 doi:10.20944/preprints202202.0081.v1

Article

Online Anomaly Detection for Smartphone-based Multivariate Behavioral Time Series Data

Gang Liu [1,†,‡], **Jukka-Pekka Onnela** [1,‡]

[1] Harvard T.H. Chan School of Public Health; gang_liu@g.harvard.edu
[2] Harvard T.H. Chan School of Public Health; onnela@hsph.harvard.edu
* Correspondence: gang_liu@g.harvard.edu
‡ These authors contributed equally to this work.

Abstract: To detect aberrant human behaviors from large volume of passive data collected by smartphones in real time, we propose an online anomaly detection method using Hotelling's T-squared test. The test statistic is a weighted average, with more weight on the between-individual component when there are little data available for the individual and more weight on the within-individual component when the data are adequate. The algorithm takes only $\mathcal{O}(1)$ run time in each update and the required memory usage is fixed after a pre-specified number of updates. The performance of the proposed method, in terms of accuracy, sensitivity and specificity, is consistently better than or equal to the offline method that it builds upon depending on the sample size of the individual data.

Keywords: Online learning; Anomaly detection; Hotelling's T-squared test; Digital phenotyping.

1. Introduction

Multivariate time series (MTS) is ubiquitous in an extensive range of real-world applications, such as weather forecast[1], health care[2,11], finance[3], manufacturing[9] and Cyber-Physical Systems[4,5,7,12]. Anomaly detection seeks outliers or suspicious observations which differ significantly from the majority of the data. The difference between anomalous and non-anomalous data can be quantified by a variety of metrics such as Euclidean distance[14], mean squared error[9], correlation[15], cosine similarity[16], or dynamic time warping[17] between two observations, or the probability that an observation is drawn from a certain distribution[5,6] or it falls in a domain derived from that distribution[18]. Although there are numerous methods to tackle the problem from distinct angles, all methods can be decomposed to two steps: (1) derive a new sequence from the original MTS using a transformation or a predictive model; (2) calculate the "difference" metric for each element in the new sequence. In step one, either a direct transformation from MTS to univariate time series (UTS), such as fuzzy integral[6], PCA[19] or subspace monitoring[20] can be applied, or a predictive model, such as generative adversarial network[7], long-short term memory[9], convolutional neural network[11], hierarchical temporal memory[5] and vector auto-regressive model[12]. In step two, either a self-defined threshold [7,9] is used to find anomalies if a predictive model was used, or an algorithm designed for anomaly detection in UTS or MTS cross-sectional data [8] is applied to the new sequence, for example, hidden Markov model[6], Bayes network[5], k-nearest neighbors algorithm[12], and one-class support vector machine[21] and ad-hoc decomposition[2,4]. All these methods address three characteristics of MTS: (1) the dependency within the same feature over time, (2) the dependency among different features, and (3) the omnipresent noise. There is no single overall best method for all settings but instead one or more suitable ones depending on the problem.

Digital phenotyping has been defined as "the moment-by-moment quantification of the individual-level human phenotype *in situ* using data from personal digital devices", in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

particular smartphones[22]. The behavioral data, passively collected by smartphones, consists of sensor data, such as the built-in global positioning system (GPS) and accelerometer, as well as phone usage data, such as call logs and screen activity. Such data streams have been shown to be predictive of relapse for patients with schizophrenia[2,23] and depressive symptoms for women at risk for perinatal depression[24]. Barnett et al.[2] proposed a semi-parametric anomaly detection method with robustness against misspecification of the distribution of time series. The method was applied to a longitudinal behavioral dataset collected passively by smartphones, which could detect an escalation of symptoms or signs of a potential relapse. The method decomposes the observed multivariate time series into a general trend, a periodic component, and an error component for each dimension. The error components are then used to build Hotelling's T-squared test statistic and identify anomalies. Henson et al.[25] applied the method and achieved 89% sensitivity and 75% specificity for predicting relapse in schizophrenia in a cohort of 126 participants followed by 3 to 12 months.

There are two main limitations to the method by [2,23]. First, the offline algorithm has been used mainly to identify anomalous behaviors in a one-time retrospective analysis, where computational performance is not critical. If however the goal is to carry out anomaly detection as data are being collected rather than identifying them at the end data collection, possibly even in real time, the method needs to scale to a large cohort of subjects followed for months or years. Second, the offline method uses only within-individual comparisons to overcome heterogeneity of participant data, and it requires at least two weeks of data to set up the individual baseline for comparisons. In many health settings, from surgery to rehabilitation, the time period immediately following patient discharge is risky. Forster et al.[26] found that nearly 20% of patients experience adverse events within 3 weeks of discharge. Many meta-analyses of suicide rates after discharge from psychiatric facilities suggest these rates remain high for several years, but are particularly high in the first few weeks and months post-discharge [27–29]. Given that this time period is most likely to have anomalies, the method should perform well especially during this period.

Here we propose an online algorithm for Hotelling's T-squared test, which incorporates both within-individual and between-individual comparisons to overcome the above limitations, using the same framework as in Barnett et al. with little sacrifice in accuracy. We review the method proposed by Barnett et al.[2] in Section 2.1; describe our new method in Section 2.2; illustrate our method using simulated and real data in Section 3; and further comment on our method in Section 4.

2. Materials and Methods

2.1. Offline Anomaly Detection Method

In the offline method of Barnett et al.[2], the authors first defined the expected behavior for each subject by decomposing the observed multivariate time series of p features into an overall trend, a weekly component, and an error component for each dimension. For a given subject, let m_i be the number of days of follow up where feature i is observed. Let $y_{ij} = \mu_{ij} + s_{ij} + \epsilon_{ij}$ be the value of the ith feature on day j of follow up, where ϵ_{ij} is the error component and μ_{ij} is the trend component estimated from a weighted average of the previous observed feature values $y_{i,j-1}, y_{i,j-2}, ...$, with more weight given to observations closer in time. These weights are specified as a t-distribution with 2 degrees of freedom and scaling parameter $10/\max m_i$. The weekly component s_{ij} is estimated to minimize the square error under the restriction $s_{ij} = s_{i,j-7}$. After estimating the decomposition of the time series as $\hat{\mu}_{ij}, \hat{s}_{ij}, \hat{\epsilon}_{ij}$, the authors transform the errors $\hat{\epsilon}_{ij}$ non-parametrically into Z-scores by sorting the errors by rank across all days of follow-up for that feature, followed by a standard normal transformation using the probability integral transform. Mathematically, the transformed error $\tilde{\epsilon}_{ij}$ can be expressed as $\tilde{\epsilon}_{ij} = \Phi^{-1}\left(\frac{\text{rank}(\hat{\epsilon}_{ij})}{m_i+1}\right)$. Let $\tilde{\epsilon}_j = [\tilde{\epsilon}_{1j}, ..., \tilde{\epsilon}_{pj}]^T$ denote the vector of transformed errors on day j and $\tilde{\epsilon}^*_k = [\tilde{\epsilon}_{k1}, ..., \tilde{\epsilon}_{k,m_i}]^T$ denote the vector of transformed errors of feature k. The covariance between the transformed errors of feature i and feature k is defined as $\Sigma_{ik} = \text{cov}(\tilde{\epsilon}^*_i, \tilde{\epsilon}^*_k)$, which is estimated empirically

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

across all days where both are observed. Hotelling's T-squared test statistic is constructed as $Q_j = \tilde{\epsilon}_j^T \Sigma^{-1} \tilde{\epsilon}_j$, where $Q_j \xrightarrow{p} \chi_p^2$ under the null hypothesis that the observation is not anomalous on day j. To correct for multiple comparisons, the method bootstraps the error components of the time series assuming stationarity to generate the null distribution for the largest test statistic across all days of follow-up, and the α-quantile of the bootstrapped values provide the threshold for significance at the α significance level.

The method was designed for retrospective analyses in studies where there is no intervention component, so detecting anomalies at the end of data collection is sufficient, and therefore computational performance is not critical. More specifically, there are four steps to the algorithm with linear or super-linear computational complexity: (1) calculating the general trend using the weighted average of all historical observations; (2) estimating the periodic term s_{ij} through linear regressions; (3) sorting the errors of each feature; and (4) computing the empirical covariance matrix of the transformed errors. The value of anomaly detection is however mainly in being able to detect anomalies and act on them close to real time. Although the offline method could be applied repeatedly, this is computationally very expensive. The method also needs to see at least two weeks of data for a given participant, but in practice these first two weeks, especially if they coincide with patient discharge from a facility, are the most likely to have anomalies as discussed above. These considerations motivate our method presented next.

2.2. Online Anomaly Detection Method

In this Section, we introduce our online anomaly method. We address the different components of the method separately.

2.2.1. Updating the general trend and periodic terms

Estimation of both periodic and non-periodic trends requires assigning weights to past observations. Even though a t-distribution with 2 degrees of freedom of the offline method has thick tails, the weights for observations far away from the current observation become negligible when m_i is large. Instead of using all historical data to compute the average, we propose to only use a subset of K most recent observations to reduce both computational time and memory use. The periodic term in the original method was estimated through linear regression, where the effect size of each day of the week is expressed as the mean observed residual on that day of the week. Here, we use sample means instead of linear regressions to estimate the periodic terms. The new estimates are identical to those from linear regression, but given that we use the classic online approach [31] to calculating the mean, its computational complexity is $\mathcal{O}(1)$ and it uses less memory as only a running sum and the number of observations need to be stored in memory.

2.2.2. Sorting the errors

Our proposed online algorithm requires the ranks of the errors within each feature in each update. The values of the errors are not fixed over time but keep changing periodically, which makes the online sorting procedure a non-trivial problem. For example, if a data point on the Wednesday of the third week is observed, then the periodic term for Wednesdays is updated and the current error is computed. Assuming the new estimate of the periodic term is larger than the previous estimate by δ, the values of the errors for the first and second Wednesday should both be decreased by δ in this update given the decomposition $y_{ij} = \mu_{ij} + s_{ij} + \epsilon_{ij}$; given that μ_{ij} is fixed once estimated, the rank of the current error cannot be obtained by locating the index of the previously sorted error. Rather than sorting the data from scratch in each update as in the offline method, we take advantage of the trackable changes in the errors estimated by sample means and propose a binning method to obtain approximate ranks of both the current error and all previous errors by examining the quantiles of the empirical distribution of all errors.

We illustrate the idea assuming a weekly period. For each feature, we initialize a histogram for the errors for each day of the week with H bins using the first M observations.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

The bin width w for each feature is determined by the corresponding maximum and minimum values $w = \frac{R(\max\{S\}-\min\{S\})}{H}$, where $S = \left\{y_k | k = 1, 2, ..., M\right\}$, y_k is the feature value on day k, and $R(> 1)$ is a hyperparameter, which controls the range that the histogram can cover on the flanks of the observed range for unobserved future values. The locations of the bins for each feature are lined up across seven days of the week. When we apply the method and observe a new value y_n on day j of the week after M days, we first update the estimate of s_j using the sample mean, then calculate the difference between the new and old estimates $\delta_j = s_{j,n} - s_{j,n-1}$ and the new error $\epsilon_n = y_n - \mu_n - s_{j,n}$. If the difference δ_j is positive, the values of all previous errors on day j of the week decrease by $|\delta_j|$, which causes the corresponding histogram of the errors to shift $\lfloor |\delta_j|/w_j \rceil$ bins to the left, where $\lfloor \rceil$ denotes rounding to the nearest integer. If the difference is negative, we shift the histogram to the right. We then locate the bin of the new error ϵ_n and update the histogram again by adding one to the count of observations in that bin. Finally, we aggregate all histograms for the seven days of the week and sum up the count in each bin to obtain a final histogram of all errors. We locate the percentile of the new error ϵ_n by dividing the sum of counts, starting from the leftmost bin and proceeding to the right, until we get to a bin for which ϵ_n falls within the total sum, and we convert the percentile to a normal random variable using the inverse standard normal cumulative distribution function. The number of bins H determines the precision of this method and, empirically, $H = 100$ appears to achieve good performance.

2.2.3. Updating the covariance matrix

The classic online approach for updating the covariance matrix is to decompose the new covariance matrix as a weighted sum of the old covariance matrix and an outer product of the new error vector $\tilde{\epsilon}_n$, which can be expressed as $\hat{\Sigma}_n = \frac{n-1}{n}\hat{\Sigma}_{n-1} + \frac{1}{n}\tilde{\epsilon}_n\tilde{\epsilon}_n^T$, where Σ_j denotes the covariance matrix after the jth update. However, due to the fact that in our setting the values of all previous $\tilde{\epsilon}$ change when a new data point is observed, we cannot apply the method directly to our problem. Note that each element in $\tilde{\epsilon}$ is obtained by the standard normal transformation, thus $\tilde{\epsilon}$ is a multivariate normal random vector with a variance of each dimension 1 by the Cramér-Wold device. That is to say, the covariance matrix of $\tilde{\epsilon}$ is essentially a correlation matrix. Motivated by this observation, we propose an approximation whereby we re-scale the covariance matrix obtained by the classic approach to a correlation matrix after each update, which is accomplished by multiplying the inverse diagonal matrix filled by the standard deviation of each dimension in the front and at the end of the covariance matrix, which can be expressed as $\text{corr} = D^{-1}\Sigma D^{-1}$, where $D = \sqrt{\text{diag}(\Sigma)}$.

So far, every step in the offline method has been modified to an online algorithm and can be described as follows. Given a new observation, we decompose it as described above and compute the test statistic using the histograms and covariance matrix from the previous update. If the corresponding p-value is smaller than the threshold, we sample a dummy variable I from Bernoulli(p). If $I = 0$, we classify the current observation as an anomaly, otherwise we consider it normal and update the histogram and covariance matrix.

2.2.4. Incorporating the between-individual comparison

As mentioned in the introduction, under the assumption of a weekly periodicity, the method proposed by Barnett et al.[2] uses only within-individual comparisons, so it requires at least two weeks of data from each individual to detect anomalies. In a clinical application, anomaly detection is most helpful at times when anomalies are most likely to occur. This is often the time period following some intervention, such as surgery or rehabilitation, and the risk of relapse is usually highest at the beginning, i.e., during the time period when there may be little to no data collected from the subject. To address this limitation, our method borrows information from other participants in the study so that it can identify anomalies starting from the first day. Though passive behavioral data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

exhibit a high level of heterogeneity among individuals, a cohort-level baseline is still an acceptable benchmark to start with if we have little to no information about a given individual. Similar to the method described in Section 3.2, we construct a cohort-level histogram of original feature values for each feature and each day of the week, and the percentile of each observation in the cohort is used to derive the chi-squared test statistic. The histograms are updated by simply adding the count of new observations in each bin; no shifting or other manipulations are needed. Let Q_b denote the test statistic derived from cohort-level (between-individual) percentiles and Q_w denote the test statistic derived from within-individual percentiles. We propose a weighted average of the two, namely, $Q = wQ_b + (1-w)Q_w$ as the final test statistic, where $Q \xrightarrow{p} \chi^2_p$ since Q_b is asymptotically independent of Q_w in the number of participants. The value of w should be 1 in the first two weeks and it should vanish gradually as more data become available for the individual. In addition, we suggest to use a dynamic significance level to identify anomalies in practice. For example, we could set $\alpha = 0.1$ for the first month and decrease it gradually to 0.05 over time. The trajectories of both w and α should be tailored for each study, and they should depend on the relative likelihood of early vs. late relapse (anomaly).

2.2.5. Software implementation

Our group has developed the open source Beiwe *data collection* platform for smartphone-based digital phenotyping, which has been in continuous development and use since 2013[32]. We have also recently released Forest, which is an open source Python data analysis library for Beiwe data. Forest can be run independently of Beiwe, but the primary use case is for the two of them to be fully integrated directly on the Amazon Web Services (AWS) back-end. Cloud-based data analysis obviates the need to move large volumes of data, and it also implements the preferred big data computing paradigm where computation is taken to data rather than vice versa. It also makes the system more readily compliant with regional data privacy regulation, such as the General Data Protection Regulation 2016/679 (GDPR), which is a regulation in European Union law on data protection and privacy in the European Union and the European Economic Area[33].

We have implemented the proposed online anomaly detection method as a module within Forest. This means that in addition to running the method on existing data, interested readers have the capability to collect their own data using Beiwe and then run the online anomaly detection algorithm as part of Forest on a daily basis. The results can be stored in a database in the AWS back-end, and the open source implementation provides an API for using Tableau or similar software to visualize the results. The Forest module that implements the method as described in this paper is called Banyan[34]. It consists of eight user configurable parameters, including the period of the data, the number of bins in the histogram, and the significance level.

3. Results

3.1. Simulation with synthetic data

The test statistic in our online method consists of a within-individual component, the counterpart of the test statistic in the Barnett et al. method[2], and a between-individual component. In the following we study two important aspects of the method. First, we compare the within-individual component of the online test statistic with the offline test statistic. Second, we compare the performance of the online method with the weighted test statistic and the offline method. Our findings show that the value of the within-individual component of the test statistic approximates the offline test statistic but is faster to compute. Our proposed method of using the weighted average of both components works well in the first two weeks, and its performance in terms of sensitivity and specificity converges to the offline method when the follow-up period is long enough.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

3.1.1. Comparison of the within-individual component of the online test statistic and the offline test statistic

The within-individual component is derived by a two-step online algorithm, where we first obtain the rank-based transformed errors from the observed features and then update the covariance matrix using these errors. We examine (1) the difference between the ranks given the same observed features, (2) the difference between the covariance matrices given the same transformed errors, and (3) the difference between the test statistics given the same observed features for the two methods. We generated the observed features using the decomposition $y_{ij} = \mu_{ij} + s_{ij} + \epsilon_{ij}$, where $\mu_{ij} = 0, s_{ij} \sim N(0, 2), s_{ij} = s_{ij+7}$. The error term $\epsilon_j = [\epsilon_{1j}, ..., \epsilon_{pj}]$ was generated in three different ways: (1) a standard multivariate normal distribution, (2) p independent gamma distributions with $\alpha = 2, \beta = 0.5$, and (3) a multivariate normal distribution with a correlation 0.7 between any two features. The number of features was set to 20, 40, and 80, and the number of bins in the histogram was set to 50, 100, and 500. The data generation procedure was repeated ten times and results shown below are averages the replications.

Comparison of ranks. In each scenario, the ranks of the errors from our online algorithm are obtained by updating the histograms as described above, whereas the ranks from the offline method are derived by sorting errors from scratch in each update. We initialize the histograms using the first 100 observations and compare the ranks of the two methods starting from observation 101. The average absolute difference, the average absolute difference divided by the sample size, and Spearman correlation between the two sets of ranks using the most recent 50 observations in each update were computed. Since the ranking procedure happens within each feature, we only study how the number of bins affects the correlation using independent normal errors. As shown in Figure 1, the absolute difference between the two sets of ranks increases as sample size increases. This happens because ranks that are close to one another end up in the same bin. However, if we divide this absolute difference by the sample size, we find the ratio converges to the reciprocal of the number of bins, which means the expected deviation in ranks is $1/H$ of the sample size. The Spearman correlation is consistently above 99.5% in all three scenarios, and the correlation is greater for more granular histograms (those with greater value of H).

Comparison between covariance matrices. As the covariance matrix depends on the transformed errors and those errors are different between our method and the offline method, for the purposes of this simulation, we used the errors from the offline method to examine the performance of the modified covariance updating algorithm for both methods. In our method, the covariance matrix is updated as described in Section 3.3, while in the offline method, it is estimated empirically from scratch in each update. We investigated the Frobenius norm of the difference of the two matrices using simulated data with different numbers of features. As presented in the upper panel in Figure 2, the Frobenius norm of the difference is small but grows with the number of updates. After sufficiently many updates, the norm converges. The norm of the difference is larger when the number of features is larger due to the higher dimensionality of the difference matrix.

Comparison of test statistics. Here we evaluate our two-step algorithm to compute the within-individual test statistic and focus on the distribution of the test statistic. The same features are used for both algorithms and the corresponding computation time and test statistics are compared in various scenarios. The lower panel in Figure 2 shows that the run time of each update increases linearly as the sample size grows for the offline method, and it is greater than that of our proposed method even when the sample size is small. The run time of our method also increases linearly but more slowly in the first 900 updates; it then becomes a constant because we chose $K = 1000$ (the size of subset in Section 3.1) as the maximal number of historical values in memory to determine the general trend in this example. Additionally, the run time is positively associated with the number of features. Since the difference in run time caused by different numbers of bins and different error distributions are too small to be seen on the graph, an average line is used to represent each scenario.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

Figure 3 shows the Spearman correlation between the test statistics from the offline method and our proposed method using most recent 50 updates in each update. In the scenarios where the errors are generated from independent or correlated multivariate normal distributions, the correlations are consistently higher than 0.95 after the first 200 updates; increasing the number of bins results in higher correlations and lower variances. In scenarios where the errors are generated from independent Gamma distributions, the correlations between the two sets of test statistics fall to 0.9 and stabilize after 2000 updates.

Figure 4 depicts the distributions of the test statistics from various methods, compared to a standard χ^2 distribution with a degree of freedom specified as the number of features. The density plots of the test statistics from the offline method and our method coincide when the errors are normally distributed. However, when the errors follow a Gamma distribution, the mean of the test statistics from our method is smaller than that from the offline method and the mean of the asymptotic distribution. The absolute value of the difference is positively associated with the number of features.

3.1.2. Comparison of the performance of the online method with the weighted test statistic and the offline method

To simulate anomalies in the features, we generated a multivariate time series using the sine function with different scale(a_j) and phase(b_j) parameters, namely, $T(t_{ij}) = a_j \sin\left(\frac{t_{ij}}{c} + b_j\right)$, where T is an intermediate variable used later to generate the observed features, i denotes ith observation, j denotes jth feature, and c is a parameter that fixes the periodicity of the function to seven days. Next, we let $y_{i1} = T(t_{i1})$ and $y_{ij} = p_1 T(t_{i,j-1}) + p_2 T(t_{ij})$ with $j > 1$ and $p_1 + p_2 = 1$ to induce correlations in the features. Gaussian noise (with zero mean and unit standard deviation) was added to each feature of the original multivariate time series to increase the difficulty of detecting the anomalies and make the data more realistic. To generate artificial anomalies, we randomly selected m% of the sample, then for each observation we again randomly selected [30%,70%] of the features and altered their magnitudes by multiplying them by a uniformly distributed random variable $u \sim U[0,3]$. We replicated the procedure 100 times to generate the observed features for 100 individuals. In each replication, we set the number of features to 10 and the number of observations to 540.

In one simulation, we set the anomaly rate to 0 and significance level to 0.05 to study the type I error of the online method. Figure 5(a) shows that the online method has first an inflated false positive rate which then decreases to the nominal level after about 100 updates. In another simulation, we set the anomaly rate to 0.05 and the significance level to 0.05. In the online method, the weight of within-individual component was 0 for the first four weeks, increased linearly to 1 on day 112, and remained at 1 afterward. Accuracy, sensitivity, and specificity were calculated after each update and are presented in Figure 5(b)(c)(d). The online method was able to detect the anomalies in the first 14 days and the corresponding average accuracy, sensitivity and specificity are 91.2%, 50.4% and 93.3%, respectively. From day 15 to day 112, the online method has a higher sensitivity but slightly lower specificity. From day 112 on, the online method only uses the within-individual component and its performance is similar to that of the offline method. Note that as expected, the run time of the online method is much faster as shown in Figure 2.

3.2. Simulation with pseudo data

Panda et al.[30] conducted a study to collect raw smartphone accelerometer data continuously for 6 months from adults who had a cancer diagnosis and were scheduled for surgery between July 2017 and April 2019. The study was designed to discover if smartphones could capture novel postoperative recovery metrics among the patients. Most patients (45, 73%) experienced no clinically significant postoperative events, and those who experienced such an event did not report the exact date of when they started to feel unwell. Since there is no reliable ground-truth available for the timing of the anomalies, we instead chose to create artificial anomalies for patients who did not experience any.

 doi:10.20944/preprints202202.0081.v1

We constructed a dataset by first calculating the mean of each metric $\boldsymbol{\mu}$ for each of the 45 individuals (those without anomalies). We then calculated the individual-specific residuals $\boldsymbol{\epsilon}$, a difference vector of the actual observation and the mean vector for each day. To create a dataset of pseudo-observations with K-days of follow-up, we bootstrapped the errors K times and added them to the mean vector. To create anomalies, in the bootstrapping step we randomly sampled $5\% \times K$ of residuals and multiplied them by an inflation factor z, where $z \in \{1, 2, 3, 4\}$. In other words, for each day, the pseudo-observation is generated as $\mu + z\epsilon$. When $z = 1$, the dataset does not have any anomalies, and we expect our method to recover the nominal false-positive rate. We repeated the procedure 50 times and calculated the average accuracy, sensitivity, and specificity every 30 days with various z values across 45 individuals. As shown in Table 1, our online method achieves the nominal sensitivity when $z = 1$. When $z > 1$, sensitivity increases with the sample size increases and then plateaus to a stable level. As expected, sensitivity is higher for greater values of z. The corresponding accuracy is listed in Table 2.

3.3. Figures, Tables and Schemes

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

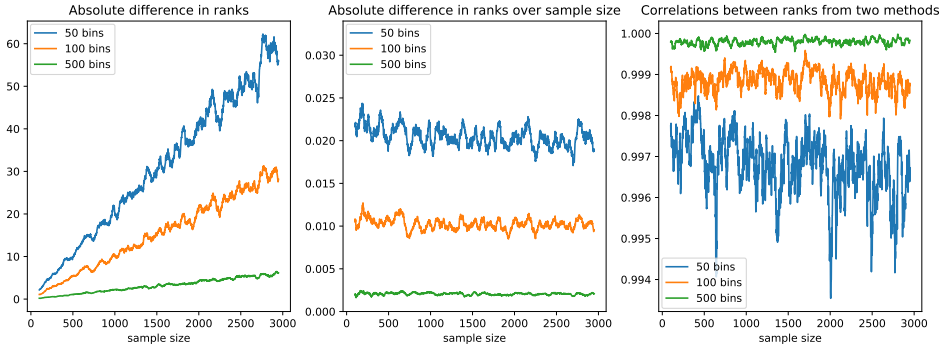


Figure 1. A comparison between ranks obtained by sorting from scratch and using the online histogram method. The left panel shows the average absolute difference between the two sets of ranks using the most recent 50 observations in each update averaged over five replications. The middle panel shows the average absolute difference over sample size, and the right panel shows the Spearman correlation in the same setting.

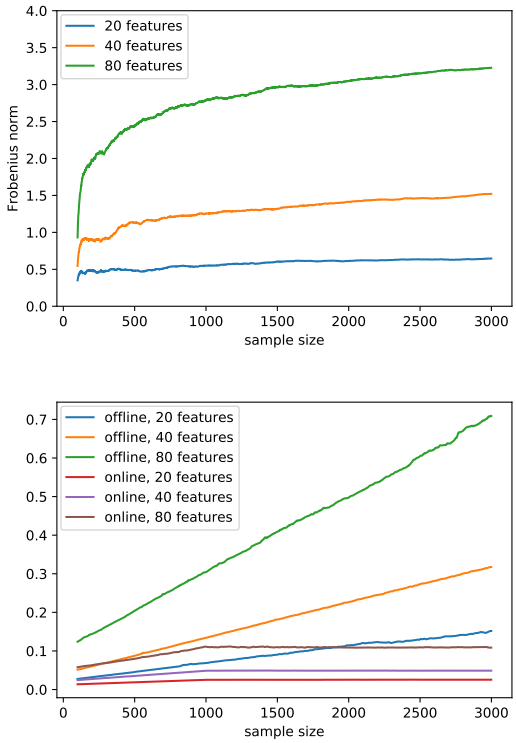
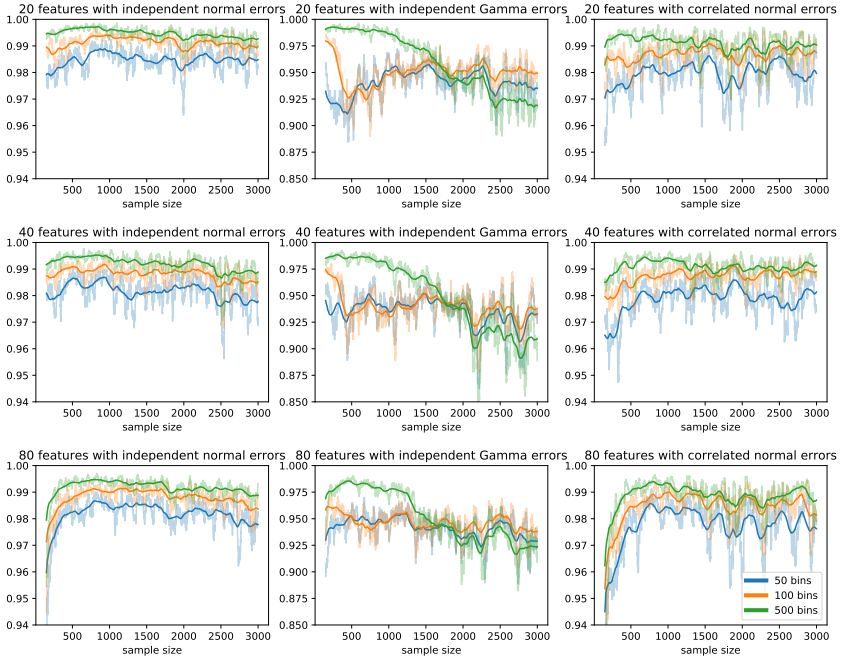


Figure 2. The upper panel is the Frobenius norm of the difference between the covariance matrices obtained from the empirical estimation and our proposed online algorithm in each update averaged over five replications. The transformed errors are taken from the offline method and used in both methods for the calculation of covariance matrices.The lower panel is a comparison of runtime between the offline method and our proposed method as the sample size increases from 100 to 3000. The run time in each update is measured in seconds with an Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz CPU. Since the difference in time caused by different numbers of bins and different distributions of errors are too small to be seen on the graph, an average line is used to represent all the scenarios given the method and the number of features.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

in



test$_s tat.pdf$

Figure 3. The Spearman correlations between the test statistics obtained from the offline method and our proposed method using the most recent 50 updates in each update. Each row represents a different number of features, each column represents a different distribution of error terms, and each color represents a different number of bins in the histogram method.

z\day	1-30	31-60	61-90	91-120	121-150	151-180
1	0.0566	0.0547	0.0514	0.0501	0.0492	0.0490
2	0.3765	0.4204	0.4143	0.4305	0.4254	0.4352
3	0.4505	0.4949	0.4849	0.4858	0.4915	0.4986
4	0.4558	0.5079	0.5071	0.5216	0.5355	0.5393

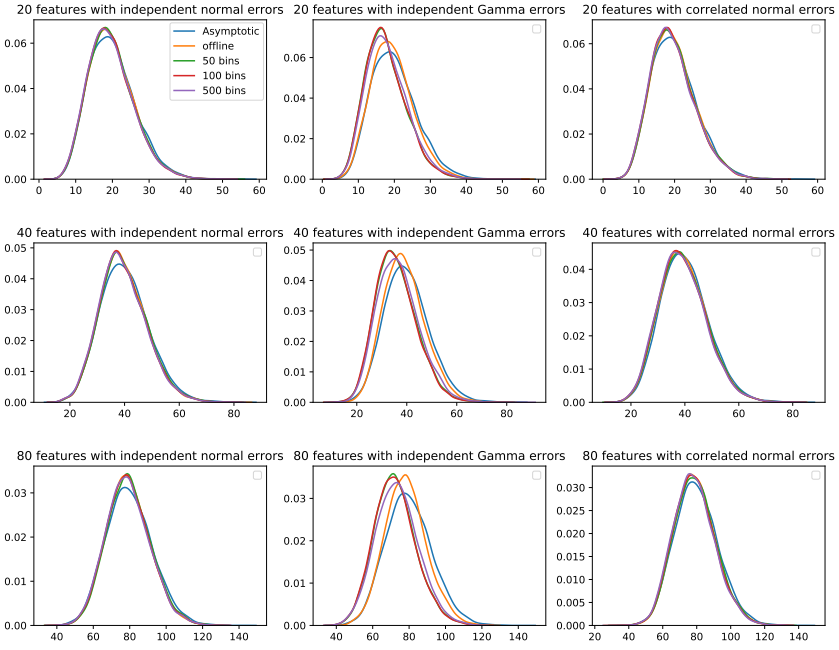
Table 1: The sensitivity of the online method to detect an artificial anomaly at different stages of follow-up among 45 patients over 50 repetitions with an anomaly rate of 0.05. For each day, the data were generated by $\boldsymbol{\mu} + z\boldsymbol{\epsilon}$, where $\boldsymbol{\mu}$ is the mean feature vector, $\boldsymbol{\epsilon}$ is the individual-specific residual generated by bootstrapping empirical residuals, and z controls the severity of the anomaly.

z\day	1-30	31-60	61-90	91-120	121-150	151-180
1	0.8916	0.8945	0.9016	0.9007	0.8998	0.9023
2	0.9405	0.9446	0.9482	0.9489	0.9485	0.9451
3	0.9482	0.9533	0.9556	0.9577	0.9586	0.9589
4	0.9490	0.9551	0.9586	0.9609	0.9646	0.9648

Table 2: The accuracy of the online method to detect an artificial anomaly, where the accuracy is defined as the rate of a correct classification.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 February 2022 doi:10.20944/preprints202202.0081.v1

of



$\text{test}_s tat.pdf$

Figure 4. The empirical distributions of the test statistics obtained from the offline method and our proposed online method, compared to a standard χ^2 distribution with a degree of freedom equal to the number of features. Each row represents a different number of features, each column represents a different distribution of error terms, and each color represents a different method.

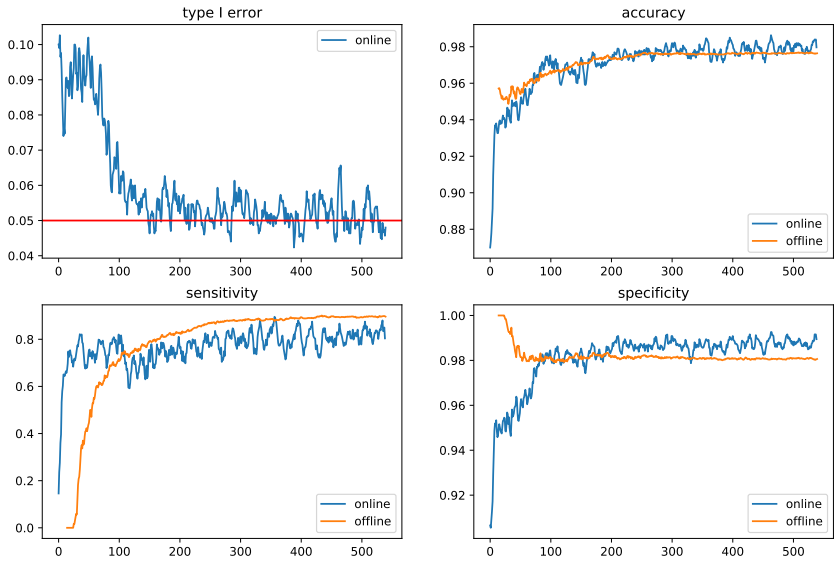


Figure 5. The average false positive rate, accuracy, sensitivity and specificity of the proposed method for 540 days across 100 synthetic dataset of 100 individuals. The underlying anomaly rate is 0 for studying type I error and the underlying anomaly rate is 0.05 for studying other metrics. The significance level is 0.05.

 doi:10.20944/preprints202202.0081.v1

4. Discussion

In this paper, we proposed an online algorithm for anomaly detection for multivariate time series where the computational complexity of each update has is $\mathcal{O}(1)$. The method is a natural extension of the offline method proposed by Barnett et al. [2], and given that it can make use of information from other subjects, not just the subject in question, the method can also detect anomalies in the first two weeks of data collection. The performance of the two methods is similar in terms of accuracy, sensitivity, and specificity when the number of observations for each individual is sufficiently large. The method has been implemented as a Python package in the Forest library[32].

The proposed method has some limitations. First, the errors derived by the decomposition may not reflect the extent to which the observation is anomalous if the feature is not self-predictive or the pattern is too complex to be approximated by a general trend and a periodic term. Second, the approach requires some expertise from the user to determine a reasonable period for transitioning from cohort-level data to individual-level data. Third, the test statistic is essentially a Mahalanobis distance, which measures the distance between the current observation and the median. It standardizes all features such that they have equal weights in the test statistic. However, some features may be more informative than others, and the current method does not provide a way for determining this. Fourth, the distribution of the test statistic is asymptotically a Chi-squared distribution, but in practice the empirical distribution is more concentrated around the mode, and sometimes the mode can even shift away from the expected value as we saw with the Gamma distributed errors. Hence, the p-value derived from the asymptotic distribution may not be accurate. However, in general, the method is robust against the mis-specification of the distribution since it is rank-based. Fifth, the anomalies are determined by a user-specified threshold, the significance level, rather than an estimate of the underlying anomaly rate. Thus, when the actual anomaly rate is very small, the method will have a large false positive rate.

Smartphones are promising for detecting anomalous behaviors given their ubiquity and the feasibility of using them for long-term follow-up, especially if relying mostly on passively collected data. Our online method for detecting anomalies in multivariate time series data is both simple and performs well in the studied settings. We believe that its transparency and interpretability are important strengths in clinical applications.

Author Contributions: Both authors developed the algorithm, tested the method and wrote the manuscript.

Funding: This research was funded by the scholarship from Graduate School of Arts and Sciences, Harvard University and NIH grants R21NR018532.

Acknowledgments: We are grateful to Dr.Nikhil Panda and Dr.Alex Haynes for their roles in the MGH cancers study which inspired the pseudo data in our simulation.

Conflicts of Interest: The authors declare no conflict of interest.

1. Shyi-Ming Chen and Shen-Wen Chen, Fuzzy Forecasting Based on Two-Factors Second-Order Fuzzy-Trend Logical Relationship Groups and the Probabilities of Trends of Fuzzy Logical Relationships. IEEE Transactions on Cybernetics *45*(3), 391–403
2. Ian Barnett and John Torous and Patrick Staples and Luis Sandoval and Matcheri Keshavan and Jukka-Pekka Onnela, Relapse prediction in schizophrenia through digital phenotyping: a pilot study. Neuropsychopharmacology *43*(8), 1660–1666
3. Mohiuddin Ahmed and Abdun Naser Mahmood and Md. Rafiqul Islam, A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems *55*, 278–288
4. Barbara Pilastre and Jean-Yves Tourneret and Stephane DEscrivan and Loic Boussouf, Multivariate Anomaly Detection in Mixed Telemetry time-series Using A Sparse Decomposition. 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP

5. Nan Ding and Huanbo Gao and Hongyu Bu and Haoxuan Ma, RADM:Real-Time Anomaly Detection in Multivariate Time Series Based on Bayesian Network. 2018 IEEE International Conference on Smart Internet of Things (SmartIoT)
6. Jinbo Li and Witold Pedrycz and Iqbal Jamal, Multivariate time series anomaly detection: A framework of Hidden Markov Models. Applied Soft Computing *60*, 229–240
7. Dan Li and Dacheng Chen and Baihong Jin and Lei Shi and Jonathan Goh and See-Kiong Ng,MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks.Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series, 703–716
8. Marco A.F. Pimentel and David A. Clifton and Lei Clifton and Lionel Tarassenko, A review of novelty detection. Signal Processing *99*, 215–249
9. Ruei-Jie Hsieh and Jerry Chou and Chih-Hsiang Ho, Unsupervised Online Anomaly Detection on Multivariate Sensing Time Series Data for Smart Manufacturing. 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)
10. Hu-Sheng Wu, A survey of research on anomaly detection for time series. 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)
11. Kai Wang and Youjin Zhao and Qingyu Xiong and Min Fan and Guotan Sun and Longkun Ma and Tong Liu, Research on Healthy Anomaly Detection Model Based on Deep Learning from Multiple Time-Series Physiological Signals. Scientific Programming *2016*, 1–9
12. Igor Melnyk and Bryan Matthews and Hamed Valizadegan and Arindam Banerjee and Nikunj Oza, Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems. Journal of Aerospace Information Systems *13*(4), 161–173
13. G. Enderlein, Hawkins, D. M.: Identification of Outliers. Chapman and Hall, London. Biometrical Journal *29*(2), 198–198
14. Edwin M. Knorr and Raymond T. Ng and Vladimir Tucakov, Distance-based outliers: algorithms and applications. The VLDB Journal The International Journal on Very Large Data Bases *8*(3-4), 237–253
15. Hans-Peter Kriegel and Peer Kroger and Erich Schubert and Arthur Zimek, Distance-based outliers: algorithms and applications. The VLDB Journal The International Journal on Very Large Data Bases *8*(3-4), 237–253
16. Yonggui Dong and Zhaoyan Sun and Huibo Jia, A cosine similarity-based negative selection algorithm for time series novelty detection. Mechanical Systems and Signal Processing *20*(6), 1461–1472
17. Diab M. Diab and Basil AsSadhan and Hamad Binsalleeh and Sangarapillai Lambotharan and Konstantinos G. Kyriakopoulos and Ibrahim Ghafir, Anomaly Detection Using Dynamic Time Warping. 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)
18. Bernhard Schölkopf and John C. Platt and John Shawe-Taylor and Alex J. Smola and Robert C. Williamson, Estimating the Support of a High-Dimensional Distribution. Neural Computation *13*(7), 1443–1471
19. Yongjun Jin and Chenlu Qiu and Lei Sun and Xuan Peng and Jianning Zhou, Anomaly detection in time series via robust PCA. 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)
20. Varun Chandola, Anomaly Detection for Symbolic Sequences and Time Series Data. The University of Minnesota
21. J. Ma and S. Perkins, Time-series novelty detection using one-class support vector machines, Estimating the Support of a High-Dimensional Distribution. Proceedings of the International Joint Conference on Neural Networks
22. Jukka-Pekka Onnela and Scott L Rauch, Harnessing Smartphone-Based Digital Phenotyping to Enhance Behavioral and Mental Health. Neuropsychopharmacology *41*(7), 1691–1696
23. Dror Ben-Zeev and Rachel Brian and Rui Wang and Weichen Wang and Andrew T. Campbell and Min S. H. Aung and Michael Merrill and Vincent W. S. Tseng and Tanzeem Choudhury and Marta Hauser and John M. Kane and Emily A. Scherer, CrossCheck: Integrating self-report, behavioral sensing, and smartphone use to identify digital indicators of psychotic relapse. Psychiatric Rehabilitation Journal *40*(3), 266–275
24. Laura J Faherty and Liisa Hantsoo and Dina Appleby and Mary D Sammel and Ian M Bennett and Douglas J Wiebe, Movement patterns in women at risk for perinatal depression: use

 doi:10.20944/preprints202202.0081.v1

of a mood-monitoring mobile application in pregnancy. Journal of the American Medical Informatics Association *24*(4), 746–753
25. Philip Henson and Ryan D'Mello and Aditya Vaidyam and Matcheri Keshavan and John Torous, Anomaly detection to predict relapse risk in schizophrenia. Translational Psychiatry *11*(1)
26. Alan J. Forster and Harvey J. Murff and Josh F. Peterson and Tejal K. Gandhi and David W. Bates, The Incidence and Severity of Adverse Events Affecting Patients after Discharge from the Hospital. Annals of Internal Medicine *138*(3), 161
27. Daniel Thomas Chung and Christopher James Ryan and Dusan Hadzi-Pavlovic and Swaran Preet Singh and Clive Stanton and Matthew Michael Large, Suicide Rates After Discharge From Psychiatric Facilities. JAMA Psychiatry *74*(7), 694
28. Janet Meehan and Navneet Kapur and Isabelle M. Hunt and Pauline Turnbull and Jo Robinson and Harriet Bickley and Rebecca Parsons and Sandra Flynn and James Burns and Tim Amos and Jenny Shaw and Louis Appleby,Suicide in mental health in-patients and within 3 months of discharge. British Journal of Psychiatry *188*(2), 129–134
29. Harriet Bickley and Isabelle M. Hunt and Kirsten Windfuhr and Jenny Shaw and Louis Appleby and Navneet Kapur, Suicide Within Two Weeks of Discharge From Psychiatric Inpatient Care: A Case-Control Study. Psychiatric Services *64*(7), 653–659
30. Nikhil Panda and Ian Solsky and Emily J. Huang and Stuart Lipsitz and Jason C. Pradarelli and Megan Delisle and James C. Cusack and Michele A. Gadd and Carrie C. Lubitz and John T. Mullen and Motaz Qadan and Barbara L. Smith and Michelle Specht and Antonia E. Stephen and Kenneth K. Tanabe and Atul A. Gawande and Jukka-Pekka Onnela and Alex B. Haynes, Using Smartphones to Capture Novel Recovery Metrics After Cancer Surgery. JAMA Surgery *155*(2), 123
31. Donald E. Knuth,The art of computer programming. Addison Wesley, 1997
32. Beiwe Research Platform. Available online: www.beiwe.org (accessed on 27 May 2021).
33. General Data Protection Regulation. Available online: https://gdpr-info.eu (accessed on 18 May 2021).
34. GitHub source code. Available online: https://github.com/onnela-lab/forest/tree/master/forest (accessed on 7 October 2021).

 doi:10.20944/preprints202202.0081.v1