Adaptive Rule Monitoring System

Alireza Tabebordbar
University of New South Wales
Sydney, Australia
alirezat@cse.unsw.edu.au

Amin Beheshti
Macquarie University and
University of New South Wales
Sydney, Australia
amin.beheshti@mq.edu.au

ABSTRACT

Rule-based techniques are gaining importance with their ability to augment large scale data processing systems. However, there still remain key challenges amongst current rule-based techniques, including rule monitoring, adapting and evaluation. Among these challenges, monitoring the precision of rules is highly important as it enables analysts to maintain the accuracy of a rule-based system. In this paper, we propose an Adaptive Rule Monitoring System (ARMS) for monitoring the precision of rules. The approach employs a combination of machine learning and crowdsourcing techniques. ARMS identifies rules deteriorating the performance of a rule based system, using the feedback receives from the crowd. To enable analysts identifying the imprecise rules, ARMS leverage machine learning algorithms to analyze the crowd's feedback. The evaluation results show that ARMS can identify the imprecise rules more successfully compared to the default practice of the system, which rely exclusively on analysts.

CCS CONCEPTS

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

KEYWORDS

Rule monitoring system, Multi-armed-bandit algorithm, Rule based systems

ACM Reference format:
Alireza Tabebordbar and Amin Beheshti. 2018. Adaptive Rule Monitoring System. In *Proceedings of SE4COG'18:IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services , Gothenburg, Sweden, May 28–29, 2018 (SE4COG'18),* 7 pages.
DOI: 10.1145/3195555.3195564

1 INTRODUCTION

The continues improvement in capabilities of data processing systems made data available from variety of sources; including web, social media, private databases and etc. This availability of data intensified the competition to increase the productivity and added value. Data curation is a critical technique in large scale data processing systems for extracting knowledge and breathing insight into raw data. However, a large number of researches on data curation rely on algorithmic approaches. For example, algorithms can be used to extract named entities from tweets (e.g. "ISIS" and "Palmyra" in "There are 1800 ISIS terrorists in Palmyra, only 300 are Syrians"), link them to entities in a knowledge base (e.g. [9, 11]), and classify tweets into a set of predefined categories (e.g., using Naive Bayes classifier). However, algorithmic approaches require large enough training data for training classifiers, which may not be available in some cases. In particular, using algorithmic approaches alone for data processing systems have various limitations [19]: (1) Algorithms are complex and difficult to interpret; (2) Algorithms are context based and can not be easily adapted to work in another context; and (3) besides the performance of machine-learning algorithms depend on the quality of training data, which is not available in some situations.

On the other hand rule-based techniques have used in different fields [1, 8, 12–14, 20]; including, information extraction, entity matching, knowledge base construction, and data classification. In particular, rule based systems have gained importance to augment algorithmic approaches in cases algorithms are not working well [26]. Rule-based techniques can provide a declarative supplement for dealing with many of the shortcomings inherent in the algorithmic approaches. Namely, algorithms are good for solving concrete tasks, yet in reality datasets are typically large and evolving over time. Using rule-based techniques have several benefits such as [19]: (1) writing rules are easier and faster than designing and training classifiers; (2) correcting mistake for rules are easier than algorithms; (3) rules can consider cases that learning algorithms cannot cover. In many cases, it is not possible to collect enough data for training classifiers. Instead, rules can solve this problem.

For example, fundamentalists ('ISIS', 'Al-Qaeda') cheat the learning based classifiers by changing their keywords and hashtags frequently. Accordingly, gathering training data for learning algorithms to capture changes will be time consuming and expensive, instead, analysts can solve this problem by adding rules, (e.g. Tweet from 'Syria'; AND tweet-owner greater than 1000 followers; AND the Tweet contains 'recruiting': implies classify as 'Need Investigations'). In this context, monitoring the precision of rules in a rule-based system is of prime importance. Typically, rules that were curated by analysts work well up to a certain time. This variability, on the quality of rules, effect on the overall accuracy of a rule-based system. Therefore, rules need to be monitored and adapted over time. Existing methods typically involves generating a validation set (a set of items manually labeled with the correct items types) to estimate the precision of a single rule [19]. However, when the number of rules is very large this often incurs prohibitive costs to create a validation set for every rule.

SE4COG'18, Gothenburg, Sweden
 978-1-4503-5740-1/18/05…$15.00
DOI: 10.1145/3195555.3195564

ID	Rules
Rule_1	**IF** tweet contains 'Mental' AND 'Health' **THEN** tag as *'Mental-Health'*
Rule_2	**IF** tweet contains 'Mental' AND 'Health' AND 'issues' **THEN** tag as *'Mental-Health'*
Rule_3	**IF** tweet contains 'Mental' AND 'Disease' AND 'Disorder' **THEN** tag as *'Mental-Health'*

Table 1: Sample Data Curation Rules

In this paper, we propose an adaptive approach for monitoring the precision of data curation rules by employing a combination of machine learning and crowdsourcing. The approach uses crowd to gather the required information for estimating rules precision, and determines the precision of rules through a Bayesian multi armed bandit algorithm, which is a subset of reinforcement learning technique. Reinforcement learning is a machine learning technique for situations a system must determine the best action among several actions, and the information about action effectiveness is only gathered by taking an action and observing the result. In a multi-armed bandit problem every action is known as an arm and the algorithm seeks to maximize the total cumulative reward over many repetition of actions. To formulate the problem of monitoring rules precision as a multi armed bandit problem the action to choose from is rules and the reward is the number of items correctly tagged by a rule.

The remainder of this document is organized as follows. Section 2 we present a motivating scenario and the problem statement. Section 3 we discuss the related work and the contribution of the paper. In Section 4 we introduce the proposed rule monitoring system. In Section 5 we evaluate the system. In section 6 we discuss limitations and future works, before concluding the paper in Section 7.

2 MOTIVATING EXAMPLE AND PROBLEM STATEMENT

Suppose the Australian government intend to improve the quality of social services [23] in healthcare by analyzing people's thought on social networks. These services may cover a wide range of health related topics including, mental health, medical research and cancer. As a motivating scenario we focus on Twitter. It is possible to use algorithmic approaches to analyze and identify tweets related to health domain. However, as Twitter data is "ever changing and never ending" [19] the accuracy of data processing systems can degrade easily, by relying on pure algorithmic techniques [7, 10]. To address this shortcoming, it is possible to use lots of rules [19, 26] to augment the algorithmic approaches. Imagine, the government use a rule-based system for processing data in Twitter. Imagine the rules presented in table 1 as a part of the system, which tags [1] Tweets with *Mental-Health*. However, one of the challenges in a rule-based system is that rules have variable precision. Thereby, monitoring the precision of rules helps an analyst to maintain the accuracy of the system. Existing methods for evaluating the precision of rules couple the analyst and crowd. In these approaches the analyst sends a sample of items tagged by rules to crowds and receives feedback over the number of items correctly/incorrectly rules tagged. Then, the analyst determines rules precision based on the crowd's feedback. However, in these techniques an analyst is not assisted in analyzing the crowd's feedback to effectively determines rules precision. Consequently, lack of a system to continually monitor the precision of rules and ensure analysts the effectiveness of rules is ongoing, is a challenge that degrade the accuracy of rule-based systems.

[1] is a keyword or term that assigned to a piece of information (e.g. Tweet) to describe the piece of information

3 RELATED WORK AND CONTRIBUTION

Rule based systems have been used in variety of domains; including [4, 19, 26] in classification systems, [3, 11, 17, 20, 24] in information extraction, and [24] in credit card fraudulent identification.

Much work has used rules as part of their systems. Suganthan and et al, proposed a rule evaluation technique by generating a validation set to estimate the precision of a single rule [19]. Tova and et al, proposed a system for interactive rule refinement for identifying credit card fraud. The approach clusters transactions and modifies conditions of rules based on a cost and benefit of algorithm [24]. Jun and et al, proposed a mechanism for adapting rules by employing crowd to verify rules accuracy [28]. Bak and et al, proposed a method for the "visual display" of rule results for the purposes of rule adjustment. The approach modifies rules based on marking of erroneous results of visual objects [3]. Gokhale and et al, proposed a crowdsourcing approach for entity matching by extracting features and training a random forest model for generating rules [20]. Chiticariu and et al. discussed the usage of rules in information extraction systems. In addition, the paper discussed research problems that already exists in rule based systems [17].

Much work has employed a combination of machine learning, crowdsourcing and rules [25, 26]. Sun and et al, classified product titles using crowds, rules and learning algorithms together. The approach uses rules for augmenting algorithmic approaches in cases algorithms are not working well (e.g. lack of enough training data). Shen and et al, proposed a hierarchal method for categorizing eBay products, by dividing the categorization process into two classification tasks using KNN and SVM classifiers [25]. Bekkerman and et al, employed crowdsourcing and machine learning techniques for classifying job titles in Linkedin. In the proposed approach, analysts determine the class of data, and extract a set of phrases (ngram) from frequently co-occurred keywords. Then, crowd is employed to enhance the learning process and labelling of phrases [15].

Multi armed bandit algorithm have been used in several applications; including experimentation in gaming [22], feature engineering [2], and education [5, 6, 18, 27]. This algorithm increasingly used in large scale randomized A/B experimentation by technology companies [21]. A line of works that extensively have used a multi-armed bandit algorithm is in field of educational learning. Joseph Jay Williams and et al. proposed a system (AXIS) to enhance online learning systems by generating explanation for questions. The system employs a combination of crowds and a multi armed

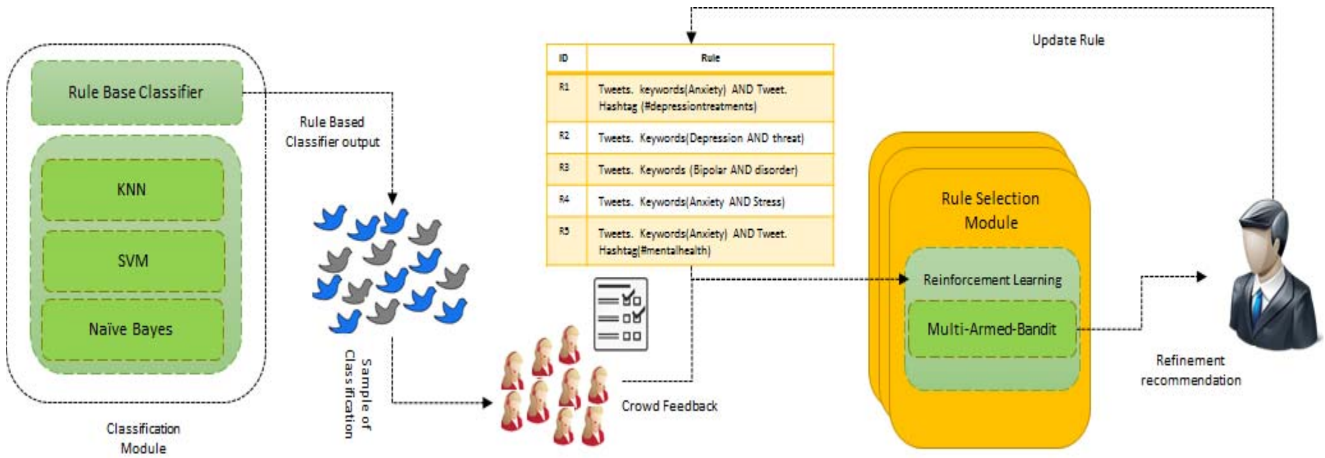

Figure 1: Overall Structure of ARMS (Adaptive Rule Monitoring System).

bandit algorithm. First, crowd workers ranks and evaluates the explanations, then the system uses a Bayesian bandit algorithm to select the highest quality explanations based on the crowds ratings [27]. Benjamin Clement and et al used a multi armed bandit algorithm in intelligent tutoring systems to choose activities that provide better learning for students. The algorithm gives reward to activities based on students learning rate [18]. M R Anderson and et al, used a multi armed bandit algorithm for input selection in feature engineering. The proposed approach improves the process of code-and-evaluate among feature engineers. In their approach instead of processing the whole input data for feature extraction, the approach identifies the most promising part of data.

Our approach, Adaptive Rule Monitoring System (ARMS), continually monitors the precision of rules, and over time learns to better identify rules deteriorating the performance of a rule based system. ARMS consists of two components: (1) observation module, and (2) evaluation module. Observation module, prepares data for evaluation by gathering a sample of items labeled by rules. Evaluation module, determines rules precision where crowds perform an evaluation over the sampled data which is relative based on the label assigned by rules. ARMS formulates training data based on the incremental and collective feedback it receives. In this manner, at every iteration ARMS becomes more efficient in identifying the imprecise rules. The selection of rules are made based on the reward system. First, at any given time, the effect of selecting a rule assessed based on the crowd's feedback. Then, ARMS assigns a reward to every rule by formulating the crowd's feedback using a reinforcement learning technique. The reward indicates rules precision. Over time, by gathering more of crowd's feedback the algorithm learns to better reward rules. ARMS works based on a subset of reinforcement learning algorithms (a machine learning technique where the information for selecting the best possible action among several actions is gathered only by trying an action and observing the results) known as multi-armed-bandit. In context of rule monitoring, by best of our knowledge no works have been used reinforcement learning before.

4 RULE MONITORING SYSTEM

Figure 1 shows how ARMS monitor rules in a rule-based system. ARMS is made up of two main components; (1) observation module; and (2) evaluation modules. Followings explain components in detail.

4.1 Observation Module

Observation module relies on crowd to gather the required information to estimate rules precision. Crowd workers[2] verify an item correctly tagged by a rule or not. To present data to crowds, first we supply crowds with an instruction on how to verify an item (e.g. if a rule tag an item with the *budget* keyword, the crowd task is to verify the item relevant to budget or not). In addition, to illustrate the instruction we show two examples, one positive and one negative. The positive example is an item that tagged with a rule and the tag correctly describes the item, and the negative example is an item that tagged with a rule, but the assigned tag is not describing the item.

4.2 Evaluation Module

A multi-armed-bandit algorithm is the core component of ARMS, which estimates rules precision by analyzing the crowd's feedback. Multi-armed-bandit algorithms contrary to supervised learning algorithms, which instruct by giving the correct actions, using the training information to evaluate the action taken. The evaluation feedback indicates how good to take an action, but not it is the worst or the best possible action.

We use Thompson sampling a Bayesian bandit algorithm that has been shown to have near optimal regret bound [16]. The algorithm provides a dynamic approach for estimating rules precision, and incorporating new observations to update estimations after observing rules reward. At each timestep rules tag a set of items, the evaluation module sends a sample of tagged items to crowds to identify how well rules are performing. After receiving the crowd's feedback the algorithm estimates a distribution θ as reward for

[2]We used a group of participants as the crowd workers

every rule. The reward indicates the precision of rules in tagging items. Rules with higher estimated reward would have higher precision, and conversely rules with lower estimated reward would be the less precise rules. The evaluation module continually by gathering more of the crowd's feedback (evidence) updates rules reward through a Bayesian bandit algorithm. In this manner, over time the module learns to better monitor the precision of rules. Initially, Bayesian bandit algorithms need to choose a prior. Prior indicates our belief before gathering any evidence[3]. Over time, by gathering more evidence prior would have minimal role in estimating rules reward. To compute prior we have chosen beta distribution

$$P(\theta = X_i) = \frac{X^{a-1}(1-X)^{\beta_i - 1}}{B(a_i, \beta_i)} = f_{a_i, \beta_i}(X).$$

The parameters β_i a_i are prior parameters. We have chosen uniform distribution $\beta_i = a_i = 1$. This means the initial reward of rules are equally likely. Then, the algorithm estimates rules reward.

$$posterior = f_{a_i + s_i, \beta_i + s_i - n_i}(X)$$

where n_i is the total number of items that has been tagged by a rule at time i, and s_i is the number of items that crowd identified the rule correctly tagged. Following explains how a Bayesian bandit algorithm estimates rules reward.

4.2.1 Example: Estimating rules precision. At each timestep rules curate a set of items, the algorithm computes a reward for every rule using *Beta(Success, Trial-Success)* distribution. *Success* is the number of items that correctly tagged by a rule. *Trial* is the total number of items that tagged by a rule. Then, over time as rules tag more items the algorithm re-estimates rules reward to update its estimation about rules precision. To put it more clearly, consider a rule based system with the rules in table 1 that tags Tweets with *Mental-Health*. Suppose at time τ_1 rules tagged a set of Tweets as the 'Mental-Health'. To determine the precision of rules, the evaluation module sends a sample of items tagged by rules to crowd, and computes rules success and trial. Suppose the evaluation module computes the success and trial for rules in table 1 as follow: $Rule_1(150 : 110), Rule_2(163 : 90), Rule_3(139 : 105)$. For example, this means at time τ_1 from 150 items that tagged by $Rule_1$, and 110 of this 150 items were relevant to mental health. The outcome of beta distribution indicates the precision of rules at time τ_1. Over time, as rules gathering more evidence through crowds feedback, the evaluation module updates rules reward and learns to better estimate their precision.

5 EXPERIMENTS AND EVALUATIONS

5.1 Experimental Environment

This section explains results obtained from deploying ARMS on two different rule based systems. First we deployed ARMS on a rule based data curation system. The system curates data in Twitter, and the goal was to assist an analyst to select the most precise rule for curating data. In addition, we have deployed ARMS in a rule based classification system. The system, contains 1103 rules and classifies tweets in health domain. The goal in this system was to identify the imprecise rules deteriorating the performance of the classification system.

5.1.1 Implementation and Datasets. The core components of techniques described in previous sections are implemented in python. As the input dataset we have used three months of Australian Tweets from May 2016 to August 2016. The applicability of ARMS is evaluated on three different domains (budget, domestic violence and mental health). As the crowd workers 12 participants were recruited, via an internal mailing list of an organization for employees interested in crowd task.

5.1.2 Experiment Scenario. We have simulated the crowdsource approaches [26, 28] to evaluate the performance of ARMS. In these methods an analyst sends a sample of curated items to crowd and receives feedback over the number of items correctly/incorrectly tagged by rules. Then the analyst estimates the precision of rules. First we implemented a method that estimates the precision of rules based on the number of items that correctly tagged by a rule. In this approach rules with the highest number of assigned correct tags is considered as the most precise rules, and rules with the highest number of assigned incorrect tag is counted as the most imprecise rules (we named this approach *rule-success*). Second, we have implemented a method that estimates rules precision by calculating the difference between the number of items correctly and incorrectly tagged by rules (we named this approach *rule-impact*). The third approach estimates rules precision based on the number of incorrect tags a rule assigned to items. In this approach the rule with the lowest number of assigned incorrect tags is considered as the most precise rule (we named this approach *rule-precise*).

5.2 ARMS: Data Curation System

This section demonstrates the performance of ARMS in a data curation system. We demonstrate how ARMS selects the most precise rule for curating data. Typically rule based data curation systems curating large amount of data. At each timestep, these systems need to select the most precise rules to avoid curating irrelevant items.

5.2.1 Data Curation System Results. This section demonstrates how ARMS learns to monitor the precision of rules in a data curation system. Figure 2 compared the performance of ARMS and other approaches in monitoring the precision of rules. As presented ARMS curated data with higher precision in all curation domains. After ten rounds of curation task the precision in budget domain reached to 78.57%, in domestic violence domain the precision is 72.22%, and this figure in mental health domain is 79.49%. On the other side; the *rule-precise* approach has the same precision with the proposed approach in two curation domains. However, contrary to ARMS which improves the precision over time; the Rule-precise approach could not main the precision of the curation task. The *rule-impact* approach in two curation domains could improve the precision to curate data more precisely, however, the performance of this approach is not as good as ARMS. In addition, we have evaluated the performance of ARMS in capturing the imprecise rules. As presented in table 2 ARMS has better or similar performance compared to other approaches. In domestic violence domain after ten rounds of curation tasks, ARMS captured a rule

[3] Evidence means the data presented to crowds for verification. If a crowd identify a rule correctly tagged an item, this indicates a positive evidence about the rule precision, and conversely if crowds verify the rule incorrectly tagged an item, this would be considered as a negative evidence

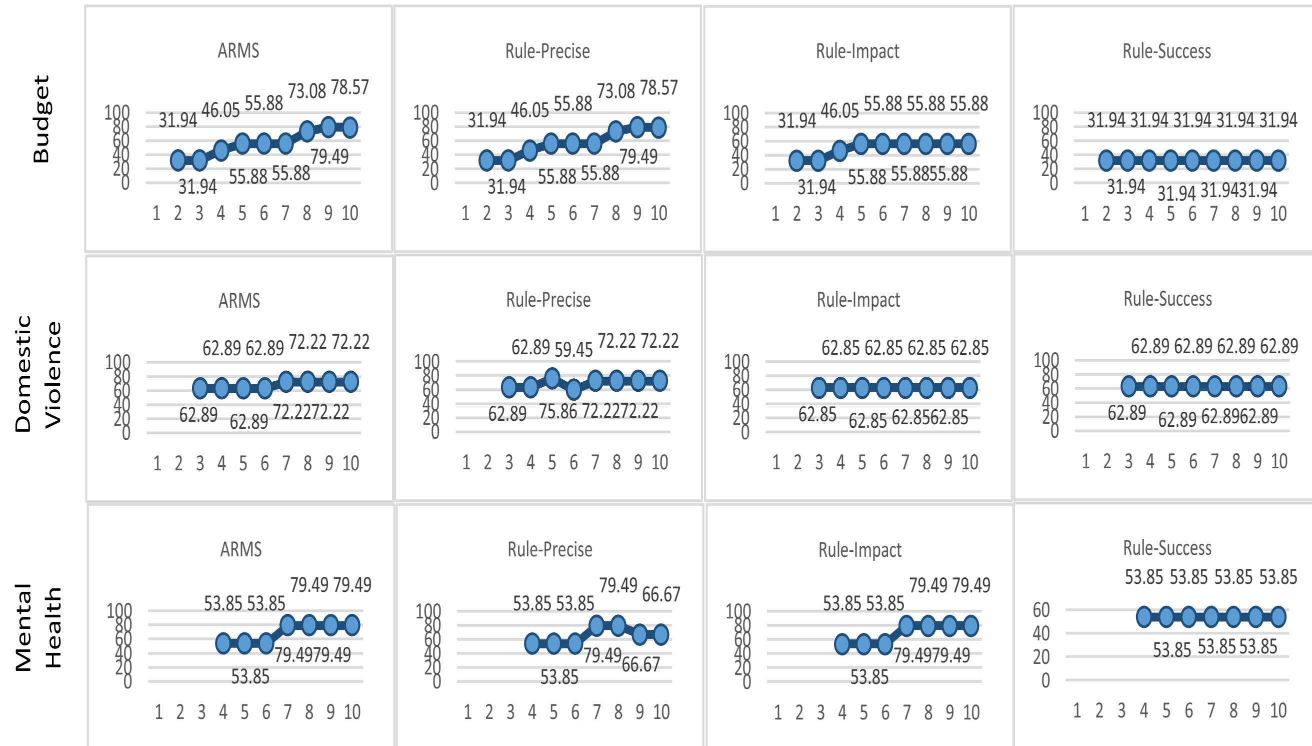

Figure 2: Performance of the proposed approach and crowdsource approaches in selecting precise rules

	Budget	Mental Health	Domestic Violence
ARMS	**26.67**	**37.23**	**09.36**
Rule-Impact	**26.67**	**37.23**	12.73
Rule-Success	31.94	47.44	12.73
Rule-Precise	31.94	47.44	12.73

Table 2: Performance of the proposed approach and crowdsource approaches in capturing imprecise rules

as the imprecise rule, which has 09.36% precision, while the rule captured with other approaches has 12.73% precision. In other domains ARMS has equal or better performance.

5.3 ARMS: Classification System

In this section we demonstrate how ARMS learns to monitor the precision of rules in a classification system. Typically, a classification system contains hundreds of rules, and every rule categorizes items with a specific label. In these systems analysts should adapt the imprecise rules to maintain the accuracy of the system. However, rules adaptation is a time consuming and difficult task, and analysts need to spend hours to find the best possible adaptation for a rule. Thereby, identifying rules that have the highest effect on deteriorating the accuracy of a classification system would help analysts to more effectively maintain the accuracy of the system. In this section we demonstrate how ARMS identifies the imprecise rules.

5.3.1 Classification System Results. To evaluate the performance of ARMS in a classification system, we divided our Twitter dataset into 16 parts (round), and at every round, we filtered out the most imprecise rules. This is similar to the rule selection policy by analysts who select and adapt the *n* most imprecise rules at every round of the classification task. However, as ARMS is not adapting rules (left as future work), we filtered out the imprecise rules. The method which improves the precision and maintains the recall of the classification system will have better performance.

5.3.2 Rule Filtering. To understand the precision of crowdsource approaches we filtered out seven [4]rules at every iteration based on the crowd's feedback. After five, 10 and 16 iterations, we roughly filtered out three, six and 10 percent of the imprecise rules. Similarly, we trained ARMS, to estimate rules reward for five, 10 and 16 iterations, and again filtered out three, six and 10 percent of the imprecise rules (based on multi-armed-bandit algorithm reward). We found out ARMS, can identify the imprecise rules more accurately, while maintains the recall and improves the precision over time. Figure 3, shows the evaluation results. In addition, we investigated the precision of rules identified as imprecise by ARMS and crowdsource approaches. We found that ARMS identified higher numbers of the

[4]We have decided to filter out 10% of imprecise rules after 16 round of classifications; so we filtered out seven rules at every iterations

Iteration & Percentage	System	Number of items labelled by rules	Precision	Recall
Filtering out 3% of rules after five iterations	Rule-Impact	1433	66.50	72.03
	Rule-Success	1404	65.02	69.01
	ARMS	1444	66.20	**72.26**
Filtering out 5% of rules after 10 iterations	Rule-Impact	1323	68.70	68.71
	Rule-Success	1273	66.84	64.32
	ARMS	1353	68.81	**70.37**
Filtering out 10% of rules after 16 iterations	Rule-Impact	1212	70.87	64.40
	Rule-Success	1169	66.88	59.11
	ARMS	1313	70.44	**69.92**

Figure 3: Precision and recall of ARMS and crowdsource approaches at different iterations.

System	Prec > 50	Prec > 20	Prec > 10	Prec < 10
Rule-Impact	33	41	11	25
Rule-Success	28	41	12	29
ARMS	5	37	10	**57**

Table 3: number of identified rules at different precisions (Prec)

imprecise rules compared to other approaches. Table 3 illustrates the precision and number of identified rules.

6 LIMITATIONS AND FUTURE WORKS.

While the proposed approach showed some initial promising results in monitoring the precision of rules, still there are some challenges that need to be addressed. ARMS could successfully identify the imprecise rules deteriorating the performance of a rule based system. Although ARMS could positively assist an analyst to monitor the precision of rules, one of the challenges that need to be solved is rule adaptation. Typically, in a rule based system after identifying imprecise rules, analysts require to modify (adapt) the rule to keep the rule applicable and precise. However, the process of adaptation is nutritionally difficult and expensive. The process is difficult because an analyst should spend hours to find the optimal modification for the rule. Also, the process is expensive as it requires an in-house analyst to adapt the rule. We believe that autonomic rule adaptation can help to reduce the cost of adapting rules by offloading analysts.

Another problem is that ARMS relies on the feedback, which gathered through crowds to update rules knowledge over changes in execution environment. However, crowd's feedback is not always accurate and contain noise. This noise, causes the evaluation module rewards rules incorrectly. The noisy feedback is not that effective to make a rule based system inapplicable. Because, over time the evaluation module by receiving more of crowd's feedback learns to better estimate rules reward. However, at the early stages the the noisy feedback may prevent the system to have the maximum possible accuracy. We plan to solve this problem by detaching from crowd's feedback over time.

7 CONCLUSION

We proposed a self Adaptive and self learning Rule Monitoring System (ARMS) for identifying the imprecise rules in a rule-based system. We employ a combination of reinforcement learning and crowdsourcing techniques. ARMS is made up of two main components; observation module and evaluation module. Observation module, observes changes in an execution environment to gather the required information for monitoring the precision of rules. Evaluation module, employ a Bayesian multi-armed-bandit algorithm for estimating rules precision. ARMS over time, learns to better identify the imprecise rules by formulating the crowds feedback through a multi-armed-bandit algorithm. The evaluation results showed ARMS successfully estimated the precision of rules in rule-based systems.

8 ACKNOWLEDGEMENTS

- We Acknowledge Prof. Boualem Benatallah for his contribution and his invaluable suggestions, comments, and supports.
- We Acknowledge the Data to Decisions CRC and the Cooperative Research Centers Programme for funding this research.

REFERENCES

[1] Bilal Abu-Salih, Pornpit Wongthongtham, Seyed-Mehdi-Reza Beheshti, and Dengya Zhu. 2015. A Preliminary Approach to Domain-Based Evaluation of Users' Trustworthiness in Online Social Networks. In *2015 IEEE International Congress on Big Data, New York City, NY, USA, June 27 - July 2, 2015*. 460–466.
[2] Michael R Anderson, Michael Cafarella, Yixing Jiang, Guan Wang, and Bochun Zhang. 2014. An integrated development environment for faster feature engineering. *Proceedings of the VLDB Endowment* 7, 13 (2014), 1657–1660.
[3] Peter Bak, Dotan Dolev, and Tali Yatzkar-Haham. 2014. Rule adjustment by visualization of physical location data. (Sept. 11 2014). US Patent App. 14/483,158.
[4] Elena Baralis and Paolo Garza. 2002. A lazy approach to pruning classification rules. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 35–42.
[5] Tiffany Barnes and John Stamper. 2008. Toward automatic hint generation for logic proof tutoring using historical student data. In *International Conference on Intelligent Tutoring Systems*. Springer, 373–382.
[6] Joseph E Beck and Beverly Park Woolf. 2000. High-level student modeling with machine learning. In *International Conference on Intelligent Tutoring Systems*. Springer, 584–593.
[7] Amin Beheshti, Boualem Benatallah, and Hamid Reza Motahari-Nezhad. 2018. ProcessAtlas: A scalable and extensible platform for business process analytics. *Softw., Pract. Exper.* 48, 4 (2018), 842–866.
[8] Amin Beheshti, Boualem Benatallah, Reza Nouri, Van Munin Chhieng, HuangTao Xiong, and Xu Zhao. 2017. Coredb: a data lake service. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2451–2454.
[9] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, and Hamid Reza Motahari-Nezhad. 2016. Scalable graph-based OLAP analytics over process execution data. *Distributed and Parallel Databases* 34, 3 (2016), 379–423.
[10] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, Sherif Sakr, Daniela Grigori, Hamid Reza Motahari-Nezhad, Moshe Chai Barukh, Ahmed Gater, and Seung Hwan Ryu. 2016. *Process Analytics - Concepts and Techniques for Querying and Analyzing Process Data*. Springer.

[11] Seyed-Mehdi-Reza Beheshti, Boualem Benatallah, Srikumar Venugopal, Seung Hwan Ryu, Hamid Reza Motahari-Nezhad, and Wei Wang. 2017. A systematic review and comparative analysis of cross-document coreference resolution methods and tools. *Computing* 99, 4 (2017), 313–349.

[12] Seyed-Mehdi-Reza Beheshti, Srikumar Venugopal, Seung Hwan Ryu, Boualem Benatallah, and Wei Wang. 2013. Big Data and Cross-Document Coreference Resolution: Current State and Future Opportunities. *CoRR* abs/1311.3987 (2013).

[13] Seyed-Mehdi-Reza Beheshti, Alireza Tabebordbar, Boualem Benatallah, and Reza Nouri. 2016. Data Curation APIs. *arXiv preprint arXiv:1612.03277* (2016).

[14] Seyed-Mehdi-Reza Beheshti, Alireza Tabebordbar, Boualem Benatallah, and Reza Nouri. 2017. On automating basic data curation tasks. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 165–169.

[15] Ron Bekkerman and Matan Gavish. 2011. High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 231–239.

[16] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.

[17] Laura Chiticariu, Yunyao Li, and Frederick R Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems!. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 827–832.

[18] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. 2014. Online optimization of teaching sequences with multi-armed bandits. In *7th International Conference on Educational Data Mining*.

[19] Paul Suganthan GC, Chong Sun, Haojun Zhang, Frank Yang, Narasimhan Rampalli, Shishir Prasad, Esteban Arcaute, Ganesh Krishnan, Rohit Deep, Vijay Raghavendra, and others. 2015. Why big data industrial systems need rules and what we can do about it. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 265–276.

[20] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 601–612.

[21] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. 2009. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery* 18, 1 (2009), 140–181.

[22] Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popovic. 2014. Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits.. In *EDM*. 161–168.

[23] Zakaria Maamar, Sherif Sakr, Ahmed Barnawi, and Seyed-Mehdi-Reza Beheshti. 2015. A Framework of Enriching Business Processes Life-Cycle with Tagging Information. In *Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings*. 309–313.

[24] Tova Milo, Slava Novgorodov, and Wang-Chiew Tan. 2016. Rudolf: interactive rule refinement system for fraud detection. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1465–1468.

[25] Dan Shen, Jean-David Ruvini, and Badrul Sarwar. 2012. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 595–604.

[26] Chong Sun, Narasimhan Rampalli, Frank Yang, and AnHai Doan. 2014. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. *Proceedings of the VLDB Endowment* 7, 13 (2014), 1529–1540.

[27] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. 2016. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. ACM, 379–388.

[28] Jun Xie, Chong Sun, Fan Yang, and Narasimhan Rampalli. 2014. Automatic rule coaching. (Sept. 2 2014). US Patent App. 14/475,470.