ORIGINAL ARTICLE

Real-Time Automated Hazard Detection Framework for Health Information Technology Systems

Olufemi A. Omitaomu^a, Ozgur Ozmen^a, Mohammed M. Olama^a, Laura L. Pullum^b, Teja Kuruganti^a, James Nutaro^a, Hilda B. Klasky^a, Helia Zandi^a, Aneel Advani^c, Angela L. Laurio^d, Merry Ward^d, Jeanie Scott^d and Jonathan R. Nebeker^d

^aComputational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA; ^bComputer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA; ^cHealth Policy and Management Department, Johns Hopkins University, Baltimore, MD, USA; ^dOffice of Health Informatics, U.S. Department of Veterans Affairs, Washington, DC, USA

ABSTRACT

An increase in the reliability of Health Information Technology (HIT) will facilitate institutional trust and credibility of the systems. In this paper, we present an end-to-end framework for improving the reliability and performance of HIT systems. Specifically, we describe the system model, present some of the methods that drive the model, and discuss an initial implementation of two of the proposed methods using data from the Veterans Affairs HIT and Corporate Data Warehouse systems. The contributions of this paper, thus, include (1) the design of a system model for monitoring and detecting hazards in HIT systems, (2) a data-driven approach for analysing the health care data warehouse, (3) analytical methods for characterising and analysing failures in HIT systems, and (4) a tool architecture for generating and reporting hazards in HIT systems. Our goal is to work towards an automated system that will help identify opportunities for improvements in HIT systems.

ARTICLE HISTORY

Received 4 September 2018
Accepted 21 March 2019

KEYWORDS

Health information technology; hazard detection; statistical process control; Markov chain; transaction process model; corporate data warehouse

1. Introduction

Health information technology (HIT) is an integral part of large-scale health care systems. The term *HIT* describes the various information and communication technologies used to collect, transmit, display, and store patient data (Sittig & Singh, 2011). Although HIT systems, such as electronic health records (EHRs), have evolved to increase the quality and safety of health care services and decrease health care overhead (Bates & Gawande, 2003; Coorevits et al., 2013; Rauscher, 2000; Ray, McEvoy, Aaron, Hickman, & Wright, 2018); recent studies have revealed possible *safety hazards* related to the use of HIT (e.g., IOM, 2012; Bowman, 2013; Brenner et al., 2016; Castro et al., 2015; ECRI, 2012; Mardon et al., 2014; Sparnon, 2013; Sparnon & Marella, 2012). Thus, there have been calls to provide a means to study HIT safety measurement, monitoring, and improvement (Singh & Sittig, 2016; Sittig & Singh, 2015).

A hazard is any condition, event, or circumstance that could induce patient harm. A HIT hazard is a hazard caused, precipitated, or propagated by HIT systems. A *safety hazard herein is defined as any activity that exhibits abnormal pattern(s) with respect to other activities in the same state and/or within a time period of interest*. This definition provides an opportunity to discover new hazards not previously known or detected. The safety concerns in HIT systems herein are

categorised into three perspectives: (1) concerns unique or specific to malfunctioning hardware or software (unsafe technology), (2) concerns related to misuse of HIT (unsafe use of technology), and (3) concerns related to lack of monitoring of safety concerns. These concerns cover hazards that pose immense threats to patient safety as reported in multiple studies (e.g., Graber, Johnston, & Bailey, 2016; Magrabi, Ong, Runciman, & Coiera, 2011; Meeks et al., 2014; Wright et al., 2016).

Related works are noted herein that focus on addressing the foregoing safety concerns by targeting specific, known hazards in the respective HIT system. To address safety concerns in HIT, this paper describes a more holistic approach to hazard detection and reporting. The proposed approach does not assume any specific hazard but rather detects potential hazards and allows analysts to confirm whether or not the hazards are of true concern. The approach then learns from the analysts' choices for future detection processes. The details of such development are presented following a review of some prior work dedicated to resolving safety concerns in HIT.

2. Related work

Detecting harm to patients via a retrospective analysis of EHR data is the first step in understanding what

happened and why. Unfortunately, with this approach, which is the current practice, patient harm occurs before it is detected and before the hazards leading to the unsafe event are recognised. This is a reactive detection and mitigation process. Before hazard detection, there is a need to (1) prevent the introduction of hazards and (2) even earlier, detect and prevent precursors to hazards. Previous work classifies HIT hazards as data-related hazards, human–computer interface hazards, and software/hardware hazards (e.g., Salahuddin & Ismail, 2015; Kim, Coiera, & Magrabi, 2017; and references therein). To detect HIT hazards, surveillance of the patient database should involve the following:

- Data-driven hazard-detection approaches to identify anomalous patterns.
- Surveillance analytical approaches to detect emerging behaviours in the system that might result in unfavourable outcomes in patient safety.

To avoid possible failures in HIT systems, possible hazards should be identified along with their causes and their effects on the system. Human-HIT interaction frequently leads to errors due to insufficient training of staff to operate the HIT, the heavy cognitive load, multi-tasking, and interruptions from operating the system (Kim et al., 2017). As is the case with humans, machines can also produce errors (Bates et al., 2001). Machine errors can be broad: general hardware and software malfunctions are common, and machine errors are possible when a user works with the interface. Commonplace issues include hardware malfunctions or slow functioning, software problems, power failure, computer viruses, or a system unable to handle an increased transaction volume (Kim et al., 2017). Machine errors can create malfunctions across a hospital if they occur frequently, so the solution must include having a hazard-detection capability in health care software and hardware systems.

In addition, complete communication and collaboration among all the entities in health care are essential to address identified concerns and improve the safety of HIT, making it a challenging problem. For example, conducting contextually rich studies of clinical workflow, recruiting partners for usability studies, and garnering support from vendor organisation leadership are all difficult tasks (Ratwani, Fairbanks, Hettinger, & Benda, 2015). Moreover, HIT vendors have economic incentives to keep their design and implementation in-house (Flows, 2017). A significant number of root causes of detected hazards boils down to the design and development stages of the software life cycle (Bowman, 2013; Horsky et al., 2012; Ratwani et al., 2015; Wright et al., 2016).

Hence, there is a need for better approaches for monitoring large-scale HIT platforms such as EHRs so that timely and effective actions can be taken against

emerging detrimental operational issues. The importance and necessity of having a hazard-detection capability in HIT systems have been investigated in the literature. In a study by Adelman et al. (2012), a framework was proposed to provide a conceptual foundation for HIT-related patient safety measurement, monitoring, and improvement. A simple method was designed to prevent wrong-patient electronic orders by using a retract-and-reorder tool. As an illustration, if a medication order for a patient is retracted before it is filled, and the same order is submitted for another patient, this scenario is categorised as a wrong-patient order. An ID-verify alert was added to the system so that providers can place the order only if they acknowledge ordering it for the right patient. Based on the experiments by Adelman et al., this method leads to 50% reduction in the odds of a retract-and-reorder session and reduces wrong-patient orders by 16%. In another study, a model was introduced to determine whether an alarm system was working properly (Wright et al., 2016). This research investigated the alert firing logs, alert rule logic, alert system configuration, and audited data. The researchers communicated with users and system developers to perform code reviews/audits. Four alerts were chosen, and data associated with each of those alerts was extracted. For each alert, the data were divided into two categories: (1) data collected during the malfunction and (2) data collected before/after the malfunction. A linear model was fitted to the data to determine when the volume of alerts increased (weekends or weekdays and the time of day). The model was then used to predict the number of times the alert would fire. Based on this model, it was found that for some period more alerts were predicted than were fired. Additional analysis was conducted to investigate why there were fewer alerts than expected.

These studies, like several others in the literature, targeted specific and known hazards in the respective HIT system. In this paper, we are interested in a more holistic approach to hazard detection and reporting. Our approach does not assume any specific hazard but rather identifies and detects potential hazards and allows analysts to confirm whether or not the hazards are of true interest. This approach then learns from the analysts' choices for future detection processes.

3. System model

The objective of the proposed framework is to develop a proof-of-concept system that will promote HIT patient safety. Specifically, we propose an end-to-end system for monitoring and detecting hazards in HIT. The schematic model for the proposed end-to-end HIT hazard-detection system is shown in Figure 1. The model consists of three components: (1) data extraction and pre-processing (DEP), (2) hazards monitoring and detection (HMD), and (3) report generation and feedback (RGF). The HMD component, in

Figure 1. Components of an end-to-end HIT hazard-detection system.

turn, comprises of two layers: event-based hazards detection and system-based reliability performance evaluation.

The proposed system model has the following capabilities:

- Data-driven hazard characterisation and detection methods to structurally query EHR data and detect anomalous patterns that might constitute hazards. These methods are implemented as a near real-time detection tool that is expected to run hourly/daily on an EHR database.
- Analytical methods that evaluate the system-level performance of HIT systems to detect emerging behaviour in the system that might result in unfavourable outcomes in patient safety. This part is implemented as a surveillance tool that is expected to run weekly/monthly on an EHR database.

The EHR systems that are used to test the proposed framework are the Veterans Affairs (VA) HIT systems, named VistA¹ and the VA Corporate Data Warehouse (CDW²) systems. This paper describes some of the initial solutions developed as part of this proposed end-to-end system.

4. Methods

This section describes the methods and techniques that drive each of the components in the system model.

4.1. Data extraction and pre-processing

The DEP component focuses on extracting relevant data from the CDW. Data extraction was initially

performed by analysis of VistA software packages. A subsequent identification of relevant data fields and flows has helped create the Transaction Process Model (TPM) shown in Figure 2a. This analysis considers two system relationships: (1) VistA user interactions (i.e., menu options and graphic user interface dialogues) and (2) VA FileMan³ database files and CDW tables. This step identifies the flow of information in a given data domain (e.g., consult, laboratory, radiology, and medications).

The DEP component includes two data mapping approaches. The first data-mapping approach involves monitoring a demo VistA server instance and provides an understanding of how the data is manipulated in the FileMan database and the VistA user interface. A generic TPM of a clinical consult is shown in Figure 2b using FileMan instances for relationship mapping.

This generic model can be leveraged to represent each data domain (consults, radiology, laboratory services, etc.) with its specific sub-processes and sub-states. For example, a “consult” has an *Order Entered* step that is followed by an *Order Signed* step in Figure 2b. Once an object is accepted, its status becomes *active*, and once it is scheduled its status changes to *scheduled*. We observed in our case of study that the workflow status terminology varies for different data domains. For instance, a laboratory order becomes *active* instead of being *scheduled*.

To gain insight on how information flows in VistA and the VA FileMan database, we developed an automated web crawler, which was implemented to extract field information for each VA FileMan file from the code repository documentation of the Open Source Electronic Health Record Alliance (OSEHRA) – <http://code.osehra.org/dox/>. The extracted information

Figure 2. (a) Generic transactional process model. (b) Information flow between FileMan files for a consult order.

includes the field name, data structure of the field (from which we can infer how it is populated), and whether or not it is a required field.

The second data-mapping approach converts data fields of interest in VA FileMan files and fields to CDW tables and columns. This dataset also includes some flags such as *required*, *human entered* (free texts), *human selected* (sets), and *system generated* (pointers and dates), along with the field and column information for all relevant data fields that are populated or revised during transactions. Additionally, we computed the proportion of rows that were not NULL (not missing) for each column in CDW.

4.2. Hazards monitoring and detection

The HMD component comprises of two layers: event-based hazards detection and system-based reliability performance evaluation.

4.2.1. Event-based hazards detection

This layer uses data-driven approaches to identify potential hazards for further investigation. The layer assumes that hazards are not known a priori. This assumption, thus, provides not only an opportunity to discover previously unknown or undetected hazards but also allows for analysts to remain in the loop. The analysts can review the detected hazards and determine whether they are true hazards or just noise in the data. These reviews will serve as feedback to the hazard-detection methods so that future occurrences are characterised correctly. This is a dynamic approach in which the detectors are continuously trained and improved using the analysts' feedback as additional data points. This approach is

analogous to the concept of incremental learning in machine learning literature. Using this approach, we developed a series of hazard detectors (HDs); one HD called *anomalous cancelled orders* is discussed in this paper.

The *anomalous cancelled orders* detector is designed to detect anomalous number of cancelled radiology orders using a statistical process control (SPC) technique. This detector uses an online process monitoring approach that detects variability in the number of cancelled orders data due to avoidable causes. A typical SPC technique is a graphical display of measured or calculated process characteristics with a centre line representing the mean of the in-control state as well as the upper control limits for allowable deviations around the mean. For this study, the daily number of cancelled orders (at a specific clinical institution) outside the SPC limit is flagged as a potential hazard. The SPC upper control limit is calculated as the mean number of orders plus three standard deviations. Although the detector runs daily, the mean and standard deviations are calculated using data for the last one year. For operational purposes, data for the weekdays and weekends are treated differently; that is, each category has a different mean and standard deviation. Previously detected anomalous cancelled orders are removed while determining the mean and standard deviations for future detections. When the detector is operational, the safety users will flag the correct detected hazards and this feedback will allow us to revise our decision criterion. Additionally, we will be able to apply more advanced supervised learning techniques on flagged data. Similar detectors on other consult activities (e.g., released and completion) and other data domains (such as auto-created radiology orders)

are also being implemented. A similar approach can be implemented for rejected orders.

4.2.2. System-based reliability performance evaluation

This layer focuses on the overall HIT system. The assumption for this layer is that it is possible for the available detectors in the first layer to miss some hazards; therefore, there is a need for an additional layer that will survey the overall HIT system for emerging degradation in system reliability as a result of hidden or emerging hazards. Such a surveillance layer will provide insights that might lead to further investigation or the development of new detectors. In this paper, we present a Markov chain (MC) model to understand system behaviours that might impact HIT reliability. This model is called *system-surveillance model*.

The MC model is a widely used analytical tool for studying the behaviour of a dynamic system in several applications including health care (e.g., Carayon et al., 2016). The MC is a discrete-time stochastic process, where the conditional probability of any future event depends only on the present state and is independent of past states (Serfozo, 2009). The Markovian property can be expressed as follows for all states $i_0, i_1, i_2, \dots, i_{t-1}, i_t$ and all time $t \geq 0$:

$$\begin{aligned} P(X_{t+1} = i_{t+1} | X_t = i_t, X_{t-1} = i_{t-1}, \dots, X_0 = i_0) \\ = P(X_{t+1} = i_{t+1} | X_t = i_t) \end{aligned} \quad (1)$$

Since the MC assumes that the conditional probability does not change over time, for all states i and j and all t , $P(X_{t+1} = j | X_t = i)$ is independent of t , as expressed in Equation (2):

$$P(X_{t+1} = j | X_t = i) = p_{ij}, \quad (2)$$

where p_{ij} is the transition probability from state i at time t to state j at time $(t + 1)$.

Specifically, we propose a time-inhomogeneous model using discrete-time MC (DTMC) to study system behaviour in HIT. Our *system-surveillance model* is motivated by an application of DTMC to high-performance computing systems (Dabrowski & Hunt, 2009). This model first defines a clinical workflow as a TPM. The TPM represents a multitude of actions by different individuals and practitioners as the clinical object is processed along the clinical flow. As an illustration, the state model for clinical consult TPM can be represented as shown in Figure 3. The DTMC is an approach of choice because it can be used to describe a detailed set of states that a system may enter and through which it can progress along different paths as found in clinical workflows.

Consequently, the TPM is then converted into a MC representation. The MC model captures the dynamics of the EHR system by using the transition counts (frequency of transition between states or nodes in the MC model) to represent execution paths in the EHR. The transition counts are used to compute the individual transition probabilities in the state transition matrix (STM). Each cell in the STM is thus the probability of transition from state i at time t to state j at time $(t + 1)$, written as $s_i \rightarrow s_j$. The probability of transitioning between any two states s_i, s_j , written as p_{ij} was estimated by computing the frequency of $s_i \rightarrow s_j$, denoted by f_{ij} , divided by the sum of frequencies of transitions from s_i to all states s_k to which s_i could transition, where k ranges from 1, ..., n and n is the number of states.

$$p_{ij} = \frac{f_{ij}}{\sum_{k=1}^n f_{ik}}. \quad (3)$$

Repeating the computation for all $i, j = 1, \dots, n$ will result in an $n \times n$ STM that concisely summarises the dynamics of the underlying TPM. The STM provides information about the probabilities of condition

Figure 3. State model for clinical consult TPM. The colour of the nodes depicts the state of the node. This representation has one starting node and two ending nodes, which is a simplification of the detailed consult TPM.

changes and can be used for prediction of the time of condition changes in a system.

The DTMC approach has been implemented for two scenarios:

- (1) A scenario of situational awareness capability: In this case, we periodically compute the transition count between states in a clinical workflow (or nodes in the MC model) to understand implications for system outputs.
- (2) A scenario of evaluation capability: In this case, we use a sequence of MC STM to simulate how a dynamic system changes by altering or perturbing the values in the STM for a fixed time duration. This will help identify cases that might impact overall system reliability. Consequently, the proposed DTMC model can be used to predict how the modelled HIT system will react when key system transition probabilities are changed. The hypothesis here is that small variations in key state variables can lead to alternative execution paths that will yield large differences in overall system reliability. Thus, large-scale simulations can help to accurately reproduce system structure and detect component behaviour that might be of concern.

4.3 Report generation and feedback

The RGF component is achieved using a software platform presented as a decision support tool to the analysts. The prototype houses the capabilities of the other two components described previously.

The platform consists of five primary parts: (1) client user interface, (2) training user interface, (3) hazard events database and workflow engine, (4) plug-gable set of HDs, and (5) HD manager and execution engine. A conceptual view of the tool architecture is shown in Figure 4.

The *client user interface* is the interface to the analysts. Through this interface analysts are presented with current information on hazards and their types and time of most recent detection (the Summary view). Upon selecting a hazard type, analysts are presented

with a map depicting the locations of the hazards, with a table indicating the numbers and locations of detected hazards (the Details view). Analysts can explore hazard(s) by selecting the hazard and using the Explore view, which presents a depiction of the reasoning behind the event (or set of events), in addition to more details related to the algorithm employed in the detector (e.g., the anomalous cancelled order detector). The analysts can further designate whether they agree with the designation of the displayed event(s) as a hazard and by doing so can also enter the rationale for confirmation of the hazard.

The *training user interface* is used to add new detectors to the HD library, as well as to set up data pre-processing as needed for the detector and then for training the detector. The *hazard event database* stores and the *workflow engine* manages the events posted by the HDs, and the details thereon. This data is used by analysts, via the client user interface described previously, to explore, accept, or reject detected potential hazards. The *HD manager and execution engine* enacts hazard activation, periodicity, and other aspects of the HDs. Associated with the HD manager and execution engine is a plug-and-play set of HDs that users can orchestrate via the engine through the user interfaces. Each detector has meta-data about its intended capabilities.

The tool platform uses a scalable microservices-based architecture to implement the capabilities described previously. For example, each detector is implemented in its own Jupyter Notebook (or C++/Java/Python/etc. code stand-alone module). Microservices for various logical routines within a detector have been developed without breaking the natural flow of the development process of the detectors. These routines are to start, prepare, compute, report, and stop a detector. The benefit of using microservices in this manner is to enable rapid microservices development of HDs with one shared codebase, while achieving compartmentation and enabling resiliency and reusability of the detectors.

The overall microservices architecture approach also enables scalability of the surveillance tool to handle the envisioned very large volume of data to

Figure 4. Conceptual view of an automated tool architecture.

be consumed by the detectors. In addition, the microservices are designed and implemented in such a way that this aspect of the architecture is seamlessly scalable. The microservices are implemented adhering to the representational state transfer principle, yet can only be accessed by the localhost, providing a measure of security.

5. Case studies

In this section, we present two case studies – *anomalous cancelled orders detector* and *system-surveillance model*.

5.1. Case study 1: anomalous cancelled orders detector

In this case study, we are interested in automatically detecting high numbers of cancelled/rejected orders in HIT systems. During our exploratory analysis, we observed that some of the radiology orders submitted into HIT systems are cancelled/rejected daily for various reasons including duplication of orders, human error, and/or machine error. However, this high number of cancelled orders is not visible to the system analysts. If orders are cancelled/rejected for valid reasons, this action is not regarded as a hazard, but when orders are cancelled/rejected for invalid reasons or if a significant number of orders are cancelled/rejected even for valid reasons, this action can be flagged as a hazard pending investigation by analysts.

The data points shown in Figure 5 represent the number of daily cancelled/rejected radiology orders observed at a clinical location for a study period of one year. Since the initial implementation does not have any labelled data, we detect

outliers in the data using the upper bound in a SPC technique as threshold (Figure 5). Data points above the threshold are regarded as outliers. Only the outliers confirmed by safety analysts are regarded as anomalous data points and then removed from the dataset.

In Figure 6, the cleaned data (data with no confirmed outliers) is used to re-compute the new mean and the new standard deviation. The data points below the Control Limit line represent the data used for SPC, and the data point above the Control Limit line represents anomalous data points. For example, the red dot in Figure 6 represents the latest anomalous number of cancelled orders that occurs on January 19. The respective clinical location is then reported to the safety analysts for further investigation.

5.2. Case study 2: system surveillance model

We used MC to represent the dynamic states of orders submitted to a HIT system. The MC is used to model the progress of a large number of orders from the time they are submitted to the time they are either completed or cancelled. In this paper, we represent the life cycle of a consult order by the seven (high-level) states shown in Figure 3. Some of the states in the model can be further decomposed into sub-states. For simplicity, we used only the high-level states for the analysis in this section. These states are “input”, “accept”, “schedule”, “revision”, “release”, “discontinue/cancel”, and “complete”. The state-to-state transition probabilities (TPs) are extracted using Equation 3 and are shown in Table 1.

Looking at Table 1, we can conclude that we have two absorbing states – cancel and complete – while

Figure 5. Initial outlier detection example for case study 1.

Figure 6. Detection of anomalous number of radiology order cancellations at a clinical location on January 19 2017.

Table 1. State-to-state transition probabilities for consult object over a period of time.

	Input (1)	Accept (2)	Schedule (3)	Revision (4)	Release (5)	Cancel (6)	Complete (7)
Input (1)	0.03	0.97	0	0	0	0	0
Accept (2)	0	0.05	0.84	0.11	0	0	0
Schedule (3)	0	0	0.03	0.03	0.70	0.04	0.20
Revision (4)	0	0	0.80	0.20	0	0	0
Release (5)	0	0	0	0.50	0.20	0.30	0
Cancel (6)	0	0	0	0	0	1.0	0
Complete (7)	0	0	0	0	0	0	1.0

Figure 7. Transition diagram for the MC model with seven high-level states based on Table 1.

the other states are transient states. The matrix in Table 1 can be illustrated as the transition diagram in Figure 7. Using this data (Table 1) and configuration (Figure 7), we obtained the results discussed in the following subsections.

5.2.1. Simulating the dynamic behaviour of submitted orders in CDW

A well-known use of stochastic STMs in a DTMC is to describe how a dynamic system changes over time in discrete time steps, where each time step represents a fixed time duration. In our experiment, we simulated 100 random walks through the chain shown in Figure 7 then compared the simulated and theoretical STMs for the chain. The STM is shown as a coloured chart. For the results shown in Figure 8, we assume that the 100 simulated walks started at state 1. For the results shown in Figure 9, we assume that the 100 simulated walks started at different states (i.e., 20 walks started from each of the transient states); this assumes that no new order can start from any of the absorbing states because it will never leave that state. This assumption is based on our experience in exploring the data in CDW. The theoretical and simulated STM for this scenario is shown in Figure 9.

As expected, the theoretical STMs in Figures 8 and 9 are the same since the model remains the same. However,

Figure 8. STM for 100 random walks starting at state 1. (a) Theoretical STM and (b) simulated STM.

Figure 9. STM for 100 random walks with different number of walks starting at different states. (a) Theoretical STM and (b) simulated STM.

a closer look at the figures shows that the simulated STMs are slightly different but still within margins of error. Therefore, we can come up with the same conclusions based on either the theoretical or simulated STM in either case. One use case of this result is that this approach could help predict the state of an object after a certain time duration as well as the probability that an object might be cancelled or completed after a finite time duration.

5.2.2. Perturbation of transition probability to enhance system performance

For this application, we are interested in changing the MC model by altering the transition probabilities between states to explore alternative execution paths. This allows analysts to see how a system might behave when certain events occur or conditions of operation

change. Using this information, the analyst can identify *failure scenarios* in which system reliability could be affected catastrophically. Perturbation analysis of DTMCs has been extensively studied in theoretical literature (e.g., Schweitzer, 1968; Delebecque, 1983; Hassin & Haviv, 1992) and computational literature (e.g., Meyer, 1989; Stewart & Dekker, 1994; Ho & Li, 1988; Suri, 1989; Cao & Zhang, 2008; Dabrowski & Hunt, 2009).

The perturbation algorithm adapted from Dabrowski and Hunt (2009) requires that a user first select a *primary row* in the STM to perturb; this is the row of interest and where the effects from proposed changes in the system are expected to show. The *secondary rows* to be perturbed are then automatically determined; these are the non-zero probability columns corresponding to the primary row. In addition, the user selects a *perturbation limit*,

Figure 10. Probability distribution of states for 20 random walks. (a) Using the original MC model and (b) using the first modified MC model.

Figure 11. Probability distribution of states for 20 random walks. (a) Using the original MC model and (b) using the second modified MC model.

L – the maximum probability value for the perturbed column. Using the primary row, the numerically lowest row element is found, and the transition probability is increased by an increment amount, ν , up to L . At the same time, the other elements in the primary row are reduced by the proportion of ν using user-defined weight factors. For each perturbation combination, a separate perturbation analysis is carried out in the primary row. The systematic perturbation of the STMs will reveal a wide range of behaviours. In this section, we present a subset of these behaviours.

Increasing the number of scheduled orders that get released: According to Table 1, 4% of orders in “scheduled” state get cancelled and another 20% get completed. In this scenario, we are interested in reducing the number of cancelled orders by increasing the probability of orders being “released”. Hence, the probability of transition from schedule to release is increased from 0.70 to 0.74. Consequently, the probability of transition from schedule to cancel is decreased from 0.04 to 0 as shown in Table 2 for the “schedule” row.

Table 2. State-to-state transition probabilities for the first modified model.

	Input (1)	Accept (2)	Schedule (3)	Revision (4)	Release (5)	Cancel (6)	Complete (7)
Schedule (3)	0	0	0.03	0.03	0.74	0	0.20

Table 3. State-to-state transition probabilities for the second modified model.

	Input (1)	Accept (2)	Schedule (3)	Revision (4)	Release (5)	Cancel (6)	Complete (7)
Schedule (3)	0	0	0	0	0.70	0	0.30

Using this modified model, we simulated 20 random walks through the chain and compared the probability distribution for the states. Figure 10 shows that distribution of states for the original model (Figure 10a) and the modified model (Figure 10b).

While the transition probability for “schedule to cancel” is decreased, the probability distribution of states for the overall first modified MC model shown in Figure 10 reveals a slight increase in the distribution probability for the complete state and a slight decrease in distribution probability for the cancel state. Although a 4% reduction in transition probability for the “schedule” row is significant, this strategy does not translate to a significant overall benefit. Hence, this scenario might not be effective by itself, but it might be more effective in combination with other scenarios.

Increasing the number of scheduled orders that get completed: For this scenario, we increase the TP for schedule to complete from 0.2 to 0.3 and set other transitions to zero except the TP from schedule to release, as shown in Table 3 for the “schedule” row.

For this scenario, the distribution of states for the overall second modified MC model shown in Figure 11 reveals that the probability is about 0.45 for both states 6 and 7; the probability for states 3–5, respectively, is slightly reduced compared with the original model. This is an increase in the number of completed orders and a reduction in the number of cancelled orders. Hence, this scenario might be more effective than the first modified MC model.

6. Discussions

An increase in the reliability of HIT system will facilitate institutional trust and credibility of the systems. The system model presented the early stages at developing new approaches for improving the reliability of HIT systems.

6.1. Strengths

The proposed *system model* comprises of three components that allowed for: (i) a multiple-stage hazard-detection framework and (ii) safety analysts involvement in the decision process. The iterative nature of the model allows for updating the logic behind the existing detectors and new insights to develop new detectors. Working with the VA’s VistA and CDW systems provides a platform for working with a large-scale EHR databases. Furthermore, the *anomalous*

cancelled order detector was shown to detect previously unknown hazards related to cancelled/rejected orders. Similar methods could be used for datasets with similar characteristic features in HIT systems. The *system surveillance model* gives encouraging results in identifying critical state transition in a clinical workflow modelled as a MC; this could serve as a basis for describing potential failure scenarios or reliability degradation scenarios.

6.2. Limitations

While working with the VA’s EHR systems is a big strength, it also poses a scalability limitation. One limitation is the ability to scale the developed model and detectors to other EHR systems. To achieve a more generalised model, the architecture for the RGF component is designed to accommodate the integration of a different EHR system. We will explore opportunities to test our model and detectors using other EHR systems.

The *anomalous cancelled order detector* provides some encouraging results, but this detector can be improved by accounting for seasonality in the data. While we have different models for weekdays and weekends respectively, the detectors may still miss some anomalous number of cancellations if they do not account for seasonality. We are currently investigating if there is a need for detectors that account for daily, weekly, or monthly variability in the number of cancelled/rejected orders data.

One limitation of the *system surveillance model* is the computational time required to estimate the probability distributions for all possible sets of perturbation combinations. Carrying out the sequence of perturbations for each perturbation combination can yield potentially interesting alternative behaviours but will be computationally expensive. Usually, though, only a subset of the perturbation combinations may be relevant; therefore, we are exploring methods for identifying critical transitions without evaluating all possible perturbation combinations.

7. Conclusion

This paper presents an end-to-end framework for detecting hazards and for monitoring the reliability of HIT systems using the VA’s VistA and CDW systems. This is a new direction in HIT patient safety research. The overall objective is to enhance the operations of large-scale HIT systems. We present the overall description of the proposed framework, describe the various

components of the system model, present two use cases studies with results. We also point out some limitations and opportunities for improvement and future work. The model and approaches presented are designed to be easily adapted for other EHR systems. This work will enable a safety manager to monitor HIT systems for a variety of hazard signals, diagnose whether the hazard is of interest, and manage the remedy for the hazard. Future studies include the development of unsupervised HDs that assume no prior knowledge of the system or data; such detectors will have the tendency to adjust to changing system configurations.

Notes

1. <https://www.data.va.gov/dataset/veterans-health-information-systems-and-technology-architecture-vista>.
2. <https://www.data.va.gov/dataset/corporate-data-warehouse-cdw>.
3. https://www.va.gov/vdl/documents/Infrastructure/Fileman/fm22_2um1.pdf.

Acknowledgments

This work is sponsored by the US Department of Veterans Affairs. This manuscript has been authored by UT-Battelle LLC under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the U.S. Department of Veterans Affairs.

ORCID

Olufemi A. Omitaomu <http://orcid.org/0000-0002-3078-7196>
 Ozgur Ozmen <http://orcid.org/0000-0002-5806-0116>
 Mohammed M. Olama <http://orcid.org/0000-0002-3890-2064>
 Laura L. Pullum <http://orcid.org/0000-0001-6045-7668>
 Teja Kuruganti <http://orcid.org/0000-0003-3704-4026>
 Hilda B. Klasky <http://orcid.org/0000-0001-7235-2521>
 Helia Zandi <http://orcid.org/0000-0003-3966-7454>
 Aneel Advani <http://orcid.org/0000-0003-0390-9749>

Angela L. Laurio <http://orcid.org/0000-0003-4142-5841>
 Jeanie Scott <http://orcid.org/0000-0003-0831-2239>

References

- Adelman, J. S., Kalkut, G. E., Schechter, C. B., Weiss, J. M., Berger, M. A., Reissman, S. H., ... Southern, W. N. (2012). Understanding and preventing wrong-patient electronic orders: A randomized controlled trial. *Journal of the American Medical Informatics Association*, 20(2), 305–310.
- Bates, D. W., Cohen, M., Leape, L. L., Overhage, J. M., Shabot, M. M., & Sheridan, T. (2001). Reducing the frequency of errors in medicine using information technology. *Journal of the American Medical Informatics Association*, 8(4), 299–308.
- Bates, D. W., & Gawande, A. A. (2003). Improving safety with information technology. *New England Journal of Medicine*, 348(25), 2526–2534.
- Bowman, S. (2013). Impact of electronic health record systems on information integrity: Quality and safety implications. *Perspectives in Health Information Management*, 10(Fall), 1–19.
- Brenner, S. K., Kaushal, R., Grinspan, Z., Joyce, C., Kim, I., Allard, R. J., ... Abramson, E. L. (2016). Effects of health information technology on patient outcomes: A systematic review. *Journal of the American Medical Informatics Association*, 23(5), 1016–1036.
- Cao, X., & Zhang, J. (2008). Event-based optimization of Markov Systems. *IEEE Transactions on Automatic Control*, 53(4), 1076–1082.
- Carayon, P., Ju, F., Cartmill, R., Hoonakker, P., Wetterneck, T. B., & Li, J. (2016). Medication error propagation. In intensive care units. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59(1), 518–521. First published 2016, December 20.
- Castro, G. B., Buczkowski, L., Hafner, J., Barrett, S., Rasinski, K., & Williams, S. (2015). *Investigations of health IT-related deaths, serious injuries, or unsafe conditions*. Final Report. The Joint Commission. Available from: <https://www.healthit.gov/sites/default/files/playbook/pdf/safety-resources-health-it-related-deaths.pdf>
- Coorevits, P., Sundgren, M., Klein, G. O., Bahr, A., Claerhout, B., Daniel, C., ... De Moor, G. (2013). Electronic health records: New opportunities for clinical research. *Journal of Internal Medicine*, 274(6), 547–560.
- Dabrowski, C., & Hunt, F. (2009). Using Markov Chain analysis to study dynamic behaviour in large-scale grid systems. *Proceedings of the 7th Australasian Symposium pm Grid Computing and e-Research (AusGrid 2009)*, Wellington, New Zealand.
- Delebecque, F. (1983). A reduction process for perturbed Markov Chains. *SIAM Journal of Applied Mathematics*, 43, 325–350.
- ECRI. (2012). *ECRI Institute PSO deep dive: Health information technology*. Plymouth Meeting, PA. Available from: <https://www.ecri.org/components/PSOCore/Documents/Deep%20Dive/Deep%20Dive%20-%20Health%20Information%20Technology%200113.pdf>
- Flows, C. (2017). Does airline deregulation offer a lesson for healthcare IT? *Forbes*, May 1. Available from <https://www.forbes.com/sites/realspin/2017/05/01/does-airline-deregulation-offer-a-lesson-for-healthcare-it/2/#73dfe757a0f>
- Graber, M. L., Johnston, D., & Bailey, R. (2016). *Report of the evidence on health IT safety and interventions*. RTI International. Available from: <http://garner>

- [health.com/wp-content/uploads/2014/02/task_8_1_final_508.pdf](https://www.health.com/wp-content/uploads/2014/02/task_8_1_final_508.pdf)
- Hassin, R., & Haviv, M. (1992). Mean passage times and nearly uncoupled Markov Chains. *SIAM Journal of Discrete Mathematics*, 5(3), 386–397.
- Ho, Y., & Li, S. (1988). Extensions of infinitesimal perturbation analysis. *IEEE Transactions on Automation Control* AC, 33(5), 427–438.
- Horsky, J., Schiff, G. D., Johnston, D., Mercincavage, L., Bell, D., & Middleton, B. (2012). Interface design principles for usable decision support: A targeted review of best practices for clinical prescribing interventions. *Journal of Biomedical Informatics*, 45(6), 1202–1216.
- IOM (2012). Health IT and patient safety – Building safer systems for better care. Institute of medicine of the national academies. Available from: http://www.nationalacademies.org/hmd/~/media/Files/Report%20Files/2011/Health-IT/HealthITandPatientSafetyreportbrieffinal_new.pdf
- Kim, M. O., Coiera, E., & Magrabi, F. (2017, March 1). Problems with health information technology and their effects on care delivery and patient outcomes: A systematic review. *Journal of the American Medical Informatics Association*, 24(2), 246–250.
- Magrabi, F., Ong, M.-S., Runciman, W., & Coiera, E. (2011). Using FDA reports to inform a classification for health information technology safety problems. *Journal of the American Medical Informatics Association*, 19(1), 45–53.
- Mardon, R., Olinger, L., Szekendi, M., Williams, T., Sparnon, E., & Zimmer, K. (2014). Health information technology adverse event reporting: Analysis of two databases (Prepared by Westat, UHC, and ECRI Institute under Contract Order No. HHSP23337024T). Washington, DC: Office of the National Coordinator for Health Information Technology.
- Meeks, D. W., Smith, M. W., Taylor, L., Sittig, D. F., Scott, J. M., & Singh, H. (2014). An analysis of electronic health record-related patient safety concerns. *Journal of the American Medical Informatics Association*, 21(6), 1053–1059.
- Meyer, C. (1989). Stochastic complementation, uncoupling Markov Chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2), 240–272.
- Ratwani, R. M., Fairbanks, R. J., Hettinger, A. Z., & Benda, N. C. (2015). Electronic health record usability: Analysis of the user-centered design processes of eleven electronic health record vendors. *Journal of the American Medical Informatics Association*, 22(6), 1179–1182.
- Rauscher, R. L. (2000). Evaluating the reliability of information systems in healthcare. In *Proceedings of the AMIA Symposium* (p. 1113). Los Angeles, CA: American Medical Informatics Association.
- Ray, S., McEvoy, D. S., Aaron, S., Hickman, T. T., & Wright, A. (2018). Using statistical anomaly detection models to find clinical decision support malfunctions. *Journal of the American Medical Informatics Association*, 25, 862–871.
- Salahuddin, L., & Ismail, Z. (2015). Classification of antecedents towards safety use of health information technology: A systematic review. *International Journal of Medical Informatics*, 84(11), 877–891.
- Schweitzer, P. (1968). Perturbation theory and finite Markov Chains. *Journal of Applied Probability*, 5(2), 401–413.
- Serfozo, R. (2009). *Basics of applied stochastic processes*. Berlin, Heidelberg: Springer Science & Business Media.
- Singh, H., & Sittig, D. F. (2016). Measuring and improving patient safety through health information technology: The health IT safety framework. *BMJ Quality & Safety*, 25(4), 226–232.
- Sittig, D. F., & Singh, H. (2011). Defining health information technology-related errors: New developments since. *To Err Is Human. Archives of Internal Medicine*, 171.14, 1281–1284.
- Sittig, D. F., & Singh, H. (2015). A new socio-technical model for studying health information technology in complex adaptive healthcare systems. In V. Patel, T. Kannampallil, & D. Kaufman (Eds.), *Cognitive informatics for biomedicine. Health Informatics*. Cham, Switzerland: Springer.
- Sparnon, E. (2013). Spotlight on electronic health record errors: Paper or electronic hybrid workflows. *Pa Patient Saf Advis*, 10(20), 55–58.
- Sparnon, E., & Marella, W. M. (2012). The role of the electronic health record in patient safety events. *Pennsylvania Patient Safety Advisory*, 9(4), 113–121.
- Stewart, W., & Dekker, M. (1994). *Numerical solution of Markov Chains*. Princeton, New Jersey: Princeton University Press.
- Suri, R. (1989). Perturbation analysis: The state of the art and research issues explained via the GI/G/I Queue. *Proceedings of the IEEE*, 77(1), 114–138.
- Wright, A., Hickman, T. T. T., McEvoy, D., Aaron, S., Ai, A., Andersen, J. M., ... Bates, D. W. (2016). Analysis of clinical decision support system malfunctions: A case series and survey. *Journal of the American Medical Informatics Association*, 23(6), 1068–1076.