

UNSUPERVISED REAL-TIME DIAGNOSIS SYSTEM FOR ECG STREAMING DATA

Eman A. Maghawry
Faculty of Computer and
Information Sciences, Ain Shams
University, Cairo, Egypt
e_maghawry@cis.asu.edu.eg

Rasha M. Ismail
Faculty of Computer and Information
Sciences, Ain Shams University, Cairo,
Egypt
rashaismail@cis.asu.edu.eg

Tarek F. Gharib
Faculty of Computer and
Information Sciences, Ain
Shams University, Cairo, Egypt
tfgharib@cis.asu.edu.eg

Received 2021- 3-27; Revised 2021-5-27; Accepted 2021-5-28

Abstract

Detecting anomalies in time series data plays a vital role in the various applications of diagnosis systems. The importance of anomaly detection is increased by its ability to detect abnormalities in Electrocardiogram (ECG) signals to generate alerts for cardiac health problems. An ECG is a time series that provides essential information about the electrical activity of the heart and is used in the diagnosis of numerous heart diseases. An accurate ECG streaming analytics approach requires continuous learning and adaptation in changing data behaviors. We aim to diagnose ECG by investigating healthy ECG and ECG with cardiological disorders by detecting anomalies in ECG signals. The main objective of this paper is to develop an efficient unsupervised diagnosing system for ECG streaming data based on an online sequence memory algorithm called Hierarchical Temporal Memory (HTM). The HTM is based on neural network and machine learning algorithm for continuous learning tasks. The proposed customization of the HTM algorithm based on our problem domain provides a significant performance results of detection of anomalies in the ECG signals.

Keywords: Anomaly detection, ECG signals streaming, ECG diagnosis system, Hierarchical temporal memory

1. Introduction

Anomaly Detection is a point-in-time detection task where the data behavior is significantly different from the previous normal behavior. Nowadays, there is an exponential increase in data availability in streaming mode because of the rise of the Internet of Things (IoT) and a huge number of real-time applications with sensors. The efficient analysis of these streams can provide valuable information for many applications and usage cases. Therefore, detecting anomalies in streaming mode requires a rapid response to change in data behavior to trigger alerts in critical scenarios.

The detection of anomalies in large volumes of data has many important real-time applications such as healthcare monitoring. Electrocardiogram (ECG) anomaly detection has been widely used to detect abnormal heartbeats that help in diagnosing many heart diseases, the leading cause of premature death worldwide [1].

* Corresponding author: Eman A. Maghawry
Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
E-mail address: e_maghawry@cis.asu.edu.eg

Streaming data can be infinite, so offline algorithms that try to store the entire stream for analysis don't adapt with it. Therefore, the fundamental ability to analyze each stream needs to be modeled in an unsupervised mode to detect unusual behaviors in real-time. Unsupervised mode does not require the storage of all data but may need to store a small subset of data versus supervised mode. However, detecting anomalies in streaming data is a challenging task, detectors should process data in real-time, and continuously update their models to adapt to change [2].

Various machine learning and signal processing techniques were used in the analysis of ECG signals. Existing works in this area can be divided into noise filtering, extraction of various ECG sub-waves, characteristics values and identification of anomalies in the ECG [3].

In [4] their algorithm used time series motif discovery to determine frequent patterns in the ECG. Authors of [5] presented an approach to identify abnormalities based on differences in heartbeat morphologies. Their approach is inspired by time-series data mining techniques and methods for statistical outlier detection.

The authors of [6] introduced an algorithm to extract the fiducial points from the ECG based on the Discrete Wavelet Transform (DWT) method. Also, in [8], they used the DWT for the ECG analysis and a support vector machine (SVM) classifier for ECG analysis and classification for heartbeat diagnosis.

Authors of [3] compared the random forest method with neural networks, k-nearest neighbour and support vector machines for congestive heart failure. Authors of [7] used a generalized feed forward neural network to investigate arrhythmia patients. Other networks such as backward projection neural networks can produce similar results but require more time for training as in [8].

Authors of [9] selected an optimal feature set by utilizing a genetic algorithm and empirical mode decomposition (EMD) where the resulting features were fed to a support vector machine for classification. Also, in [10], the authors used a combination of supervised active learning and unsupervised learning with autoencoder and softmax for feature learning and a deep neural network for classification according to the standards of Association for the Advancement of Medical Instrumentation (AAMI). A wavelet coherence technique (WTC) was used in [11] and long-short term memory (LSTM) algorithm was used in [12] for identifying anomalies. Authors of [13] they deal with ECG beat as a binary image rather than a set of numeric features in detecting the abnormalities.

In [14], their proposed algorithm was an unsupervised method based on brute force discord detection (BFDD) and applied on the ECG. Authors of [15] proposed a supervised approach based on Convolutional Neural Networks for ECG classification; they proposed a modification of CNNs in which the convolutional layers are trained by unsupervised methods.

There are many anomaly detections approaches, both supervised (e.g. support vector machines and decision trees [16]) and unsupervised (e.g. clustering). However, most anomaly detection methods are for batch data processing, and unsuitable for real-time streaming applications. The online analysis of data streams poses challenges, such as the learning of new concepts and developing algorithms that work in time sensitive and computationally constrained environments [17].

Common previous solutions for anomaly detection use machine learning methods. Usually, they are supervised learning techniques that require a training sample of normal behavior as training model, and in other cases, an unusual set of patterns may be required. However, in many real-time settings, it is a difficult task to obtain this prior knowledge. Therefore, unsupervised learning techniques are used to attempt to identify abnormal patterns, which adjust the knowledge model directly from the input stream [18].

The detection performance of these statistical techniques can be sensitive to thresholds and window size. This sometimes results in many false positives in the detection of anomalies as data changes, requiring frequent updates of thresholds. Therefore, there are still opportunities to improve existing methodologies.

Consequently, we consider these properties for effective extraction of ECG segments in real time and detection of anomalies in the streams. More recently, cortical-like algorithms such as the Hierarchical Temporal Memory

(HTM) have been a promising technique for several real-world prediction tasks. However, very little work has been done on applying cortical-like algorithms to time series data such as ECG. We utilize the HTM in our ECG diagnosis system.

The main objective of this paper is to detect changes in data behavior from incoming ECG data. Our aim is to diagnose ECG by investigating healthy ECG and ECG with cardiological disorders by detecting anomalies in ECG signals. The ECG signals are presented in a streaming fashion and predictions are needed continuously to assist in the diagnosis of many heart diseases. As such, a continuous online learning technique is required to meet the research objective. ECG anomaly detection technique based on customized online sequence memory algorithm called Hierarchical Temporal Memory (HTM) [19] is proposed. The proposed anomaly detection technique is a part of our ECG diagnosis system to determine whether a real-time ECG stream contains any abnormal heartbeats. In case of abnormalities, the diagnosed anomalies and the diagnostic report will be transmitted via a network to a physician for appropriate actions.

2. Overview of HTM

A Hierarchical Temporal Memory (HTM) model was configured and built using Numenta NuPic software [19]. The HTM is implemented in the form of a series of algorithms known as Cortical Learning Algorithms (CLA) using neurobiological principles [20]. The HTM is based on machine intelligence algorithm and neural network for continuous learning tasks [14]. This section presents a brief outline of the HTM. The official literature [21] provides a more detailed description of algorithms and the basic concepts of HTM technique. Also, authors of [25] introduce a comparison with existing machine learning techniques like neural networks such as Bayesian networks against the HTM [26].

The HTM consists of layers and regions as shown in Fig. 1. Each layer learns in unsupervised manner by finding patterns in the entry sequence and patterns over time. An HTM region is logically includes a set of columns. Each column in a region is linked to a single subset of the input bits.

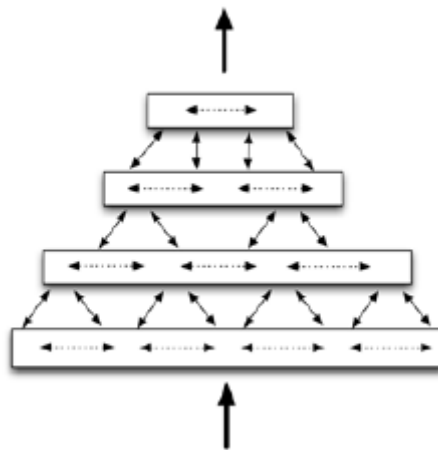


Fig. 1. Simple hierarchy figure of four HTM regions arranged in a four-level [23]

In each layer, the columns composed of cells as in Fig. 2 that are interconnected with other cells from other columns. Each cell is activated by an input in its column [20]. The cells of each column are connected to the cells of several other columns, likewise the columns are connected to the output of several columns in their input spaces. Their initial connections are usually set on random basis [20]. The gray cells in Fig. 2 introduces active cells.

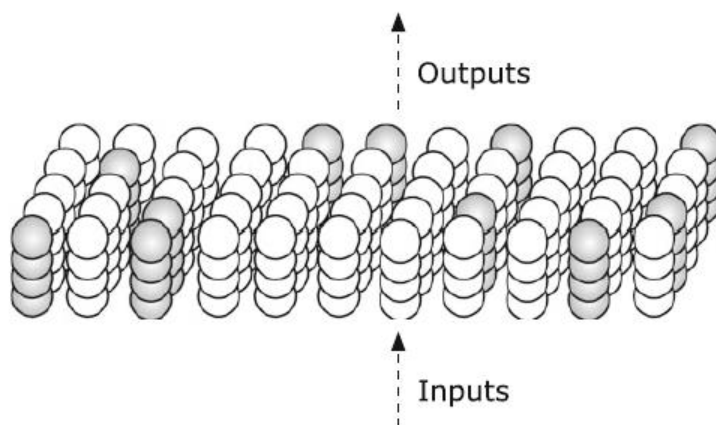


Fig. 2. From a conceptual perspective, the HTM columns are arranged in a two-dimensional array [20]

When the HTM gets a new pattern, it tries to match it to stored patterns. Predictions can only be done based upon previously received patterns sequences. The HTM does not understand the meaning of patterns; it only knows which patterns are likely to follow specific observed patterns [21]. The HTM is a distributed memory model that utilizes temporal information and requires less training compared to other conventional neural networks [20, 22]. It has three basic functions: encoding, learning and prediction.

The encoding is a representation of the data set. It is based on the formation of sparse distributed representations (SDRs) from continuous stream of input patterns. The SDR is a binary array that most bits are zeros, and few are ones, the ones bits carry important semantic features of the data. The memory mechanism of the HTM is relied on the SDR. When the HTM receives a new pattern, it will be distributed to those individual cells to be memorized as SDR. The importance of using SDRs is to match similar inputs to similar representations and only needs to match a portion of the pattern is sufficient to ensure significant matching.

The HTM learns what patterns are likely to follow each other, so learning how to predict future patterns. Each column of the HTM has possible connections to a random selection of the bits in an input space to the HTM (*called potential bits*). The HTM uses the concept of *permanence* to modify the connectivity between the input space and the columns. Permanence is a scalar value between zero and one and represents the strength of the connection.

The learning function is composed of two basic algorithms [20]:

1. **HTM Spatial Pooler:** converts an input of region into a sparse pattern. It learns the connections to each column from a subset of the inputs and uses inhibition to select a sparse set of active columns. *Spatial term* means that inputs share many active bits and *pooled term* means inputs are grouped into a common representation.
2. **HTM Temporal Pooler:** makes predictions on the SDRs created by the HTM Spatial Pooler. Each cell has three conditions: active, predictive, or deactivated based on its connections. If the cell is active, it means it connects with other cells that were active just before. Cells can then be predictive when they are active by examining their connections. If all the cells do this, collectively they can store and recall sequences, and they can predict what will happen afterwards. When a column gets an input, only the predictive will then be activated. If the input is unexpected, all the cells in the column will be activated.

Once completion of the learning, the transition between the SDRs can be memorized by the HTM and has the ability to predict a next input sequence based on the previous input. This makes the HTM very efficient for noise and missing data in a way that the prior input will help to generate a context of what needs to be expected. The following steps summarize how the HTM prediction performs new inputs:

1. Create an SDR of the input by activating whole columns.
2. Place the input into context by selecting among cells of the active columns.
3. Predict future patterns based on learned transitions among SDRs.

3. The Proposed Real Time ECG Diagnosis System

The main objective of the proposed system is to provide diagnostic decision for the ECG in real-time. As shown in Fig. 3, the input of the proposed system is an ECG signal stream, each stream is segmented into single cardiac cycles. The proposed system includes three modules. The first module is the *Heartbeat Encoder Module* which extracts the ECG wave fiducial points to convert the ECG characteristics into encoded bits (SDR). The second module is the *Anomaly Detector Module* which exploits encoded inputs in the automatic detection of heart abnormalities to help in the diagnosis of heart disease.

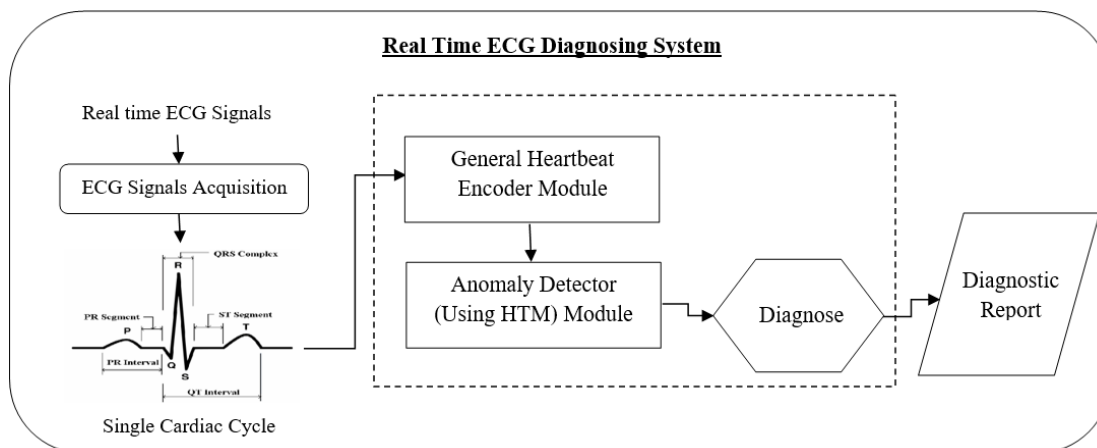


Fig. 3. The Proposed Real Time ECG Diagnosing System

3.1 General Heartbeat Encoder Module

When ECG signals are streamed to the proposed system, the process of the ECG Signals Acquisition is responsible for segmenting these signals into a single beat cycle to be sent to the Encoder Module. Then, the signal smoothed by Band Pass Filter process [23] to remove undesired frequency noise components arising from the operation of other organs. Since the presence of noise makes it more difficult to reliably detect the fiducial points (QRS, P wave and T wave as shown in Fig. 4). Base-line wandering can mask some important features of the ECG, it is caused by variable electrode-skin impedance, patients movements and breathing [23]. When ECG signal noise is removed, Multilevel Teager Energy Operator algorithm [MTEO] algorithm [23] is used to detect ECG signal fiducial points as shown in Fig. 4. The Teager energy operator (TEO) is a time-frequency analysis algorithm and has been employed in many different signal processing applications. It uses principal component analysis (PCA) to reduce the number of templates and fiducial points and, as a result, reduce the correlation computation load and improve calculation speed [24].

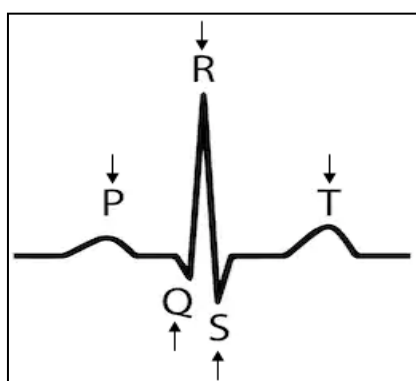


Fig. 4. Peaks of Single Beat

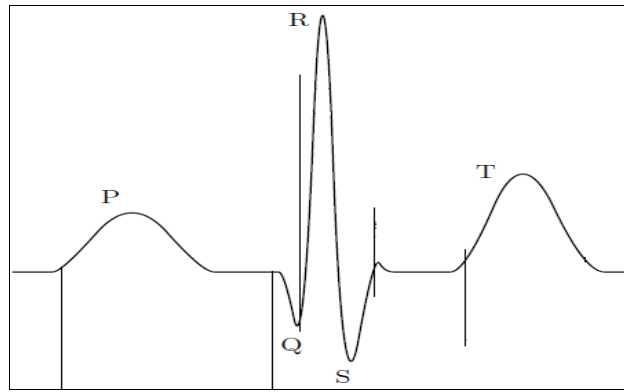


Fig. 5. The Starting Points for Each Wave

After the peaks are detected, they are used to find the beat characteristics by detecting the starting positions for each wave as shown in Fig. 5. The beat segments and intervals can be fetched as shown in Fig. 6. These values can describe the ECG features that indicate the electrical activity of the heart and are used for additional investigations or diagnosis.

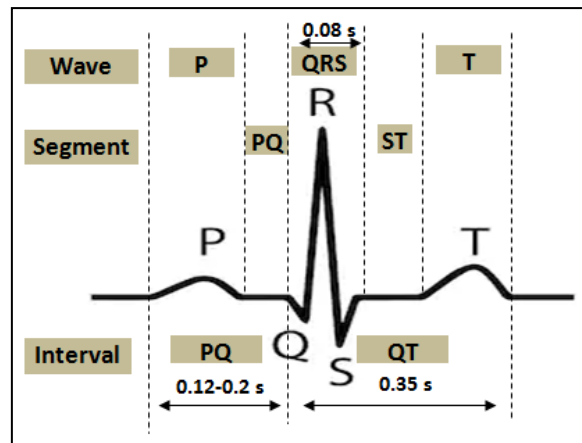


Fig. 6. Examples for Normal Values Ranges in Segments and Intervals

The algorithm used to detect the start and end points of the wave takes advantage of the fact that the wave contributes to a significant derivative, except around its extrema [25]. The algorithm begins at the peak of the wave and looks for the maximum derivative near the beginning and end of the wave. There are three steps in this algorithm:

- 1) **Detection:** use the maxima value of the wave, then follow this direction until the absolute derivative falls below a certain fraction of the maximum.
- 2) **Windowing:** include the portion of the wave between the maximum segment and the segment with the same electric level.
- 3) **Detecting:** the lower the threshold, the higher separation points relative to the wave maximum.

All extracted features are converted into an SDR that can be fed into the HTM. An SDR is given by a binary array that has a fixed number of bits. Different array lengths can be used. Only 2% of bits are ones, referred to as active bits. The zero bits are the inactive bits. In an SDR, two inputs with similar semantic meaning must have similar SDR representations. Consequently, they must have same equal bits because SDRs arrays are compared position by position [26]. The encoding process is responsible for determining which output bits should be ones, and which should be zeros in SDRs, for given ECG signal features to capture the important semantic characteristics of the data.

Certain aspects are taken into consideration when implementing this conversion process [27]:

1. Semantically similar beats should result in SDRs with overlapping active bits.

The Anomaly Detector Module is responsible for detecting abnormal values for ECG features to aid in the diagnosis of heart disease using HTM. HTM adopts a continuous learning model, it does not need to store a training dataset like most other anomaly detection algorithms. Instead, it learns from each data point using unsupervised Hebbian-like learning rule [29].

The proposed anomaly detection module computes a raw anomaly score that measures the deviation between the model's expected input and the actual input. It can be calculated based on the intersection between the expected and actual sparse vectors.

As described earlier, spatial pooling is about adjusting the connections between columns and the input space. Also, temporal pooling is about adjusting the connection between cells of columns. Both the spatial and temporal pooling works according to the principle that cells or columns that activate together reinforce their connections (known as permanence).

The HTM spatial pooler forms connections from spatial pooler columns to a subset of the input space (known as potential connections). Different input patterns lead to different levels of columns activation. Thus, an inhibition mechanism is used to ensure that a small fraction of the columns that receive most inputs is active within the inhibition radius. The inhibition function is defined to achieve a relatively constant percentage of active columns, even when the number of active input bits varies significantly. In addition to inhibition, another mechanism known as boosting allows less activated columns to be more activated and the more activated neurons to be less activated, thus allow more complex patterns. The permanence of the connections is adjusted according to the Hebbian rule; for each active column, active inputs are reinforced, and inactive inputs are punished [30]. After a defined number of iterations, these connections are adjusted, and temporal pooling is performed based on a cell state (active, predicting or deactivated).

Each cell will examine its connections and if some of them are active then it will change to a predictive state. When a column received an input, only cells in predictive mode are then activated. If no cells are at this stage, then all the cells in the column will be activated because the input is unexpected. When a cell is activated, it examines other cells in other columns that were active earlier and then strengthen the connection with them. The cell's connections with other inactive cells will be weakened. If any of the cells in a column have not been predicted, then all the cells are activated (known as *bursting*). HTM calculates the anomaly score which corresponds to the fraction given by the number of bursting columns divided by the total number of active columns [26].

HTM specifies an *anomaly score* (s) for each new pattern it receives as follows:

1. If an incoming pattern was predicted, then $s = 0$.
2. If a pattern was not predicted at all, then $s = 1$.
3. If a pattern is partially predicted, then $0 < s < 1$

The actual score depends on whether the actual pattern received is "similar" to the expected pattern. The similarity is determined by the SDR. The greater the overlap between the actual and predicted bits in the column space, the lower the anomaly score. At the beginning of the training, the anomaly score will be high because most of the patterns will be new. As HTM learns, the anomaly score decreases until there is a change in pattern stream [26].

There are still no specific standards for selecting parameters values in a HTM implementation as described in [22, 29]. As presented by [29], the parameters should be adjusted to fit into the problem domain. Not all parameters are covered in this section due to excessive number of parameters. HTM is customized and configured to be applicable to our proposed anomaly detection module. The module follows all layer configurations and recommendations by [19].

4. Experimental Environment

The MATLAB 2018 software was used for ECG analysis and Numenta NuPIC [19] software was used for the HTM model. Experiments were run on Windows7, Inter Core 7, 6 GB of main memory, 2.60GHz CPU and 1TB hard disk.

4.1 Dataset

The proposed system was evaluated using QT datasets [31] taken from PhysioNet (<http://www.physionet.org>), which is used by most researchers. It contains a total of 105 cases, each fifteen minutes long with two channel ECGs. It consists of T-wave morphologies in cases chosen from the supraventricular arrhythmia, MIT-BIH arrhythmia, sudden death, long-term, and ST change, European ST-T databases and BIH normal. This dataset includes records of healthy people and records of patients with cardiological disorders. All annotations were manually reviewed to select twenty-two records with a normal ECG rhythm and the rest of records has abnormal ECG rhythm. All records were sampled at 250 Hz. In each record, between 30 and 100 representative heartbeats were manually annotated by cardiologists. The beginning, peak and end of the P-wave, beginning and end of the QRS-complex, peak and end of the T-wave are annotated. Thirty consecutive beats of the dominant morphology were annotated in each case. An example of a record with normal beats (annotated by N) and contains cardiovascular disorder (Premature ventricular contraction) annotated by V is shown in Fig. 8.

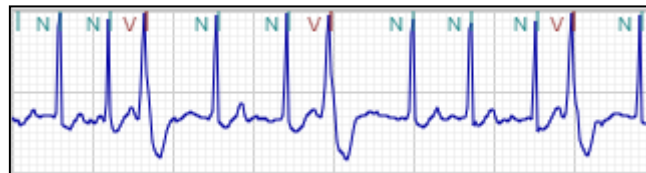


Fig. 8. Part of QT Database Signal

4.2 Experimental Setup

Multiple parameters of the HTM algorithm should be set to evaluate our system like the number of columns, the number of cells in each column and their connections. The parameters of the spatial pooler function which overlapping input patterns map to the same sparse distributed representation are summarized in the following which Table 2.

Table 2. Spatial Pooler Specifications

Spatial Pooler Specifications	
W	Number of adjacent block of 1's
Min Value	The minimum value of the input value.
Max Value	The maximum value of the input value.
Range	Equals (Maxval – Minval)
Radius	Inputs separated by more than, or equal to this radius value will have non-overlapping representations.
N	Number of bits in the SDR: $N = (W * (Range/Radius)) + (2 * (W-1)/2).$
Resolution	Inputs separated by more than, or equal to this resolution value will have different representations. $Resolution = Radius / W$
Column	Group of one or more cells that work as a unit in an HTM Region.
Input Dimension	A list representing the dimensions of the input, where each value represents the size of the dimension.
Column Dimension	Total number of columns.
Cells per Column	Total number of cells per column
Potential Radius	The percent of the inputs that column can be connected to. If set to 1, the column will be connected to every input within its potential radius.
Max Boost	Each column's overlap gets multiplied by a boost factor before it gets considered for inhibition.

The used parameters for temporal memory function which the process of forming a representation of a sequence of input patterns are presented in Table 3.

Table 3. Temporal Memory Specifications

Temporal Memory Specifications	
Permanence	<ul style="list-style-type: none"> - A scalar value that indicates the connection state of the cells. The acceptable values are [0,1]. - A cell can be in the following states: <ul style="list-style-type: none"> • Connected: permanence value is above a threshold. • Potential: permanence value is below a threshold. • Unconnected: does not have the ability to connect. - Learning in an HTM region is accomplished by modifying permanence values of potential synapses.
Initial Permanence	<ul style="list-style-type: none"> - Spatial pooler start to pool the initial permanence values for cells and connected inputs. - Permanence values are increased for cells connected to input bits that are turned on and decreased for cells connected to inputs bits that are turned off. - Initial permanence of a new input
Connected Permanence	If the permanence value for adjacent cells is greater than this value, it is said to be connected.
Key Window Size	Number of inputs used in the computation of the moving average
Min Threshold	Minimum segment activity for learning.

5. Evaluation of the Proposed Anomaly Detection Technique

The confusion matrix is used to assess the performance of our module. The performance based on how our system can detect the abnormal beats in the tested signals [32].

The main metrics that are used to track the performance of different configurations for our model are presented in Table 4, TP, TN, FP, and FN, respectively represent True Positive, True Negative, False Positive, and False Negative

- True positive = correctly identified
- False positive = incorrectly identified
- True negative = correctly rejected
- False negative = incorrectly rejected

Table 4. The Main Performance Metrics

Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Recall	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$

Several experiments were run in different configurations to study the influence of the number of parameters and their values on our ECG dataset and to get the best configuration. The correct configuration of the algorithm to our problem area would significantly improve the results that improve system performance.

From Table 5 to Table 9 present different setup settings for the HTM. 1st configuration evaluates the default parameters values for the HTM. The second configuration evaluates the modifications of the potential radius and max boost values. The third configuration increases the initial permanence and connected permanence values, changing the min threshold value but not affecting the results. The fourth and fifth configurations set the same permanence and connected permanence values, changing the min threshold value and max boost didn't affect the results of the evaluation.

Table 5. First Configuration Parameters with Results

1st Configuration Parameters						
Column Dimension	Cells Per Column	Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
512	8	9	10	0.4	4	0.5
Evaluation for 1st Configuration						
Accuracy (%)		Recall (%)		Precision (%)		
50		83		55		

Table 6. Second Configuration Parameters with Results

2nd Configuration Parameters						
Column Dimension	Cells Per Column	Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
512	8	3	5	0.4	4	0.5
Evaluation for 2nd Configuration						
Accuracy (%)		Recall (%)		Precision (%)		
75		93		78		

Table 7. Third Configuration Parameters with Results

3rd Configuration Parameters						
Column Dimension	Cells Per Column	Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
512	8	3	5	0.8	1	0.8
Evaluation for 3rd Configuration						
Accuracy (%)		Recall (%)		Precision (%)		
26		44		39		

Table 8. Fourth Configuration Parameters with Results

4th Configuration Parameters						
Column Dimension	Cells Per Column	Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
512	8	3	10	0.8	4	0.8
Evaluation for 4th Configuration						
Accuracy (%)		Recall (%)		Precision (%)		
100		100		100		

Table 9. Fifth Configuration Parameters with Results

5th Configuration Parameters						
Column Dimension	Cells Per Column	Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
512	16	3	5	0.8	4	0.8

Evaluation for 5 th Configuration		
Accuracy (%)	Recall (%)	Precision (%)
100	100	100

As shown in Fig. 9, the fourth and fifth configurations are the best ones that improve the results. The results showed that the increasing value of the permanence or cells connection strength improves the accuracy and overall system performance. Furthermore, when the number of other parameters is considered, the accuracy of the system has increased.

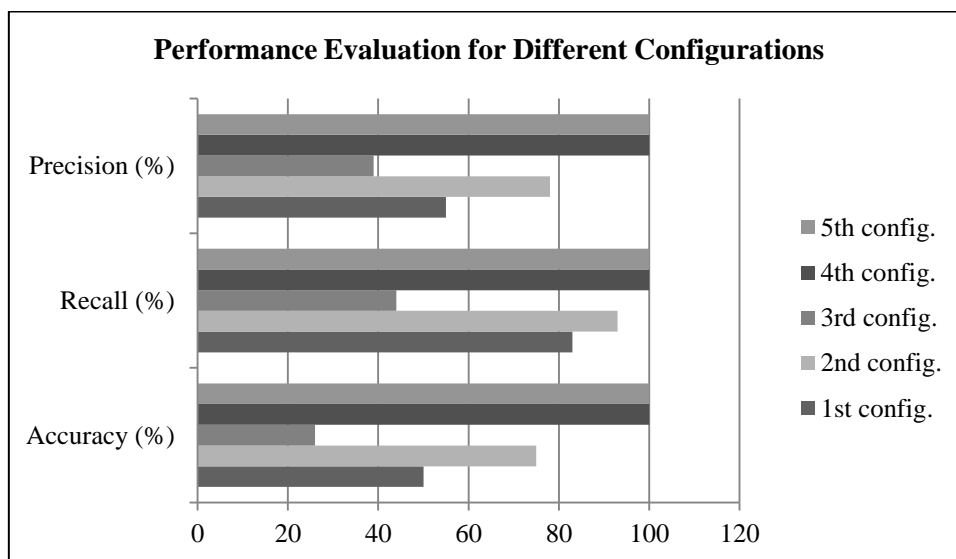


Fig. 9. Performance Evaluation for Different HTM Configurations

The configurations in Table 10 are used to study the effect of increasing or decreasing the column dimensions on the result relative to the best configuration. The tested dimensions are 128, 512, 1024 and 2048 dimensions. As shown in Table 11 and Fig. 10, performance decreases with both the largest and smallest column dimensions.

Table 10. Configuration Parameters for Different Column Dimensions

Configuration Parameters				
Potential Radius	Max Boost	Initial Permanence	Min Threshold	Connected Permanence
3	5	0.8	4	0.8

Table 11. Performance Metrics for Different Column Dimensions

Column Dimension	Cells Per Column	Accuracy (%)	Recall (%)	Precision (%)	Specificity (%)
2048	32	33	100	33	0
1024	16	46	62	59	12.5
512	8	100	100	100	100
128	8	68	82	78	20

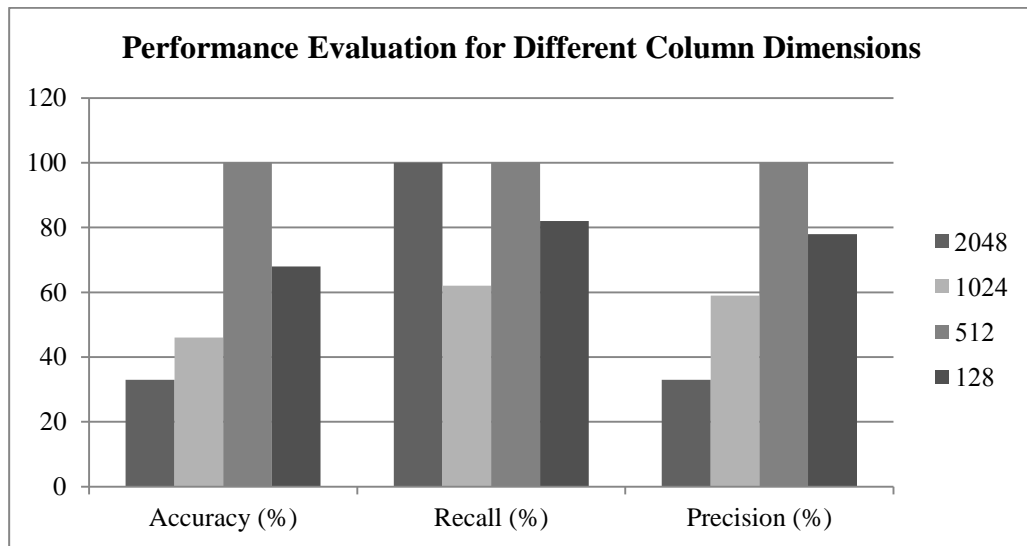


Fig. 10. Performance Evaluation for Different Column Dimensions

As shown in Fig. 10, performance achieves low accuracy at higher dimensions values because of the complex input representation due to the large pooling space. In addition, selecting very small column dimensions decreases the system performance, as the very small pooler space causes inaccurate predictions for the next input during the temporal memory learning phase. Table 11 demonstrates a comparison of accuracy between our approach against other supervised and unsupervised techniques for ECG anomaly detection. The results show an improvement in the accuracy of the anomaly detection module in our proposed approach.

Table 11. Comparison of Accuracy between our Approach and Other Literatures

Literature	(S/U) *	Method	Accuracy
Batra et al. [33]	S	SVM	83.04%
Pereira et al. [34]	S	SVM	98.43%
Nasiri et al. [35]	S	SVM and Genetic Algorithm	93%
Karim et al. [36]	S	F-t ALSTM-FCN	94.96%
Malhotra et al. [37]	S	SAE-C	93.40%
Pereira et al. [34]	U	VRAE+Clust/W	95.22%
Pereira et al. [34]	U	K-means	95.96%
Ceylan et al. [38]	U	Fuzzy Clustering Neural Network Algorithm	99%
Our approach	U	HTM	100%

*S= Supervised, U=Unsupervised

6. Conclusion

The main goal of this paper is to detect changes in data behavior from the incoming ECG data streaming. Therefore, an effective streaming anomaly detection mechanism is required to achieve our goal. The proposed anomaly detection technique is a part of our ECG diagnosis system to help in diagnosing cardiac disorders. In this paper, we applied an online sequence memory algorithm called Hierarchical Temporal Memory (HTM) to ECG anomaly detection problem domain. Various configuration parameters for the HTM have been tested and evaluated. The customized configuration proves a significant improvement in the performance of the anomaly detection module in our proposed ECG diagnosis system.

7. References

[1] P. Sahoo et al., "A cardiac early warning system with multi channel SCG and ECG monitoring for mobile

- health,” *Sensors*, vol. 17, p. 711, 2017.
- [2] S. Ahmad and S. Purdy, “Real-Time Anomaly Detection for Streaming Analytics,” *arXiv:1607.02480*, 2016.
- [3] V. Mithal, S. Chaturvedi, M. Jain and H. Karnick, “Diagnosis by Exclusion: Normal is the Absence of the Abnormal,” 2009.
- [4] H. Sivaraks and C. A. Ratanamahatana, “Robust and accurate anomaly detection in ECG artifacts using time series motif discovery,” *Comput. Math. Methods Med.*, 2015.
- [5] D. Ngo and B. Veeravalli, “Design of a real-time morphology-based anomaly detection method from ecg streams,” in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015.
- [6] C. V. Banupriya and S. Karpagavalli, “Electrocardiogram beat classification using probabilistic neural network,” *Int. J. Comput. Appl. (IJCA)*, vol. 1, pp. 31-37, 2014.
- [7] Z. Masetic and A. Subasi, “Congestive heart failure detection using random forest classifier,” *Computer Methods and Programs in Biomedicine*, vol. 130, p. 54–64, 2016.
- [8] S. Jadhav, S. Nalbalwar and A. Ghatol, “Performance evaluation of multilayer perceptron neural network based cardiac arrhythmia classifier,” *IJCSI International Journal of Computer Science*, vol. 9, 2012.
- [9] H. Liu et al., “Practical Method for ECG Classification Using Weighted ELM,” *Applied Mechanics and Materials*, Vols. 738-739, pp. 832-835, 2015.
- [10] D. Rohan and A. Patil, “Layered Approach for ECG beat classification utilizing Neural Network,” *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, pp. 1495-1500, 2012.
- [11] M. M. A. Rahhal, H. A. Y. Bazi, N. Alajlan, F. Melgani and R. R. Yager, “Deep learning approach for active classification of electrocardiogram signals,” *Information science*, vol. 345, pp. 340-354, 2016.
- [12] P. Kora and K. Krishna, “ECG based heart arrhythmia detection using wavelet coherence and bat algorithm,” *Sensing and Imaging*, vol. 17, pp. 1-16, 2016.
- [13] S. Chauhan and L. Vig, “Anomaly detection in ECG time signals via deep long short-term memory networks,” in *Proc. IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015.
- [14] R. K. Rasmussen, “Patient identification based on unfiltered ECG using Hierarchical Temporal Memory,” *Biomedical Engineering and Informatics*, 2017.
- [15] M. Chuah and F. Fu, “ECG anomaly detection via time series analysis,” in *Proc. Int. Symp. Parallel and Distributed Proc. and Appl.*, 2007.
- [16] I. A. Dmitrievich, “Deep Learning in information analysis of electrocardiogram signals for disease diagnostics,” *The Ministry of Education and Science of The Russian Federation Moscow Institute of Physics and Technology*, 2015.
- [17] M. Seera, C. P. Lim, W. S. Liew, E. Lim and a. C. K. Loo, “Classification of electrocardiogram and auscultatory blood pressure signals using machine learning models,” *Expert Systems with Applications*, vol. 42, p. 3643–3652, 2015.
- [18] S. Enríquez and H. Ysaías, “Anomaly detection in streaming multivariate time series,” *phD thesis in science computation, university of chile, faculty of physical sciences and mathematics*, 2017.

- [19] “Numenta, NuPIC 0.7.0.dev API Documentation,” 2017. [Online]. Available: <http://nupic.docs.numenta.org/>.
- [20] J. Hawkins, S. Ahmad and D. Dubinsky, “Hierarchical temporal memory including HTM cortical learning algorithms,” *Palo Alto, CA:Numenta, Inc*, 2010.
- [21] D. George and B. Jaros, “The htm learning algorithms. Technical report, Numenta,” 2007. [Online]. Available: http://www.numenta.com/for-developers/education/ Numenta_HTM_Learning_Algos.pdf.
- [22] J. Casarella, “The Classification of Un-preprocessed ECG Waveforms through the Application of the Hierarchical Temporal Memory Model,” *Doctor thesis, Pace University*, 2012.
- [23] H. Sedghamiz and D. Santonocito, “Unsupervised detection and classification of motor unit action potentials in intramuscular electromyography signals,” in *Proc. 5th IEEE Int. Conf. E-Health Bioeng. (EHB)*, 2015.
- [24] Z. Nenadic and J. W. Burdick, “Spike detection using the continuous wavelet transform,” *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 74-87, 2005.
- [25] Pattern recognition basics I: time domain methods, [Online]. Available: http://pub.ist.ac.at/~schloegl/lv/biosig/biosig2005_chp5.pdf.
- [26] Hole K.J., “The HTM Learning Algorithm. In: Anti-fragile ICT Systems,” *Simula SpringerBriefs on Computing, Springer*, vol. 1, 2016.
- [27] S. Purdy, “Encoding data for HTM systems,” *arXiv.1602.05925*, 2016.
- [28] Normal Electrocardiography (ECG) Intervals, Buck Christensen. [Online]. Available: <https://emedicine.medscape.com/article/2172196-overview>.
- [29] Y. Cui, S. Ahmad and J. Hawkins, “Continuous online sequence learning with an unsupervised neural network model,” *Neural Comput*, vol. 28, pp. 2474-2504, 2016.
- [30] Y. Cui, S. Ahmad and J. Hawkins, “The HTM spatial Pooler: A neocortical algorithm for online sparse distributed coding,” *bioRxiv*, 2017.
- [31] AL. Goldberger, L. Amaral, Glass, JM. Hausdorff, P.Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, H. Stanley, “PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” *Circulation*, 2000.
- [32] Ting K.M., “Confusion matrix,” *Encyclopedia of Machine Learning and Data Mining, Springer*, 2017.
- [33] Batra, A. and Jawa, V., “Classification of arrhythmia using conjunction of machine learning algorithms and ECG diagnostic criteria,” *International Journal of Biology and Biomedicine*, vol.1.1, pp.1-7, 2016.
- [34] Pereira, J. and Silveira, M., “Unsupervised representation learning and anomaly detection in ECG sequences,” *International Journal of Data Mining and Bioinformatics*, vol. 22, pp.389–407, 2019.
- [35] Nasiri J A, Naghibzadeh M., Yazdi H. S., and Naghibzadeh B., “ECG Arrhythmia classification with support vector machines and genetic algorithm,” *Third UKSim European Symposium on Computer Modeling and Simulatio*, 2009.
- [36] Karim F., Majumdar S., Darabi H., and Chen S., “LSTM fully convolutional networks for time series classification,” *CoRR, abs/1709.05206*, 2017.

- [37] Malhotra P., Ramakrishnan A., Anand G., Vig L., Agarwal P., and Shroff G., "LSTM-based encoder-decoder for multi-sensor anomaly detection," *33rd International Conference on Machine Learning*, 2016.
- [38] Ceylan R., Ozbay, Y. and Karlik, B., "A novel approach for classification of ECG arrhythmias: Type-2 fuzzy clustering neural network," *Expert Systems with Applications*, vol.36, pp.6721-6726, 2009.