Note: This is the accepted author manuscript, the published version can be found here: https://link.springer.com/article/10.1007/s00146-020-00985-1

Healthcare and Anomaly Detection: Using Machine Learning to predict Anomalies in Heart Rate Data

Edin Šabić[1,2], David Keeley[1,2], Bailey Henderson[1], Sara Nannemann[1,2]

[1]Electronic Caregiver, Las Cruces, New Mexico, United States of America

[2]New Mexico State University, Las Cruces, New Mexico, United States of America

Emails in order of authorship:

esabic@ecg-hq.com

dkeeley@ecg-hq.com

bhenderson@ecg-hq.com

snannemann@ecg-hq.com

Corresponding author: Edin Šabić (esabic@ecg-hq.com)

Keywords: Anomaly detection, healthcare, heart rate, machine learning

Author Contributions

In addition to the specified contributions below, all authors met the ICMJE author criteria. The great majority of effort, including all analyses and drafting, was accomplished by Edin Šabić. The design of the work and revisions were primarily led by David Keeley. Additionally, interpretation of the data and revisions were also performed by Bailey Henderson. Lastly, substantial feedback during early drafting as well as subject matter knowledge concerning nursing and patient management was provided by Sara Nannemann.

Abstract

The application of machine learning algorithms to healthcare data can enhance patient care while also reducing healthcare worker cognitive load. These algorithms can be used to detect anomalous physiological readings, potentially leading to expedited emergency response or new knowledge about the development of a health condition. However, while there has been much research conducted in assessing the performance of anomaly detection algorithms on well-known public datasets, there is less conceptual comparison across unsupervised and supervised performance on physiological data. Moreover, while heart rate data is both ubiquitous and noninvasive, there has been little research specifically for anomaly detection of this type of data. Considering that heart rate data is indicative of both potential health complications and an individual's physical activity, this is a rich source of largely overlooked data. To this end, we employed and evaluated five machine learning algorithms, two of which are unsupervised and the remaining three supervised, in their ability to detect anomalies in heart rate data. These algorithms were then evaluated on real heart rate data. Findings supported the effectiveness of local outlier factor and random forests algorithms in the task of heart rate anomaly detection, as each model generalized well from their training on simulated heart rate data to real world heart rate data. Furthermore, results support that simulated data can help configure algorithms to a degree of performance when real labeled data is not available and that this type of learning might be especially helpful in initial deployment of a system without prior data.

Keywords: Anomaly detection, healthcare, heart rate, machine learning

1. Introduction

The detection of anomalies, also known as outliers, has been an integral practice across a diverse range of disciplines. Anomalies can indicate a breach of a system or network (Garcia-Teodoro, Diaz-Verdejo, Maciá-Fernández, & Vázquez, 2009; Omar, Ngadi, & Jebur, 2013), signal abnormal physiological levels (Jothi, Rashid, & Husain, 2015), or even flag the occurrence of fraud (Liu et al., 2016). Regardless of the particular application, analytics and machine learning models have the potential to provide both predictive and descriptive value. Descriptive value can include new understandings of data interactions, an example being the use of visualizations to help researchers and practitioners alike better understand a phenomenon. In the context of healthcare, this could manifest as new insights into the development of a health complication or condition. Predictive value, where the focus lies not in describing data but predicting future states or events, can potentially improve patient outcomes through active intervention. Indeed, a great promise of healthcare anomaly detection is that the process can be used to alert health practitioners to anomalous physiological data that can be indicative of health complications. A greater emphasis on the application and visualization of both supervised and unsupervised anomaly detection models will improve practice in applied settings such as healthcare – and, in turn, improve patient outcomes.

Technology is ubiquitous in healthcare, but the application of machine learning to physiological data is still in its early stages. Adding to the already difficult and demanding work of health care professionals is the shortage of workers in the public health sector. Improving our strategic use of the petabytes of physiological data collected during patient monitoring will enhance both patient care and alleviate some of the burden carried by health care professionals. Simply put, the amount of vigilance needed to provide care and track patient vitals in this context

is impossible to maintain for many healthcare operations. Machine learning provides a potential solution to this dilemma. Through the automatic detection of rare or substantial deviations from normal levels, health care professionals can be provided with more useful information with which to make clinical decisions, as well as respond more quickly to an adverse event. Whether anomaly detection algorithms are implemented to detect patient falls at home (Albert, Kording, Herrmann, & Jayaraman, 2012), monitor health vitals through wearable sensors (Banaee, Ahmed, & Loutfi, 2013), or assist with other patient care-related tasks, these algorithms provide an avenue to automate some components of patient care. However, as has been continuously supported in the literature, what is considered by the system as an anomaly is often difficult to define.

Clearly, the process of anomaly detection depends on an operational definition for what is to be considered an anomaly. Anomaly detection methods generally assume two things; namely: 1) anomalies are rare in occurrence and 2) anomalies are different – in some sense or another – from normal data. Further complicating this operational definition is the existence of multiple types of anomalies (Liu, Ting, & Zhou, 2008). For instance, data points can be anomalous in respect to nearby data points (local anomalies) or in respect to the dataset as a whole (global anomalies). While it is impossible to determine exactly to which of these classes an anomaly belongs, the distinction is still useful in reminding data scientists that anomalies can differ not just from the data – but from each other as well.

Reducing the number of false alerts is a challenging problem in anomaly detection (Haque, Rahman, & Aziz, 2015; Omar, Ngadi, & Jebur, 2013), as an excessively high false alarm rate is detrimental to both the potentially efficiency of such a system and for the signal value of the alarm. That is, a high false alarm rate can increase the chances of alarm fatigue, a

problem that is especially prevalent for nurses in wards where auditory alarms continuously bombard the environment (Cvach, 2012). Alarm fatigue is the result of desensitization to a particular alarm or set of alarms – often a byproduct of continuous presentation leading to decreased salience. False alarms can be the result of an erroneous sensor reading, or just a random – but benign – fluctuation in the physiological data. Regardless of the nature of the false alarm, these situations decrease both trust in the alarm itself and the efficiency of the system as a whole.

In many cases, deciding which machine learning approach to implement is informed by both experimentation and the fundamental nature of the task. Indeed, there is much variability in the machine learning models used in healthcare applications. For instance, researchers have applied neural networks to problems in healthcare (Wang et al., 2016), while others utilize clustering techniques (Bose et al., 2018). Still others use multi-layer perceptrons (Adnan et al., 2017) or long-term memory networks (Malhotra, Vig, Shroff, & Agarwal, 2015). Unsurprisingly, the choice of algorithm is partly determined by the type of data or amount of data available – often a result of the type of device that is capturing physiological or movement data as input. Researchers have investigated the potential of smart phones to capture data in this regard (Amin et al., 2016), and, in general, wearable sensors for health monitoring appear to have a promising future in healthcare (Banaee et al., 2013).

Identifying an optimal machine learning approach to a problem is also dependent on whether ground truth labels exist on the data. If the data is labeled, supervised techniques are appropriate, while unsupervised techniques allow for analysis of unlabeled data (see Goldstein & Uchida, 2016, for an excellent comparison of unsupervised models). However, it should be briefly mentioned that semi-supervised anomaly detection algorithms provide yet another option.

In semi-supervised anomaly detection, the dataset includes both labeled and, often a large amount of, unlabeled data. While this type of learning has a great potential as labeled data can often be difficult to obtain, these types of models have unique assumptions and limitations (Zhu & Goldberg, 2009).

While there are myriad models that can be successfully tuned to handle anomaly detection problems, the present research focuses on the following five: local outlier factor (LOF), isolation forests (IF), support vector machines (SVMs), *k*-nearest neighbors (*k*-NNs), and random forests (RF). These algorithms were chosen for (1) their widespread prevalence in the field of anomaly detection and machine learning tasks in general, and (2) because they constitute both supervised and unsupervised approaches. One advantage of this approach is the ability to critically evaluate both unsupervised and supervised approaches to the task of anomaly detection.

Briefly, it is important to fundamentally overview the machine learning models chosen. While it is beyond the scope of this paper to exhaustively describe the techniques and algorithmic details of each model in detail, we nevertheless overview each model and note some use cases.

1.1. Random Forests (RF)

RF is an ensemble decision tree method that can be used for both prediction and classification tasks (Breiman, 2001). The trees of the model each cast a weighted decision or vote to produce an overall determination (Dietterich, 2000). In classification tasks, the prediction of new data is determined by aggregating the predictions of *n* trees. Ensemble methods function best when there is some variance across how the individual classifiers make their predictions (Hansen & Salamon, 1990). There are two main sources of randomness within the RF algorithm. First, RF only use a subsample of the data during each tree creation, and approximately 36.8% of

the data is unused by each tree (Grömping, 2009). This process, known as bootstrap aggregating or bagging, samples subsets of the data with replacement (Breiman, 1996). Second, instead of assessing all features at each tree split, only a random set of samples is used to make the eventual decision – helping to counteract overfitting (Liaw & Wiener, 2002). Impressively, even shallow decision trees have been proven effective in applications such as handwritten character recognition (Amit & Geman, 1997).

1.2. Local Outlier Factor (LOF)

The LOF algorithm was introduced by Breunig, Kriegel, Ng, and Sander (2000). As the name suggests, the algorithm primarily assesses how isolated each data point is with respect to its neighbors – and as such the algorithm depends on *k*-nearest neighbors. While other algorithms produce a binary classification of *outlier-ness*, LOF can also produce the degree to which a point is an outlier. This local density approximation is calculated through *k*-nearest neighbors, and data points residing in relatively lower density areas are classified as outliers or given a higher degree of *outlier-ness*.

1.3. Isolated Forests (IF)

IF is an algorithm that uses a process called isolation to determine anomalies from normal data points (Liu, Ting, & Zhou, 2008). Unlike other anomaly algorithms that are trained on normal instances, IF is an unsupervised ensemble method where anomalies are those data points which are easier to isolate from normal data points. A process known as isolation is used to partition each data point until each instance is isolated. Data points which are more quickly isolated, in terms of steps, are considered more likely to be anomalies.

1.4. Support Vector Machines (SVMs)

SVMs classify by attempting to derive a hyperplane that maximizes the distance between classes within the data (Hu, Liao, & Vemuri, 2003). Said differently, this hyperplane is an optimal boundary that divides the data to minimize the misclassification error. This algorithm has been used successfully in intrusion detection (Mukkamala, Janoski, & Sung, 2002), and system health estimation (Sotiris, Tse, & Pecht, 2010).

1.5. *K*-Nearest Neighbors (*k*-NNs)

k-NNs is a very simple and popular machine learning method which heavily depends on a parameter, *k*, to determine how to classify data in reference to nearby data. In the simplest instance where $k = 1$, the class of each data point will simply be determined based on the nearest point to the given data point as measured by Euclidean distance. *k*-NNs have been used successfully in network anomaly detection (Muniyandi, Rajeswari, Rajaram, 2012), intrusion detection (Yassin, Udzir, Muda, Sulaiman, 2013), and in general have been shown to be quite powerful across a series of datasets (Goldstein & Uchida, 2016).

2. Methods

2.1 Model Development

Five models were created within Python (Anaconda distribution) using many packages from the popular sci-kit learn library (https://scikit-learn.org). We chose the same general approach for the development of each algorithm, although each was tuned separately.

2.1.1. Parameter Tuning

To determine the parameters of each model, we used a series of loops which changed one parameter of the model at a time and evaluated performance through *k*-fold cross-validation ($k = 5$).

2.1.2. Ground Truth

The simulated data was inherently unlabeled; however, in order to tune and develop the models we chose to label any point outside of a range (60bpm – 100bpm) as anomalous. This range has been used by others as a general rule-of-thumb (Liu et al., 2014), and was necessary for evaluation of the algorithms as well as the development of the supervised algorithms. This labeled array was used when calling the .fit() function for all supervised algorithms, and used as a form of evaluation for all unsupervised algorithms.

2.1.3. Training and Validation Data

A peripheral goal of the present research was to investigate the potential of simulated heart rate data in model training. Simulated data carries the disadvantage, of course, of not including the noise and variation of actual data. However, a major advantage of using simulated data is an increased amount of control in terms of both the number of anomalies and the distribution of the data.

For our purposes, simulated normal data was created through a C script written for a RR interval time series modeling challenge (https://archive.physionet.org/challenge/2002/generators/rrgen-171.c). We then randomly sampled at which locations that an outlier would appear for two datasets that would consist of 10,000 simulated heart rate values each. In one dataset, anomalies constituted 0.5% of the entire dataset (defined as residing outside the heart rate range of 60bpm – 100 bpm), while in the other dataset anomalies constituted 2.5% of the entire dataset. To determine the heart rate value of the anomaly, we alternated between randomly sampling from 101 – 120 beats per minute (bpm) and 40 – 60 bpm. A Python function was used to sample without replacement for where in time an anomaly would appear, and all values in the range were equally likely. Similarly, another Python

function was used to sample with replacement for anomaly values, generating a total of 250 values for the 2.5% anomaly dataset and 50 values for the 0.5% anomaly data set.

2.1.4. Testing Data

Test data consisted of a single patient from the MIT-BIH database distribution (http://ecg.mit.edu/time-series/) that matched the general heart rate pattern of those using our company products. We chose this approach as our models will be implemented on an individualized basis. Note that there is no ground truth label with which to evaluate the algorithms in this testing dataset. To arbitrarily create labels would be counterproductive, as, again, these do not exist in the real world unless they are retroactively added after new information is gained on the situation – potentially through clinical appointments or consultation with a physician or doctor. Indeed, this ambiguity concerning the actual severity of a detected anomaly reflects the difficulties faced by those developing anomaly detection algorithms for healthcare use. Nevertheless, this ambiguity is a reality. As a result, we have maintained this ambiguity in the evaluation of the models. For evaluation purposes, visualizations and comparisons across the algorithms constituted the majority of the testing results. Importantly, this inability to be certain of whether a detected anomaly is actually anomalous has been echoed by others in the literature (Wang et al., 2016).

3. Analysis

The following analysis encompasses the training and testing of five algorithms (two unsupervised, three supervised) on two simulated heart rate datasets – each including 10,000 heart rate samples. As mentioned earlier, we simulated heart rate data to include one dataset which included a total of 50 anomalies (.5% of the dataset) and another which included a total of 250 anomalies (2.5% of the dataset). For our purposes, we defined a point as anomalous if it was

outside the range of 60 – 100 bpm. While an individual can certainly be outside this range without any significant health risk, this approach nevertheless allows for a general rule through which the algorithm can be tuned. We then turn to exploratory evaluation through visualizations of these models on actual heart rate variability to assess the behavior of the algorithms.

3.1. Feature Engineering

Our final feature set included only three total features outside of the raw data, these included: 1) difference between current heart rate value and the last heart rate value, 2) difference between current heart rate value and the moving average of the last five heart rate values, and 3) a *k*-means clustering feature based on 100 clusters on the aforementioned second feature (moving average difference) using the MiniBatchKMeans() function. We did not normalize the data as initial normalization efforts decreased the performance of the models.

3.2. Simulated Data Performance

Train and test performance for each algorithm trained on either the 0.5% anomaly dataset or the 2.5% anomaly dataset can be seen below (see Table 1). We implemented an 80:20 split of the data to create the training and testing datasets, respectively. We tuned model parameters through a series of loops that iterated through changes in each relevant parameter of the given model and assessed performance using a 5-fold cross-validation approach. This also allowed for us to evaluate the significance of features to the model in order to drop any irrelevant features. Heart rate data was approximately normal with a slight negative skew (see Figure 1 below).

Table 1

Performance of all five algorithms across both datasets.

	0.5% Anomaly Dataset		2.5% Anomaly Dataset	
	CR	HR	CR	HR
k-NN	99.98%	95%	99.96%	97.62%
SVM	99.96%	95%	99.97%	98.73%
RF	100%	92.5%	100%	100%
LOF	98.94%	100%	96.89%	100%
IF	94.22%	100%	94.76%	100%

Note. CR = Correct rejection, HR = Hit rate, *k*-NN = *k*-nearest neighbors, SVM = support vector machine, RF = random forests, LOF = local outlier factor, IF = isolation forests.

--- Figure 1 here ---

Fig. 1. Distribution of heart rate data for the MIT-BIH database patient.

3.3. Evaluation on MIT-BIH Database

Beginning first with the models trained on the 0.5% anomaly dataset, both the *k*-NN and SVM algorithms did not make many anomalous predictions. The *k*-NN algorithm classified one point as anomalous after a sudden spike in the data from around 83bpm to 100 bpm, while the SVM algorithm classified that same point as anomalous in addition to one more point at roughly 105 bpm. Moreover, the LOF algorithm classified more points as anomalies, both at the upper and lower ends of the heart rate data (see Figure 2). The RF model predicted only values at the upper end of heart rate data as anomalous and appeared to learn the rule (anomalous points are those that are outside of the 60 – 100bpm range). The IF model began flagging anomalies at

around the 95 bpm range and increased with frequency for points above 100 bpm. Both RF and IF visualizations are shown below (see Figure 3).

--- Figure 2 here ---

Fig. 2. Detected anomalies (each shown as a red x) for the local outlier factor model trained on the 0.5% anomaly dataset.

--- Figure 3a and 3b stacked here ---

Fig. 3. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 0.5% anomaly dataset.

Turning now to the models fitted to the 2.5% anomaly dataset, the *k*-NN and SVM algorithms performed similarly to those trained on the 0.5% anomaly dataset and predicted only a few anomalies in the data. The LOF model classified anomalies at both ends of the heart rate range and made most anomalous predictions at large spikes in the heart rate data. Interestingly, the LOF model predicted a cluster of points as anomalies – each of which was above 100 bpm and followed a particularly large spike in the heart rate data (see Figure 4).

--- Figure 4 here ---

Fig. 4. Detected anomalies (each shown as a red x) for the local outlier factor model trained on the 2.5% anomaly dataset.

When comparing the RF and IF algorithms (see Figure 5), both models predict anomalies mostly at the upper end of the heart rate data. However, the IF model did classify a few heart rate samples at the lower end of the heart rate data as anomalous. Further, the IF algorithm exhibited a higher anomaly classification rate than the RF model, although this is not surprising given that the IF model is unsupervised and was never given any labels during training.

--- Figure 5a and 5b stacked here ---

Fig. 5. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 2.5% anomaly dataset.

4. Discussion

The automatic detection of anomalies in physiological data, such as heart rate measurements, constitutes a challenging but important venture. Indeed, some of the difficulty in detecting health data anomalies is the degree of randomness associated with human physiological data. For instance, the anomaly detection system might flag an individual's current heart rate as anomalous, but in reality they might just be beginning a high-intensity workout regimen. The present research provides insight into how five machine learning algorithms perform in detecting anomalies after being fitted to two simulated datasets of heart rate data. Furthermore, we were able to evaluate the five models on one patient's heart rate data provided by the MIT-BIH database.

Performance on the simulated heart rate dataset was quite high for many of the algorithms. In the 0.5% anomaly dataset, the RF model learned the ground truth rule (anomalies are data points outside of the normal range of 60 – 100bpm) very quickly, with a hit rate of

92.5% and a correct rejection rate of 100%. Impressively, the unsupervised counterpart to the RF algorithm, IF, exhibited a hit rate of 100%. However, this was somewhat marred by the high false alarm rate as evidenced by a correct rejection rate of around 94%. Clearly, in real-world contexts false alarms will decrease both trust in the anomaly detection system and its efficiency. Unless the penalty for a false alarm is quite low, these prediction errors can have a significant impact on the usability of such a system. Both the *k*-NN and SVM algorithms performed best at predicted anomalies in the 0.5% anomaly dataset, with each displaying above 99% correct rejection performance and a 95% hit rate performance.

Unlike findings observed in the 0.5% simulated dataset, the RF model performed best in terms of overall performance when trained and tested on the 2.5% dataset. Here, the RF model exhibited ceiling performance (CR = 100%, HR = 100%), while its counterpart, IF, exhibited a high hit rate but, again, displayed a lower correct rejection rate (CR = 94.22%, HR = 100%). The *k*-NN and SVM models performed similarly to those that were trained on the 0.5% simulated dataset, with the exception that their hit rates were higher on the 2.5% simulated dataset ($HR_{k\text{-NN}}$ = 97.62%, HR_{SVM} = 98.73%). The LOF model was the best performing unsupervised algorithm, with a perfect hit rate and an impressive correct rejection rate of 96.89%. Perhaps unsurprisingly, the unsupervised algorithms performed worse than supervised algorithms at the given task.

4.1. Anomaly Detection for Real-World Data

However, as our focus was healthcare and the actual implementation of machine learning into heart rate monitoring, we believed it was integral to evaluate how the present algorithms would perform in predicting anomalies for a sample of real heart rate data. Here, we turned to visualizations to evaluate how frequently the algorithms flag outliers and to understand how the previous training step on simulated data influenced anomaly detection performance on real heart

rate data. It is integral to keep in mind the importance of high levels of both specificity and sensitivity with these algorithms, as prediction methods that exhibit low sensitivity will not function well in their primary task (e.g., the detection of anomalies), while prediction models that exhibit low specificity will lead to a high false alarm rate. As these anomaly detection models are implemented into real-world contexts, maintaining high values across both of these system performance measures will ensure that practitioners and patients alike can place trust in the system to make accurate predictions.

Beginning first with the algorithms fitted to the 0.5% anomaly dataset, both the SVM and K-NN algorithms did not make many anomalous predictions in the MIT-BIH data. This reflects either a very conservative approach or that the real-world data was too dissimilar to our simulated data to model adequately. The LOF algorithm classified anomalies at both ends of the heart rate range as anomalies, and exhibited a conservative approach that would be useful in some real world contexts.

The RF model, on the other hand, predicted many points as anomalies but only at the upper end of the data. The RF model appeared to learn the rule rather quickly, and appeared to simply classify points that were above 100 as anomalies. In contrast, the IF model classified points at both ends as anomalous. However the IF rate of anomaly classification was far too high and, therefore, the model would most likely exhibit a high false alarm rate in the real world. Turning to the models fitted on the 2.5% anomaly dataset, the LOF algorithm predicted many more points at the upper end of the heart rate range as anomalous. However, the LOF rate of anomaly classification did not appear to be too high (e.g., it did not constitute a high false alarm rate) and the model's performance on the task was, overall, impressive. Again, the SVM and K-NN algorithms did not predict many anomalies when trained on the 2.5% anomaly data.

Interestingly, the RF model trained on the 2.5% anomaly dataset performed identically to the RF model trained on the 0.5% anomaly dataset. It should be noted that all classified anomalies for the RF models were above the 100 bpm range. The IF model trained on the 2.5% anomaly dataset performed similarly to its counterpart that was trained on the 0.5% anomaly dataset. Specifically, the IF model exhibited a high hit rate but also a moderately high false alarm rate – which would decrease its efficiency in many real world applications. One could argue that both LOF algorithms performed best – classifying only data points as anomalies after sudden spikes or falls and exhibiting conservative behavior (low false alarm rate). Of course, it is important to briefly mention that the ideal balance between hit rate and false alarm rate will depend on the task-related penalties of the anomaly detection. That is, if the given anomaly detection application has heavy penalties for false alarms, such as high costs for an emergency response, then it would clearly be important to prioritize a low false alarm rate.

Many of the algorithms did not label sudden drops in the MIT-BIH data as anomalies. It should be noted that many of these drops were within the normal range as specified during simulated data fitting, so this could be an artifact of this process. Interestingly, the RF and IF models performed similarly across both the 0.5% and 2.5% anomaly dataset. That is, the pattern of classification for both models did not change much as a result of a different training dataset. Lastly, it should be stressed that the entirety of the model evaluation here is only based on the behavior of the algorithms through visualizations and not based on ground truth labels. Our objective with this task was to evaluate the potential of simulated data in training an algorithm based on anomaly prevalence. We then visualized the performance of the models on a novel and real-world heart rate sample, noted differences across the models, and finally differentiated performance based on which data the model was trained. Future research is needed to evaluate

alternatives to our approach, as well as to extend the results using a large database of real, ideally labeled, heart rate data. Furthermore, while we chose to specifically focus on anomaly detection in heart rate data because of the lack of research in this regard, future investigations should also evaluate how other physiological measures – such as blood pressure, temperature, and so on – can be used in tandem to predict anomalies at a greater scale.

We present two important considerations in terms of real-world implementation of these algorithms. First, the algorithms utilized in the present research scale well with large amounts of data, as they all allow for parallel processing. Specifically, the speed of these algorithms is benefitted by the fact that their processing of the data can be distributed across many computing resources (Bekkerman, Bilenko, & Langford, 2011), and also by the algorithms' parallel, rather than serial, development. For instance, the random forests algorithm discussed within the present research can construct decision trees in parallel, rather than developing each tree one after another – significantly expediting the process. Second, these algorithms necessitate a degree of tuning and monitoring to ensure proper functioning in real-world applications. The present research accomplished this using a classification rule to determine anomalies, and through a validation step on actual heart rate data. In real-world implementation, the performance of these algorithms will have to be monitored during deployment to ensure that data classified as anomalies are, indeed, anomalous. If unwanted performance is occurring (e.g., a very high false alarm rate), the algorithm can be differentially tuned through proper testing and validation to allow for improved performance.

5. Conclusion

The present research evaluated five algorithms tuned for anomaly detection. Through our unique approach, we were able to evaluate performance of the models based on whether they

were trained with a dataset that included 0.5% anomalies or 2.5% anomalies. Performance on the simulated dataset was quite high, with *k*-NN and SVM algorithms performing best on the 0.5% dataset and random forests performing best on the 2.5% dataset. Moreover, evaluation on the MIT-BIH heart rate sample data demonstrated the efficacy of these models after they had been trained on simulated data. When trained on either dataset, the LOF algorithm appeared most promising through its conservative classification (low anomaly detection rate) and also its ability to classify anomalies at both lower and upper heart rate ranges. We conclude that simulated data can help tune algorithms to some degree of performance when real labeled data is unavailable, and this type of imposed rule-based learning might be especially helpful when initially employing a system without any prior data.

Acknowledgement

The authors would like to thank both New Mexico State University and Electronic Caregiver for supporting this research. Additionally, the authors would like to thank Hannah Rich, Andrew Washburn, Morgan Beasley, and Taylor Bunker for their comments on the manuscript.

References

Adnan, J., Daud, N. N., Mokhtar, A. S. N., Hashim, F. R., Ahmad, S., Rashidi, A. F., & Rizman, Z. I. (2017). Multilayer perceptron based activation function on heart abnormality activity. *Journal of Fundamental and Applied Sciences*, *9*(3S), 417-432.

Albert, M. V., Kording, K., Herrmann, M., & Jayaraman, A. (2012). Fall classification by machine learning using mobile phones. *PloS one*, *7*(5), e36556.

Amin, M., Banos, O., Khan, W., Muhammad Bilal, H., Gong, J., Bui, D. M., ... & Chung, T. (2016). On curating multimodal sensory data for health and wellness platforms. *Sensors*, *16*(7), 980.

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation*, *9*(7), 1545-1588.

Banaee, H., Ahmed, M., & Loutfi, A. (2013). Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges. *Sensors*, *13*(12), 17472-17500.

Bekkerman, R., Bilenko, M., & Langford, J. (Eds.). (2011). Scaling up machine learning: Parallel and distributed approaches. Cambridge University Press.

Bose, E. L., Clermont, G., Chen, L., Dubrawski, A. W., Ren, D., Hoffman, L. A., ... & Hravnak, M. (2018). Cardiorespiratory instability in monitored step-down unit patients: using cluster analysis to identify patterns of change. *Journal of clinical monitoring and computing*, *32*(1), 117-126.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In ACM sigmod record (Vol. 29, No. 2, pp. 93-104). ACM.

Cvach, M. (2012). Monitor alarm fatigue: an integrative review. *Biomedical instrumentation & technology*, *46*(4), 268-277.

Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In *International workshop on multiple classifier systems* (pp. 1-15). Springer, Berlin, Heidelberg.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, *28*(1-2), 18-28.

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, *11*(4), e0152173.

Grömping, U. (2009). Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*, *63*(4), 308-319.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10), 993-1001.

Haque, S., Rahman, M., & Aziz, S. (2015). Sensor anomaly detection in wireless sensor networks for healthcare. *Sensors*, *15*(4), 8764-8786.

Hu, W., Liao, Y., & Vemuri, V. R. (2003, June). Robust Support Vector Machines for Anomaly Detection in Computer Security. In *ICMLA* (pp. 168-174).

Jothi, N., & Husain, W. (2015). Data mining in healthcare–a review. *Procedia Computer Science*, *72*, 306-313.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413-422). IEEE.

Liu, J., Bier, E., Wilson, A., Guerra-Gomez, J. A., Honda, T., Sricharan, K., ... & Davies, D. (2016). Graph analysis for detecting fraud, waste, and abuse in healthcare data. *AI Magazine*, *37*(2), 33-46.

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015, April). Long short term memory networks for anomaly detection in time series. In *Proceedings* (p. 89). Presses universitaires de Louvain.

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)* (Vol. 2, pp. 1702-1707). IEEE.

Muniyandi, A. P., Rajeswari, R., & Rajaram, R. (2012). Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm. *Procedia Engineering*, *30*, 174-182.

Omar, S., Ngadi, A., & Jebur, H. H. (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, *79*(2).

Sotiris, V. A., Peter, W. T., & Pecht, M. G. (2010). Anomaly detection through a bayesian support vector machine. *IEEE Transactions on Reliability*, *59*(2), 277-286.

Wang, K., Zhao, Y., Xiong, Q., Fan, M., Sun, G., Ma, L., & Liu, T. (2016). Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals. *Scientific Programming*, *2016*.

Yassin, W., Udzir, N. I., Muda, Z., & Sulaiman, M. N. (2013, August). Anomaly-based intrusion detection through k-means clustering and naives bayes classification. In *Proc. 4th Int. Conf. Comput. Informatics, ICOCI* (Vol. 49, pp. 298-303).

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), 1-130.

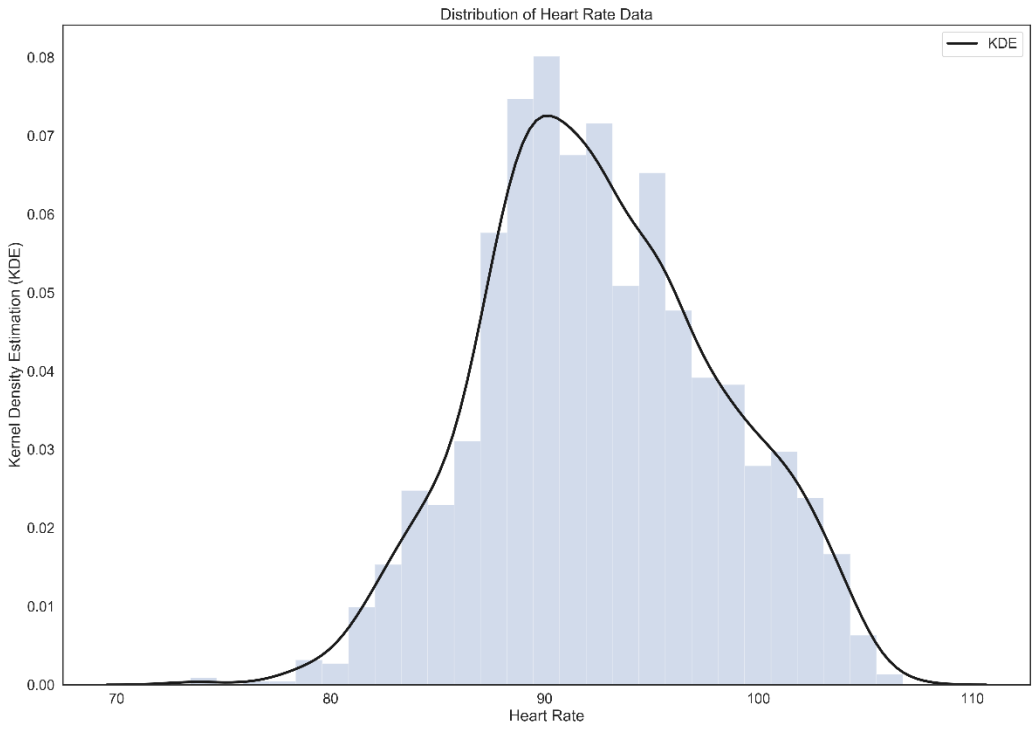

Fig. 1. Distribution of heart rate data for the MIT-BIH database patient.

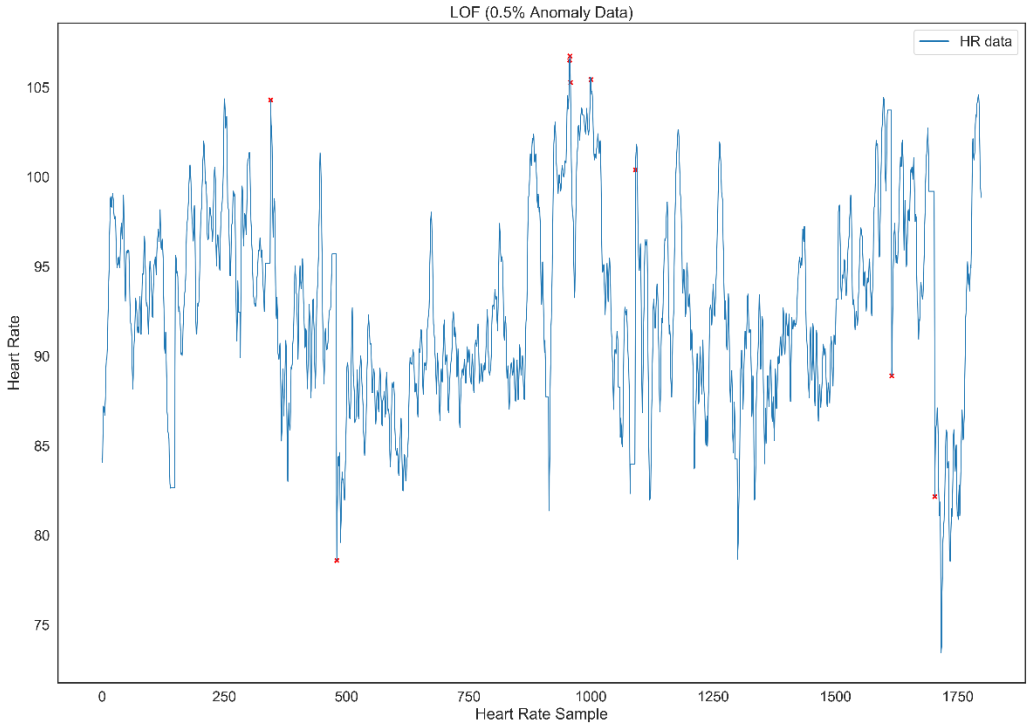

Fig. 2. Detected anomalies (each shown as a red x) for the local outlier factor model trained on the 0.5% anomaly dataset.

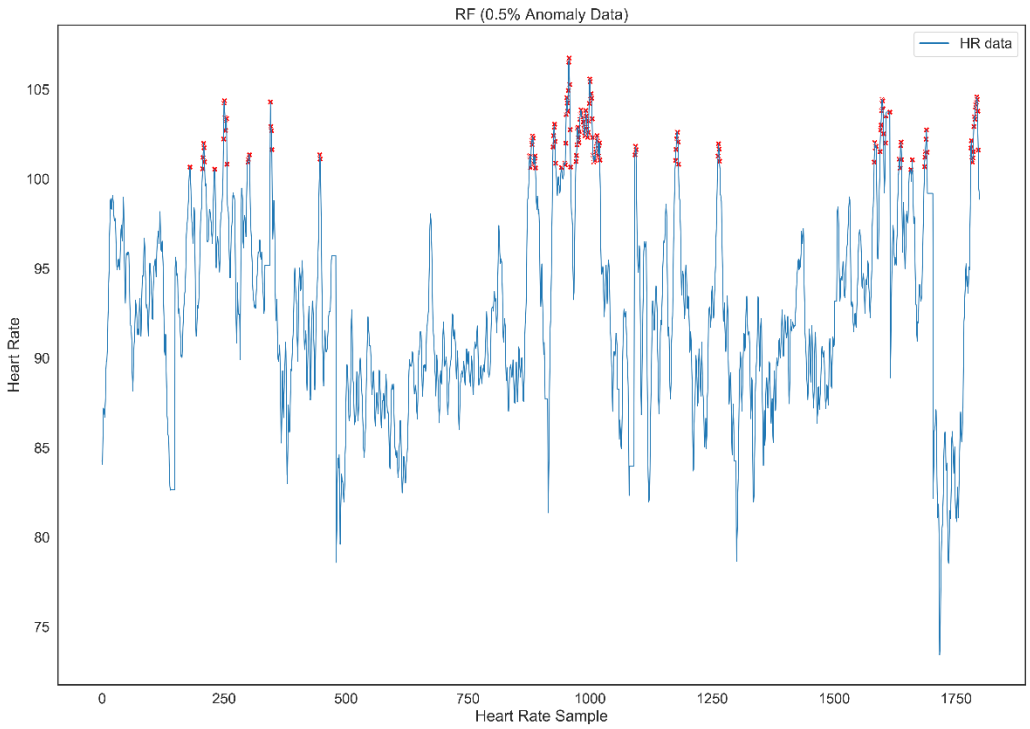

Fig. 3a. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 0.5% anomaly dataset.

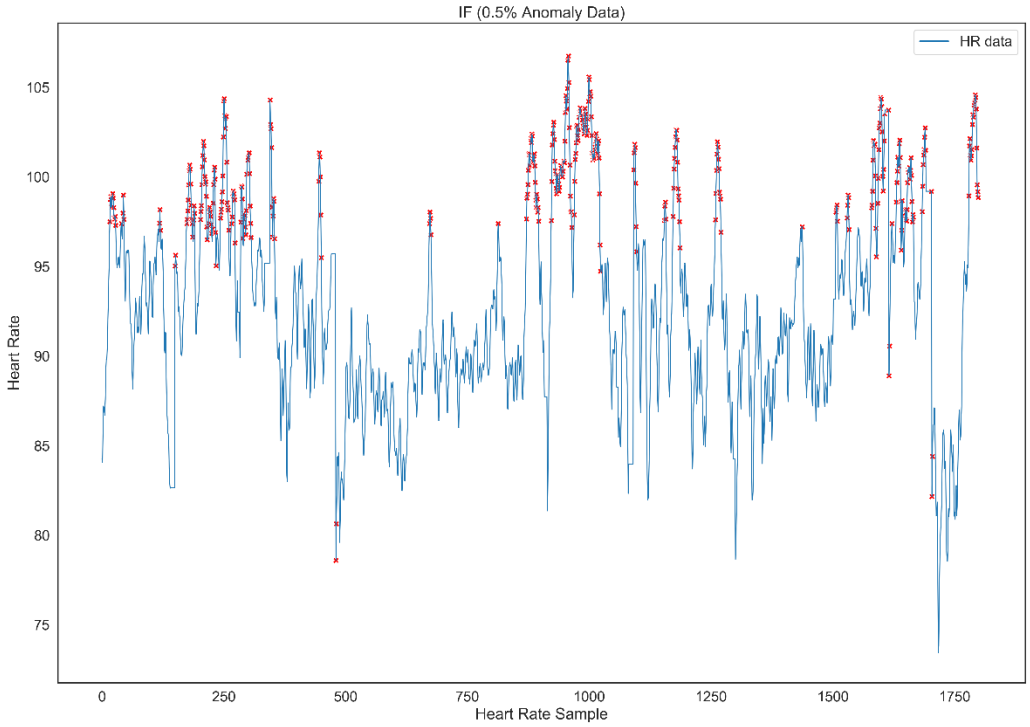

Fig. 3b. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 0.5% anomaly dataset.

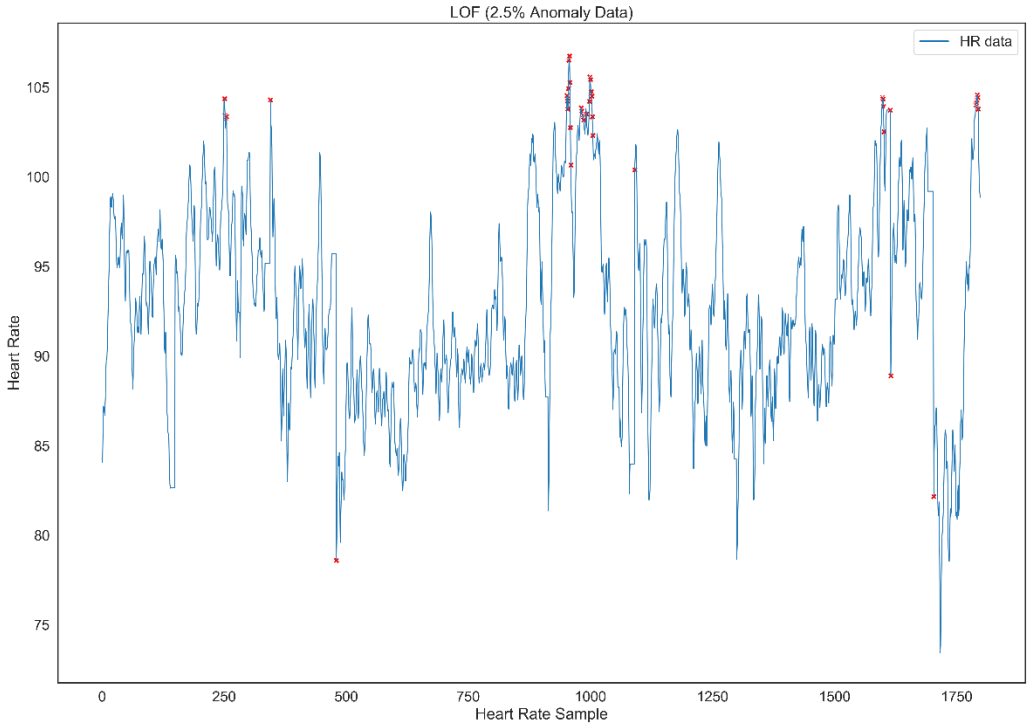

Fig. 4. Detected anomalies (each shown as a red x) for the local outlier factor model trained on the 2.5% anomaly dataset.

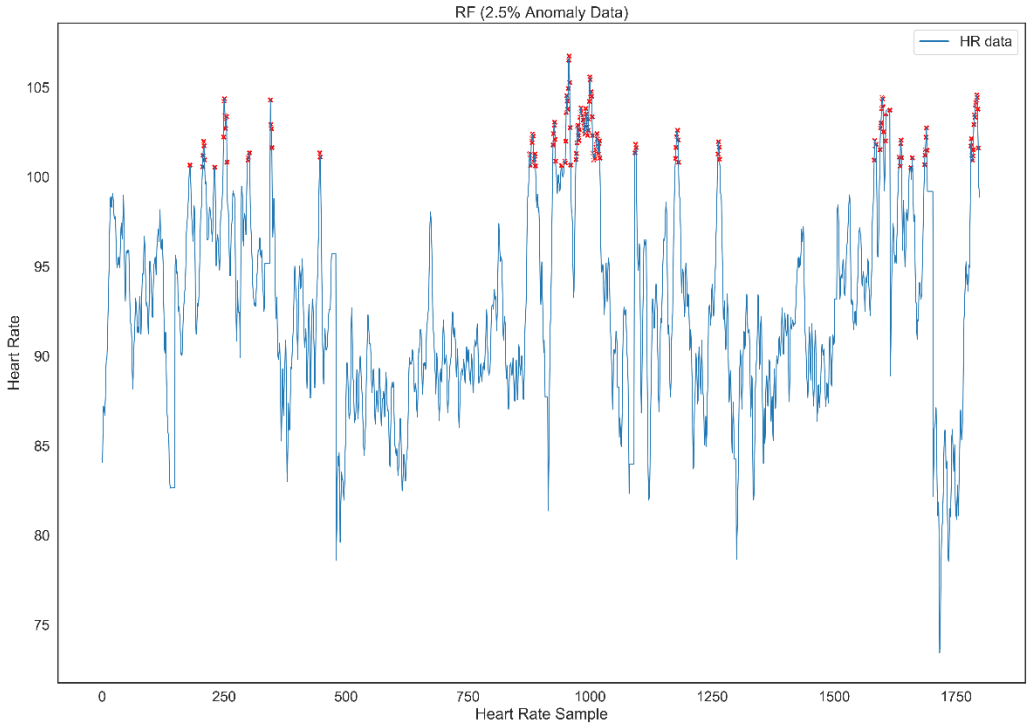

Fig. 5a. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 2.5% anomaly dataset.

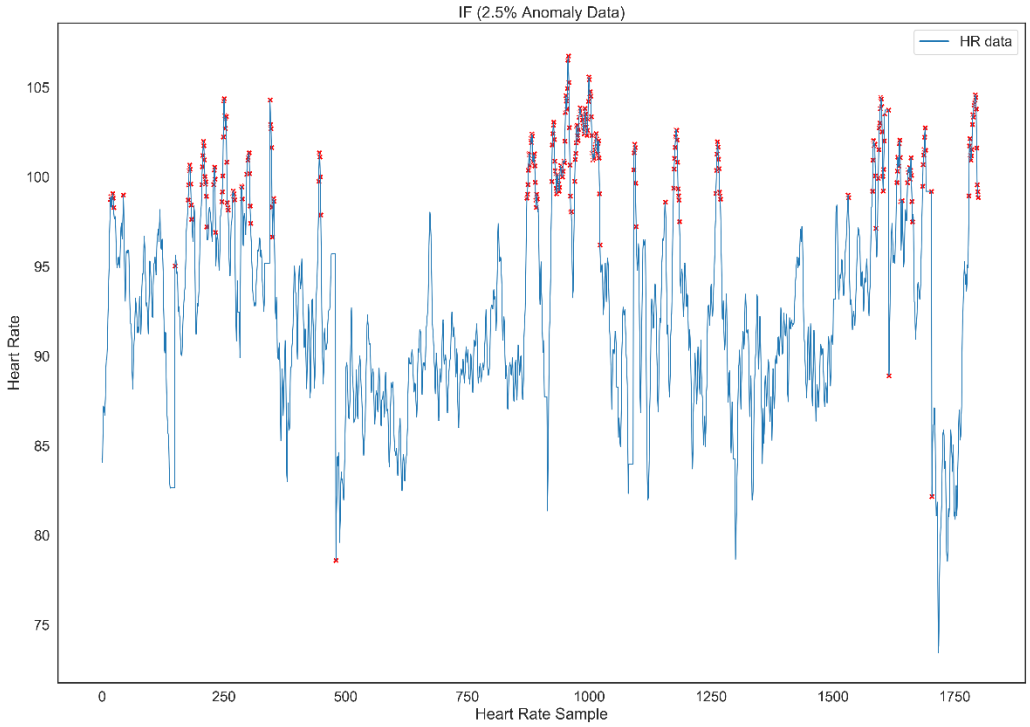

Fig. 5b. Detected anomalies (each shown as a red x) for the random and isolation forests models trained on the 2.5% anomaly dataset.