Received: 15 July 2016 | Revised: 28 January 2017 | Accepted: 14 April 2017
DOI: 10.1111/exsy.12218

WILEY Expert Systems

ARTICLE

Performance evaluation of rule-based expert systems: An example from medical billing domain

Umair Abdullah[1] | Antoni Ligęza[2] | Kashif Zafar[3]

[1] Barani Institute of Information Technology, Rawalpindi, Pakistan
[2] AGH University of Science and Technology, Krakow, Poland
[3] National University of Computer & Emerging Sciences, Islamabad, Pakistan

Correspondence
Umair Abdullah, Barani Institute of Information Technology Rawalpindi, Pakistan.
Email: umair@biit.edu.pk

Abstract

Most of the research in the area of performance evaluation of rule-based expert systems (RBESs) is focused on verification and validation issues. Many researchers discuss usability, usefulness, portability, and response time for the evolution of RBES. The final goal of all such studies is to construct a system with optimal, accurate knowledge base. Arguably, a system with best knowledge base is actually worthless if it is never utilized in the real world. We have proposed "benefit" as a measure of evaluation and suggested some guidelines for performance evaluation of RBESs. The proposed measure has been demonstrated for performance evaluation of an RBES applied in the medical billing domain. Results showed that the system has saved hundreds of working hours during the evaluation period of 3 months. Moreover, other associated measures have also been considered. Associated measures in the medical billing domain are "claim rejection rate"—reduced by 54%—and "claim aging," which has decreased from 34 to 28 days due to the RBES. Guidelines proposed by this research can be applied for the evaluation of expert systems implemented in other application domains, including in the first place business decision support systems.

KEYWORDS

health care IT, medical billing, medical claim processing, rule-based expert system

1 | INTRODUCTION

Expert systems (ESs ; Hopgood, 2001; Jackson, 1999; Liebowitz, 1998; Negnevitsky, 2002) are perhaps one of the best-known example of successful stories in the domain of practical applications of artificial intelligence (Norvig & Russell, 2009). Such systems use efficient solutions for knowledge representation and reasoning; typically, knowledge is represented with rules, and reasoning is performed via execution of selected rules. In fact, ESs constitute a symptomatic example of a successful combination of human expert knowledge elaboration and formal representation, human inference mimic, explanations, and limited dialog. A number of handbooks present theoretical foundations (Ligęza, 2006), tools and applications (Giarratano & Riley, 2005; Jackson, 1999; Liebowitz, 1998), and design and formal analysis issues (Liebowitz, 1998; Ligęza, 2006; Coenen & Vermesan, 1999; Delaere, Mues, Wets, & Vanthienen, 1998). A recent comprehensive book makes an attempt at summarizing the state-of-the-art (Giurca, Gasevic, & Taveter, 2009). In this paper, terms "rule-based system (RBS)" and "rule-based ES (RBES)" have been used interchangeably.

Although the domain of ESs started in the late 1960s, however, became an active area of research in late 1980s and early 1990s. The main interest was in some logical characteristics of such systems (e.g., such as consistency and completeness) assuring correct design and reliable work of implemented systems. Since then, analysis, validation, and verification of ESs have been part of active research globally (Coenen et al., 2000; Ligęza, 1999; Vermesan & Coenen, 1999; Vermesan, 1997). These early efforts are summarized in Preece (1991). However, after technology shift in the 1990s—with increasing scale of the working knowledge bases (KBs)—now focusing on extraction of knowledge from huge volumes of data and feeding the extracted knowledge to ES (Tomczak & Gonczarek, 2013). The focus points on evaluation, aims, and methods of ESs have also been changed. Large volumes of data do not undergo simple logical analysis; instead, experimental testing, knowledge extraction (from text and data warehouses), aggregation, and statistical methods are more and more frequently used. Methods of machine learning, case-based reasoning, and belief networks remained very popular (Jackson, 1999).

RBESs store detailed knowledge of a specific domain as production rules (i.e., if <condition> then <action>). It is a very convenient and efficient way of implementing business rules in software. Business rules are a collection of knowledge oriented, interrelated, complex conditions with the associated actions in a particular business domain (Halle & Ronald, 2001). Terms "production rule" and "rule" have been used synonymously in this paper.

In medical billing domain, there exist thousands of business rules for acceptance and rejection of medical claims. Few sample rules, expressed in restricted natural language, are listed below:

Rule 1: IF procedure code = 00952 AND patient gender! = female THEN Block claim AND Show message "procedure code 00952 is only applicable on female."

Rule 2: IF procedure code = 87880 AND modifier QW not applied THEN Apply modifier QW AND Show message "modifier QW applied with procedure code 87880."

Rule 3: IF place of service! = IH AND procedure code in ("99231," "99232," "99233") THEN Block claim AND Show message "Procedure codes are related to the hospital subsequent care. So the place of service should be IH."

Rule 4: IF patient age < 18 AND diagnosis code = V70.0 THEN Block claim AND Show message "For patient age less than 18 years, use diagnosis code V20.2 instead of V70.0."

Rule 5: IF diagnosis code = <DXCODE> and patient gender! = <GENDER> THEN Block claim AND Show message "diagnosis code inconsistent with patient gender."

Rule 5 given above is a table-based generic rule. A table with two columns <DXCODE> and <GENDER> has already stored diagnoses codes with their proper gender values. So if, for any claim, a diagnosis code exists but the patient gender is not the same as stored in the table then it means diagnosis code has not been used with its proper gender. So, the claim will be blocked by the system.

With conventional programming techniques, it is not possible to implement thousands of such medical billing-related business rules in the form of a software system (Abdullah, Ahmed, Asghar, & Zafar, 2015). Such a system would be impossible to maintain, and the knowledge management would be a laborious task. Instead, knowledge engineering-based approach is to be applied. An RBES has been implemented to maintain and apply these business rules on medical claims at the real time. In order to remove data inconsistencies from medical bills (before the submission of bills to insurances for reimbursement of medical services provided at doctor clinic), careful management of these rules is performed, and the rules in the KB undergo an evaluation process.

Note that ESs, and their subtype RBSs varies in many aspects from other software systems implemented with conventional programming languages (Ligęza & Nalepa, 2011); the most important ones influencing the design of the medical billing ES are recapitulated below:

First, the ESs employ declarative knowledge representation instead of procedural knowledge. In RBSs, the knowledge representation component is clearly separated from the operational part. This enables easier maintenance of the KB.

Second, the knowledge is represented with use of some specific knowledge representation language. The language must assure satisfactory expressive power and be capable of automated processing.

Third, the ultimate functionality of such systems is somewhat specific; the same KB is used for a variety of cases.

Fourth, an RBS must incorporate an inference engine assuring efficient inference strategy; the rules are to be used purposefully, and the process should be driven to the desired end.

Fifth, the body of knowledge undergoes dynamic maintenance: it can be modified on demand (enlarged, reduced, or updated) during operation. The changes can influence the work of the system and the produced conclusions.

Sixth, the KB can be vastly diverse and complicated. The chunks of knowledge, etymologically different, can be originating from different sources. They should be checked for satisfying strict formal requirements during some analysis and verification stage. For intuition, it should be coherent, consistent with the real world, complete, unambiguous, and nonredundant. This is so to assure both quality and reliability of the ES.

Finally, the inference process should be transparent to users (e.g., to enable error localization), that is, the system should be able to provide any required justifications and explanations about its conclusions and the way they were obtained. This is crucial for validation of the produced hypotheses and for monitoring and understanding the way of conclusion derivation, thus ensuring confidence of users in the produced results.

Although the form and internal structure of RBSs are relatively simple, it is quite complex and tedious task to design and implement a working system (Ligęza, 2006). It should be noticed that due to the fundamental differences between RBSs and other software systems, classical software engineering methods and tools cannot be used in a direct way (Nalepa, 2009) for the design, development, and evaluation of RBSs.

Accuracy and validity of business rules and their correct implementation are vital for the success of an RBES. Many researchers have worked on verification and validation issues of RBESs (Coenen, 1998; Ligęza, 2006; Preece, 1992). However, the focus of this research paper is beyond the creation of a 100% accurate and complete system.

Arguably, a complete, accurate, and very efficient system (in lab environment) will have no worth if it is never used in real-world environment. The focus of this paper is the evaluation of an RBES in the context of practical, real time, operational environment (not the lab environment). There can be many parameters for evaluation of realtime performance such as "adequacy," "accuracy," "efficiency," "coverage" or "usefulness" of a system and its rules. Moreover, business rules in real-world keep changing; therefore, an RBS will become obsolete if its KB is not updated regularly. Updating the KB may affect the real-time performance of the system. Therefore, it is strongly suggested/postulated that every RBS should be reevaluated after a specific period of use.

Section 2 presents related work and state-of-the-art in RBS technology. Section 2.2 presents the proposed measures and methodology for evaluation of RBESs. Section 3 gives a brief outline of the RBES applied in medical billing domain, which is going to be evaluated on the basis of measures proposed in Section 2.2. Section 4 presents a use-case example for better understanding of the system. In Section 5, details of the evaluation process of the RBES applied in medical billing domain have been described. Section 6 is devoted to results and discussion. Finally, concluding remarks and conclusions are given in Section 7.

2 | THEORETICAL FRAMEWORK

In this section, the state-of-the-art in development and application of ESs in medical billing is presented, and related work is reported briefly. Medical billing is the process of sending claims (i.e., medical bills) to insurance companies in order to receive reimbursements for services rendered to patients by the health care practitioners. Medical billing is a complicated, knowledge-based, and dynamic process. According to Healthcare Management Systems (Jones, 2014), about 30% of all claims are rejected first time, and of those first-time rejected claims, 60% are never resubmitted, which means medical practices lose 18% of the revenue due to faulty claims or errors in medical billing process, and 90% of those errors are avoidable (Jones, 2014).

All medical billing companies try to avoid medical billing errors and have built specialized software for "claim-scrubbing," that is, to verify claim's data for potential errors before it is sent to insurance for reimbursement. By scrutinizing the claims before submission, a practice can speed cash receipts and reducing denied claims and time spent reworking claims. *Athenahealth*[1] has developed an RBS for scrubbing medical claims (Monegain, 2009). Their experts are transforming medical billing knowledge to production rules. Similarly, many other health care organizations are trying to incorporate expert level capability in their business cycle so that claim rejections could be avoided and thus enhanced service quality and increased customer satisfaction could be achieved.

2.1 | State-of-the-art in medical billing ESs

Generally speaking, the computer systems supporting the process of medical billing are of three basic types:

- Simple systems, based on the use of typical tools, such as spreadsheets and/or simple databases (e.g., Excel, Access);
- Specialized systems including relational databases;
- Expert-like systems with clearly defined knowledge component.

The first two solutions are not of interest here; we focus on the third class, that is, ESs for medical billing. Medical billing ESs are a very special type of ESs located in a relatively narrow domain of knowledge concerning medical billing.

Claim Inspector of AdvancedMD (Pocatello, 2015) automatically scrubs claims for potential errors. It is a powerful coding tool that automatically checks claims for errors, based on the most current coding edits for major insurance payers. It runs more than 3.5 million edits on each claim before the claim is submitted. As a result, the first-pass claim acceptance rate is 95% or better (Pocatello, 2015).

3M Audit ES is state of the art tool for scrubbing medical claims (3M Audit Expert System, 2013). It provides robust code auditing, reporting, and management tool for medical coding compliance issues at the point of coding. It includes clinical edits to evaluate consistency with coding guidelines, resource edits to review length of stay and related charges, compliance specific sequencing edits, interactive edits that can be reviewed retrospectively, background edits that can be set to run without interrupting the user (coder), and user-defined edits, which are based on facility/hospital needs. However, 3M Audit ES is specific for inpatient records only (3M Audit Expert System, 2013).

2.2 | Related work

Evaluation of ESs is mostly related to verification of some formal properties, such as completeness, consistency, or testing for subsumption. A number of taxonomies defining the hierarchy of interesting characteristics have been proposed. For example, according to Preece (1992), the lack of work in area of ES evaluation becomes more apparent when ESs are compared to conventional software technology, where much effort has been made to develop powerful and practical methodologies of assessing and assuring the quality of software during all stages of its life cycle.

Generally desired features of an ES, including a medical billing one, can be summarized in three main points:

- All the input should be processed, no unserved cases should be left; alternatively, the ratio of such cases should be minimized;
- All the cases should be processed correctly, and thus the output should be as per expectation;
- The productivity of the system, measured by processing efficiency and quality of the output, should be as high as possible.

These features mentioned above can be achieved by removing certain anomalies and fine tuning of the rule base and inference engine. The critical analysis of the related studies reveals that automated systems foster consistency in automatic checking at multiple levels of the process, as inclusion of new rule is simple and fully automated in RBESs, so mostly rule order is inconsequential for goal oriented systems. Although in conventional programming, the rule or check order is of critical importance, as it directly affects the system performance at runtime.

A method to verify consistency and completeness of production rules was presented in the early RBESs (Suwa, Scott, & Shortliffe, 1982). The work on verification, testing, and validation of rules present in rule base continued in 1990s (O'keefe & Preece, 1996; Tepandi, 1990). However, performance at runtime, beyond the testing, and validation remained unaddressed area till 1996. When first time a model for the relationship between the goal states, the set of interrelated rules, and the data items required for the rule to fire (execute), was proposed and applied for validation of runtime performance of an RBES in Preece, Grossner, and Radhakrishnan (1996). A detailed

[1]A United States-based health care IT company working in medical billing domain.

reference on basic theory and practice of validation and verification of ESs is given by Vermesan (1997). An intuitive initial definition of completeness is suggested by Ligęza (1998) that an RBS is complete if there is at least one rule succeeding for every possible input. A useful study on integrity issues of ESs and database systems was performed by Bench-Capon et al. (1999). Similarly, a combined study on KB of ESs was done from the perspective of verification and validation (Coenen et al., 2000). Issues related to verification of tabular RBSs have been analyzed in Ligęza (2001). A detailed study on all aspects of RBSs, foundations, inference engine design, verification and validation of KB, and so forth is presented in Ligęza (2006). The study (Filip, Vanthienen, & Baesens, 2013) provides a formal grounding and an evaluation of the comprehensive rule-based compliance checking methodology. Some widely accepted taxonomy of faults in RBSs emerging from above mentioned studies are as follows:

1. **Inconsistency and inaccuracy (leads to incorrect results):**
    - (a) Inconsistency/inaccuracy of a single rule,
    - (b) Inconsistency/inaccuracy of a pair of rules,
    - (c) Inconsistency/ inaccuracy of the rule-base.
2. **Incompletion (missing rules, no coverage of the input):**
    - (a) Incompleteness of a group of rules,
    - (b) Incompleteness of the whole rule-base,
    - (c) Missing rules for specific goals.
3. **Redundancy (lack of efficiency):**
    - (a) Identical rules,
    - (b) Equivalent rules,
    - (c) Subsumption within a single rule,
    - (d) Subsumption for a pair of rules.
4. **Incorrect and unused rules and conditions:**
    - (a) Unused input fact,
    - (b) Impossible to prove precondition,
    - (c) Unused conclusion (dead end),
    - (d) Impossible to prove goal,
    - (e) Incorrect syntax,
    - (f) Free variables in conclusion.
5. **Indeterminism (deterioration of the produced output):**
    - (a) Single rule case,
    - (b) Multiple rule cases (the so-called conflict sets).
6. **Not minimal representation (loss of efficiency):**
    - (a) Possibility of rule reduction (gluing two or more rules in one,
    - (b) The possibility of replacement of a set of simple rules with reduced set of more complex ones.

However, these characteristics, although well-defined from a logical point of view—and important for assuring correct work of ESs—are hardly applicable and verifiable in the case of big software projects including rules. Here, we mention them as a kind of inspiration rather than definite, formal requirements.

3 | PROPOSED MEASURE AND METHODOLOGY

In order to propose a methodology for evaluating the system, one should reconsider the potential problems specified in Section 2.2 and, moreover, take into account the efficiency issues. In cognizance with the above foregoing general guidelines for evaluations are as follows:

- The system should cover (process) as many cases as possible (compare to completeness).
- The processed cases should lead to correct results (compare to consistency).
- The processing time should be minimal, that is, only rules necessary to obtain the results should be processed (compared to redundancy).

Simple measures like coverage ratio and correctness ratio can be used to show the performance of an RBS. In the context of the RBS under evaluation, coverage ratio will be equal to (no. of errors identified by RBS/total no. of errors actually present in data). Ideally, RBS should be able to identify all the erroneous data (i.e., covering all errors) so that no case is rejected by insurances after it has passed through the RBS. Similarly, correctness ratio will be 100% if wrong/incorrect rules are not present in the KB of the system.

However, to be more specific and to quantify these simple counts, we have proposed "usefulness" or "benefit" as the measure of performance evaluation of an RBES, which can be measured in terms of time and/or money saved by the system. A "rule" has been proposed as the granularity of benefit analysis. The benefit provided by individual rules is very important to measure. If a rule is correct but it does not provide any benefit (i.e., never applied), then it is worthless. It is only costing processing time to the system. The total benefit of an RBES is equal to the sum of benefits provided by all of its rules present in rule base. A simple intuitive equation for the system for the measurement of benefit is as follows:

$$Benefit_{RBS} = \sum_{i=1}^{N} Benefit(R_i), \tag{1}$$

where R_i represents the i-th rule, and Benefit (R_i) is the benefit (in terms of time and/or money) provided by the i-th rule. N is the total number of active production rules present in the system. Some of these rules can be incorrect, which will have negative impact, that is, time saved by wrong/incorrect rule is actually time wasted by the system. Therefore, Equation 1 can be refined as follow:

$$Benefit_{RBS} = \sum_{i=1}^{C} Benefit(R_i) - \sum_{j=1}^{W} Benefit(R_j), \tag{2}$$

where "C" is the total number of correct rules, and "W" represents the total number of incorrect (or wrongly implemented) rules present in the system. Benefit (R_j) is a negative benefit (i.e., time/money wasted) due to the j-th rule being incorporated in the system but leading to incorrect results. In proceeding sections, we have applied these proposed measures on the RBES implemented in the domain of medical billing.

To further evaluate the time and cost efficiency due to the deployment of RBS, statistical package for social sciences (SPSS-Version 23.0) is used. Multiple statistical tests including one sampled *t* test to measure the mean time utilized by users to do corrective actions on claim in comparison to standard mean time taken by RBS for autocorrection of claims, paired sample *t* test to measure the key performance indicator (KPI) rejection rate prior and after the deployment of RBS and descriptive statistics depicting measures of central tendency and measures of dispersion for the time saved by RBS have been applied on the data.

4 | THE MEDICAL BILLING-RELATED ES: A BRIEF OUTLINE

Medical billing-related ES differs from classical ESs both with respect to internal composition, as well as its functionality and operation modes. To better understand its working, let us briefly underlay the most striking differences with respect to conventional ESs:

- the system is knowledge-extensive; a vast, diversified, KB is defined for realistic applications,
- the system makes intensive use of data; data processing is one of its important roles,
- the knowledge of the system is regularly updated and so is subject to regular maintenance
- knowledge acquisition is supported with data mining techniques,
- the system operates on different input cases but has to process a large number of them with maximal efficiency and minimal of errors.

The rule-based ES developed using structured query language (Abdullah, Ahmed, & Sawar, 2009) blocks the faulty claims by performing knowledge oriented checks and actions on medical claims. Typical input and output of the system are shown in Figure 1. A batch of medical claims is passed to the RBES for identifying knowledge oriented data inconsistencies. The system returns the error-free medical claims, and erroneous claims are either blocked by the system or corrected automatically. A list of errors is also produced so that manual corrective actions may be carried out.

The architecture of the RBES as published by Abdullah (2012) is shown in Figure 2. The dotted box, containing four components, namely, Rule-Based Engine, Rule Learning Module (comprising of a warehouse and production rule-mining module), KB, and Knowledge Editor, constitutes the RBES, which is being evaluated in this research paper. Medical claims' data from different sources are inserted into the operational database of the company. Domain users (billing executives and medical providers), shown on the left side of Figure 2, use medical billing-related software to enter medical claims' data in operational database of the company. Domain experts (persons normally with more than 3 years of experience in medical billing), shown at the top left corner of Figure 2, also use medical billing-related software in order to monitor medical billing activities within the company. After applying the RBS on medical claims' data, claims are submitted to insurance companies electronically via internet. Claims without any billing mistakes are immediately signed-off by the RBS and submitted to their respective insurances, which are then paid by the insurance companies.

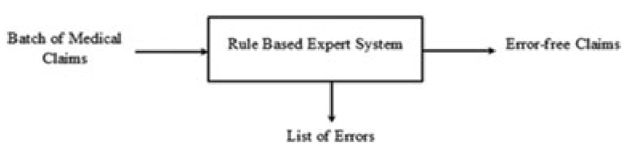

FIGURE 1 Input and output of the rule-based expert system (Source: Abdullah et al., 2015)

Currently, the system has around 2,000 production rules in the KB. The KB is a crucial part of the system; it is relatively flat and strongly diversified. Production rules present in the KB of the system—to be evaluated—can be categorized into different types as shown in Figure 3. There are two major categories: "meta-rules" and "rules." Meta-rules are evaluated first, and they activate relevant normal production rules.

Normal rules can be further divided into three categories: warning rules, autocorrection rules, and claim blocking rules. Warning rules are not critical, just display message for missing optional data. Violation of a warning rule does not result in claim rejection. Autocorrection rules are most useful. They identify the critical mistakes in the data and then instead of blocking the claims, these rules correct the faulty claims by themselves. In this way, manual correction time of faulty claims is also saved.

Claim blocking rules implement critical checks. These rules do not allow faulty claims to pass to insurance. Claims blocked by these rules wait for manual correction by the billing executive or the provider. Claim blocking rules can be further divided into two categories: table-based rules and singular rules. Singular rules compare data of a claim to the data values present within the rule itself, and table-based rules compare data of a claim to multiple sets of values stored in a relational database table. Sample rules of above mentioned types have been given below in descriptive format:

Rule 1 (Meta Rule): IF insurance = Medicare THEN Activate all Medicare related rules.

Rule 2 (Meta Rule): IF practice code = 1001 THEN Activate rules of practice 1001.

Rule 3 (Warning Rule): IF procedure code = 95117 THEN Show message "Use procedure code 95115 if single injection applied (95117 is for two or more injections)."

Rule 4 (Warning Rule): IF procedure code between "00100" and "01999" AND modifier = LT THEN Show message "Modifier LT should not be used with anesthesia code."

Rule 5 (Auto-correction Rule): IF electronically corrected checkbox = unchecked THEN Check print center check box AND Show message "Checked print center, as electronically corrected claim box is unchecked."

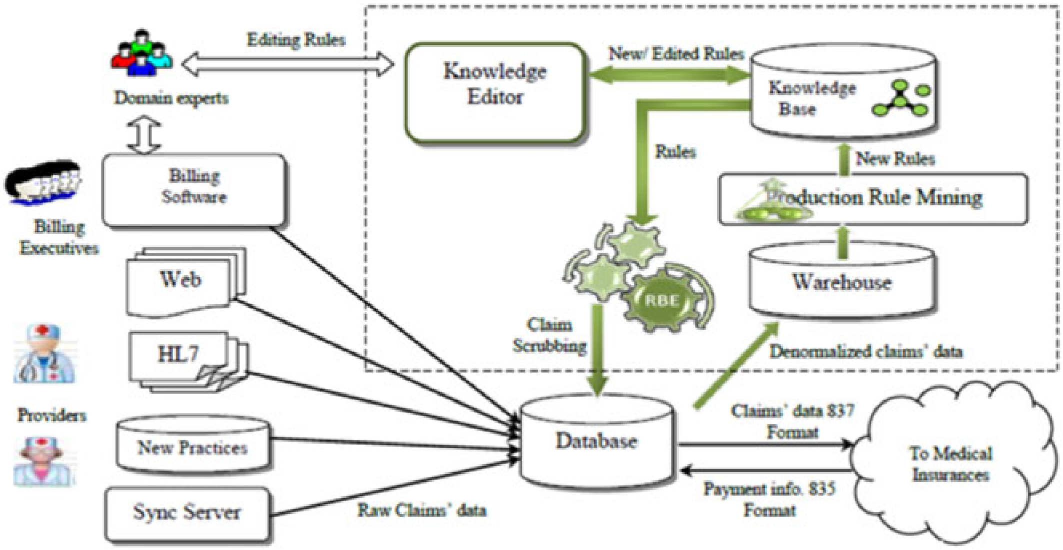

FIGURE 2 Architecture of the rule-based expert system applied in medical billing domain (Source: Abdullah, 2012)

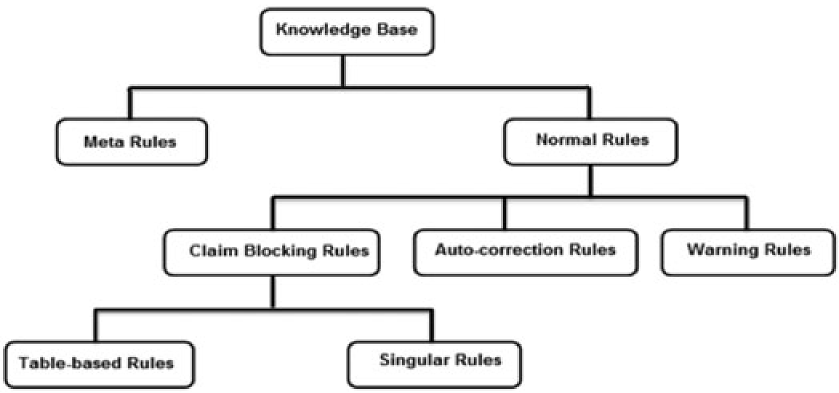

FIGURE 3 Taxonomy of production rules present in knowledge base of the system under evaluation

Rule 6 (Auto-correction Rule): IF modifier! = 26 AND procedure code in (59025, 76801, and 76805) THEN Apply modifier 26 AND Show message "Modifier 26 auto updated for procedure codes 59025, 76801, 76805."

Rule 7 (Singular Rule): IF patient age not between 29 days to 2 years AND procedure code = 99472 THEN Block claim AND Show message "Subsequent inpatient pediatric critical care procedure 99472 can only be used for patient with age between 29 days and 2 years."

Rule 8 (Singular Rule): IF procedure code = 77427 AND units >1 THEN Block claim AND Show message "Radiation treatment procedure 77427 can be billed with one unit only."

Rule 9 (Table-based Rule): IF procedure code 2 = <COLUMN 2 CODE > AND procedure code 1! = <COLUMN 1 CODE > THEN Block claim AND Show message "Component code being used without comprehensive code."

Rule 10 (Table-based Rule): IF procedure code 1 = <PROC1> AND procedure code 2 = <PROC 2 > THEN Block claim AND Show message "procedure 1 and procedure 2 are mutually exclusive codes, therefore, cannot be combined with one claim."

Last two table-based rules are the implementation of National Correct Coding Initiative. Rule 9 shows that a component code cannot be billed without its comprehensive code. Hundreds of such pairs have been stored in a relational database table. Inference engine of the system access those pairs with the help of <COLUMN 1 CODE > and <COLUMN 2 CODE > variables and compares them with procedure codes 1 and 2 present in the claim being processed. Similarly, <PROC 1 > and <PROC 2 > of a relational database table store hundreds of mutually exclusive codes (i.e., which are not allowed to be billed together in a single medical claim). Rule 10 compares procedure codes 1 and 2 of the claim with the table data and blocks the claim if both mutually exclusive codes are present in the claim.

5 | USE CASE EXAMPLE

In order to provide some further intuitions a simple use case, example is presented as follows. Suppose a patient—aged 16—visits a medical provider for having a respiratory problem. Provider, after examining the patient, applied an anti-allergic injection. After the patient visit, the provider prepared a medical bill/claim for reimbursement from an insurance company of the patient guarantor (as the patient is a dependent, i.e., under 18 years of age). The provider sent the claim to the medical billing company for further submission to insurance on his behalf––with the following diagnosis codes:

V70.0 Routine general medical examination at a health care facility, and

477.2 Allergic rhinitis due to animal (cat) (dog) hair and dander.

The claim has the procedure codes 95004 for skin testing and 95117 for professional services for allergen immune therapy. RBES deployed in the medical billing company processes the claim for any inconsistencies and quickly blocks the claim and shows error message "For patient having age less than 18 years, use diagnosis code V20.2 instead of V70.0." The system also suggests using procedure code 95115 instead of 95117 in case single injection is used (as 95117 is for two or more injections). The claim is corrected at the billing company (i.e., prior to its submission to insurance). It would have caused a considerable delay if the erroneous claim has got submitted to insurance. Moreover, using procedure code 95117, and actually one injection is used, may result in a medical fraud case on the provider.

Claim detail form along with portion of claim error window is shown in Figure 4. Critical errors are displayed in red font, and blue font is used for warnings, and green font color is used for suggestions. User has three actions for each rule: can ignore the rule, can manually edit the claim, or can select auto apply option so that RBS my do the required change automatically. Auto-apply option saves the manual working time on claim.

6 | EVALUATION OF THE RBES FOR MEDICAL BILLING

This section presents performance evaluation of the RBES in accordance with the methodology proposed in previous section. The RBS is operational in medical billing domain since January 2010 (Abdullah et al., 2015). The system is being used at medical billing company for "scrubbing" (i.e., performing small legitimate actions in order to maintain accuracy) of medical claims. Knowledge-oriented data inconsistency errors are identified in the medical claims and corrected by the system before submitting them to medical insurances (Abdullah et al., 2015).

6.1 | Organizational setting

The company has an office in a custom-built software technology park, which governs under United States health care laws such as Health Insurance Portability Accountability Act. There are approximately 3,000 providers of more than 480 practices signed up with the company. Approximately, 1,000 employees are working in different departments, with IT and Operations being two major departments of the company. Employees of the company work in three shifts (8 hr each). Company remains operational 24 hr a day, 6 days a week. On Sundays, maintenance tasks are performed. The company offers a suite of fully integrated, end-to-end services, which help physicians and practices simplify every step of the practice management process, from the initial scheduling of an appointment to the billing and remittance tasks. The company has its custom-built medical billing software, websites for medical providers, desktop-based, and a web-based certified electronic health record (EHR).

Prior to the implementation of RBS, claim rejection rate was high due to medical billing and coding errors in data. RBS developed and deployed to support billing executive in making decisions such as which modifier should be used with a specific procedure, what other attribute values are required for certain diagnosis codes or procedure codes. RBS also performs National Correct Coding Initiative edits such as comprehensive component code pairs, Mutually Exclusive Edits, Medically Unlikely Edits (MUE), along with other medical billing compliance related errors.

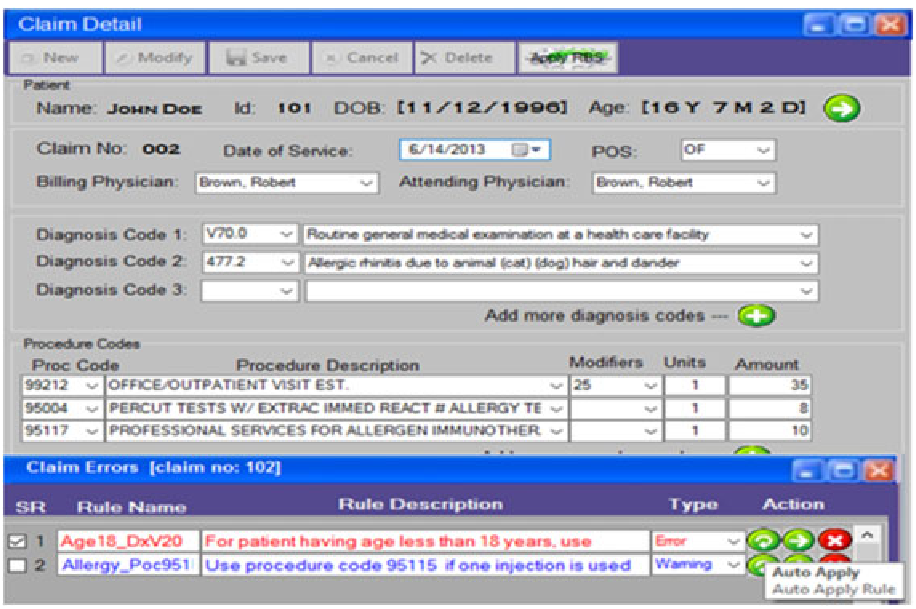

FIGURE 4 Claim detail form with portion of claim error window showing critical error and a warning

6.2 | Medical billing process and the RBES

Medical billing is the process of submitting medical claims to insurance companies and following up on the submitted claims in order to receive payment for services rendered by a health care provider. It is an interaction between a healthcare provider and the insurance company (payer). The entirety of this interaction is known as the billing cycle. This process can take several days to several months for completion and requires several interactions before a resolution is reached. As published by Abdullah et al. (2015), the simplified medical billing process is shown in Figure 5. The process starts with the office visit of a patient on a preset appointment date and time.

A provider or staff typically creates or updates the patient's medical record. The record contains information about treatments and demographic information including patient's name, social security number, home telephone number, office telephone number and address, and his/her insurance policy identity number. If the patient is a dependent, then guarantor information is included in patient's record. On the first visit, providers usually assign one or more diagnoses to the patient. Diagnosis codes and procedure codes, along with patient information constitute a medical claim, which is sent to billing company or a clearinghouse for onward submission to medical insurances. The insurance company (i.e., payer), processes the claims to assess the payment level. Three factors: patient eligibility, provider credentials, and medical necessity, are used to determine payments. Payment rates are already agreed upon by the provider and the insurance company.

Until a claim is fully paid or the provider accepts an incomplete reimbursement, exchange of claims, and rejections continues several times between the provider and the insurance via medical billing company. In medical billing domain, due to similarities in diagnoses codes and high complexity of medical billing knowledge, the frequency of rejections and overpayments is high (often reaching 50%).

Insurance companies also deny certain services that are out of their scope. In such cases, adjustments are made, and the claim is resubmitted. A certain portion of billed amount, approved by the insurance but not paid, is identified as patient's responsibility. It is sent to the patient for reimbursement.

Figure 6 and Figure 7 show processing time of correct claims and faulty claims, respectively. Note that after the date of service, billing time and time taken by insurance for processing correct claim are the same (with and without RBS). However, faulty claim gets delayed (shown as insurance aging 1 in Figure 7). RBS has been implemented to reduce this delay in payment of a faulty claim by timely identifying the errors. Figure 6 shows that after date of service, claims are received by the billing team, billing team enters the claim in the billing software. If the claim is clean, it gets paid by the insurance within 5 to 12 days. A clean claim has insurance aging of 5 to 12 days.

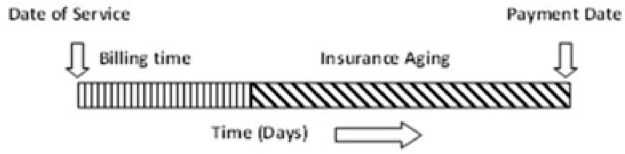

FIGURE 6 Processing time of correct claim comprise of 2 to 3 days of billing time plus 5 to 12 days (approx.) of insurance processing time (Source: Abdullah, 2012)

Figure 7 shows that the processing of erroneous claim, which gets rejected at insurance office. This rejection is known as first-level rejection, and if rejected second time, then it is called second-level rejection. Insurance aging 1 is the time taken by insurance for processing the erroneous claim for the first time (and rejecting it). The billing team after receiving rejection for the claim rectifies the error and resubmits the claim to insurance. Insurance aging 2 in Figure 7 represents the time taken by insurance for reprocessing the claim second time and paying for it.

After applying RBS, time taken by insurance for processing faulty claim is saved (shown as gray area in Figure 7 and the gray area is absent in Figure 8), as RBS identifies the mistake prior to its submission to insurance, and user is required to correct the claim before submitting it to insurance.

6.3 | Parameters for the measurement of benefit (the proposed measure)

The following two values have been calculated for measurement of benefit (i.e., the proposed measure) for evaluation of the system under consideration;

- The estimated working time saved, and
- The estimated claim aging time saved.

6.3.1 | Estimated working time saved

Time which would have been spent by billing executives to perform corrective actions on the faulty claims, which have been corrected by RBS automatically with the help of “auto-correction” rules.

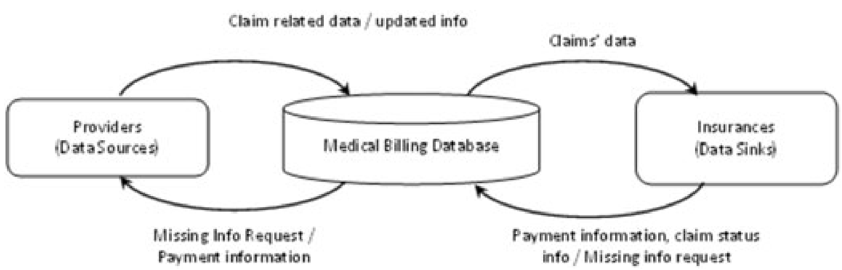

FIGURE 5 Typical medical billing cycle (Source: Abdullah, 2012)

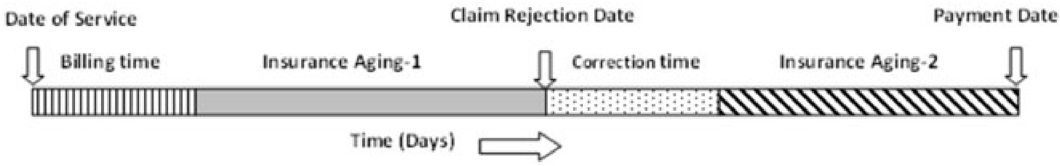

FIGURE 7 Processing time of faulty claim without rule-based system is sum of billing time (2 to 3 days), insurance processing time (5 to 12 days approx.), which results in rejection, correction time of 2 to 3 days, and insurance processing time of 5 to 12 days (approx.) again (Source: Abdullah, 2012)

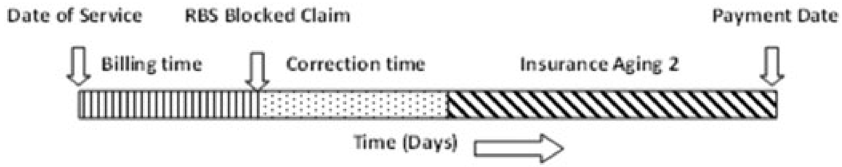

FIGURE 8 Claim processing time (days) of faulty claim with rule-based system (RBS) does not include 5 to 12 days of insurance first processing time as error is identified, and claim is blocked by the RBS on same day (Source: Abdullah, 2012)

6.3.2 | Estimated claim aging time saved

It is the difference between time spent by insurance on processing of an erroneous claim and time spent by the same insurance on a claim, which has the same type of error but corrected by RBS.

Instead of letting the erroneous claims pass to insurance and get rejected, RBS blocks faulty claims from submission, and user is asked to correct the faults by making a suggested action. In this way, a lot of time is saved, as insurance may take many days to process a faulty claim and finally reject it.

$$\text{Total\_Working\_Time\_Saved} = \sum_{i=1}^{C} \left(\text{Time\_Saved\_per\_Claim}(R_i) \times \text{Claims\_Processed\_by}\ (R_i)\right), \tag{3}$$

6.4 | Methods for data acquisition and measurements

Data stored in log tables populated by the RBS has been used for calculation of the time saved by the system. Log tables store all the details about the processing of claims by the RBS, that is, when a claim is blocked or auto-updated and due to which reason. "Claim no," "applied rule," "applied date," "result," and "error description" are important columns of log tables, which are of the interest of this study. Data of last quarter (i.e., October, November, and December) of 2014 is being used for performance evaluation of the applied system. Counts of claims processed during this period give us an overview of the effectiveness of the system.

6.4.1 | Method to estimate working time saved

To estimate working time saved an experiment was performed. Five billing executives selected, having the variable skill of handling company's medical billing software (least efficient to the most efficient) participated in the experiment. Different users work at different speeds. Novice users take more time to open a claim, edit it, and save it back to the billing software. Experienced users perform data editing tasks more quickly. An average has been taken to calculate the time consumed by a user for updating the data. The experiment lasted for 10 working days. All the corrective actions carried out by RBS during the evaluation period were done manually. In this way, manual working time was recorded for each rule. To calculate total working time saved, we have used the following formula:

where R_i is i-th rule, and i goes from 1 to C (i.e., total number of correct rules).

The experiment to calculate manual working time was performed for 120 corrective actions, which RBS has performed during the evaluation period. As a sample, a list of top 20 corrections with an average manual working time of five billing executives is shown in Table 1. With 95% confidence each average has margin of error $\pm$ 5 s.

Manual average working time for doing a change is actually the time saved per claim by an auto-correction rule, which performs the correction within milliseconds (i.e., less than 1 s). Time has been calculated from opening a claim to when the claim is saved successfully (assuming that user has already logged into the billing software).

Table 2 depicts the results of time consumed for performing corrective action manually for 20 different checks. The mean time consumed by user to perform corrective action manually is calculated as 94.25 s. As different time was required to do corrective

TABLE 1 Time consumed by average user for performing corrective actions done by the RBS automatically (Source: Abdullah, 2012)

SR	Action/change description	Time in seconds
1	Apply specific modifier with specific CPT	110
2	Set billing physician for specific location	90
3	Apply place of service of a specific CPT	105
4	Insert insurance notes and check send notes	90
5	To open a claim and insert claim notes	40
6	Exclude any CPT from submission	80
7	Block a claim for carrier hold instruction	105
8	Check print center for specific insurance	70
9	Check include in EDI for a specific CPT	85
10	Apply G codes for E-prescription claims	180
11	Apply specific DX code for specific CPT	130
12	Apply specific CPT for specific DX code	135
13	Insert service authentication code for a claim	85
14	Check "Is Capitated" box for a claim	65
15	Adjust the claim and exclude it from submission	125
16	Replace procedure codes	130
17	Set referring physician of a claim	60
18	Change attending physician of a claim	65
19	Clear group number for a Medicare patient	70
20	Update CLIA check box for a claim	65

Note. CPT = Current Procedural Terminology; CLIA = Clinical Laboratory Improvement Amendments; RBS = rule-based systems.

TABLE 2 One-sample statistics

	N	Mean	Std. deviation	Std. error mean
Time consumed for performing corrective action manually	20	94.25	33.454	7.480

action for each check manually, so data depicts huge variation with 33.454 as standard deviation. To check the significance one sample *t* test was performed, and following hypotheses were developed.

H_o (Null Hypothesis): $\mu_1 = 0.001$ (Mean time to do corrective actions manually is 0.001 s)

H_1 (Alternate Hypothesis): $\mu_1 \neq 0.001$ (Mean time to do corrective actions manually is not 0.001 s)

RBS takes milliseconds or less than millisecond to accomplish the task or autocorrects; hence, test value was taken as .001.

Table 3 depicts that one sample *t* test statistic is 12, and *p* value is .00000000011, which is far less than .05. Such *p* value indicates that average time consumed for corrective action by users manually is significantly different by the time consumed (0.001 s) by RBS in auto-correction job. So the Null hypothesis is rejected at 95% level of significance.

6.4.2 | Method to estimate claim aging time saved

Terms claim aging, claim insurance aging, and insurance aging have been used interchangeably in this paper, which refers to the amount of time (normally in days) taken by insurance to pay/process a medical claim. Claim aging time saved is the difference between the amount of time consumed to get a payment of a claim (by insurance) without prior processing of the claim by the RBS and with the prior processing of the claim by RBS. By analysis of historic medical billing data of the company, it has been found that insurance time for processing a faulty claim and rejecting it (which becomes the time saved by RBS) depends upon following factors.

- Type and nature of mistake or fault in the claim;
- Insurance payer itself (which is processing the faulty claim);
- Other factors such as provider setup, patient insurance plan, and claim amount.

The first and the second factors given above are most crucial in determining insurance processing time. Some insurances process claims within 7 days, and some insurances take up to 30 days for processing a claim. Similarly, if some basic and important information is missing, then claim is rejected spontaneously. For knowledge-oriented mistakes, claim is rejected after several days. Types and nature of mistakes or errors can be associated with production rules of RBS. Normally for every billing error or mistake, we have one rule. Therefore, we can relate insurance time for processing erroneous claim to production rule as time saved by the production rule. During analysis phase of the research work, we have estimated the time saved by each rule with the following methods:

- The erroneous claim first rejected by insurance, later on, corrected by RBS and paid by insurance company. First processing time taken by insurance, when it rejected the claim, is noted.
- Time taken by similar claims (having the same fault), some submitted to insurance without RBS, and some processed by RBS,

TABLE 3 One-sample test

	Test value = .001					
					95% confidence interval of the difference	
	T	Df	Sig. (two-tailed)	Mean difference	Lower	Upper
Time consumed for performing corrective action manually	12.599	19	.000	94.249	78.59	109.91

TABLE 4 Top 10 auto-correction rules in terms of number of corrections made during the evaluation period

SR	Rule description	No. of claims	Saving time per claim (s)	Processing time saved (hr)
1	Office and well care code in the claim, modifier 25 applied with the office visit code.	3,540	180	177
2	G-code used with E-prescription claim	3,012	210	175.7
3	Separate lines for each unit of CPT L3020 entered by RBS	2,963	210	172.8
4	Modifier 26 is auto updated whenever any CPT code is used from the list (59025, 76801, 76805, 76811, 76815, 76816, 76817, 76818, 76819, 76820, and 76830)	2,551	110	77.9
5	For the selected location, RBS changed POS to OH	1,245	95	32.9
6	Service-authentication exception code is set to 7 for anesthesia practices (1010884, 1010887, and 1010892)	925	85	21.8
7	Unchecked print center, because the electronic corrected claim is checked at insurance payer's level	840	85	19.8
8	Patient age is less than 18 years and financial guarantor exists, RBS has checked the guarantor info for patient billing at patient demo	677	65	12.2
9	Missing CLIA check automatically updated by RBS	389	65	7
10	Checked print center, because the electronically corrected claim box is unchecked at insurance payer level	246	70	4.8
	TOTAL	16,388		702

Note. CLIA = Clinical Laboratory Improvement Amendments; POS = Place of Service; RBS = rule-based system; OH = Outpatient (Hospital).

corrected and then submitted to insurance. Average difference in time until all the claims are paid is stored, as time saved by the rule per claim.

- For every rule, we identified claims from historic data (pre RBS data) having the same error (for which a rule is built). Then averaged the first insurance aging (i.e., processing time before the rejection) to calculate claim aging time saved by the rule per claim. Time saved calculated on the basis of above mentioned methods have been presented in the next section along with the brief analysis.

7 | RESULTS AND DISCUSSION

During the last quarter of 2014, approximately 96,000 errors in 84,000 (7%) erroneous claims have been identified out of 1,260,000 total claims. As shown in Figure 9, approximately 1,176,000 (93%) claims have been marked correct and were signed-off by the system for further submission to insurances. Out of 96,000 errors, 30,000 corrections were made in 24000 claims by the auto-correction rules, 52,000 errors in 48,000 claims were critical (i.e., identified by claim blocking rules), which were later corrected by billing executives manually. Approximately, noncritical 14,000 errors (in 12,000 claims) were found, which resulted in displaying of warning messages only (however, claims were not blocked by the RBS).

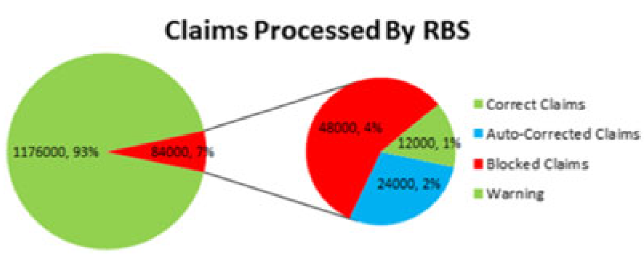

FIGURE 9 Claims processed by rule-based system (RBS) during last quarter of 2014

Note that one erroneous claim can have multiple errors; therefore, error count is greater than the number of distinct erroneous claims. Findings and outcome data after the analysis are presented in the following subsections:

7.1 | Estimated working time saved

Top 10 auto-correction rules have been shown in Table 4. These rules have corrected 16,388 erroneous claims and saved 702 working hours (which billing executives would have spent in order to correct those erroneous claims). Currently, there are approximately 140 auto-correction rules in the KB of the system. By using the formula given in Equation 3, we have found estimated working time saved is 1,853 hr. This means RBS has saved 1,853 hr of manual work by doing 30,000 corrections. Hourly rate of a medical biller/coder range from approximately 11 to 19 USD; hence, by saving 1,853 man hours during last 3 months of 2014, RBS has saved an amount ranging from 20,383 to 35207 USD. This is a considerable amount, which proves the usefulness of the applied RBES.

Working of top 10 auto-correction rules of RBS was analyzed; Table 5 shows descriptive statistics of the analysis. In 2014, auto-correction rules of RBS had saved total 701.9 or 702 hr of manual work. On average, each of the above mentioned rules saved 70 hr.

7.2 | Estimated claim aging time saved

Claims with critical errors are rejected in 1 day (i.e., rejected at first level). Claims rejected after some processing are known as rejected at the second level, and they take approximately 2 to 14 days of processing time before rejections. Different claims can be under processing at the same time by different insurance payers or even by same insurance payer. Insurance processes claim in batches, and as described earlier, claim aging depends upon insurance itself and the type of error in the claim. The following analysis shows how much

TABLE 5 Descriptive statistics

	N	Minimum	Maximum	Sum	Mean	Std. deviation
Processing time saved	10	4.8	177.0	701.9	70.190	75.3008
Valid *N* (list-wise)	10					

amount has been paid and how many days earlier due to RBS. Claim aging has been calculated from historic data of the rejections.

During the last quarter of 2014 (October, November, and December), the total amount paid earlier (as RBS saved claim aging by timely identifying the errors) is 7 million USD. By grouping, the amount paid in aging groups (buckets) amount paid per number of days earlier has been calculated as shown in Figure 10. From Figure 10, it is clear that maximum amount of 1,360,127 USD have been paid to medical providers 5 days earlier (due to RBS). The second highest amount is 1,208,647 USD paid 8 days earlier. Similarly, 39 days have been saved for the payment of 339,322 USD by insurances to various medical providers due to RBS.

Estimated time saved (in payment of claims earlier) shown in this section has been validated when we observe actual claim aging and claim rejection reports, presented in the following section.

7.3 | Associated measures

Associated measures are those regular business reports, which should be effected when an ES is introduced/deployed in a company. These reports provide the evidence of the effectiveness of the applied RBS and play an instrumental role in validating system's evaluation results. In the domain of medical billing compliance, important measures are claim rejection report and claim aging report, which should be associated with every claim scrubbing software to validate its effectiveness.

7.3.1 | Claim aging report

As described earlier, time spent by insurance for processing of the claim is termed as claim aging. A daily report is generated by the operations department of the company regarding claim aging. It has been observed that average claim aging has dropped from 34 days to 26 days after the implementation of RBS.

7.3.2 | Claim rejection report

Like all multinational organizations, the health care IT Company where the study was practically carried out used to track the KPI rejections on daily basis. With conventional programming, only a few checks, based on KPI rejections, were implemented, which did not reduce the KPI rejection rate significantly. And due to the implementation of ICD-10, it was required to get an accurate picture of how well the revenue cycle is performing. As the negative impacts of KPI rejections directly affect the cash flow of the company, so tremendous effort was made in order to develop the system, which can stop the data entry operators from further manual mistakes.

RBS has taken most of the credit for the reduction in rejection rate. To evaluate the statistical significance of the data, the following Null and Alternate hypotheses were developed for the study:

H_o (Null Hypothesis): $\mu_1 - \mu_2 = 0$ or $\mu_1 = \mu_2$ (the paired population means are equal);

H_1 (Alternate Hypothesis): $\mu_1 - \mu_2 \neq 0$ or $\mu_1 \neq \mu_2$ (the paired population means are not equal),

where μ_1 denotes the average KPI rejection rate of October 2009 (prior to the deployment of RBS), and μ_2 denotes the average KPI rejection rate of December 2014 (After the deployment of RBS). The Null hypothesis states that average KPI rejections prior to and after the deployment were statistically equal (i.e., equal with some acceptable margin). Paired sample *t* test was applied on paired sample size of 27. As both October and December comprise of 31 days and excluding 4 Sundays, remaining 27 days were taken in account.

In October 2009, when the RBS was not executed and conventional programming software was in use, the company had KPI rejection rate of 5.18% (i.e., 5 claims were rejected out of every 100 claims), which was an alarming stage. RBS decreased the KPI rejection rate of the company from 5.18% to 2.75% in December 2014 thus showing 53% reduction in rejection rate. Table 6 depicts the arithmetic mean of both samples, that is, KPI rejections of October 2009 and December 2014.

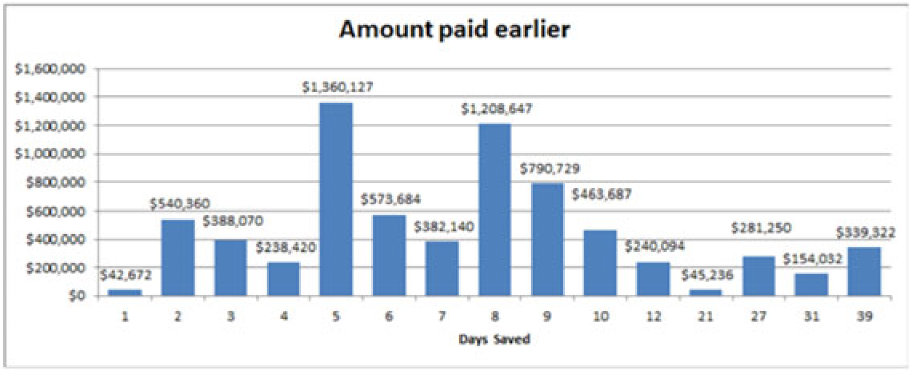

FIGURE 10 Amount paid earlier due to rule-based system

TABLE 6 Paired samples statistics

	Mean	*N*	Std. deviation	Std. error mean
KPI rejections—October 2009	5.1830	27	1.64691	0.31695
KPI rejections—December 2014	2.7574	27	1.43608	0.27637

Note. KPI = key performance indicator.

As it is vital for study to know how strongly two variables are correlated with each other, so paired sample correlation has been calculated. Table 7 depicts that correlation of both the variables is positive but yet insignificant. As the significance value .0292 > .05, which means correlation is by chance.

TABLE 7 Paired samples correlations

	N	Correlation	Sig.
KPI rejections‐‐October 2009 and KPI rejections‐‐December 2014	27	0.211	0.292

Note. KPI = key performance indicator.

Table 8 depicts that paired sample *t*-test value 6.484 does not lie within the critical region of lower limit 1.65661 and upper limit 3.19450; hence, Null hypothesis μ1−μ2 = 0 or μ1 = μ2 is rejected at 95% level of significance. The two-tailed significance value of .0000007 is less than .05, so we can conclude that there is significant difference between mean KPI rejections in October 2009 (prior to deployment of RBS) and mean KPI rejections in December 2014 (after the deployment of RBS).

We can conclude that RBS has decreased KPI rejections significantly. The below graphs depict the KPI rejection rate of the company prior and after the deployment of RBS. Claim rejection rate for the month of October 2009 is shown in Figure 11 and of December 2014 in Figure 12.

Other benefits provided by the RBS have been briefly discussed in the following section.

7.4 | Other benefits

RBS has provided two indirect benefits; claim blocking report and customer satisfaction report. RBS claim blocking report contains

TABLE 8 Paired samples test

	Paired differences						
			95% confidence interval of the difference				
	Std. deviation	Std. error mean	Lower	Upper	T	Df	Sig. (two-tailed)
KPI rejections‐‐October 2009 and KPI rejections‐‐December 2014	1.94381	.37409	1.65661	3.19450	6.484	26	0.0000007

Note. KPI = key performance indicator.

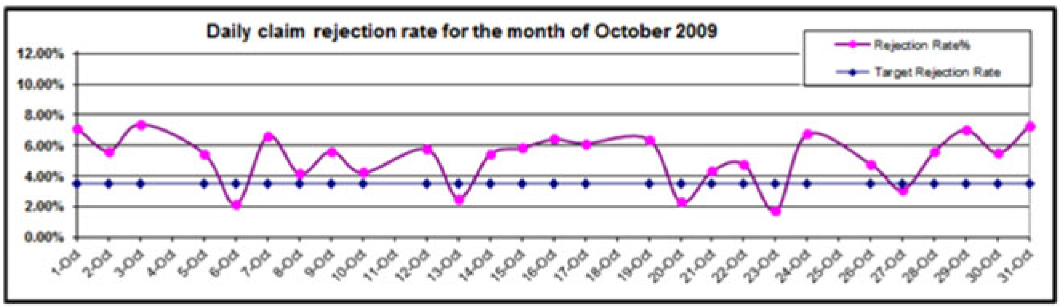

FIGURE 11 Daily claim rejection graph prior to implementation of the rule-based system (Source: Abdullah, 2012)

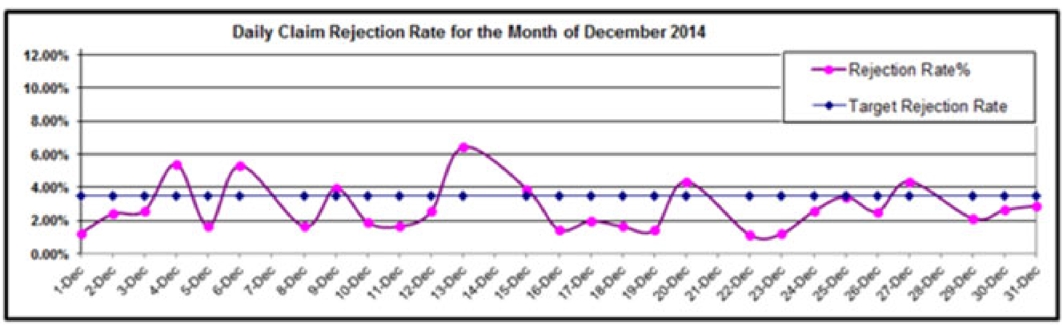

FIGURE 12 Daily claim rejection graph of December 2014, that is, after implementation of the rule-based system (Source: Abdullah, 2012)

summary and details of claim blocked by RBS. This report reflects the quality of work done in operation department of the medical billing company. If billing executives make fewer mistakes in operations, fewer claims will be blocked by RBS. Claim blocking report is sent by RBS team to team leaders and managers of operation department on daily basis. Secondly, results of RBS are displayed on EHR of the company. Medical providers use suggestions of RBS and update their claims in order to avoid medical billing errors. This feature in EHR has increased customer satisfaction. Further, RBS has elevated the client satisfaction, as claim rejection rate has been decreased by 51%, and average claim aging has reduced from 34 to 26 days.

8 | CONCLUSIONS

The RBES applied in medical billing domain—currently with 2,000 production rules—has been proved to provide considerable benefit to the company. It has saved 1,853 hr of manual working time by auto-correcting erroneous claims, with estimated worth of 20,383 to 35,207 USD (as medical coder average hourly pay range from 11 to 19 USD). Another main advantage of RBS is claim aging time saved. During last quarter of 2014, approximately 7 million USD have been paid earlier due to the RBS—with average claim aging reduced from 34 to 26 days. Customer satisfaction has been increased as claim rejection rate—an associated measure of RBS—has decreased from 5.18% to 2.76%. On the basis of statistical analysis administered through SPSS (V 23.0), we can conclude that RBS deployment had made significant decrease in KPI rejections of the company. Average time consumed by the user to do corrective action and by auto-correction rules of RBS is significantly different. Briefly, the study reveals cost and time efficiency of RBS deployed in health care IT Company. Deployment of RBS has considerably saved the working time of the employees and has improved the revenue cycle of the company by reducing claim aging days. Other benefits such as performance assessment of operation department (using RBS claim blocking report) are also being provided by the system.

This study demonstrates the "benefit" (in the form of time saved) as a proposed measure of evaluation of real-time performance of an RBES, which has been calculated as the sum of the benefits provided by individual rules present in the KB. As KB of a system changes with use therefore periodic reevaluation of the system has been recommended.

Previous studies in the area of ES evaluation mostly focus on verification and validation issues with an end goal of 100% accurate system, which is arguably not possible in the ever-changing real world environment. Even if a perfect system is constructed, it would be worthless if never used in the real-world operational environment. Technique and method proposed and demonstrated in this research paper (for performance evaluation of the RBES applied in medical billing domain) can effectively be used for all types of ESs, especially RBESs.

ACKNOWLEDGEMENT

This research work has been supported by Higher Education Commission of Pakistan under "5000 indigenous PhD Fellowships Scheme." Warming thanks to Health care IT Company "MaxRemind" (http://www.mremind.com) for providing excellent research environment.

REFERENCES

3M Audit Expert System. (2013, October 1). Retrieved November 1, 2015, from Health Information System Website: http://multimedia.3m.com/mws/media/218379O/3m-audit-expert-system-fact-sheet.pdf

Abdullah, U. (2012). *Data mining driven learning apprentice system for medical billing compliance*. Foundation University, Islamabad, FUIEMS. Islamabad, Pakistan: Higher Education Commission of Pakistan URL: http://eprints.hec.gov.pk/9612/1/1841S.htm.

Abdullah, U., Ahmed, A., & Sawar, M. J. (2009). Design of a rule-based system using structured query language. In *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC09)* (pp. 223–228). Chengdu, China: IEEE Press.

Abdullah, U., Ahmed, A., Asghar, S., & Zafar, K. (2015). Data mining driven rule-based expert system for medical billing compliance: A case study. In *Improving knowledge discovery through the integration of data mining techniques* (pp. 267–296). Hershey, Pennsylvania, USA: IGI Globel.

Bench-Capon, T., Castelli, D., Coenen, F., Devendeville-Brisoux, L., Eaglestone, B., Fiddian, N., … Vermesan, A. (1999). Validation, verification and integrity issues in expert and database systems. *Expert Update*, *2*(1), 31–35.

Coenen, F. (1998). Verification and validation in expert and database systems: The expert systems perspective. In *A key note presentation in ninth European international workshop on database and expert systems applications* (pp. 16–21). Vienna, Austria: IEEE.

Coenen, F., Bench-Capon, T., Boswell, R., Dibie-Barthélemy, J., Eaglestone, B., Gerrits, R., et al. (2000). Validation and verification of knowledge-based systems: Report on eurovav99. *The Knowledge Engineering Review*, *15*(2), 187–196.

Filip, C., Vanthienen, J., & Baesens, B. (2013). Comprehensive rule-based compliance checking and risk management with process mining. *Decision Support Systems*, 1357–1369.

Giarratano, J. C., & Riley, G. (2005). *Expert systems*. Boston, MA, USA: PWS Publishing Co.

Giurca, A., Gasevic, D., & Taveter, K. (2009). *Handbook of research on emerging rule-based languages and technologies: Open solutions and approaches*. New York: Information Science Reference.

Halle, B. V., & Ronald, G. (2001). *Business rules applied: Building better systems using the business rules approach*. New York, NY, USA: John Wiley & Sons, Inc.

Hopgood, A. A. (2001). *Intelligent systems for engineers and scientists* (2nd ed.). Boca Raton London New York Washington, D.C.: CRC Press.

Jackson, P. (1999). *Introduction to expert systems* (3rd ed.). New York, NY, USA: Addison–Wesley.

Jones, K. (2014, March 1). Denial management process. Retrieved Nov 16, 2015, from Healthcare Management Systems: http://www.hcmsnapa.com/wp-content/uploads/2012/02/HCMS-DenialManagement-2-24-12.pdf

Liebowitz, J. (1998). *The handbook of applied expert systems*. Boca Raton: CRC Press.

Ligęza, A. (1998). Towards logical analysis of tabular rule-based systems. In R. R. Wagner (Ed.), *Proceedings of the ninth European international workshop on database and expert systems applications* (pp. 30–35). Vienna, Austria: IEEE Computer Society.

Ligęza, A. (1999). Intelligent data and knowledge analysis and verification; towards a taxonomy of specific problems. In *Validation and verification of knowledge based systems: Theory, tools and practice* (pp. 313–325). Boston, Dordrecht, London: Kluwer Academic Publishers.

Ligęza, A. (2001). Toward logical analysis of tabular rule-based systems. *International Journal of Intelligent Systems*, *16*(3), 333–360.

Ligęza, A. (2006). *Logical foundations for rule-based systems*. Berlin, Heidelberg: Springer-Verlag.

Ligęza, A., & Nalepa, G. J. (2011). A study of methodological issues in design and development of rule-based systems: Proposal of a new approach. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2* (1), 117–137.

Monegain, B. (2009, November 13). Athenahealth's rules engine scores a patent. Retrieved January 16, 2016, from http://www.healthcareitnews.com: http://www.healthcareitnews.com/news/athenahealths-rules-engine-scores-patent

Nalepa, G. J. (2009). Languages and tools for rule modeling. In A. Giurca, D. Gaševi'c, & K. Taveter (Eds.), *Handbook of research on emerging rule-based languages and technologies: Open solutions and approaches* (pp. 596–624). Hershey, New York: IGI Global.

Negnevitsky, M. (2002). *Artificial intelligence. A guide to intelligent systems*. Harlow, England; London; New York: Addison-Wesley.

Norvig, P., & Russell, S. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Prentice-Hall.

O'keefe, R. M., & Preece, A. D. (1996). The development, validation and implementation of knowledge-based systems. *European Journal of Operational Research*, *92*(3), 458–473.

Pocatello, I. (2015, March 10). Claim inspector. Retrieved January 15, 2016, from http://www.advancedmd.com: http://www.advancedmd.com/uploads/files/cs-tmj-sleepcenter.pdf

Preece, A. D. (1991). Methods for verifying expert system knowledge bases. Concordia University Canada: Report for Bell Canada, Centre for Pattern Recognition and Machine Intelligence.

Preece, A. D. (1992). Verification, validation, and test of knowledge-based systems. *AI Magazine*, *13*(4), 77.

Preece, A. D., Grossner, C., & Radhakrishnan, T. (1996). Validating dynamic properties of rule-based systems. *International Journal of Human-Computer Studies*, *44*(2), 145–169.

Suwa, M., Scott, C. A., & Shortliffe, E. H. (1982). An approach to verifying completeness and consistency in a rule-based expert system. *AI Magazine*, *3*(4), 16.

Tepandi, J. (1990). Verification, testing, and validation of rule-based expert systems. In *Proceedings of the 11-th IFAC World Congress* (pp. 162–167). Tallinn, Estonia: IFAC.

Tomczak, J. M., & Gonczarek, A. (2013). Decision rules extraction from data stream in the presence of changing context for diabetes treatment. *Knowledge and Information Systems*, 521–546.

Vanthienen, J., Mues, C., Wets, G., & Delaere, K. (1998). A tool-supported approach to inter-tabular verification. *Expert Systems with Applications*, *15*(3), 277–285.

Vermesan, A. (1997). Foundation and application of expert system verification and validation. In *The handbook of applied expert systems* (pp. 277–295). Boca Raton, FL: CRC Press.

Vermesan, A., & Coenen, F. (1999). *Validation and verification of knowledge based systems. Theory, tools and practice*. Boston: Kluwer Academic Publisher.

Umair Abdullah is working as an associate professor at Barani Institute of Information Technology, Rawalpindi, Pakistan ("http://www.biit.edu.pk"). Prior to this, he has served as an assistant professor at Foundation University Islamabad, Pakistan. He has 15 plus years of teaching experience and 7 years of industry experience. His research interests include rule-based expert systems, knowledge representation, and data mining.

Antoni Ligęza is working as a professor of Computer Science at AGH University of Science and Technology in Kraków, Poland (http://home.agh.edu.pl/~ligeza/wiki/). He is also serving as vice president of the Polish Artificial Intelligence Society. His research is focused on knowledge engineering, rule-based systems, automated diagnosis, attribute logic, and automated planning.

Kashif Zafar is working as a professor and head of the Computer Science Department at National University of Computer and Emerging Sciences, Lahore, Pakistan (http://www.nu.edu.pk/campus/Employee/Kashif-Zafar_1238). Before joining FAST-NU, he worked as research associate or lecturer in the Faculty of Computer Science and Engineering at GIK Institute of Engineering Sciences and Technology Topi, Pakistan. He has 18 plus years of experience both in industry and academia. His research interests include machine learning, swarm intelligence, computational intelligence, evolutionary algorithms, data mining, expert systems, and so forth.

How to cite this article: Abdullah U, Ligęza A, Zafar K. Performance evaluation of rule-based expert systems: An example from medical billing domain. *Expert Systems*. 2017;0:e12218. https://doi.org/10.1111/exsy.12218