**Use of FURIA for Improving Method of Task-Mining**

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

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:**

Firmy, které využívají robotickou procesní automatizaci RPA, řeší velmi často problematiku výběru vhodného procesu k automatizaci. Manuální výběr vhodného procesu je velmi časově náročný. Proto se část oboru process miningu se specializovala na oblast výběru vhodných procesů k automatizaci na základě procesních dat. Této specializaci se velmi často označuje jako task-mining. Tato práce se zabývá vylepšením již existující metody task-miningu za pomoci odstranění limitujících omezení současné metody a také rule learning algoritmu FURIA pro detekci pravidel.

**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 se mnoho organizací snaží minimalizovat náklady a zbavit se zvyšujícího počtu administrativních úkonů. Některé úkony je nutné plnit například z legislativních důvodů nebo protože jsou důležité pro chod organizace. Rutinní a administrativních úkoly je možné automatizovat pomocí současných technologií, jako je například propojení aplikací přes API nebo pomocí technologie RPA (Robotic Process Automation). RPA roboti jsou schopní jednoduše vykonávat rutinní činnosti stejně jako uživatel počítače. RPA technologie v poslední době získává mnoho pozornosti. Nicméně i RPA technologie má svoje limity a jeden z problémů je výběr vhodné činnosti nebo procesu k automatizaci takzvaní task-mining (Syed et al., 2020).

Task-mining je podoblast process miningu, která se zaměřuje na nálezní vhodných procesů a úkolů k automatizaci. Srovnání přístupu task-miningu je popsáno v tab 1, kde jsou popsány použité postupy, použitá data a autoři práce. Většina autorů využívá UI logy pro výběr kandidátů k automatizaci.

Tento výzkum vychází z práce Bosco et al., (2019) a následně navazuje na jeho práci. V tomto výzkumu budou představeny a otestovány další metody, které by mohly přinést lepší výsledky a tím vylepšit současný algoritmus od Bosco et al., (2019).

Bosco et al., (2019) se snaží objevit deterministické procesy k automatizaci pomocí porovnání UI logů s předchozími logy 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. Ve výzkumu tento předpoklad upravujeme v návaznosti na poznatky dalších vědců a také praktických poznatky. 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 (J. C. Hühn & Hüllermeier, 2010). FURIA využívá fuzzy logiku pro hledání pravidel v datech, z toho důvodu se její použití jeví jako lepší RLA než RIPPER.

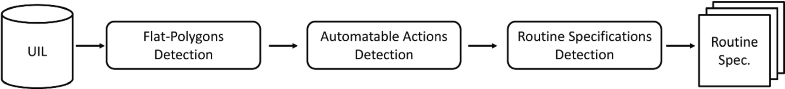
Cíle tohoto článku jsou otestovat možnost využití ne striktně deterministických pravidel pro zjištění kandidátů k automatizaci a také srovnání algoritmu FURIA s RIPPER.

|  |  |  |  |
| --- | --- | --- | --- |
| **Použité postupy** | **Použitá data/příklad, kde je princip využit** | **Jsou vybraní kandidáti striktně determinističtí?** | **Autoři** |
| Využívají striktně deterministická pravidla. Pro objevení pravidel využívají algoritmus Foofah (Jin & Anderson, 2017). | UI logs – Snaží se objevit pravidla, pro transformaci dat mezi excelovýma tabulkami. | Ano | (Leno, Augusto, et al., 2021a) |
| Snaží se objevit vzorec v datech, který má největší podobnost a na základě toho vygenerují rutiny pro automatizaci.  Využívají PM4PY a a-priori. | UI logs – Přesun dat z excelové tabulky do webového formuláře. | Ne | (Agostinelli et al., 2020) |
| Představují metodu, jak vybrat vhodné kandidáty k automatizaci založené na vizualizaci procesu a představených vzorcích. | UI logs – Stažení dat z ERP do excelu a provedení transformace dat v excelu. | Ne | (Choi et al., 2021) |
| Představují metody pro ohodnocení kandidátů k automatizaci na základě vybraných faktorů, které jsou hodnoceny pomocí známek. Výsledné známky slouží jako hodnotící faktor vhodnosti. | Procesní logy (není přesně specifikováno) – Testováno na 9 různých příkladech. | Ne | (Viehhauser & Doerr, 2021) |
| Objevování rutin a sekvencí, které jsou striktně deterministické. Využití algoritmů RIPPER, FOOFAH. | UI logs – Přepisování vstupních dat ze studijního oddělení do webového formuláře a excelu. | Ano | (Bosco et al., 2019) |
| Vybrání a objevování rutin na základě CloFast algoritmu (Fumarola et al., 2016). Algoritmus využívají k nalezení podobných sekvencí, které splňují určitý práh. Sekvence následně hodnotí na základě určitých kritérií. | UI logs – příklad založen na datech z webového formuláře a excelové tabulky. | Ano | (Leno, Augusto, et al., 2021b) |
| Využití AI pro zpracování jazyka takzvané NPL na základě textových vstupů předpovídají, jestli je činnost automatizovatelná. | Procesní logy – 47 různých procesů z 10 různých zdrojů | Ne | (Leopold et al., 2018) |

**Tab. 1**. Srovnání přístupů k task-miningu. Source: Authors

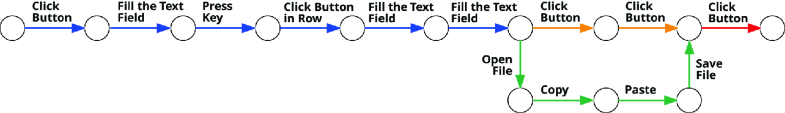
## Bosco approach

Jak už v úvodu bylo naznačeno, tento výzkum rozšiřuje práci autorů Bosco et al., (2019) a v této části je popsán detailněji přístup, který využívají. Jejich přístup je popsán viz fig. 1., kde v první fázi mají setříděné UI logy z procesu. Následně z UI logů detekují flat-polygons pomocí metody DAFSA (Deterministic Acyclic Finite State Automaton) – což mapuje všechny procesní cesty, které se v procesu nachází. Výstup z části flat-polygon dection je vidět ve fig. 2. Po zjištění všech procesních cest následuje detekce, jestli daná akce je automatizovatelná včetně všech parametrů, které daná akce má. Každá akce více parametrů (minimálně jeden) v závislosti na dané uživatelské činnosti a využívané aplikaci. Z pohledu autorů je akce automatizovatelná pokud se v každém jejím případě dají určit všechny její parametry, v takovém případě můžeme akci považovat za deterministickou. Ke zjištění parametrů využívají 3 přístupy. První přístup je založen na opakování parametrů akce v průběhu procesu. To znamená, že parametr akce je vždycky stejný ve všech případech. Pro složitější případy, které nelze určit první postupem, využívají autoři dva algoritmy. Jeden pro nalezení pravidel v datech s názvem RIPPER. Druhý algoritmus se nazývá FOOFAH a využívá se pro nalezení pravidel k transformaci dat. Je využívaný především v souvislosti s tabulkovými procesory.



**Fig. 1.** Bosco approach to discover automatable routines. Source (Bosco et al., 2019).

Čtvrtá část postupu spočívá v doplnění informací k akcím a získání dodatečných informací pro vytvoření specifikace k rutině kandidáta. Ve čtvrté části algoritmu jsou akce spojeny do navazujících rutin. K těmto rutinám jsou následně hledány pravidla pro jejich spuštění. K hledání pravidel využívají algoritmus RIPPER. Pokud není pravidlo nalezeno pomocí RIPPERu, tak vytvoří triviální podmínku tím, že zkrátí rutinu o první akci, která bude tím pádem v daném případě spouštěčem.



**Fig. 2**. Working example in Bosco article. Source (Bosco et al., 2019)

### Bosco working example

Příklad, na kterém Bosco et al., (2019) testuje svůj přístup, je vidět na Fig. 2., který je založen na práci studijního oddělení na univerzitě v Melbourne. Kde pracovníci studijního oddělení aktualizují informace o studentech. První část (modrá) je vždycky stejná, student přijde na studijní oddělení, student potvrdí své informace a pracovníci vyplní informace do informačního systému. Následně pokud je student australské národnosti, tak ještě potvrdí svoji adresu, kterou studijní oddělení zálohuje do excelu (zelená varianta). Pokud je jiné národnosti, tak pracovníci zaškrtnou v informačním systému volbu zahraniční student (oranžová varianta). Poslední část (červená) procesu je jen ukončení vyplňování kliknutím na tlačítko.

## Approach

Jak je naznačeno v úvodu článku, tak Bosco et al., (2019) přístup je založen na striktně deterministických pravidlech. Podle jeho interpretace je možné danou činnost automatizovat jen v případě striktně deterministického procesu. Na základě těchto kritériích jsou vybíraní kandidáti od Bosco et al., (2019). Tento přístup nicméně velmi omezuje výběr potenciálních kandidátů k automatizaci. Především pokud pravidla musí být objeveny algoritmem, protože v mnoha případech, jak vyplývá z rozhovorů s RPA experty a i z vědecký článků v této oblasti, RPA roboti často pracují s aplikacemi, které fungují jako „black-box“. To znamená, že uživatelé aplikací sami nevědí, podle jakých pravidel aplikace generuje výsledky. Toto je typické pro legacy systémy, které bývají velmi často automatizovány pomocí RPA. Další důvod, proč v našem přístupu nebudeme využívat jen striktně deterministická pravidla, je, že se dost často automatizují jen určité části procesu a zbytek procesu obstarává člověk. 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. 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, označí a pokračuje dál, nebo robot spadne do výjimky v případě, kdy nedokáže daný případ splnit. Označené případy a případy co skončili výjimkou následně dokončí člověk. (Leno, Polyvyanyy, et al., 2021; Soeny et al., 2021; Syed et al., 2020)

Na výše uvedeném příkladu od Bosco et al., (2019) je možné ukázat, jak je využívání striktně deterministických procesů omezující. Příklad je možné ukázat na parametru 30, který je v tab 2. Parametr 30 má hodnoty Australia, nebo country s náhodným číslem. Pravidlo objevené algoritmem RIPPER je přepsáno do pseudokódu pro lepší pochopní, originální pravidlo je možné najít v apendix A.

If Parameter 8 == ID1103

Parameter 30 = Country5

Else:

Parameter 30 = Australia

Algoritmus RIPPER byl schopný tímto pravidlem určit přibližně 50 % případů správně. Z čehož lze předpokládat, že 50 % všech případů lze automatizovat. I když v tomto případě pravidlo, které našel RIPPER, není vhodné, jelikož toto určení 50 % je velmi náhodné. Lepší pravidla vygeneroval algoritmus FURIA. Originální pravidlo je možné najít v appendix A, pravidlo bylo přepsáno do pseudokódu. Přesnost pravidel algoritmu FURIA je 50 %.

If paramenter36 == C:/Customers/Australia/

Parameter30 = Australia

If Parameter8 == ID1103

Parameter30 = Country5

If Parameter8 == ID396

Parameter30 = Country140

Na těchto příkladech je ukázáno, že striktně deterministická pravidla, která splní podmínku confidance == 1.0 z Bosco et al., (2019) jsou omezující. Především proto některé přístupy, které jsou v tab. 1., nevyužívají pro výběr vhodných kandidátů striktně deterministická pravidla.

Pro přístup, který využívá Bosco et al., (2019) by se dal okruh kandidátů jednoduše rozšířit změnou podmínky například na confidance == 0.5. Hodnota confidance se bude lišit v závislosti na oboru, počtu případu, celkové rentabilitě a dalších výhodách, které automatizace může přinést viz (Aguirre & Rodriguez, 2017; Syed et al., 2020; van der Aalst et al., 2018).

V našem přístupu rozšíříme okruh potenciálních kandidátů k automatizaci o kandidáty, kteří nejsou striktně determinističtí, jelikož předpokládáme, že alespoň část procesu může být automatizována, i když neznáme pravidla, která by splňovala 100 % všech případů.

# Metodologie

Tento výzkum, jak již bylo napsáno, navazuje na postupy Bosco et al., (2019), kde je pomocí jiných metod zjištěno, jestli je možné vylepšit současný přístup. V tomto výzkumu se zaměřujeme jen na část postupu od Bosco et al., (2019) speciálně na klíčovou část celého přístupu, a to detekci akcí k automatizaci popsanou v kapitole 1.1. Tato část je klíčová z důvodů, že mnoho akcí je zde určeno jako neautomatizovatelné a to z důvodu nenalezení pravidel pro stoprocentní určení parametrů dané akce. Pro tento výzkum toto omezení odstraníme, jelikož jak vyplívá z literární rešerše a z komunikace s RPA specialisty, tak mnoho procesů je jen částečně automatizovaná a jen pro některé případy. V tomto výzkumu pracujeme s předpokladem, že akce jejíž parametry, které nelze stoprocentně určit, budeme i nadále považovat za automatizovatelnou, z důvodu že RIPPER nebo jiný RLA algoritmus dokáže správně určit některé parametry například v 50 % případů.

V tomto výzkumu otestujeme RLA algoritmus FURIA na příkladu a datech, které jsou zmíněně v kapitole 1.1. Cílem výzkumu je srovnat, jestli je RLA FURIA lepší než RLA RIPPER.

**Research questions:**

*Je algoritmus FURIA lepší než RIPPER při hledání pravidel v UI logs?*

Pro zodpovězení této otázky využijeme UI logs od log6.mxlm (Bosco et al., 2019) a UI logs představené v kapitole 2.1. Pro výpočet a hledání pravidel byla využita platforma WEKA (Waikato Environment for Knowledge Analysis) ve verzi 3.8.5, která byla nainstalovaná na počítači s Windows 10. Platforma WEKA v sobě integruje oba RLA algoritmy jak RIPPER (jRip) tak i FURIA. Pro zjišťování pravidel je využíváno základní nastavení v programu WEKA.

Data[[1]](#footnote-1) log6.mxlm byla převedena pomocí tohoto programu[[2]](#footnote-2) do formátu CSV a jsou k nalezení v repositáři pod názvem Log6Result.csv. Ukázka transformovaných dat s názvem Log6Result.csv je v tab. 2. Akce jsou rozloženy do parametrů a jsou ve stejné posloupnosti jako v původním formátu. Každý jeden řádek udává jeden procesní záznam, který je tvořen více akcemi. Tento formát/přístup je používán v celém výzkumu kvůli zjednodušení a pracuje se vždy jen s rozloženými akcemi na parametry.

Pro data Log6Result.csv bude testován především parametr 36, jelikož z tohoto parametru vychází jediné pravidlo, které bylo objeveno původním algoritmem. Dále budou testovány parametry, které neobsahují časový údaj (timestamp), anebo nemají shodné hodnoty v celém sloupci. Poslední kritérium, je že sloupec musí obsahovat hodnoty ve všech řádcích, protože pro tyto výsledky pro tyto zbývající parametry by byly jednoznačně určeny na 100 % krom hodnoty pro Timestamp a parametr 51. Všechny otestované parametry jsou vidět v tab. 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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. 2**. Ukázka dat – log6.csv. Source Transformovaná data do CSV dle (Bosco et al., 2019).

U datasetu, který je popsán v kapitole 2.1., jsou data již ve formátu CSV, takže žádná další transformace není nutná. U těchto dat má největší hodnotu testovat parametr 37, jelikož zbylé parametry jsou shodné, anebo se jedná o vstupní data viz parametr 7 až parametr 31.

**2.1 Working example**

Dále pro výzkum byla vytvořena data založená na reálném procesu, co byl již automatizován pomocí RPA, nicméně v minulosti tuto činnosti vykonával zaměstnanec. Proto již není možné získat reálné UI logy, na kterých by bylo možné provést výzkum. Data[[3]](#footnote-3) byla vytvořena z datasetu od (Bosco et al., 2019) a 1985 Auto Imports Database od Schlimmer. Jednalo se o proces, kdy zaměstnanec otevřel excelový formulář, načetl a zkopíroval data do webové aplikace, která data zpracovala (jako black-box), zobrazila výsledek a uložila data a podle výsledku webová aplikace odeslala email. Poté se připravila na zadání nových dat, nebo se ukončila již v předcházejícím kroku. Vizualizovaní proces viz fig. 3.

**Graphical user interface, text, application

Description automatically generatedFig. 3.** Working example of process for our data. Source: authors

# Results

Zjištěné výsledky při testování parametru 36 na datech s názvem Log6Result.csv jsou vidět v tab. 3. Přesnost algoritmu FURIA a RIPPER je 100 %, takže pro tento typ dat má FURIA stejné výsledky jako RIPPER. Počet záznamů neboli instancí je v datech 999 a oba algoritmy určili všechny instance správně, díky pravidlům uvedeným níže.

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

**Tab. 3.**Výsledky algoritmů na datech log6.mxml.Source Authors.

Objevená pravidla v datech Log6Result.csv algoritmem RIPPER byla jen dvě, a ta byla objevena i výchozím programem. Tato pravidla lze napsat takto:

If Parameter 30 == Australia:

Parameter 36 = C:/Customers/Australia/

If Parameter 30 != Australia:

Parameter 36 = UnbindFile

FURIA na stejných datech objevila trochu jiná pravidla se totožním výsledkem a to znamená, že algoritmy pro tento druh dat jsou identicky přesné.

If Parameter 37 == Web:

Parameter 36 = UnbindFile

If Parameter 30 == Australia:

Parameter 36 = C:/Customers/Australia/

V tab. 4 jsou výsledky přesnosti všech pravidel, které dokázali algoritmy najít pro daný parametr. Ve většině případů je přesnost algoritmů totožná až na parameter42, kde je FURIA o 0.7 % přesnější než RIPPER. FURIA dokázala určit o 7 případů více než RIPPER správně. Všechny tyto výsledky jsou zaokrouhlené s přesností na jedno desetinné místo.

|  |  |  |
| --- | --- | --- |
|  | **FURIA** | **RIPPER** |
| **Parameter8** | 8.3 % | 8.3 % |
| **Parameter19** | 8.3 % | 8.3 % |
| **Parameter25** | 0 % | 0 % |
| **Parameter30** | 49.8 % | 49.8 % |
| **Parameter36** | 100 % | 100 % |
| **Parameter37** | 100 % | 100 % |
| **Parameter38** | 100 % | 100 % |
| **Parameter39** | 100 % | 100 % |
| **Parameter40** | 49.8 % | 49.8 % |
| **Parameter41** | 50.2 % | 50.2 % |
| **Parameter42** | 51.4 % | 50.7 % |
| **Parameter43** | 100 % | 100 % |
| **Parameter44** | 52.3 % | 52.3 % |
| **Parameter45** | 49.8 % | 49.8 % |
| **Parameter46** | 100 % | 100 % |
| **Parameter47** | 100 % | 100 % |
| **Parameter48** | 50.2 % | 50.2 % |
| **Parameter49** | 100 % | 100 % |
| **Parameter50** | 49.8 % | 49.8 % |

**Tab. 4.** Výsledky srovnání RIPPER a FURIA. Source Authors.

Na datech z kapitoly 2.2 s názvem Auto2Mail.csv byly zjištěny tyto výsledky viz tab. 5.

|  |  |  |
| --- | --- | --- |
|  | **FURIA** | **RIPPER** |
| **Přesnost** | 88,8 % | 80 % |
| **Správně určené instance** | 182 | 164 |
| **Nesprávně určené instance** | 23 | 41 |

**Tab. 5.**Výsledek ukázková data. SourceAuthors.

Na těchto datech jsou pravidla algoritmu FURIA jasně přesnější než RIPPER. Pravidla pro tyto případy jsou k nalezení v Apendix A. FURIA určila správně 182 případů z 205 případů, to znamená přesnost 88,8 %. RIPPER určil správně 164 případů z 205, takže přesnost je 80 %. Na těchto datech je tedy algoritmus FURIA přesnější než RIPPER. V tab. 5 jsou výsledky zaokrouhleny na jedno desetinné místo.

# Discussion

Task-mining je žhavé téma a v okamžiku kdy 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 výzkumníci v komerční sféře snaží najít univerzální algoritmus pro hledání rutin k automatizaci. Této problematice se věnují start-upy a hlavně velcí hráči v RPA automatizaci nebo process miningu jako je UiPath anebo Celonis. Bohužel jejich postupy task-miningu jsou vědecké komunitě nepřístupné a jsou střeženým obchodním tajemstvím.

Jeden z problémů task-miningu jsou data a jejich kvalita (Leno et al., 2020). Z pohledu dat je problematické, že výše zmíněné metody využívají UI logy uživatele. U UI logů je problém 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 vede k tomu, že zaměstnanec může záměrně sabotovat zaznamenávání logů. Kvalita dat je pak 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 vyrušovaní 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., 2021b)

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 proces automatizovatelný, jsou založeny na textových vstupech/datech a AI algoritmu.

Taky pro náš přístup je důležité zmínit, že výsledky RLA algoritmů se mohou lišit na základě vstupních dat, a každý algoritmus je jinak přesný pro různá data a nelze říct, že FURIA je lepší v každém případě pro všechny druhy dat (J. Hühn & Hüllermeier, 2009; Manghai & Jegadeeshwaran, 2019). 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 komplexnější pohled na tento přístup.

## Conclusion

## Výzkum ukázal, že využívání striktně deterministických pravidel je omezující a limituje to počet kandidátů k automatizaci. Reálný příklad a přístupy dalších odborníků potvrzují, že je možné automatizovat procesy jen pro určité případy, že pravidla procesu nemusí objevit žádný algoritmus a že je vhodné rozšířit počet možných kandidátů k automatizaci. Dále se ukázalo, že RLA algoritmus FURIA je lepší než RIPPER na testovacích datech, nicméně jak zmiňují další odborníci značně záleží na datech. Dále FURIA dokázala na testovacích datech najít pravidla, která jsou vhodnější pro RPA automatizaci. Z tohoto výzkumu tedy vyplývá, že rozšíření potenciálních kandidátů o nestriktně deterministické a využití algoritmu FURIA je přínosné.

**Bibliografie**

Agostinelli, S., Lupia, M., Marrella, A., & Mecella, M. (2020). Automated Generation of Executable RPA Scripts from User Interface Logs. In A. Asatiani, J. M. García, N. Helander, A. Jiménez-Ramírez, A. Koschmider, J. Mendling, G. Meroni, & H. A. Reijers (Eds.), *Business Process Management: Blockchain and Robotic Process Automation Forum* (Vol. 393, pp. 116–131). Springer International Publishing. https://doi.org/10.1007/978-3-030-58779-6\_8

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

Choi, D., R’bigui, H., & Cho, C. (2021). Candidate Digital Tasks Selection Methodology for Automation with Robotic Process Automation. *Sustainability*, *13*(16), 8980. https://doi.org/10.3390/su13168980

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

Fumarola, F., Lanotte, P. F., Ceci, M., & Malerba, D. (2016). CloFAST: Closed sequential pattern mining using sparse and vertical id-lists. *Knowledge and Information Systems*, *48*(2), 429–463. https://doi.org/10.1007/s10115-015-0884-x

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

Hühn, J., & Hüllermeier, E. (2009). FURIA: An algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, *19*(3), 293–319. https://doi.org/10.1007/s10618-009-0131-8

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. (2021a). Discovering executable routine specifications from user interaction logs. *ArXiv:2106.13446 [Cs]*. http://arxiv.org/abs/2106.13446

Leno, V., Augusto, A., Dumas, M., La Rosa, M., Maggi, F. M., & Polyvyanyy, A. (2021b). 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

Manghai, A. & Jegadeeshwaran. (2019). *Application of FURIA for Finding the Faults in a Hydraulic Brake System Using a Vibration Analysis through a Machine Learning Approach*. *20*, 9. http://www.phmsociety.org/node/2614

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

Viehhauser, J., & Doerr, M. (2021). Digging for Gold in RPA Projects – A Quantifiable Method to Identify and Prioritize Suitable RPA Process Candidates. In M. La Rosa, S. Sadiq, & E. Teniente (Eds.), *Advanced Information Systems Engineering* (Vol. 12751, pp. 313–327). Springer International Publishing. https://doi.org/10.1007/978-3-030-79382-1\_19

**Appendix A**

**JRIP rules:**

(Parameter8 = ID1103) => Parameter30=Country5 (2.0/0.0)

=> Parameter30=Australia (997.0/499.0)

**FURIA rules:**

(Parameter36 = C:/Customers/Australia/) => Parameter30=Australia (CF = 1.0)

(Parameter8 = ID1103) => Parameter30=Country5 (CF = 0.5)

(Parameter8 = ID396) => Parameter30=Country140 (CF = 0.4)

**Appendix B**

**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. Data is possible download here: https://github.com/Scherifow/SGS-Task-Mining [↑](#footnote-ref-1)
2. Program for transformation is possible download here: https://github.com/Scherifow/SGS-Task-Mining/tree/main/mxml2csv-Transformation [↑](#footnote-ref-2)
3. Data is possible to find in repository: https://github.com/Scherifow/SGS-Task-Mining [↑](#footnote-ref-3)