn & Hüllermeier, 2010). FURIA využívá fuzzy logiku pro hledání pravidel v datech, z toho důvodu se její použití jeví jako lepší RLA než RIPPER, který spolehlivě najde pravidla v datech, která obsahují deterministická pravidla.

Cíle tohoto článku jsou otestovat možnost využití ne striktně deterministických pravidel pro zjištění kandidátů k automatizaci a také srovnání algoritmu FURIA s RIPPER, jestli je predikce pravidel u ne striktně deterministických pravidel lepší.

# Approach

Rozšíření potenciálních kandidátů k automatizaci o kandidáty, kteří nejsou striktně determinističtí. V několika případech není nikdy možné najít pravidla, které by pokrývala všech sto procent případů. Bosco et al., (2019) využívá jen pravidla, které splní podmínku confidance == 1.0, to znamená, že jsou použita jen pravidla, která jsou deterministická. V RPA odvětví se běžně pracuje s procesy, které nejsou na celém procesu automatizované a využívají se takzvaní attended RPA roboti, kde RPA robotovi asistuje člověk (Leno, Polyvyanyy, et al., 2021; Soeny et al., 2021; Syed et al., 2020). Existuje více scénářů a způsobů, jak tato spolupráce probíhá, například člověk vykoná část procesu a následně činnosti dokončí robot, nebo další možnost je, že činnosti, které robot nedokáže zpracovat, tak robot případ označí a pokračuje dál nebo robot spadne do výjimky a nedokáže daný případ splnit. Označené případy a případy co skončili výjimkou následně dokončí člověk. Okruh kandidátů se dá jednoduše rozšířit změnou podmínky na confidance == 0.5[[1]](#footnote-1). Hodnota confidance se bude lišit v závislosti na oboru a počtu případu a celkové rentabilitě a dalších výhodách, které automatizace může přinést (Aguirre & Rodriguez, 2017; Syed et al., 2020; van der Aalst et al., 2018).

Další způsob, jak objevit delší rutiny pro automatizaci, také souvisí s confidance prahem. Protože pro některé případy procesu RIPPER nebo FURIA dokáže najít stoprocentní pravidlo, které platí ve všech případech pro část daného případu, nicméně pokud je část procesu jiná, nebo má neobvyklé hodnoty v předcházejících částech procesu, tak neprojde podmínkou a bude zamítnuta viz ukázka dat v tabulce 1[[2]](#footnote-2).

If Parameter 36 == C:/Customers/Australia/

Parameter 30 == Australia

Else:

Parameter 30 == Country169

Pravidlo na prvním a druhém řádku kódu na ukázkových datech bude platit v 100 % případů pro tento proces, nicméně na řádku 4 je vidět, že RIPPER nedokázal najít pravidlo, které by pokrylo zbytek v parametru 30 (Value), když se jedná o country a náhodné číslo. Proto tento parametr nikdy nebude mít confidance 1.0, ale bude nižší, a tím pádem nesplní podmínku pro přidání do souboru pravidel.

Výpočet a hledání pravidla proběhlo na platformě WEKA (Waikato Environment for Knowledge Analysis) ve verzi 3.8.5, která byla nainstalovaná na počítači s Windows 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter 28 | Parameter 29 | Parameter 30 | Parameter 31 | Parameter 36 |
| Action type | **Timestamp** | **Value** | **Label** | **Path** |
| insertValue | 2019-03-05T11:48:53.780+11:00 | Country169 | country | UnbindFile |
| insertValue | 2019-03-05T11:49:58.732+11:00 | Country183 | country | UnbindFile |
| insertValue | 2019-03-05T11:50:51.115+11:00 | Australia | country | C:/Customers/Australia/ |
| insertValue | 2019-03-05T11:51:53.245+11:00 | Country164 | country | UnbindFile |
| insertValue | 2019-03-05T11:52:53.824+11:00 | Country190 | country | UnbindFile |
| insertValue | 2019-03-05T11:53:28.975+11:00 | Australia | country | C:/Customers/Australia/ |
| insertValue | 2019-03-05T11:54:34.283+11:00 | Country87 | country | UnbindFile |
| insertValue | 2019-03-05T11:55:17.999+11:00 | Australia | country | C:/Customers/Australia/ |
| insertValue | 2019-03-05T11:56:42.956+11:00 | Country68 | country | UnbindFile |

**Tab. 1**. Ukázka dat – log6.csv. Source Transformováná data do CSV dle (Bosco et al., 2019).

Data v tabulce 1 byla převzata z výzkumu Bosco et al., (2019) a byla transformována z MXML do formátu CSV[[3]](#footnote-3), který je vyžadován programem WEKA. Dále tato data byla použita na srovnání FURIA a RIPPER. WEKA má v sobě integrované oba algoritmy. RIPPER se jmenuje (JRip) v platformě WEKA. Pro zjišťování pravidel využívá základní nastavení v programu WEKA.

**2.1 Working example**

Dále pro výzkum byla vytvořena data založená na reálném procesu, co byl automatizován pomocí RPA, nicméně v minulosti tuto činnosti vykonával zaměstnanec. Proto již není možné získat reálné UI logy, na kterých by mohl být proveden výzkum. Data byla vytvořena z datasetu od (Bosco et al., 2019) a 1985 Auto Imports Database od Schlimmer.

**Obrázek příkladu**

# Results

Objevená pravidla v datech log6.mxml algoritmem RIPPER byla jen dvě ve výchozím programu. Tato pravidla lze napsat takto:

If Parameter 30 == Australia:

Parameter 36 = C:/Customers/Australia/

If Parameter 30 != Australia:

Parameter 36 = UnbindFile

V originálním programu tato pravidla byla jako spouštějící podmínky pro automatizovatelné rutiny. RIPPER s těmito pravidly má přesnost 1.0.

FURIA na stejných datech objevila trochu jiná pravidla se stejným výsledkem a jsou tedy stejně přesná.

If Parameter 37 == Web:

Parameter 36 = UnbindFile

If Parameter 30 == Australia:

Parameter 36 = C:/Customers/Australia/

Přesnost FURIA je 1.0 stejně jako má RIPPER, takže pro tento typ dat má FURIA stejné výsledky jako RIPPER viz tabulka 2. Počet záznamů neboli instancí v datech byl 999 a oba algoritmy určili díky těmto pravidlům všechny správně.

|  |  |  |
| --- | --- | --- |
|  | **FURIA** | **RIPPER** |
| **Přesnost** | 100 % | 100 % |
| **Správně určené instance** | 999 | 999 |
| **Nesprávně určené instance** | 0 | 0 |

**Tab. 2.**Výsledky algoritmů na datech log6.mxml.Source Authors.

Jelikož pravidla jsou striktně deterministická ve všech datech, které Bosco et al., (2019) poskytnul, tak v tomto výzkumu jsou použita jiná data, aby bylo možné srovnat algoritmy i na jiných datech a jiném příkladu viz kapitola 2.1. Na datech z kapitoly 2.1 byly zjištěny tyto výsledky viz tabulka 3. Na těchto datech jsou pravidla algoritmu FURIA jasně přesnější než RIPPER. Pravidla pro tyto případy jsou k nalezení v Apendix A. Na těchto datech je vidět, že přesnost není ani u jednoho algoritmu 100 %, takže by v originální verzi nebylo nalezeno žádné pravidlo. FURIA určila správně 182 případů z 205 případů. Robot fungoval správně v 88,9 % a s touto přesností by doručoval správné výsledky, kdyby automatizace byla nasazena. RIPPER určil správně 164 případů z 205, takže přesnost je nižší oproti algoritmu FURIA, jen 80 %. Takže, aby v původním algoritmu mohla být využita tato pravidla, tak by se confidance level musel snížit na 0.889 potažmo minimálně na 0.8 včetně, tak by pravidla od RIPPER byla přidána do souboru pravidel.

|  |  |  |
| --- | --- | --- |
|  | **FURIA** | **RIPPER** |
| **Přesnost** | 88,9 % | 80 % |
| **Správně určené instance** | 182 | 164 |
| **Nesprávně určené instance** | 23 | 41 |

**Tab. 3.**Výsledek ukázková data. SourceAuthors.

# Discussion

Task mining je žhavé téma a až někdo představí univerzální algoritmus, který dokáže najít v již existujících datech procesy k automatizaci, bude tato technologie hojně využívána napříč organizacemi, které za to budou ochotny platit nemalé peníze. Jelikož se jedná o řešení lukrativního problému, tak se i v komerční sféře a se zkoumá univerzální algoritmus pro hledání rutin k automatizaci. Této problematice se věnují start-upy a speciálně velcí hráči v RPA automatizaci nebo process minugu jako je UiPath anebo Celonis. Bohužel jejich postupy task-minigu jsou vědecké komunitě nepřístupné a jsou střeženým obchodním tajemstvím.

Jeden z problémů task minugu jsou data a jejich kvalita (Leno et al., 2020). Z pohledu dat je problematické, že výše zmíněné metody využívají UI logy uživatele. UI logy jsou problematické především v tom, že nejsou ve většině společností standardně sledovány jako například procesní logy z ERP systémů. To vede k tomu, že zaměstnanci, jejichž aktivity, které provádí na počítači, jsou sledovány, mají špatný pocit z toho, že jejich aktivity jsou monitorovány. (Razaghpanah et al., 2018)

Při špatné komunikaci ze strany organizace to může vést k tomu, že zaměstnanec může záměrně sabotovat zaznamenávání logů. To vede k tomu, že kvalita dat je velmi špatná a příprava dat pro analýzu je náročná, někdy i nemožná. Nicméně datová kvalita je problematická i bez záměrné sabotáže, jelikož lidé jsou často vyrušováni při práci, dále stejné úkony provádí v jiném pořadí se stejným finálním výsledkem. Dále i jednoduché rozhodnutí, které člověk provádí, může být velmi těžké pro odhalení umělou inteligencí. (Leno, Augusto, et al., 2021)

Dalším důležitým faktorem je, že UI logy mají více parametrů než klasické procesní logy, které jsou v mnoha případech velmi strohé.

Zajímavý přístup pro zjištění automatable routines je využití textových popisů k procesu, které je možné zpracovat, a na základě nich identifikovat komplexitu procesu a jestli je vhodný pro automatizaci (Leopold et al., 2018). Jejich odhady, jestli je process automatizovatelný, jsou založeny na textových vstupech/datech a AI algoritmu, který je schopný určit automatizovatelnost procesu jen v určitém procentu, což je podobný přístup s nestriktními determistickými pravidly jako byl využit v našem výzkumu. Jinak většina přístupu k hledání rutin k automatizaci využívá UI logy (viz citace všech ostatních UI log článků.

Taky pro náš přístup je důležité zmínit, že výsledky RLA algoritmů se mohou lišit na základě vstupních dat, a každý algoritmus je silnější pro různá data a nelze říct, že FURIA je lepší v každém případě pro všechny druhy dat. citace Proto výsledky našeho výzkumu mají jisté limitace spojené s použitým datasetem. Proto je vhodné pro budoucí výzkum tento přístup otestovat i na dalších datech, která poskytnou další pohled na tento přístup.

**Bibliografie**

Aguirre, S., & Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. In J. C. Figueroa-García, E. R. López-Santana, J. L. Villa-Ramírez, & R. Ferro-Escobar (Eds.), *Applied Computer Sciences in Engineering* (Vol. 742, pp. 65–71). Springer International Publishing. https://doi.org/10.1007/978-3-319-66963-2\_7

Bosco, A., Augusto, A., Dumas, M., La Rosa, M., & Fortino, G. (2019). Discovering Automatable Routines from User Interaction Logs. In *Business Process Management Forum* (Vol. 360, pp. 144–162). Springer International Publishing. https://doi.org/10.1007/978-3-030-26643-1\_9

Cohen, W. W. (1995). Fast Effective Rule Induction. In *Machine Learning Proceedings 1995* (pp. 115–123). Elsevier. https://doi.org/10.1016/B978-1-55860-377-6.50023-2

Hühn, J. C., & Hüllermeier, E. (2010). An Analysis of the FURIA Algorithm for Fuzzy Rule Induction. In J. Koronacki, Z. W. Raś, S. T. Wierzchoń, & J. Kacprzyk (Eds.), *Advances in Machine Learning I* (Vol. 262, pp. 321–344). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-05177-7\_16

Jin, Z., & Anderson, M. R. (2017). *Software for Foofah: Transforming Data by Example* [Data set]. https://doi.org/10.1145/3218889

Leno, V., Augusto, A., Dumas, M., La Rosa, M., Maggi, F. M., & Polyvyanyy, A. (2021). Discovering data transfer routines from user interaction logs. *Information Systems*, 101916. https://doi.org/10.1016/j.is.2021.101916

Leno, V., Augusto, A., Dumas, M., Rosa, M. L., Maggi, F., & Polyvyanyy, A. (2020). *Identifying candidate routines for Robotic Process Automation from unsegmented UI logs*.

Leno, V., Polyvyanyy, A., Dumas, M., La Rosa, M., & Maggi, F. M. (2021). Robotic Process Mining: Vision and Challenges. *Business & Information Systems Engineering*, *63*(3), 301–314. https://doi.org/10.1007/s12599-020-00641-4

Leopold, H., van der Aa, H., & Reijers, H. A. (2018). Identifying Candidate Tasks for Robotic Process Automation in Textual Process Descriptions. In J. Gulden, I. Reinhartz-Berger, R. Schmidt, S. Guerreiro, W. Guédria, & P. Bera (Eds.), *Enterprise, Business-Process and Information Systems Modeling* (Vol. 318, pp. 67–81). Springer International Publishing. https://doi.org/10.1007/978-3-319-91704-7\_5

Razaghpanah, A., Nithyanand, R., Vallina-Rodriguez, N., Sundaresan, S., Allman, M., Kreibich, C., & Gill, P. (2018, February 18). *Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem*. The 25th Annual Network and Distributed System Security Symposium (NDSS 2018). https://dspace.networks.imdea.org/handle/20.500.12761/507

Soeny, K., Pandey, G., Gupta, U., Trivedi, A., Gupta, M., & Agarwal, G. (2021). Attended robotic process automation of prescriptions’ digitization. *Smart Health*, *20*, 100189. https://doi.org/10.1016/j.smhl.2021.100189

Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J. J., Ouyang, C., ter Hofstede, A. H. M., van de Weerd, I., Wynn, M. T., & Reijers, H. A. (2020). Robotic Process Automation: Contemporary themes and challenges. *Computers in Industry*, *115*, 103162. https://doi.org/10.1016/j.compind.2019.103162

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, *60*(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

**Appendix A**

**FURIA rules:**

(Parameter11 = two) and (Parameter1 in [70, 73, inf, inf]) => Parameter37=Sent email (CF = 0.98)

(Parameter16 in [-inf, -inf, 1732, 1867]) and (Parameter15 in [-inf, -inf, 953, 963]) and (Parameter7 in [102, 103, inf, inf]) => Parameter37=Sent email (CF = 0.98)

(Parameter11 = two) and (Parameter31 in [-inf, -inf, 16500, 20970]) and (Parameter25 in [-inf, -inf, 347, 358]) => Parameter37=Sent email (CF = 0.99)

(Parameter25 in [-inf, -inf, 34, 219]) and (Parameter7 in [-inf, -inf, 164, 192]) => Parameter37=Sent email (CF = 0.91)

(Parameter8 = saab) => Parameter37=Sent email (CF = 0.89)

(Parameter18 in [-inf, -inf, 516, 528]) and (Parameter25 in [335, 346, inf, inf]) => Parameter37=Sent email (CF = 0.88)

(Parameter11 = four) and (Parameter24 in [319, 327, inf, inf]) and (Parameter7 in [-inf, -inf, 103, 104]) => Parameter37=- (CF = 0.97)

(Parameter11 = four) and (Parameter13 = rwd) => Parameter37=- (CF = 0.95)

(Parameter15 in [945, 957, inf, inf]) and (Parameter17 in [-inf, -inf, 652, 654]) and (Parameter11 = four) => Parameter37=- (CF = 0.91)

(Parameter16 in [1736, 1768, inf, inf]) and (Parameter18 in [-inf, -inf, 555, 557]) and (Parameter18 in [548, 549, inf, inf]) => Parameter37=- (CF = 0.9)

**RIPPER rules:**

(Parameter11 = four) and (Parameter24 >= 327) and (Parameter7 <= 103) => Parameter37=- (31.0/0.0)

(Parameter11 = four) and (Parameter13 = rwd) => Parameter37=- (25.0/1.0)

(Parameter7 <= 115) and (Parameter25 >= 327) and (Parameter17 >= 639) => Parameter37=- (14.0/0.0)

(Parameter11 = four) and (Parameter7 <= 91) => Parameter37=- (11.0/1.0)

(Parameter16 >= 1778) and (Parameter18 <= 555) and (Parameter12 = sedan) => Parameter37=- (6.0/0.0)

=> Parameter37=Sent email (118.0/7.0)

1. Changed code from Bosco et al., (2019) can be found on GitHub: [↑](#footnote-ref-1)
2. Data is possible download here: [↑](#footnote-ref-2)
3. Program for transformation is possible download here: [↑](#footnote-ref-3)