2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA) | 978-1-6654-0345-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/SAUPEC/RobMech/PRASA52254.2021.9377017

Anomaly Detection Monitoring System for Healthcare

Tlou Boloka, Gerrie Crafford, Windy Mokuwe, and Beatrice Van Eden
Industrial Robotics
Council for Scientific and Industrial Research
Pretoria, South Africa
TBoloka, GCrafford, MMokuwe, BvEden {@csir.co.za}

***Abstract*—Most developing countries suffer from inadequate health care facilities and a lack of medical practitioners as most of them emigrate to developed countries. The outbreak of the COVID-19 pandemic has left these countries more vulnerable to facing the worse outcome of the pandemic. This necessitates the need for a system that continuously monitors patient status and detects how their physiological variables will change over time. As a result, it will reduce the rate of mortality and mitigate the need for medical practitioners to monitor patients continuously. In this work, we show how an autoencoder and extreme gradient boosting can be merged to forecast physiological variables of a patient and detect anomalies and their level of divergence. An accurate detection of current and future anomalies will enable remedial action to be taken by medical practitioners at the right time and possibly save lives.**

I. Introduction

Critical patients admitted to hospitals are typically connected to systems that provide continuous monitoring of multiple physiological variables. In most developing countries, constant monitoring is used by medical practitioners to keep track of patient condition deteriorating. Early detection of patient condition deterioration will enable remedial action to be taken by medical practitioners at the right time which will reduce the need for patients to be transferred to the higher acuity units, reduce their length of stay at the hospital, and improve their survival rates [1], [2].

The vast majority of hospitals in developing countries employ a traditional approach of bed-side monitoring and rule-based monitoring. In a bed-side monitoring approach, the medical practitioner observes the patient's physiological variable(s) to know the patient's status. This approach can be time-consuming and tedious. On the other hand, in a rule-based approaches, the normal range is set; any value outside the range is deemed abnormal otherwise healthy [3]. This approach has been proven to be not accurate and produces a large number of false positives leading to alarm fatigue. Furthermore, it does not capture the correlation of the variables [4].

Recently there has been an increasing body of work concerning data-driven approaches that combines machine learning and the Internet of Things (IoT) to enable autonomous continuous monitoring of physiological variables [5], [6], [7], [8], [9]. This growth is due to the recent advancement in the IoT technologies such as wireless communication and sensors [10].

Anomaly detection is a data-driven technique that serves as the basis of applications across a diverse variety of domains, such as fault detection, intrusion and fraud detection, and process control. The goal of anomaly detection is to identify patterns in data that do not conform to a well-defined notion of normal behavior [11]. In [12], they employ Gaussian Processes to estimate the future trajectory of a patient's vital signs. However, the Gaussian process is known to suffer from the curse of dimensionality, which makes their approach infeasible when using high dimensional features [13].

Work by [14] uses a single physiological variable state to perform anomaly detection. In contrast to the sequential state anomaly detection, single state anomaly detection methods are known to perform poorly. They perform poorly because because they do not take into consideration how the patient variables were changing over time, instead, it uses the current variable to detect if the patient is in a normal state or not [15]. A review of the literature reveals that data-driven approaches relying on supervised learning have demonstrated promising results in various applications [16], [17]. However, the supervised learning approach requires data from both normal and anomaly classes. This is a limitation of supervised methods because it is almost impossible to obtain every possible type of anomaly that could happen in the system.

In scenarios where labeled data are scarce or unavailable, unsupervised anomaly detection approaches are usually applied, because only normal data are required to train a detection model [18]. In this work, we propose a system that continuously monitors the patient's condition using physiological variables and predicts when the patient will require attention from the medical practitioners. The proposed system merges both the supervised and unsupervised approaches and uses normal data only.

978-1-6654-0345-0/21/$31.00 ©2021 IEEE

II. Anomaly Monitoring System for Healthcare Approach

Our methodology comprises of four steps, namely: pre-processing, anomaly detection, forecasting physiological variables, and using anomaly detection on the forecasted physiological variables.

A. Pre-processing

When physiological variables are recorded they may be null values due to sensors malfunctions. To deal with null values, we replaced them with a mean value of all patients at that time. However, for a patient with the total number of null values above 25 we deleted the entire record. To enable model robustness, zero mean Gaussian noise is added to the training data.

We then calculate the correlation between the physiological variables, and drop one of the variables if the correlation is above 8.5. For instance, heart rate and pulse rate are highly correlated, that mean their contribution to the learning process is the same.

B. Anomaly Detection

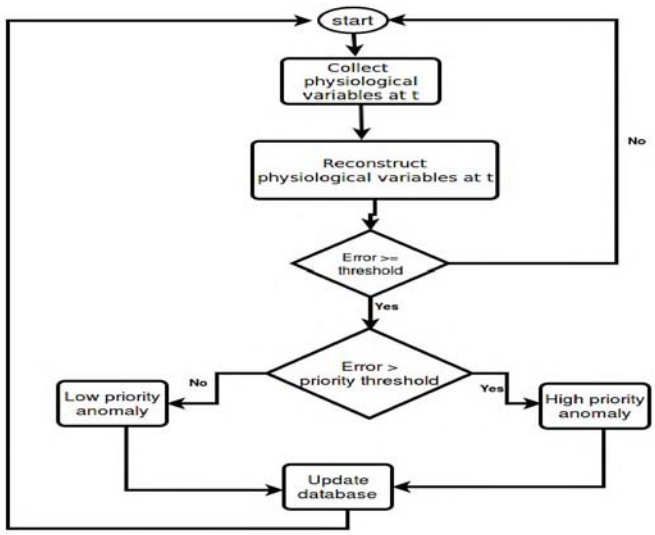

Fig. 1. Anomaly detection process.

For anomaly detection, we employ an autoencoder. Autoencoders learn a representation (encoding) for a set of data, typically for dimensionality reduction [19]. It consists of a reduction side (encoder) and a reconstructing side (decoder). Both the encoder and decoder are fully-connected feed-forward neural networks.

First, the input passes through the encoder, which produces the code and then goes through the decoder, which has a similar structure. The decoder is responsible for reconstructing the input using the code. The goal is to get an output identical to the input as shown in Equation 1

$$F_\theta(h_t, r_t) \approx h_t, r_t \tag{1}$$

where $F_\theta(h_t, r_t)$ represents the model. Autoencoders are considered an unsupervised learning technique since they don't need explicit labels during training. We use the reconstruction error (shown in Equation 2) to detect anomalies. A significant error in reconstruction is a sign of an anomaly. An anomaly detection threshold is used to separate anomalies from normal data points.

$$\underbrace{\{h_t, r_t\}}_{\textbf{Model Input}} - \underbrace{\{h_t, r_t\}}_{\textbf{Model Output}} \tag{2}$$

Figure 1 shows our anomaly detection process. Firstly, the system collects the current physiological variables of the patient, then we use the autoencoder to reconstruct the variables, if the difference (error) between the actual and reconstructed variables is greater than or equal to the preset anomaly detection threshold the variables are considered as anomalies otherwise normal. When the variables are detected as an anomaly we further assess if the error is greater than the preset priority threshold. If that is the case, the anomaly is considered high priority else it is considered low priority anomaly. Anomaly detection means that the patient requires medical attention from the medical practitioners. A low priority anomaly means the patient condition is slightly different from normal. A high priority anomaly means the patient condition is significantly distinct from normal. Then the variables values and their predictions are stored in the database for future maintenance.

C. Forecasting Physiological Variables

For forecasting physiological variables, we employ XGBoost supervised learning approach. XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework [20]. Artificial neural networks are considered best when using unstructured data (i.e., images or text). However when it comes to structured data, decision tree based algorithms are known to be best performers. Hence, we selected the XGBoost algorithm. The model take in physiological variables at time t_i as input and output physiological variables at time t_{i+1} as shown in Equation 3.

$$F_\theta\{(h_t, r_t), \ldots, (h_n, r_n\} \approx \{h_{n+1}, r_{n+1}\} \tag{3}$$

D. Using Anomaly Detection on the Forecasted physiological Variables

In Figure 2 we show how anomaly detection is used on the forecasted physiological variables. The process is similar to the anomaly detection process except that in this process, an autoencoder is applied on the forecasted variables instead of the actual values.

III. Experiments and Results

To assess the performance of our proposed system, we utilize physiological parameter data from the Multiple Intelligent Monitoring in Intensive Care (MIMIC) database [21]. It contains thousands of recordings of multiple physiologic signals ("waveforms") and time series of

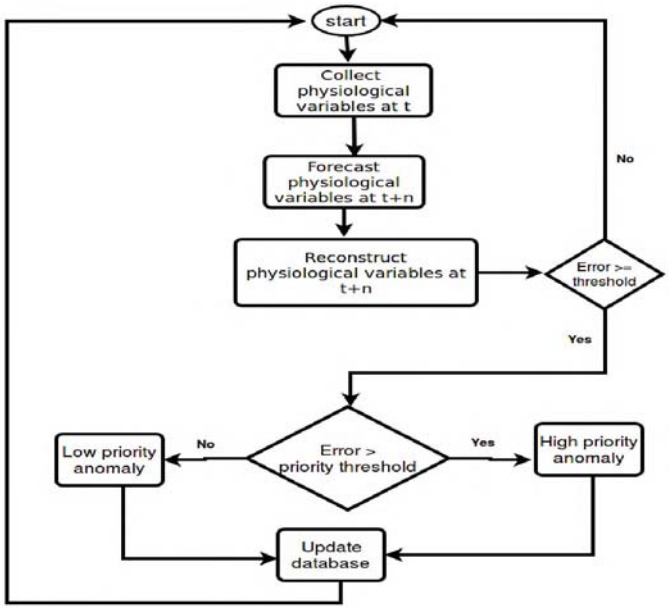

Fig. 2. Forecasting physiological variables and anomaly detection process.

physiological variables ("numerics") collected from bedside patient monitors in adult and neonatal intensive care units. This data is not labeled.

In this work, we focus on the numerical data for two physiological variables: namely, heart rate (HR) and Respiration rate (RESP). Our experiments are divided into three parts, namely: anomaly detection, forecasting physiological variables, and the combination of forecasting and anomaly detection.

A. Anomaly Detection

We start by exploring the training data, Table I shows the number of data points per batch; we observe that in HR, most data points lie between 41 and 180. While in RESP, most data points lie between 0 and 80. We also note that there are fewer data points above 120 in RESP. The total number of data points we used for training were 58960 while for testing we used 14740 data points.

Batch	HR	RESP
0-20	0	27017
21-40	20	23043
41-60	3491	6224
61-80	16021	2103
81-100	16652	492
101-120	8464	75
121-140	4764	6
141-160	5706	0
161-180	2958	0
181-200	727	0
≥201	157	0

TABLE I
TRAINING DATA IN BATCHES.

In this subsection, we evaluate the effectiveness of anomaly detection. In Figure 3 we show the autoencoder mean square error (MSE) during learning, we observe how the model improves with an increase in the number of epochs. The model converges after 40 epochs.

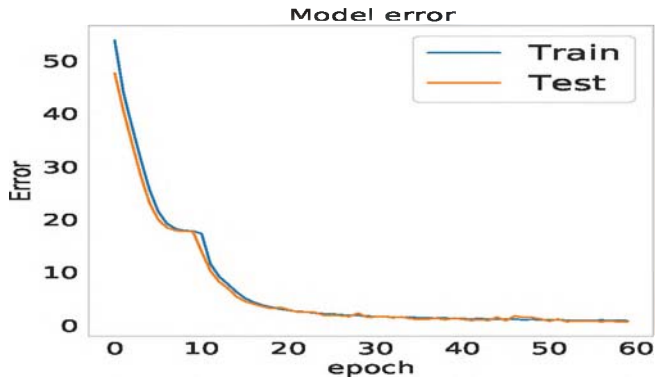

Fig. 3. Autoencoder learning MSE.

To select an anomaly detection threshold, we used the training data of 58960 samples to evaluate how their error values are distributed. In Figure 4 we show the results; we observe that most error values are between 0 and 0.5. From this experiment, we selected 1 as an anomaly detection threshold, which means when the reconstruction error of the model is greater or equal to 1, the data point is detected as an anomaly; otherwise, they are detected as normal.

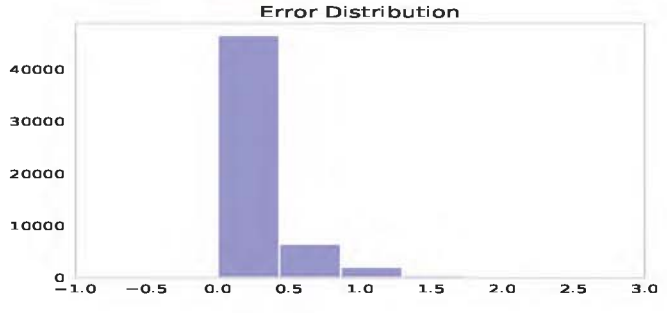

Fig. 4. Training data error distribution after the model was fully trained.

We then evaluated how the model performs with an anomaly detection threshold of 1 using the testing data. We show a comparison of the model reconstruction (detected) values with the ground truth values. For visualization simplicity, we chose to visualize 100 samples detected as anomalies from the testing data. Figure 5 shows the heart rate reconstructed (detected) compared to the ground truth values. In Figure 6, we show similar results for the respiration rate. We observe that in most cases, the model reconstructed values do not match the ground truth values. Hence the model detected the data points as anomalies.

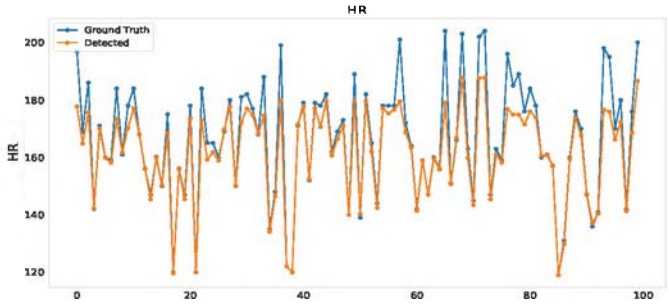

Fig. 5. Comparing the ground truth and the anomaly detected HR values.

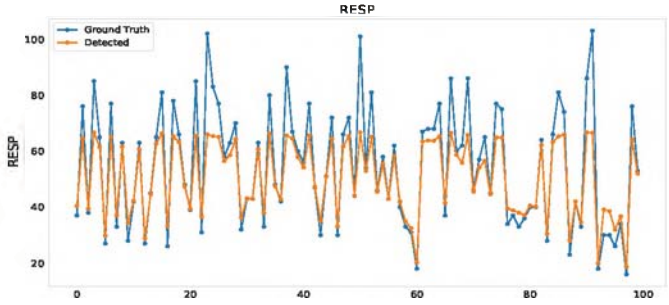

Fig. 6. Comparing the ground truth and the anomaly detected RESP values.

Figure 7 shows the error of the above data points; we observe that the error of all the data points is high or equal to the anomaly detection threshold (red horizontal line).

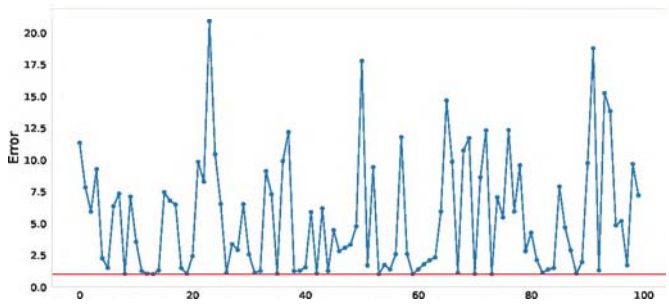

Fig. 7. Anomaly detected values error.

We repeated the experiments, but this time we used the data points detected as normal. We observe that the model was able to reconstruct the values similar to the ground truth as shown in Figures 8 and 9. Furthermore, we observe that the ground truth points are not visible as most of them are under the reconstructed (detected) values.

We then show the reconstruction error of the data points detected as normal in Figure 10. We observe that the error of all the data points is below the anomaly detection threshold.

In Table II we show the number of data points detected as anomalies or normal when using different threshold on the testing data. We observe that when the threshold is at 1, the model detects 1143 data points as anomalies and 13597 as

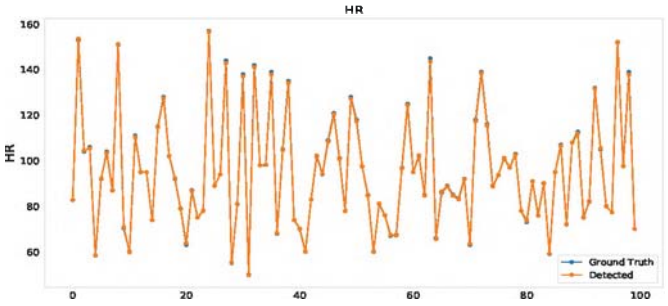

Fig. 8. Comparing the ground truth and the normal detected HR values

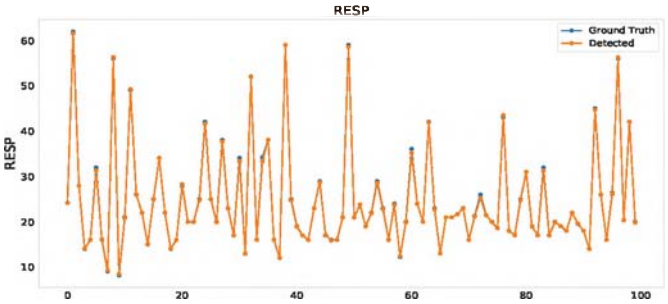

Fig. 9. Comparing the ground truth and the normal detected RESP values.

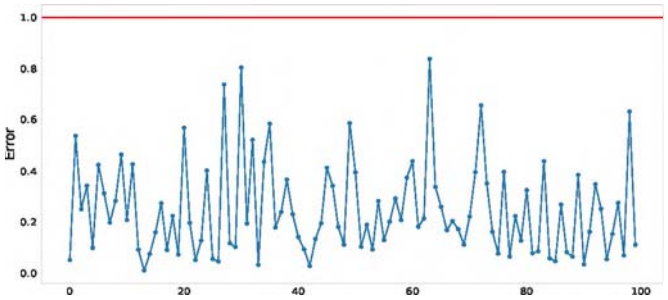

Fig. 10. Normal detected values error.

normal data points. While when we choose threshold of 1.5, the model detect 897 as anomalies and 13843. The lower the threshold, the more points will be detected as anomalies.

Anomaly detection threshold	Anomalies	Normal
1	1143	13597
1.5	897	13843

TABLE II
NUMBER OF DATA POINTS DETECTED AS AN ANOMALY OR NORMAL USING DIFFERENT ANOMALY DETECTION THRESHOLD.

Figure 11 and 12 show data points detected as anomalies and others as normal using anomaly detection thresholds of 1 and 1.5 on testing data, respectively. From the two figures, we observe that there is a clear separation between anomalies and normal data points. The model flags data points that are

not similar to the data points it has seen during training as anomalies. Furthermore, we observe that most normal data points are in regions where most training data resides as shown in Table I.

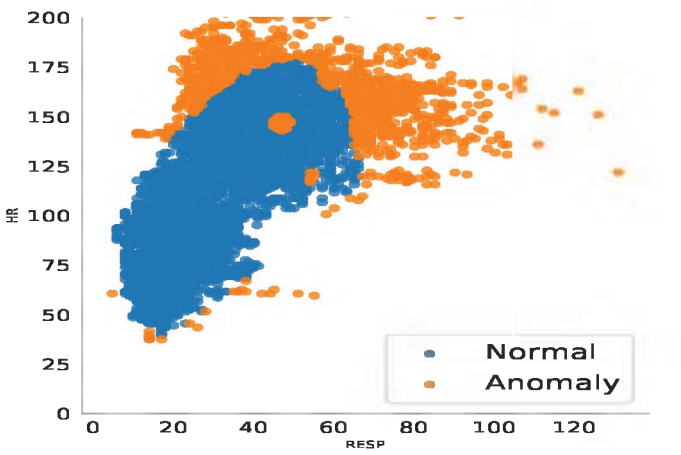

Fig. 11. Values detected as normal or anomalies using 1 threshold.

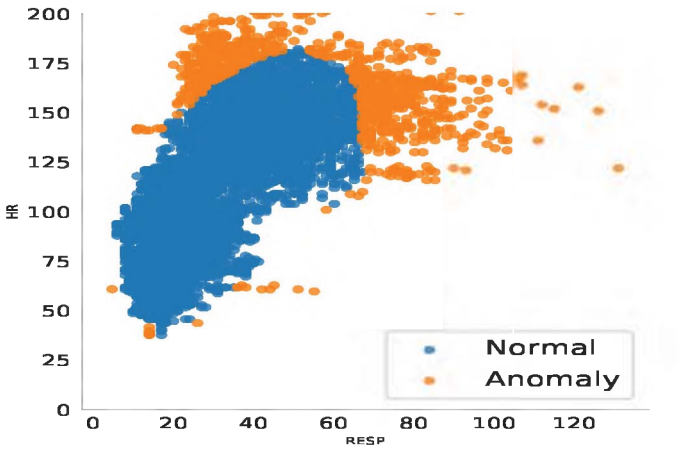

Fig. 12. Values detected as normal or anomalies using 1.5 threshold.

In Figure 13 we visualise high and low priority anomalies. Data points with error between 1 (red horizontal line) and 4 (green horizontal line) are considered low priority anomalies, while those with error above 4 are considered high priority anomalies. This approach will help medical practitioners to understand the level of seriousness of the patient's condition. Furthermore, this approach can help with detecting malfunctions of data gathering sensors. Intuitively, we expect malfunctioning sensors to produce bizarre data points.

B. Forecasting Physiological Variables

In this subsection, we evaluate the model's performance concerning forecasting physiological variables (HR and RESP). The models takes in n previous physiological

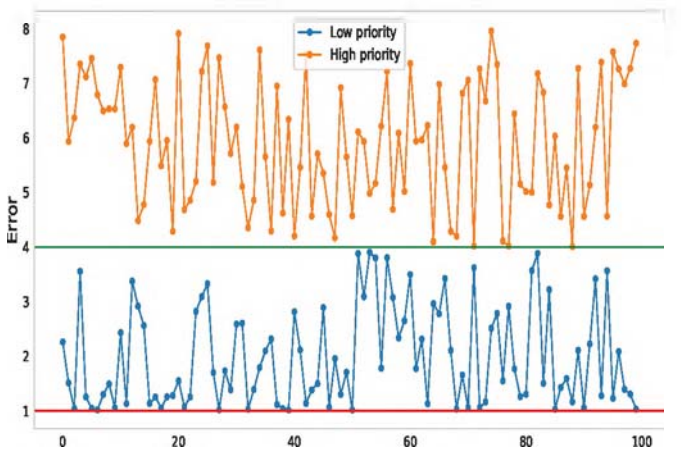

Fig. 13. High and low priority anomalies.

variables at different time t_i as input and output physiological variables at time t_{i+1} as shown in Equation 3. We use the root mean squared error (RMSE) to assess this attribute. We compare random forent (RF) [22], k-nearest neighbor (KNN) [23], XGBOOST [20] and feed-forward long short-term memory (LSTM) [24]. To enable a fair comparison amongst the models, we used the same number of training (17688) and testing (4421) data points.

In Table III we show RMSE for each model when using different input (sample) size we observe that RF and XGBoost perform better than KNN and LSTM. KNN and LSTM perform badly with an increase in sample size, while on the other hand, XGBOOST and RF are shown to be less affected by an increase in the sample size.

Sample size (n)	KNN	RF	XGBOOST	LSTM
1	3.4	3.8	4.5	3.0
5	3.5	3.1	2.9	2.9
10	3.8	3.1	2.9	2.9
30	4.6	3.1	2.9	3.0
50	5.2	3.1	2.9	3.5

TABLE III
FORECASTING PHYSIOLOGICAL VARIABLES RMSE USING DIFFERENT SAMPLE SIZE.

C. Forecasting and anomaly detection

In this subsection we evaluate the integration of the forecasting and anomaly detection models. We have selected XGBoost with the sample size of 5 for forecasting as it has been demonstrated to perform best with the lowest RMSE of 2.9 (as shown in Table III). To evaluate this attribute, we used 4421 testing data points.

Figure 14 and 15 show the forecasted values detected as normal, low anomaly, and high anomaly. The figures demonstrate a clear separation amongst normal, low anomaly, and high

anomaly data points. High anomaly are further away from normal. The separation is similar to the separation we have shown in Figure 11 and 12.

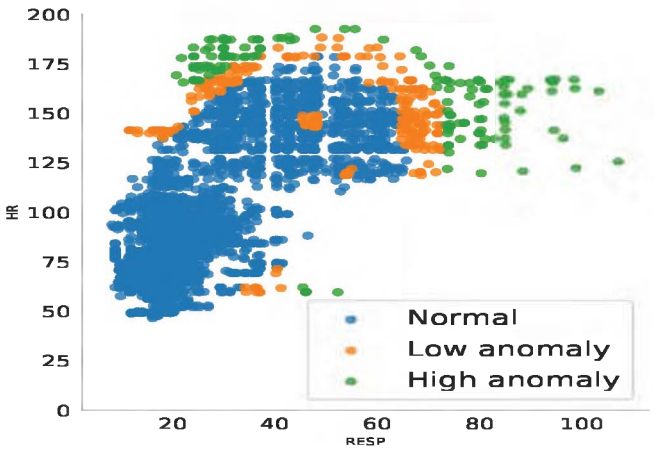

Fig. 14. Forecasted values detected as normal or low/high anomalies using 1 threshold.

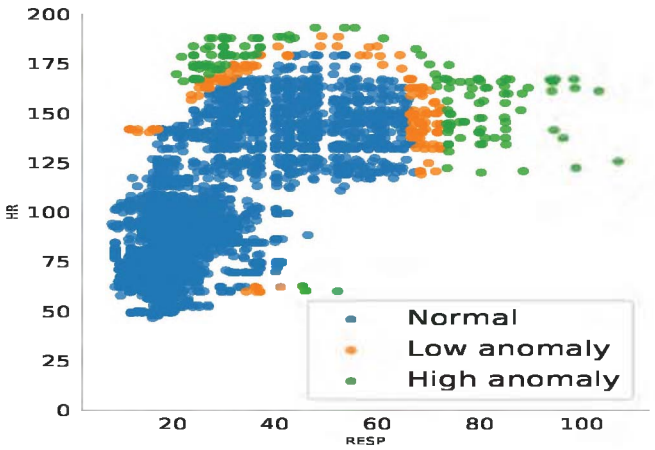

Fig. 15. Forecasted values detected as normal or low/high anomalies using 1.5 threshold.

IV. Conclusion

This paper has shown how an autoencoder and XGBoost can be combined to forecast physiological variable of a patient and detect anomalies with their level of divergence. Furthermore, we have shown how anomalies can be detected from unlabelled data. Merging anomaly detection and forecasting approaches can be vital in reducing the mortality rate and mitigating the tedious constant monitoring of the patients in hospitals done by our medical practitioners.

References

[1] H. Brown, J. Terrence, P. Vasquez, D. W. Bates, and E. Zimlichman, "Continuous monitoring in an inpatient medical-surgical unit: a controlled clinical trial," *The American journal of medicine*, vol. 127, no. 3, pp. 226–232, 2014.

[2] C. P. Subbe, B. Duller, and R. Bellomo, "Effect of an automated notification system for deteriorating ward patients on clinical outcomes," *Critical Care*, vol. 21, no. 1, p. 52, 2017.

[3] B. H. Cuthbertson and G. Smith, "A warning on early-warning scores!" 2007.

[4] C. P. Subbe, M. Kruger, P. Rutherford, and L. Gemmel, "Validation of a modified early warning score in medical admissions," *Qjm*, vol. 94, no. 10, pp. 521–526, 2001.

[5] A. Muaremi, J. Seiter, G. Tröster, and A. Bexheti, "Monitor and understand pilgrims: Data collection using smartphones and wearable devices," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 2013, pp. 679–688.

[6] D. Teichmann, A. Kuhn, S. Leonhardt, and M. Walter, "The main shirt: A textile-integrated magnetic induction sensor array," *Sensors*, vol. 14, no. 1, pp. 1039–1056, 2014.

[7] F. Zhang, J. Cao, S. U. Khan, K. Li, and K. Hwang, "A task-level adaptive mapreduce framework for real-time streaming data in healthcare applications," *Future generation computer systems*, vol. 43, pp. 149–160, 2015.

[8] E. Årsand, M. Muzny, M. Bradway, J. Muzik, and G. Hartvigsen, "Performance of the first combined smartwatch and smartphone diabetes diary application study," *Journal of diabetes science and technology*, vol. 9, no. 3, pp. 556–563, 2015.

[9] S. Weyer, F. Weishaupt, C. Kleeberg, S. Leonhardt, and D. Teichmann, "Rheostim: Development of an adaptive multi-sensor to prevent venous stasis," *Sensors*, vol. 16, no. 4, p. 428, 2016.

[10] D. Gil, A. Ferrández, H. Mora-Mora, and J. Peral, "Internet of things: A review of surveys based on context aware intelligent services," *Sensors*, vol. 16, no. 7, p. 1069, 2016.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey. acm computing surveys (csur), 41 (3)," 2009.

[12] G. W. Colopy, M. A. Pimentel, S. J. Roberts, and D. A. Clifton, "Bayesian optimisation of gaussian processes for identifying the deteriorating patient," in *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2017, pp. 85–88.

[13] R. Bellman, "Dynamic programming and lagrange multipliers," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 42, no. 10, p. 767, 1956.

[14] B. Gyunka and S. Barda, "Anomaly detection of android malware using one-class k-nearest neighbours (oc-knn)," *Nigerian Journal of Technology*, vol. 39, no. 2, pp. 542–552, 2020.

[15] M. Mozaffari and Y. Yilmaz, "Online multivariate anomaly detection and localization for high-dimensional settings," *arXiv preprint arXiv:1905.07107*, 2019.

[16] B. Jin, D. Li, S. Srinivasan, S.-K. Ng, K. Poolla, and A. Sangiovanni-Vincentelli, "Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks," in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019, pp. 1–8.

[17] D. Li, G. Hu, and C. J. Spanos, "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis," *Energy and Buildings*, vol. 128, pp. 519–529, 2016.

[18] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[19] D. H. Ballard, "Modular learning in neural networks." in *AAAI*, 1987, pp. 279–284.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[21] A. Johnson, T. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, and R. Mark, "Data descriptor: Mimic-iii, a freely accessible critical care database sci," 2016.

[22] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[23] L. Devroye, L. Gyorfi, A. Krzyzak, and G. Lugosi, "On the strong universal consistency of nearest neighbor regression function estimates," *The Annals of Statistics*, pp. 1371–1385, 1994.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.