Proceedings of the 2020 IISE Annual Conference
L. Cromarty, R. Shirwaiker, P. Wang, eds.

Detection of Anomalous Events in Electronic Health Records

Jordan J. Pellett[1], Olufemi A. Omitaomu[1,2,*], Mohammed M. Olama[2], Ozgur Ozmen[2], Hilda B. Klasky[2], Laura L. Pullum[2], Teja Kuruganti[2], Merry Ward[3], Angela Laurio[3], Jean M. Scott[3], Frank Drews[3], Jonathan R. Nebeker[3]

[1]University of Tennessee, Knoxville, TN, USA
[2]Oak Ridge National Laboratory, Oak Ridge, TN, USA
[3]US Department of Veterans Affairs, Washington, DC, USA
[*]omitaomuoa@ornl.gov

Abstract

Over the past decade, Health Information Technology (Health IT) has enabled an explosion in the amount of digital information stored in electronic health records (EHRs). According to recent studies, safety-related issues in healthcare can present themselves as anomalies in EHR data. Motivating examples of anomalous events in EHRs include clinical events related to invalid order cancellations or rejections, which may be initiated by clinical staff or automatic software routines in Health IT systems. Such events may be detected using anomaly detection or change point detection methods. In this paper, we explore the use of a forecasting approach to detect anomalies in EHR data using an online Support Vector Regression technique. Specifically, the proposed approach uses temporal frequency of activities in EHRs, coupled with dynamic robust confidence intervals, to characterize events as normal or anomalous. Once an event is characterized as an anomaly, our approach suppresses its effects in subsequent time intervals. The proposed approach shows encouraging results using real-world EHR data from the Veterans Affairs' corporate data warehouse.

Keywords
Anomaly detection, Health IT, time-series data, robust confidence intervals, support vector regression

1. Introduction
Health Information Technology (Health IT) is the hardware and software used in large-scale healthcare systems to support healthcare services [1]. In addition to facilitating efforts to decrease healthcare overhead and increase the quality of healthcare services, Health IT systems have enabled massive amounts of digital information to be stored in electronic health records (EHRs). Traditionally, EHR data has been used for archiving patient information and performing administrative tasks. Recently, EHRs are being used for more complex tasks such as aiding in clinical research advances and informing improved clinical decision making. In light of recent studies (e.g., [2–4]) revealing new safety issues related to Health IT use, there is an opportunity to explore an additional use of EHR data – detection of safety-related issues (anomalies) in healthcare.

Some of the existing methods for anomaly detection using EHR data rely on retrospective approaches. For example, in [5], EHR data was used to evaluate six offline statistical anomaly detection models in their ability to detect malfunctions in clinical decision support systems. These retrospective approaches serve as an initial step in identifying what went wrong and why. Recently, in [6], an online process monitoring approach uses a statistical process control (SPC) method to detect high numbers of invalid cancellations of radiology orders using EHR data. In this paper, we propose an improvement to the online process monitoring approach through a real-time detection of anomalies in Health IT systems. Specifically, we present a real-time forecasting approach to detect hazardous events in streaming EHR data. This approach uses temporal frequency of activities in EHRs such as order submission, order cancellation, and order rejection, among others. In particular, we use a regression technique to forecast the next value based on historical patterns in the data. To allow for variation in the predictions, we compute robust confidence intervals around the predicted values using a data-driven, dynamic procedure. This computed confidence interval is then used to evaluate the next activity frequency in real-time, determining whether it is normal or abnormal. If an activity frequency is characterized as abnormal, the approach suppresses its effects in subsequent time intervals. Therefore, our contributions in this paper are the: (i) application of a forecasting approach for estimating future frequency of activities in EHRs using an online Support Vector Regression (SVR) technique; (ii) development of an approach which allows for real-time anomaly detection in EHRs, rather than relying on retrospective analysis; and (iii) development of a dynamic robust confidence interval technique suitable for online SVR.

The remainder of the paper is structured as follows: Section 2 presents the foundational techniques used in our proposed approach. Section 3 discusses the detailed proposed approach. Section 4 presents and discusses some numerical results. Finally, Section 5 presents the conclusions from our study.

2. Methods

In this section, we discuss the base components of our proposed approach.

2.1 Accurate-Online Support Vector Regression

The proposed approach uses the accurate-online support vector regression (AOSVR) algorithm [7]. Unlike conventional batch implementations of support vector regression, AOSVR allows for real-time forecasting without having to retrain the model from scratch every time the training set is modified. With AOSVR, when a new point is added to the existing dataset, the trained SVR model is efficiently updated rather than retrained. Highlights of the theoretical framework for SVR are provided below. For more detailed theory for SVR and AOSVR algorithms see [7].

From the training set $T = \{(\mathbf{x}_i, y_i), i = 1 \ldots m\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, establish a linear regression function by

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{1}$$

on a feature space $\mathcal{F}$. In Equation (1), $\mathbf{w}$ is a vector in $\mathcal{F}$ called the weight vector, b is the bias term, and $\phi(\mathbf{x})$ maps $\mathbf{x}$ to a vector in the higher dimensional feature space $\mathcal{F}$. The weight vector $\mathbf{w}$ and the bias b are obtained by solving the following primal optimization problem

$$\text{Minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-) \qquad \text{Subject to } \begin{cases} y_i - \mathbf{w}^T\phi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i^+ \\ \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \geq 0, \end{cases} \tag{2}$$

where C is a positive regularization constant that penalizes y-values that differ from $f(\mathbf{x}_i)$ by more than ε. The slack variables ξ_i^+ and ξ_i^- represent the excess deviation size for upper and lower deviations, respectively. Data points with $|f(\mathbf{x}_i) - y_i| \leq \varepsilon$ are defined to have no contribution to the regression model (their coefficients equal 0). Alternatively, data points with $|f(\mathbf{x}_i) - y_i| > \varepsilon$ (called support vectors) have nonzero coefficients and thus contribute to the regression function.

The dual formulation of the primal optimization problem is crucial for extending SVR to nonlinear functions. Once the related Lagrangian function is obtained and the Karush-Kuhn-Tucker conditions are applied, we arrive at the dual formulation. See [7] and references therein for more detailed information about the dual formulation. Finally, resolving the dual problem gives following regression function

$$f(\mathbf{x}) = \sum_{i=1}^{m}(\alpha_i^+ - \alpha_i^-)K(\mathbf{x}_i, \mathbf{x}) + b. \tag{3}$$

Here, α_i^+ and α_i^- are Lagrange multipliers and $K(\mathbf{x}_i, \mathbf{x})$ is known as the kernel function. The kernel function allows non-linear function approximations to be made with SVR, but maintains the computational efficiency present when making linear approximations [7]. Our current analysis utilizes the radial basis kernel.

2.2 Simulated Annealing

To compute the SVR parameters (C and ε), we integrate simulated annealing (SA) into the AOSVR algorithm. SA was first introduced in 1983 by Kirkpatrick et al. [8] as a probabilistic technique for approximating global optima in discrete combinatorial optimization problems. Since its introduction, SA has become a widely used tool for both discrete and continuous problems in several application areas including medicine, engineering, and business [9].

Generally, local search techniques begin with a current solution which is slowly improved as neighbor solutions obtained from small perturbations to the initial solution are considered. If a neighbor solution has a lower cost, it replaces the current solution. This process continues until no improved solutions are found. The problem with these techniques is that the search often stops prematurely at a local, rather than global optimum [9]. SA attempts to overcome this by allowing, with a certain probability, acceptance of some non-improved higher cost solutions. Given a current solution S_c and current temperature T_c, the acceptance probability of a neighbor solution S_i is

$$P_{T_c} = \begin{cases} 1, & \text{if } E_i < E_c \\ \exp\left(-\Delta E/T_c\right), & \text{otherwise} \end{cases} \tag{4}$$

where E_c and E_i are costs of S_c and S_i respectively and $\Delta E = E_i - E_c$. That is, if a neighbor solution has a lower cost than the current solution, it is always accepted. Otherwise, it is accepted with probability $\exp\left(-\Delta E/T_c\right)$. For the latter case, a random number $rd \in (0,1)$ is generated. If $rd < \exp\left(-\Delta E/T_c\right)$, the neighbor solution is accepted. Observe that $\lim_{T_c \to 0} \exp\left(-\Delta E/T_c\right) = 0$ and thus the probability of accepting a higher cost solution decreases as T_c decreases. In the context of AOSVR, SA seeks values of C and ε which minimize the mean squared error over the training set. The steps of the SA algorithm are summarized below.

Algorithm 1: Simulated Annealing

Result: S_c
Initialization: T_f, T_c, r, N, S_c ;
while $T_c \geq T_f$ **do**
    for $i = 1$ **to** N **do**
        $\Delta E = E_i - E_c$
        if $\Delta E < 0$ **then**
            $S_c \leftarrow S_i,\ \ E_c \leftarrow E_i;$
        else if $rd < \exp(-\Delta E/T_c)$ **then**
            $S_c \leftarrow S_i,\ \ E_c \leftarrow E_i;$
        end
    end
    $T_c = r * T_c$
end

3. Prediction and Detection Workflow for the Proposed Approach

The workflow for our approach is summarized in Figure 1. The proposed approach modifies the AOSVR algorithm by incorporating the computation of robust confidence intervals (RCIs) using a data-driven approach. The RCI methodology is based on [10]; however, in this paper we implement two RCI versions – static RCI and dynamic RCI.

The base approach uses the static RCI procedure and consists of all the steps shown in Figure 1, except the step shown with gray background. The approach begins with data normalization followed by model training. Once the trained SVR model is obtained, it is used to compute the training and validation error set described in Section 3.1. Next, the trained model forecasts a prediction from test set data. The model prediction and previously obtained error set are then used to compute the corresponding confidence interval. If the observed value falls outside of the confidence interval, it is classified as an anomaly and we implement a replacement strategy in order to suppress its effects in subsequent time intervals. That is, prior to updating the model to reflect the new test point, we replace the anomalous value with the average of the previous ℓ values, where ℓ is the embedding length of the time series. If, on the other hand, the observed value falls within the confidence interval, it is not classified as an anomaly, the replacement step is skipped, and the observed value is used in updating the model. The updated model is then used to predict the next point, from which the process continually repeats until the end of the test set. The compared approach uses the dynamic RCI procedure and follows the same workflow, but with an additional step (the gray box in Figure 1). The following subsections describe the detailed procedures for computing static and dynamic RCIs.

3.1 Static RCI

The procedure for computing static RCIs (same technique as in [10]) is as follows. After obtaining the trained forecasting model, errors are computed on both the training and validation sets. Suppose the combined training and validation set S (maintaining order) consists of values $x_1, \ldots, x_n$ such that $S = \{x_1, \ldots, x_n\}$. Starting with x_1, the first ℓ values are fed into the trained model to predict $\hat{x}_{(\ell+1)}$. Taking the difference between the predicted value and the known

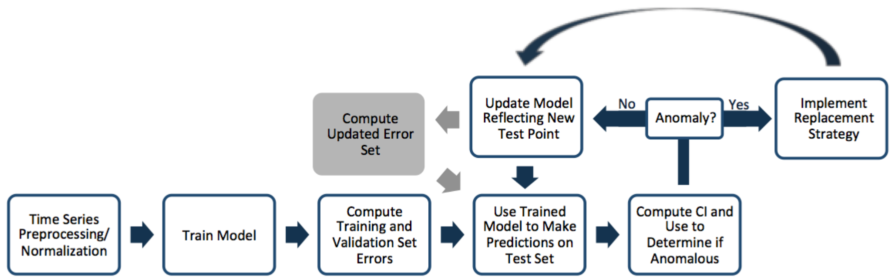


Figure 1: The prediction and detection workflow for the proposed approach

observed value, we obtain the first error $e_1 = \hat{x}_{(\ell+1)} - x_{(\ell+1)}$. This procedure is repeated such that $e_i = \hat{x}_{(\ell+i)} - x_{(\ell+i)}$, until the end of the set S is reached. After the set of errors $E = \{e_1, \ldots, e_{n-\ell}\}$ is obtained, the errors are sorted in ascending order by value. The set of errors, minus a number of sample errors from each extreme is used to compute the RCI. For small datasets, the number of sample errors to be removed from each extreme of the error set E is $((n-\ell) \times p - 1)$ where $n-\ell$ is the cardinality of E and $2p$ is the corresponding significance level of the confidence interval. If $((n-\ell) \times p - 1)$ is not a whole number, it is truncated. We denote this new error set by E_{CI}. The confidence interval for each new predicted value, $\hat{t}_j$, of the test set, is then formed as $\{\hat{t}_j + \min(E_{CI}), \hat{t}_j + \max(E_{CI})\}$.

3.2 Dynamic RCI

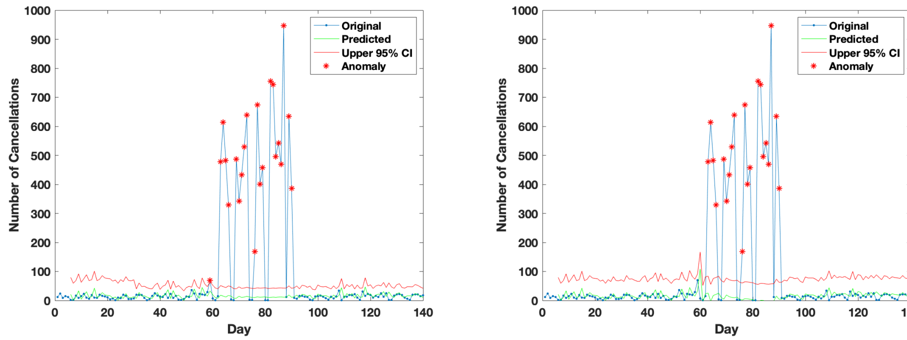
As described in Section 3.1, the static RCI computes, from the training and validation sets, a single error set used to estimate the confidence interval. Thus, the width of the confidence interval around a given predicted point is constant with time as new predictions do not change the original error set. This technique is appropriate for conventional regression approaches. However, since with AOSVR, the regression function may be updated when new data points are added, we propose a modified version of the static RCI called dynamic RCI. The dynamic RCI allows for modifications to the training and, consequently, the error set. Thus, the width of the computed confidence interval can change with time. The dynamic RCI method is as follows. First, the original error sets E and E_{CI} are computed from the training and validation sets in the same manner as was described for the static RCI. The confidence interval around the first predicted value $\hat{t}_1$ is $\{\hat{t}_1 + \min(E_{CI}), \hat{t}_1 + \max(E_{CI})\}$. The difference between the predicted value $\hat{t}_1$ and observed value t_1, call it e_{t_1}, is then added to the error set E. Next, E_{CI} is recomputed using the usual procedure and denoted as E_{CI_2}. The trained model is then used to predict $\hat{t}_2$ and the corresponding confidence interval is $\{\hat{t}_2 + \min(E_{CI_2}), \hat{t}_2 + \max(E_{CI_2})\}$. This process is continued such that for $j > 1$, the corresponding confidence interval around $\hat{t}_j$ is $\{\hat{t}_j + \min(E_{CI_j}), \hat{t}_j + \max(E_{CI_j})\}$.

4. Results

Our proposed approach was tested using data acquired from the Veterans Affairs' corporate data warehouse for 130 stations. In this section, we compare results obtained using static versus dynamic RCIs in establishing the threshold for anomaly detection. In this context, an anomaly is defined as any observed data point whose value is above the upper confidence interval for the corresponding forecast value. We also compare results obtained by our approach to those obtained by the traditional SPC method.

4.1 Dynamic vs. Static Robust Confidence Interval

Figure 2 depicts a time series corresponding to a stream of EHR data (specifically, number of cancelled consult orders) and provides an illustration of the differences in behavior observed when using the dynamic RCI method (Figure 2a) as opposed to the static RCI method (Figure 2b). Illustrated is the proof of concept that the dynamic RCI method allows the width of the confidence interval to change with time, while the static RCI width remains constant. Additionally, we observe for this specific subset of EHR data with $C = 100$ and $\varepsilon = 0.1$, that the first point classified as an anomaly (around day 60) with the dynamic RCI method is not classified as an anomaly using the static method. This additional classified anomaly occurs four time steps prior to the cluster of anomalous behavior from days 64 through 90 and thus could serve as a warning for Health IT safety managers. In general, it appears that the dynamic RCI method is more conservative in that it classifies more lower values as anomalies than the static RCI method. Thus, the dynamic RCI method is used in the remaining analysis.



(a) Prediction and detection approach: dynamic RCI (b) Prediction and detection approach: static RCI

Figure 2: Dynamic vs. static RCI method

4.2 Dynamic RCI vs. Statistical Process Control

Figure 3 compares results obtained using our proposed approach to those obtained using SPC. For station A the proposed approach classifies cancellation numbers above approximately 15 as anomalous (Figure 3a), whereas SPC classifies cancellation numbers above approximately 95 as anomalous (Figure 3b). For station B the proposed approach classifies numbers of cancellations greater than approximately 30 as anomalous (Figure 3c), while SPC classifies numbers of cancellations greater than approximately 175 as anomalous (Figure 3d). Analyzing between approaches, we find that our proposed approach has a lower threshold for detection than SPC. Without ground-truth data, we hypothesize that 15 invalid cancellations is a high number that must be detected and reported; hence, the proposed approach seems to be an improvement over SPC. Future studies will further analyze the performance of these methods. Analyzing the between-stations results, we find that the results appear to be station dependent. That is, the lower bound for what is considered anomalous varies by station.

5. Conclusions

In this paper, we proposed a novel approach for utilizing EHR data to detect safety issues in Health IT systems that is suitable for both batch and streaming EHR data implementations. Further, we find that the proposed AOSVR approach coupled with dynamic robust confidence intervals gives encouraging results for real-time detection of anomalies in EHR data; thus showing improvement over conventional methods such as SPC. Future studies include training the AOSVR model with a set of data characteristic of normal operations in representative stations to potentially eliminate the cause of station-dependent results.

Acknowledgements

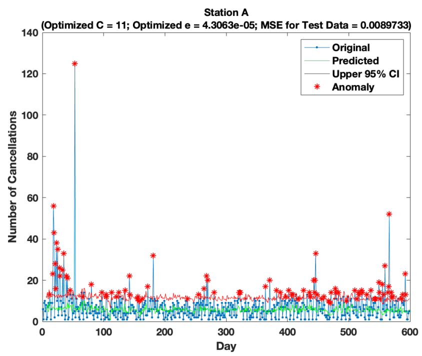
This work is sponsored by the US Department of Veterans Affairs. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the 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).

References

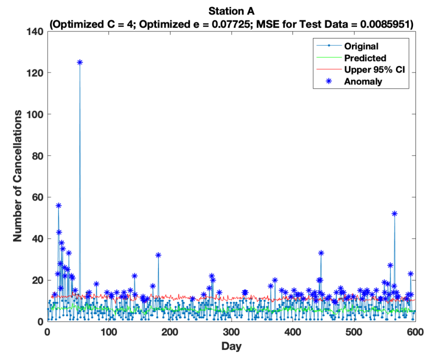
[1] M. O. Kim, E. Coiera, and F. Magrabi, “Problems with health information technology and their effects on care delivery and patient outcomes: A systematic review,” *Journal of the American Medical Informatics Association*, vol. 24, no. 2, pp. 246–250, 2017.

[2] I. of Medicine (US). Committee on Patient Safety and H. I. Technology, *Health IT and patient safety: Building safer systems for better care*. National Academies Press, 2012.

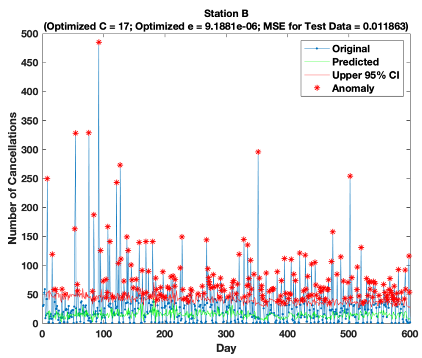
[3] G. Castro, L. Buczkowski, J. Hafner, S. Barrett, K. Rasinski, and S. Williams, “Investigations of Health IT-



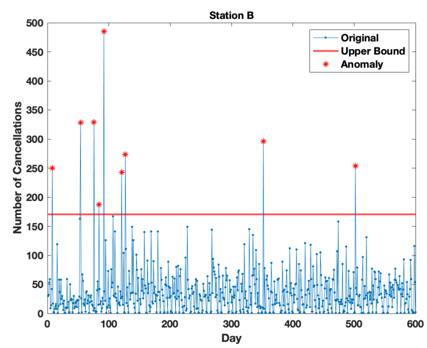
(a) Station A: proposed approach with dynamic RCI



(b) Station A: statistical process control



(c) Station B: proposed approach with dynamic RCI



(d) Station B: statistical process control

Figure 3: Proposed approach with dynamic RCI vs. statistical process control approach

related deaths, serious injuries, or unsafe conditions," *Prepared by Joint Commission under Contract No. HHSP233201300019C). Washington, DC: Office of the National Coordinator for Health Information Technology*, 2015.

[4] S. K. Brenner, R. Kaushal, Z. Grinspan, C. Joyce, I. Kim, R. J. Allard, D. Delgado, and E. L. Abramson, "Effects of health information technology on patient outcomes: A systematic review," *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 1016–1036, 2015.

[5] S. Ray, D. S. McEvoy, S. Aaron, T.-T. Hickman, and A. Wright, "Using statistical anomaly detection models to find clinical decision support malfunctions," *Journal of the American Medical Informatics Association*, vol. 25, no. 7, pp. 862–871, 05 2018. [Online]. Available: https://doi.org/10.1093/jamia/ocy041

[6] O. A. Omitaomu, O. Ozmen, M. M. Olama, L. L. Pullum, T. Kuruganti, J. Nutaro, H. B. Klasky, H. Zandi, A. Advani, A. L. Laurio, M. Ward, J. Scott, and J. R. Nebeker, "Real-time automated hazard detection framework for health information technology systems," *Health Systems*, vol. 8, no. 3, pp. 190–202, 2019. [Online]. Available: https://doi.org/10.1080/20476965.2019.1599701

[7] J. Ma, J. Theiler, and S. Perkins, "Accurate on-line support vector regression," *Neural Comput.*, vol. 15, no. 11, pp. 2683–2703, Nov. 2003. [Online]. Available: http://dx.doi.org/10.1162/089976603322385117

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. [Online]. Available: https://science.sciencemag.org/content/220/4598/671

[9] K. A. Dowsland and J. M. Thompson, "Simulated annealing," *Handbook of natural computing*, pp. 1623–1655, 2012.

[10] A. Oliveira and S. Meira, "Detecting novelties in time series through neural networks forecasting with robust confidence intervals," *Neurocomputing*, vol. 70, no. 1-3, pp. 79–92, 2006.