Heart Rate Anomaly Detection Using Contractive Autoencoder for Smartwatch-Based Health Monitoring

Keerthana Mannarapparambil Sivan[1], Shanfeng Hu[1], Nauman Aslam[1], Xiaomin Chen[1], Pradorn Sureephong[2], Suwit Wongsila[2], Sabeela Q. Ahmed[3]

[1]*Department of Computer and Information Sciences, Northumbria University, United Kingdom*

[2]*College of Arts, Media and Technology, Chiang Mai University, Thailand*

[3]*NHS Health Trust*

keerthanamss14@gmail.com, (shanfeng2.hu, nauman.aslam, xiaomin.chen)@northumbria.ac.uk, dorn@camt.info, suwit@kirly.org, sabeela.ahmed2@nhs.net

Abstract— **The widespread adoption of wearable devices enables continuous monitoring of physiological parameters like heart rate, offering valuable insights into health. However, consumer-grade wearable data exhibits real-world noise, variations, and discontinuities across diverse populations, posing significant challenges for anomaly detection models. This paper proposes a novel deep learning approach to address these challenges, utilizing a Contractive Autoencoder (CAE) model optimized and applied specifically to noisy temporal heart rate data from wearable devices. By incorporating a contractive regularization penalty in the loss function, the model learns more robust and stable representations of the irregular data with high accuracy. Comprehensive experiments on a real-world Fitbit dataset demonstrate the proposed CAE model accurately identifies anomalous heart rate patterns missed by traditional thresholding techniques. The research encountered key challenges in ensuring model generalizability across diverse populations with natural heart rate variations, handling missing and sparse data from unreliable real-world wearable devices, and obtaining properly labelled anomaly data for robust training. Although the current model achieved promising anomaly detection results, further extensive validation on diverse datasets is essential to fully assess its capabilities across expanded demographics and use cases. Overall, this research provides an important foundation for optimizing deep learning approaches on noisy real-world wearable data through rigorous evaluation.**

Keywords— contractive autoencoder, contractive loss, anomaly detection, heartrate anomaly, outlier detection

I. Introduction

Anomaly detection in heart rate data is a crucial aspect of modern healthcare, aiming to identify and flag any abnormal or irregular heart rate patterns that may indicate potential health problems. With the widespread adoption of wearable devices, such as smartwatches and fitness trackers, continuous heart rate monitoring has become accessible and convenient for individuals of all ages and lifestyles, providing valuable insights into cardiovascular health conditions. These wearable devices are equipped with advanced sensors that capture heart rate data continuously, enabling the tracking of heart rate variations and trends over time [1]. The heart rate data collected by wearable devices encompasses a rich and dynamic time series, often referred to as continuous or time series data. Unlike traditional sporadic measurements, continuous heart rate readings are taken at regular intervals, spanning from a few seconds to a few minutes, hours or days. This continuous data stream offers a comprehensive and granular view of an individual's heart, capturing fluctuations, patterns, and rhythms that might otherwise go unnoticed with sporadic measurements [2]. By continuously monitoring heart rate, healthcare professionals and individuals can gain insights into various heart rate parameters and assess how they evolve over different time frames.

Researchers and medical practitioners have been increasingly leveraging machine learning and data-driven techniques to develop robust anomaly detection methods for heart rate data collected from wearable devices. However, existing techniques face challenges in handling noise, sparse sampling rates, and inability to generalize across diverse populations exhibiting natural variations in normal heart rate ranges. This presents a research gap for developing robust anomaly detection models that can learn effective data representations despite real-world noise and inter-personal variations.

To address these limitations, we present a novel CAE model optimized specifically for consumer-grade wearable heart rate data. Unlike existing work exploring raw PPG or ECG signals, our approach focuses on quantified beats per minute (bpm) heart rate readily available from consumer wearables. This CAE goes beyond the standard autoencoders that solely minimize reconstruction error by incorporating a contractive regularization penalty that encourages learning invariant representations robust to noise and minor perturbations. Experiments on real-world Fitbit data show over 90% accuracy in classifying normal and anomalous patterns, highlighting capabilities in handling noise compared to existing techniques. By optimizing for wearable data, we aim to unlock consumer devices' potential for preventive monitoring through early anomaly detection, with implications for real-time exercise adjustment to mitigate health risks.

By integrating the CAE model into wearable devices, we aspire to contribute to the advancement of digital health in the region and empower elderly individuals to take charge of their mobility disorder prevention.

The key contributions of this research are as follows,

1. Proposing a novel CAE model for accurate and robust anomaly detection in heart rate data.
2. Demonstrating the superiority of CAE model over other ML approaches on a real-world wearable data.

3. Enabling early irregular heart rate detection with wearables for preventive health.

II. Related work

Several studies have explored different machine learning techniques to detect anomalies in heartrate data and were able to accurately detect anomalies. Several commercial wearable devices, such as Fitbit and Apple Watch, have also implemented algorithms for anomaly detection in heart rate data [3]. For instance, Alugubelli et al. [4] discussed wearables' potential for remote heart rate and HRV monitoring, including device accuracy, limitations and future directions. In a study by Liu et al. [5], a convolutional autoencoder was employed to estimate COVID-19 symptoms and anomalies. Furthermore, Abir et al. [6] proposed a deep learning framework with CNN, VAE, and LSTM for COVID-19 detection from smartwatch data. It detected COVID-19 in 74% of subjects, showing potential as a supplementary screening tool using heart rate and step count from consumer wearables.

Beyond heart rate anomaly detection, machine learning models have been applied to predict chronic obstructive pulmonary disease based on physiological time series patterns [7]. Additionally, researchers explored using Fitbit-assessed behaviour as a predictor for readmission of postsurgical cancer inpatients, building a predictive machine learning model with Fitbit activity data [8].

In the domain of CAE, researchers have leveraged gradient-based activation penalties and sparse activations to reflect data's intrinsic properties [9]. CAEs have found application in diverse tasks, including cloud Intrusion Detection [10], document clustering [11], recognition of pilots' Fatigue Status [12], Online spike sorting [13], and more. For ECG denoising, Banerjee et al. [14] proposed a convolutional sparse contractive autoencoder incorporating sparsity, contractive regularization, and L2 norm.

Some prior studies developing anomaly detection models for wearable devices have utilized ECG data for training and evaluation. ECG provides detailed waveform data useful for research purposes. For example, Zhong et al. [16] proposed an unsupervised approach using convolutional autoencoders and Gaussian mixture models to estimate beat-to-beat heart rate from ECG data from wearable sensors. However, consumer smartwatches mostly rely on photoplethysmography (PPG) sensors which measure blood volume changes to estimate heart rate. The PPG sensors output periodic heartbeat waveforms, which can be processed to derive a quantified beats-per-minute (bpm) value. Prior studies like Gu et al. [15] developed lightweight convolutional neural network to detect anomalies directly from the raw PPG waveform data.

While studies have used ECG and PPG data, for many commercial wearables only the quantified bpm heart rate is available to users, not the raw PPG waveform. Our work focuses specifically on analysing the bpm heart rate data readily available from consumer wearables. This quantified bpm data provides a direct measurement of heart rate in beats per minute. We propose a CAE model designed to work with the bpm heart rate time series data for effective anomaly detection, unlike unsupervised techniques in some prior work.

Existing anomaly detection techniques using deep learning, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have faced challenges related to expert feature engineering, handling high-dimensional data, and interpretability issues [17]. Anomaly detection remains an active research area for healthcare applications. This work proposes a novel CAE architecture for supervised anomaly detection in heart rate data from wearables. The approach provides robust anomaly detection, advancing digital health monitoring.

III. Methodology

A. Dataset and preprocessing

This research utilizes a public dataset of heart rate time series data collected from Fitbit users [23]. The data is structured with each row representing a heart rate measurement at a specific timestamp.

In total, the dataset contains 1,048,576 heart rate measurements across the 14 users. The time intervals between measurements range from 1 second to over 1 hour. This variable sampling rate is common in real-world wearable device datasets.

Initial data analysis surfaced several key observations - the dataset exhibited imbalance with uneven sample distribution per user, discontinuities from missing observations, and varying data lengths among users.

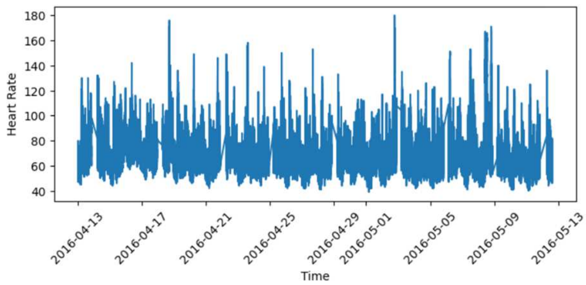

Fig. 1. The temporal heartrate data for a specific individual over a specific period, ranging from 2016-04-12 to 2016-05-12, in its raw, unprocessed form.

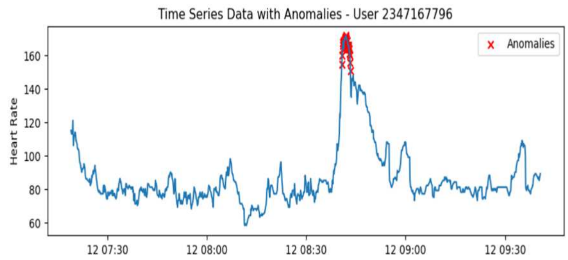

Fig. 2. An example of an identified irregular or unusual variation in the heart rate of an individual, indicating a potential anomaly.

To better understand the data, we visualized heart rate plotted against time (Fig. 1). This plot would display a fluid line reflecting fluctuations influenced by physiological factors. Patterns like gradual shifts, oscillations, and spikes/drops can be revealed. An anomaly could be represented by a sudden spike or drop-in heart rate that stands out from surrounding points (Fig. 2). This irregularity often indicates a potential health issue or error.

To ensure the quality and suitability of the data for the machine learning algorithm, a pre-processing step was carried out. This involved several key actions, including noise removal, handling missing values, data normalization, and converting the data into a suitable format. By performing these necessary adjustments, the data was prepared optimally for accurate and effective analysis using the machine learning algorithm.

The raw data was initially filtered to include the complete data records from just five users spanning the entire time duration. The data was then resampled to 10 second intervals for each user, with missing values imputed via forward fill. To smooth out noise, a rolling 30-sample average was applied to the heart rate readings for each user. The sequences were then sliced to contain only complete intervals of 5 minutes. This process yields a substantial volume of data with a consistent length, ensuring adequacy for training, validation, and testing purposes. Thresholds of 60 to 140 bpm were set to identify anomaly labels - any heart rates outside this range were labelled as 'Anomaly' while others were labelled as 'Normal'. The dataset was split into training, validation and testing sets for modelling.

Training an autoencoder using normal data is essential to establish a baseline understanding of regular patterns and features within the dataset. This process allows the autoencoder to learn the inherent structures and representations of normal data, which in turn enables it to distinguish anomalies or deviations when presented with unfamiliar or anomalous instances. Therefore, the dataset has been partitioned in a manner where 80% of the normal data is allocated for training purposes. The remaining data has been combined and then divided into two subsets: validation and test data.

B. Autoencoder

An autoencoder is a type of artificial neural network architecture used for learning efficient data representations in an unsupervised manner [20]. It contains an encoder that maps input data to a latent representation, and a decoder that reconstructs the input. Autoencoders can be used for various applications, including dimensionality reduction, feature learning, noise reduction, and anomaly detection [20-22].

The encoding process transforms the input x into a hidden representation y through the encoding function f, and can be mathematically represented as,

$$y = f(x) = \phi(W_x + b_h) \quad (1)$$

Here, the hidden layer is denoted as h. W_x are the weights applied to the input, b_h is the bias term, and φ is the activation function.

The decoding process then maps the hidden representation y back to a reconstructed input r through the decoding function g, the decoding operation can be represented as,

$$r = g(y) = \phi^0(W_y + b_r) \quad (2)$$

Here, W_y are the weights applied to y, b_r is the decoding bias, and ϕ^0 is the activation function. In summary, the encoder f transforms the input to a hidden representation y, which the decoder g then uses to reconstruct the input as r.

The autoencoder training minimizes the reconstruction error between input and output. The reconstruction error, also called the reconstruction loss, measures how well the autoencoder can reproduce the original input after it has been encoded and decoded [22]. If we have a dataset of inputs $D_i = [x_1, x_2, x_3, \ldots . x_n]$, then the cost function with reconstruction error R can be expressed as,

$$J_{AE}(\theta) = \sum_{x \epsilon D_i} R\,(x, r) \quad (3)$$

The reconstruction error $R(x, r)$ is the mean squared error of the input x and output y and can be represented as,

$$R(x, r) = \| \ x - r \ \|^2 \quad (4)$$

C. Contractive Autoencoder

The CAE introduces a novel explicit regularization term into the traditional autoencoder cost function. This contractive penalty sets the CAE apart from standard autoencoders that solely minimize reconstruction error between the input and reconstructed output. The contractive term provides a unique form of regularization that specifically promotes robustness and stability in the learned feature representations [22].

By penalizing the Frobenius norm of the Jacobian matrix of encoder activations, the CAE cost function uniquely penalizes the model's sensitivity to minor perturbations in the input data. This encourages the model to discover encodings that are invariant and unaffected by small changes or noise in the inputs. In effect, the contractive regularization enables the model to focus on robust features that represent the underlying causal factors rather than superficial noise patterns.

Unlike common regularization techniques like early stopping and dropout, the contractive penalty directly builds invariance and robustness into the optimization process through the cost function. This novel regularization approach improves generalizability and stability compared to basic autoencoders trained only to minimize reconstruction error $R(x, r)$.

The total CAE cost function can be represented as,

$$J_{CAE}(\theta) = \sum_{x \epsilon D_i} (R\,(x, r) + \ \lambda || J_f(x) ||_f^2) \quad (5)$$

Here, λ controls the weighting of the contractive term and $J_f(x)$ is the Jacobian matrix of hidden layer activations with respect to x.

D. Proposed Method

The input data is a 30 time step sequence. So the encoder input dimension per sample is 30. This is passed through stacked dense layers that compress the data into a reduced encoding. The decoder then reconstructs back to the original 30 dimension.

Encoder: The encoder consists of three dense layers that decrease in size to compress the data (Fig. 3). Each layer uses the ReLU activation for nonlinearity, avoiding vanishing gradients. ReLU enables better generalization and training speed compared to other activations. L2 regularization on the first layer encourages robust encodings. A dropout layer follows for regularization to mitigate overfitting. The encoder output is 30 neurons, capturing a compressed encoding of the data. This reduced encoding dimension controls model complexity.

Decoder: The decoder mirrors the encoder architecture in reverse order to reconstruct the input data. It has three dense layers with 60, 120, and 240 neurons, using the same ReLU

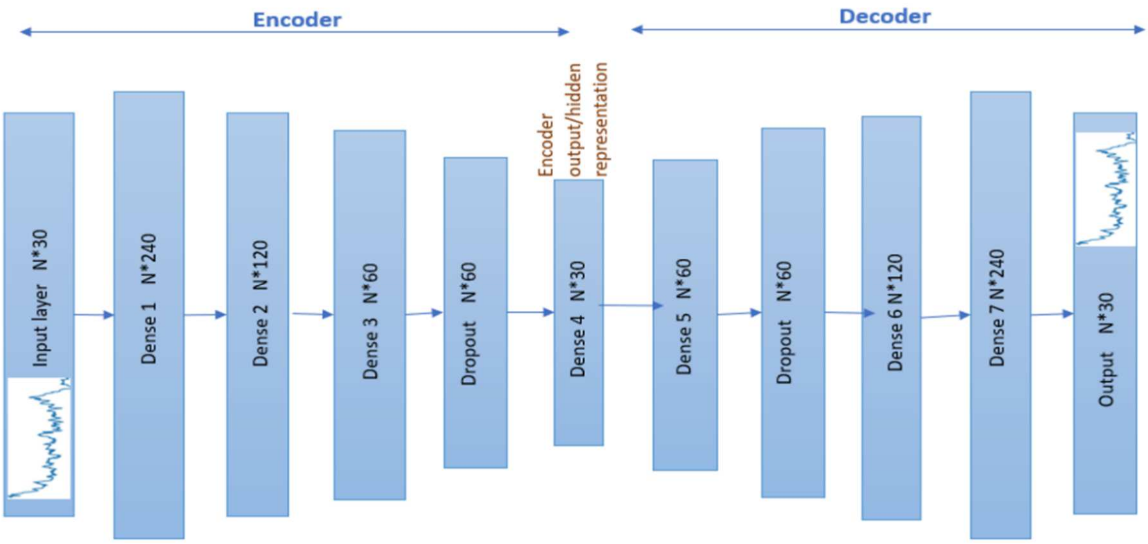

Fig. 3. Detailed CAE architecture with dense layers in encoder and decoder. N denotes the number of samples

activation and dropout as the encoder (Fig. 3). The last layer contains neurons matching the original input size, allowing the model to map the encodings back to the original dimensions. The decoder learns to reverse the dimensionality reduction performed by the encoder.

Loss Function: The loss function for the contractive autoencoder has two key components - the reconstruction loss and the contractive regularization loss.

The reconstruction loss measures how well the model reconstructs the input data after it has been encoded and decoded. This is calculated as the mean squared error (MSE) between the input data and the reconstructed output.

The contractive loss regularizes representations. It is calculated by obtaining the Jacobian matrix of outputs w.r.t inputs via gradient tape. Taking the Frobenius norm penalizes encodings where small input changes cause large output changes, encouraging contractive encodings.

The final loss function is a weighted sum of the MSE reconstruction loss and the contractive regularization loss.

Training: The CAE model is trained by minimizing a combined loss of reconstruction error and contractive regularization penalty using the Adam optimizer. A small learning rate of 0.00001 is set to control update sizes. For each batch, the input is passed through the encoder and decoder to reconstruct the output. The loss function calculates the mean squared reconstruction error and contractive regularization penalty. The combined losses are differentiated via backpropagation to determine the parameter gradients. The gradients are provided to the Adam optimizer, which uses them to update the weights and biases to minimize the loss.

The model is trained on batches of 32 samples over 50 epochs. The input data is provided as both input and target, so the model learns to reconstruct it. Validation data assesses performance during training. The training and validation losses decrease over epochs, indicating the model successfully learns to reconstruct the sequences.

E. Comparative Analysis

The research approach taken to detect anomalies in heartrate can be described as an inductive approach. This involves inferring a general rule from a specific set of data and applying it to new, unseen data to identify any anomalies. The approach involves an iterative and self-reflective process [19] of designing, building, and testing a system to detect anomalies in the heart rate data, evaluating its performance, and finetuning model hyperparameters to optimize its performance. The proposed model was implemented in Anaconda3, Tensorflow 2.12, and python 3.9 (CPU version) environments. The model was implemented on a laptop computer that featured an Intel Core i5-1135G7 CPU clocked at 2.4 GHz. The data used in this research is the data collected from Fitbit users. The data was pre-processed and labelled into normal and anomalies based on threshold values. Around 80% of the normal data was used for training, while the remaining data was merged and used for validation and testing purposes.

The CAE is compiled with Adam optimization to minimize the custom loss of reconstruction error and contractive regularization. Training proceeds by feeding batches of 32 samples to update weights and minimize loss over 50 epochs. Scaled training data is input as both input and target so the model learns reconstruction. Validation data assesses performance. EarlyStopping stops training early if validation loss plateaus after 5 epochs to prevent overfitting. By iterating batches and applying gradient optimization, the model minimizes the combined loss and learns to compress the input into a stable encoding while accurately reconstructing the original input.

The CAE demonstrated stable optimization and excellent performance. The training loss decreased steadily from 0.0009 to 0.00014, indicating successful reconstruction learning.

Training time per epoch was consistent, demonstrating reliable optimization. Early stopping could trigger shortly after validation loss plateaued around epoch 20. Achieving a low validation reconstruction error of ~0.0022 highlights robust learned representations.

Fig. 4 visualizes the model's reconstruction capability on a sample normal heart rate sequence. The first plot depicts the input sequence fed into the model. The second plot reveals the sequence reconstructed by the model from the encoded representations. The remarkably close alignment between the input and output sequences demonstrates that the model has successfully learned the fundamental patterns characterizing normal heart rate data, enabling it to accurately regenerate the input sequence despite compressing it into a low-dimensional encoding. This exemplifies the model's proficiency in extracting meaningful representations to precisely recreate normal heart rate sequences.

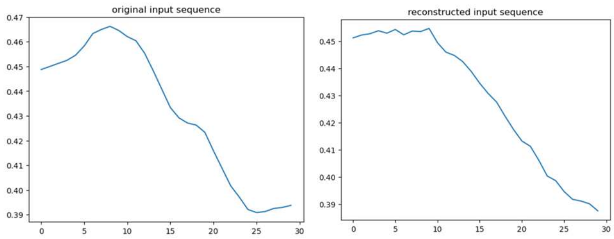

Fig. 4. CAE model reconstruction of a random heart rate sequence.

.To evaluate the trained autoencoder's generalization, its reconstruction competence is validated on an unseen test set. The model reconstructs each test sequence, then the deviation between the original and predicted sequences is quantified using mean squared error. A threshold of 0.00007, identified through iterative testing of multiple values, classifies the sequences as either normal or anomalous based on the reconstruction error. Test sequences accurately reconstructed below the set threshold are designated as normal patterns learned by the model. Meanwhile, sequences poorly reconstructed and exceeding the threshold are identified as unlearned anomalies. By comparing these predicted labels to the true labels, the threshold's efficacy in distinguishing anomalies is assessed. This iterative process determined 0.00007 as the optimal threshold for segregating normal and anomalous heart rate sequences based on reconstruction error, maximizing classification performance.

TABLE 1

PERFORMANCE MATRIX FOR CLASSIFICATION OF ANOMALAOUS VS NORMAL SEQUENCES

	Precision	Recall	F1-Score
Anomaly	0.94	0.91	0.93
Normal	0.91	0.94	0.93
Accuracy			0.93

The evaluation results are presented in the classification report as depicted in Table 1. The CAE model achieves an overall accuracy of 93% in discriminating between normal and anomalous heart rate sequences. The precision of 0.94 for detecting anomalies indicates a low false positive rate, while the recall of 0.91 shows the model correctly identifies the vast majority of anomalies.

The F1-score, balancing precision and recall, reaches 0.93 for anomalies. Similarly for normal sequences, precision of 0.91 and recall of 0.94 result in a strong 0.93 F1-score. The high accuracy, precision, recall and F1-scores validate the effectiveness of the autoencoder-based anomaly detection approach on unseen heart rate data.

Fig. 5 shows the reconstruction error on the test set varies between normal and anomalous sequences. Normal patterns in green have low, tight clustered errors, indicating accurate reconstruction. Anomalous sequences in red show a broader spread of larger errors.

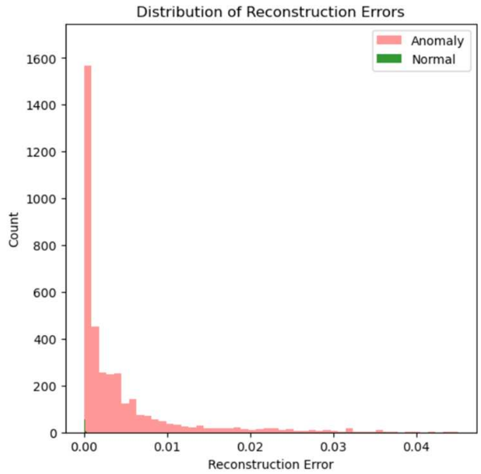

Fig. 5. Distribution of reconstruction errors in test data.

The ROC curve and AUC metric evaluate anomaly detection performance (Fig. 6). Reconstruction error per sample is used with true labels to derive AUC. The high AUC of 0.98 indicates excellent detection. The ROC curve's proximity to the upper left corner shows strong performance, confirming the effectiveness of the autoencoder-based anomaly detector.

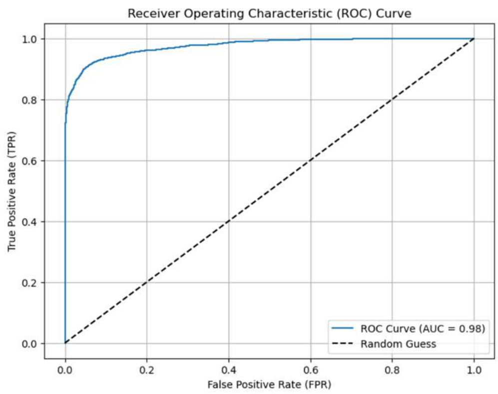

Fig. 6. ROC curve highlighting model's tradeoff between true and false positives.

IV. CONCLUSION

This study presented a novel CAE model for heart rate anomaly detection, showcasing its potential in handling noise and variability challenges in widely available consumer-grade wearable data. The CAE demonstrates high accuracy, precision, and recall in anomaly detection from noisy wearable data. The model was trained on a dataset of normal heart rate samples, and its performance was evaluated on a separate test dataset which contains both normal and anomalous data. The results demonstrated that the CAE model achieved a high accuracy of 90% on the test data, with balanced precision and recall scores for both the 'Anomaly' and 'Normal' classes. The area under the ROC curve reached 0.98, further validating anomaly detection proficiency. The reconstruction error distributions clearly differentiated anomalies on unseen test data. Building on these promising anomaly detection capabilities, this study lays the foundation for ancillary innovations like personalized interventions based on predicting future heart rate trajectories.

However, evaluation was limited to one dataset from a single source. More robust validation on larger, diverse datasets is needed to fully assess capabilities. Additionally, the labelling of anomalies was done manually by setting a threshold, which may miss outliers that fall within the normal range. While this research focused on binary classification, the CAE approach shows promise for multi-class classification of diverse cardiac irregularities, enabling more granular anomaly detection. Exploring this presents an exciting future direction. The future work should explore alternative architectures, like extending for time-series forecasting to enable proactive monitoring. With advances in anomaly detection and data availability, the CAE model lays groundwork for innovative healthcare approaches. However, obtaining properly labelled anomaly data posed a key challenge. Next research phase demands collecting and labelling real-world data.

Overall, the model has demonstrated its potential as an effective tool for heart rate anomaly detection, offering valuable contributions to the field of healthcare monitoring.

V. Acknowledgement

We gratefully acknowledge the generous support and funding provided by the Digihealth Asia project (https://digihealth-asia.eu/), which has been instrumental in enabling us to conduct this impactful research. Erasmus Mundus Grant ID: 619193-EPP-1-2020-1-BE-EPPKA2-CBHE-JP

VI. References

[1] C. S. McLachlan and H. Truong, “A narrative review of commercial platforms offering tracking of heart rate variability in corporate employees to detect and manage stress,” Journal of Cardiovascular Development and Disease, vol. 10, no. 4, p. 141, 2023. doi:10.3390/jcdd10040141

[2] G. M. Peters et al., “Detecting patient deterioration early using continuous heart rate and respiratory rate measurements in hospitalized COVID-19 patients,” Journal of Medical Systems, vol. 47, no. 1, 2023. doi:10.1007/s10916-022-01898-w

[3] P. C. Demkowicz et al., Physician responses to Apple Watch-detected irregular rhythm alerts: A case-based survey, 2022. doi:10.1101/2022.08.02.22278237

[4] N. Alugubelli, H. Abuissa, and A. Roka, “Wearable devices for remote monitoring of heart rate and heart rate variability—what we know and what is coming,” Sensors, vol. 22, no. 22, p. 8903, 2022. doi:10.3390/s22228903

[5] S. Liu et al., “Fitbeat: Covid-19 estimation based on wristband heart rate using a contrastive convolutional auto-encoder,” Pattern Recognition, vol. 123, p. 108403, 2022. doi:10.1016/j.patcog.2021.108403

[6] F. F. Abir et al., “PCovNet+: A CNN-VAE anomaly detection framework with LSTM embeddings for smartwatch-based COVID-19 detection,” Engineering Applications of Artificial Intelligence, vol. 122, p. 106130, 2023. doi:10.1016/j.engappai.2023.106130

[7] Yang Xie et al., “Prediction of chronic obstructive pulmonary disease exacerbation using physiological time series patterns,” 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013. doi:10.1109/embc.2013.6611114

[8] S. Bae, A. K. Dey, and C. A. Low, “Using passively collected sedentary behavior to predict hospital readmission,” Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016. doi:10.1145/2971648.2971750

[9] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 833–840, 2011.

[10] W. Wang, X. Du, D. Shan, R. Qin, and N. Wang, “Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine,” IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1634–1646, 2022. doi:10.1109/tcc.2020.3001017

[11] B. Diallo et al., “Deep embedding clustering based on contractive autoencoder,” Neurocomputing, vol. 433, pp. 96–107, 2021. doi:10.1016/j.neucom.2020.12.094

[12] E. Q. Wu, X. Y. Peng, C. Z. Zhang, J. X. Lin, and R. S. Sheng, “Pilots’ fatigue status recognition using deep contractive Autoencoder Network,” IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 10, pp. 3907–3919, 2019. doi:10.1109/tim.2018.2885608

[13] M. Radmanesh, A. A. Rezaei, M. Jalili, A. Hashemi, and M. M. Goudarzi, “Online spike sorting via deep contractive autoencoder,” Neural Networks, vol. 155, pp. 39–49, 2022. doi:10.1016/j.neunet.2022.08.001

[14] R. Banerjee, A. Mukherjee, and A. Ghose, “Noise cleaning of ECG on edge device using convolutional sparse contractive Autoencoder,” 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), 2022. doi:10.1109/percomworkshops53856.2022.9767313

[15] M. Gu et al., “A lightweight convolutional neural network hardware implementation for wearable heart rate anomaly detection,” Computers in Biology and Medicine, vol. 155, p. 106623, 2023. doi:10.1016/j.compbiomed.2023.106623

[16] J. Zhong et al., “Convolutional autoencoding and gaussian mixture clustering for unsupervised beat-to-beat heart rate estimation of electrocardiograms from wearable sensors,” Sensors, vol. 21, no. 21, p. 7163, 2021. doi:10.3390/s21217163

[17] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” ACM Computing Surveys, vol. 54, no. 3, pp. 1–33, 2021. doi:10.1145/3444690

[18] Julenaranguren, “Bellabeat - case study,” Kaggle, https://www.kaggle.com/code/julenaranguren/bellabeat-case-study (accessed Aug. 6, 2023).

[19] K. Lewin, “Action research and minority problems (1946).,” Resolving social conflicts and field theory in social science., pp. 143–152. doi:10.1037/10269-013

[20] D. Bank, N. Koenigstein, and R. Giryes, Autoencoders, 2020. doi:10.1017/9781108955652.006

[21] M. Aamir, N. Mohd Nawi, F. Wahid, and H. Mahdin, “A deep contractive autoencoder for solving multiclass classification problems,” Evolutionary Intelligence, vol. 14, no. 4, pp. 1619–1633, 2020. doi:10.1007/s12065-020-00424-6

[22] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, Contractive Auto-Encoders: Explicit Invariance During Feature Extraction, 2011.

[23] Julenaranguren, “Bellabeat - case study,” Kaggle, https://www.kaggle.com/code/julenaranguren/bellabeat-case-study.