Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/87/20241569

The survey and discussion of research on heart disease prediction based on Apache Spark

Junyu Hu

The Department of Artificial Intelligence, South China Normal University, Foshan, 528225, China

20224001049@m.scnu.edu.cn

Abstract. Heart is the most important part of the human body, diseases that are related to heart cause a huge thread to human health. In this paper, methods that applied Apache Spark to heart disease related works would be shown and discussed in order to classify these methods and make a conclusion about the innovations and shortcomings of these works. These works are defined into two categories: the ones that adopted traditional machine learning method and the ones that used deep learning methods. By classifying these works into two types, commonalities and similar innovative approaches in the same category of methods can be better observed and summarized, facilitating a clearer comparison of the similarities and differences in the innovative focuses among similar yet distinct methods. By doing so, conclusions were made to show that apart from enhancing operational efficiency and reliability of models for diagnosing and treating heart diseases, current research utilizing Apache Spark in this field also identifies areas for improvement such as expanding sample data representation, speeding up data processing, and addressing concept drift issues with proposed solutions. By addressing these challenges, researchers aim to optimize existing methods using Apache Spark and advanced data analytics techniques to combat heart diseases.

Keywords: Apache Spark, machine learning, deep learning, monitoring and prediction of heart disease, electrocardiogram analysis.

1. Introduction

As one of the most significant organs of the human body, the heart is like an engine of a car, it delivers oxygen and other nutrition that other organs need to keep functioning to every corner of the body through blood. Heart disease is hard to be diagnosed when it doesn't seizure while its seizures suddenly without an obvious omen, making it hard to prevent. Therefore, diseases that are related to hearts have now become the number one killer of human health: About 695,000 people died from heart disease in 2021—that's 1 in every 5 deaths. Heart disease costs about $239.9 billion each year from 2018 to 2019 [1].

Luckily, there have been new ways to prevent heart disease. By applying wearable medical devices and other devices, medical workers are now able to monitor patients' real-time situation of their hearts, which allows doctors to react timely once heart disease occurs. This method will surely help in the problems that the suddenness of heart disease brings, but it costs too much labor costs for a medical worker to keep monitoring the patient's heart. Which is why applying artificial intelligence into

DOI: 10.54254/2755-2721/87/20241569

monitoring and predicting heart disease is brought up. To achieve this goal, the program should be able to measure real-time heart rate with low latency, and it should have the ability to resist interference since there will be inevitable interference in real life usage. Moreover, the huge amount of data for real-time monitoring demands a program that has a strong ability to deal with big data.

Apache Spark, as a fast and universal computing engine designed specifically for large-scale data processing, meets these requests. Apart from that, it has been used in many other practical fields such as text mining、human face recognition and so on, thus proving its practicality. Therefore, many researchers have applied Apache Spark on heart disease area and have focused on different aspects.

In this article, the following works would be mentioned and analyzed. Ilbeigipour et al. [2] presents the implementation of a machine learning pipeline for real-time heart arrhythmia detection using a structured streaming module built on the open-source Apache Spark platform. The study evaluates the impact of employing this new module on classification performance metrics and the latency rate of heart arrhythmia detection. Alarsan et al. [3] proposes an Electrocardiogram (ECG) classification method using machine learning on Apache Spark, addressing irregularities in ECG signals. Implemented in Scala with MLlib, it achieved high accuracy using algorithms like Gradient-Boosted Trees and Random Forests, leveraging Spark's scalability for processing large datasets. The approach was validated on MIT-BIH Arrhythmia and Supraventricular Arrhythmia databases, demonstrating efficient ECG signal classification. Carnevale et al. [4] proposed a tool based on Apache Spark and the Menard algorithm for handling this electrocardiogram data. To validate their solution, they conducted a series of experiments, implementing the algorithm to detect heart diseases. The experimental results demonstrated the superiority of their approach in terms of performance. Abdullah, et al. [5] proposes a real-time heart rate prediction system based on Apache Spark. By integrating Apache Kafka and Apache Spark, the online phase predicts heart rate in advance using the best model, aiding healthcare providers and patients in real-time avoidance of heart rate risks. With the hope of providing categorized references for relevant researchers, this article will provide a review of the relevant studies on Spark in the field of cardiovascular diseases, aiming to offer insights for fellow professionals.

This paper will investigate the following aspects: firstly, it will present research related to heart disease using Spark with machine learning and deep learning. Subsequently, the paper will analyze and discuss the mentioned research efforts to identify their innovations and strengths. Following this, a summary of these works will be provided. Finally, the paper will conclude with a review of the application of Apache Spark in the field of heart disease.

2. Methods

2.1. Introduction of Spark

Spark is a general-purpose big data processing framework. Similar to traditional big data technologies like Hadoop's MapReduce, Hive engine, and the Storm real-time streaming engine, Spark encompasses various common computing frameworks in the big data field. The working mechanism involves a master-slave architecture where the master, called the driver, coordinates the execution of tasks on worker nodes. When a user submits an application, the driver obtains resources from the cluster manager and then divides the tasks into smaller units of work called tasks. These tasks are then dispatched to the worker nodes for parallel execution. Spark employs in-memory computation, which enhances processing speed by caching intermediate results. It operates on Resilient Distributed Datasets (RDDs), in-memory data structures allowing fault-tolerant distributed processing of large datasets. The Directed Acyclic Graph (DAG) scheduler optimizes task execution, while the use of lazy evaluation minimizes unnecessary computations. These principles—distributed task execution, in-memory computation, RDDs, and optimized task scheduling—form the basis of Spark's operational framework and underpin its efficiency in processing large-scale data.

DOI: 10.54254/2755-2721/87/20241569

2.2. Traditional machine learning methods

2.2.1. Heart arrhythmia detection

The study developed a real-time pipeline for atrial fibrillation and RBBB (Right Bundle Branch Block) arrhythmia detection using ECG (electrocardiogram) signals [2]. It employed online segmentation and feature extraction, with random forest classification shown in Figure 1 [2]. Data from MIT/BIH database were preprocessed for noise removal, R-peak detection, and feature extraction. An Apache Spark pipeline with Pandas-UDF was implemented for data preprocessing. The pipeline used Spark structured streaming for real-time processing.

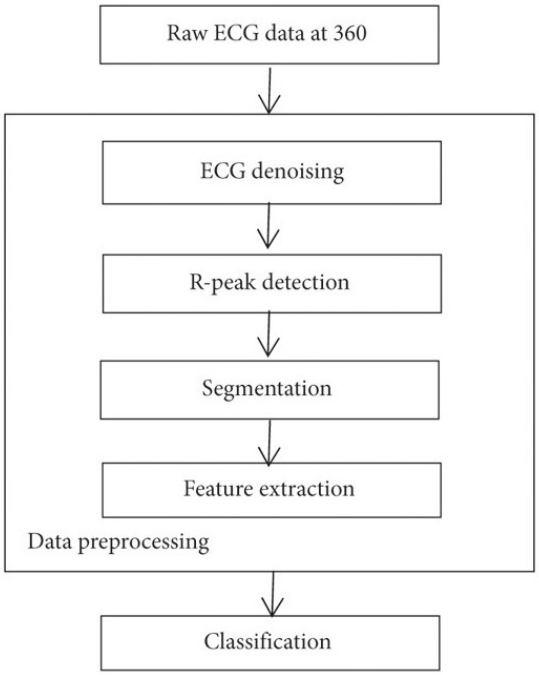

Figure 1. Block diagram of data preprocessing and classification [2].

The classification process involves learning from training tuples to describe class labels and evaluating the model for classification. Various classifiers like decision trees and random forests are trained, leveraging Apache Spark's functionalities for real-time ECG classification. Ilbeigipour et al. innovatively developed an online pipeline integrating preprocessing and classification steps on Spark structured streaming. This approach reduces latency in arrhythmia diagnosis using parallel computing on big data platforms. It also enhances multi-class classification performance and can incorporate static patient data for result reliability.

2.2.2. Heart diseases classification

This study [3] proposes an ECG classification method using Apache Spark, MLlib, and Scala, achieving high-precision signal classification. Evaluated on MIT-BIH Arrhythmia and Supraventricular Arrhythmia databases, it utilizes Discrete Wavelet Transform for feature extraction. Three feature types (Summits, Temporal, Morphological) are categorized. Spark is employed for feature processing.

Due to the large size of the dataset in this case, and the necessity to implement decision trees, random forests, and gradient boosting trees, machine learning algorithms were not easily implementable in Matlab due to performance concerns. Hence, Spark-shell and Scala were utilized on a local host PC. Following the data processing, the authors employed Gradient-Boosted Trees (GBT) and Random Forest (RF) models for machine learning, achieving classification accuracies of 96.75%

DOI: 10.54254/2755-2721/87/20241569

and 97.98%, respectively. The GBT and RF models developed in this study are capable of classifying various types of ECG heartbeats, thus enabling their implementation in CAD ECG systems for fast and reliable diagnosis.

2.2.3. Heart disorder detection

In this paper, Carnevale et al. [4] discuss a tool utilizing the Menard algorithm based on Apache Spark. The core idea of the Menard algorithm is to locate specific peaks (such as QRS peaks) within the ECG signal and determine their positions. Regarding the dataset, the authors utilized the European ST-T database. In the proposed method, the Menard algorithm has been employed for calculating the QRS complex using Apache Spark. During the preprocessing stage, it's necessary to handle multiple samples simultaneously, requiring the organization of a suitable set of samples on a file line because Apache Spark treats each sample as a string RDD. Additionally, Spark distributes the workload across tasks involving the processing of multiple lines of RDDs. The implementation utilizes a set of samples with a duration of 10 seconds, where the only information required for computing the QRS complex is represented by detected peak indices. This scientific work addresses the issue of distributed processing of electrocardiogram (ECG) signals and it resolves the issue of local preprocessing of ECG signals. Apache Spark was utilized as a tool for extensive data preprocessing in this study, enhancing the efficiency of preprocessing.

2.3. Deep learning methods

2.3.1. Prediction for heart rate

In this study, Alharbi et al. [5] proposed a real-time heart rate prediction system for preemptive heart risk avoidance [5] shown in Figure 2. The system comprises two stages: an offline stage and an online stage. The objective of the offline stage is to develop models using various prediction techniques to minimize the root mean square error. In the online stage, Apache Kafka and Apache Spark are employed to predict heart rates in advance based on the best-developed model.

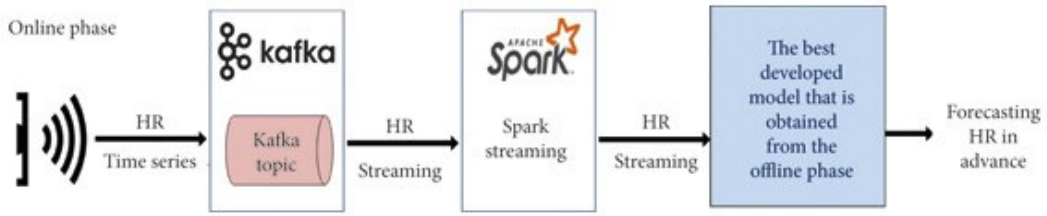

Figure 2. The architecture of HR forecasting system [5].

In terms of dataset, this study utilized the open healthcare dataset called Medical Information Mart for Intensive Care (MIMIC-II). From this dataset, a patient's heart rate time series (univariate dataset) was extracted on a per-minute basis. In the preprocessing stage, the authors first converted the raw data into fixed data, then transformed it into supervised learning format, and finally augmented the data. For model training, four deep learning models were employed for heart rate prediction: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BI-LSTM), and Gated Recurrent Unit (GRU). These steps belong to the offline stage. In the online stage, Apache Kafka and Apache Spark were utilized as the streaming processing platforms. Apache Spark Streaming API was used to process medical data (i.e., HR streams) from pre-created Kafka topics.

3. Discussion

Although many innovative and effective methods have been proposed shown in the method section, there are still limitations and areas for improvement in the approaches mentioned.

DOI: 10.54254/2755-2721/87/20241569

In the context of Heart Arrhythmia Detection, as noted by the authors, there is room for improvement in enhancing the reliability of patient outcomes by incorporating diverse features into the data. The methods mentioned in the paper also suffer from the high computational complexity associated with newer techniques such as deep neural networks, which have not yet been utilized. Additionally, the issue of concept drift due to variations in data caused by factors like noise remains unaddressed. This phenomenon can impact the effectiveness of new features in online learning and potentially affect the accuracy of results. The methods outlined in the paper still do not adequately handle concept drift in real-world scenarios, highlighting the need for a solution to address this challenge.

In Heart diseases classification, optimizing feature selection and engineering processes can further enhance the performance of classification algorithms. As data volume increases, there is a need for further optimization of algorithms to improve computational efficiency and scalability, particularly for real-time applications. Moreover, since this experiment has not been validated in real-world scenarios, the effectiveness of these models across different datasets or in practical applications remains to be further verified. Lastly, similar to what was mentioned in Heart Arrhythmia Detection, ECG data may experience concept drift due to changes in patient conditions, equipment variations, or other factors, which could impact model performance. Future work could explore better methods to handle concept drift, such as online learning or dynamic model updating.

As for the Heart Disorder Detection, there would be potential challenges in local preprocessing of large files, especially when handling extensive datasets, which could constrain the application's performance. To address this, leveraging a distributed file system for data processing with Spark is recommended to boost efficiency and speed, thereby circumventing single-point bottlenecks. Additionally, while the R-R interval method is commonly used for heart rhythm analysis, it may not comprehensively detect all types of arrhythmias present in complex ECG signals. Therefore, enhancing the analysis by integrating additional signal features and employing deep learning models can effectively identify more intricate arrhythmia patterns. This approach not only improves the accuracy of arrhythmia detection but also expands the scope of analysis to include more nuanced cardiac abnormalities. By combining distributed computing capabilities with advanced analytical techniques, such as deep learning, the article proposes a robust framework for enhancing the precision and scalability of heart disorder detection systems based on ECG data processing.

In terms of the Prediction for Heart Rate section cited from the paper, there are still opportunities for improvement in the stability of the real-time prediction system. During the online phase, simulated sensor-generated heart rate time-series data are sent to a Kafka topic and then processed through Spark streaming before being fed into the best model. However, this architecture may encounter delays or insufficient processing capacity when dealing with large-scale or high-frequency data. Enhancements could involve optimizing the data transmission and processing architecture, such as introducing higher-performance data streaming systems or scaling up server resources. Additionally, the paper does not address the system's capability to handle concept drift, a challenging issue often encountered in real-world scenarios that could affect model performance. Future work could explore better methods to handle concept drift effectively.

In summary, current applications of Apache Spark in conjunction with heart disease detection can benefit from several improvements. Firstly, increasing data diversity by incorporating diverse features can enhance model reliability [6, 7]. Secondly, addressing concept drift challenges more effectively in real-world applications remains a priority for further optimization [8-10]. Lastly, enhancing the performance of classification algorithms can improve computational efficiency and scalability of the algorithms.

4. Conclusion

In this article, the primary focus lies in exploring the application of Apache Spark within the field of cardiology. It delves into various studies that employ both machine learning and deep learning techniques. These research efforts are predominantly geared towards enhancing the operational

DOI: 10.54254/2755-2721/87/20241569

efficiency and reliability of models used in diagnosing and treating heart diseases, each contributing its own unique innovations. Despite the significant strides made in these areas, there remain several avenues for improvement. For example, there is a pressing need to diversify sample data to better represent diverse patient populations. Additionally, improving the speed of data processing could further streamline diagnostic processes and treatment planning. Moreover, researchers have highlighted the concept drift issues that models encounter in real-world applications, underscoring the need for more robust solutions to maintain model accuracy over time. Looking ahead, future research in this domain could benefit from focusing on these areas for enhancement. By addressing these challenges, researchers can potentially refine existing methodologies and develop more effective tools and models for combating heart diseases using Apache Spark and advanced data analytics techniques.

References

[1] CDC 2024 Heart Disease Facts https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html

[2] Ilbeigipour S et al. 2021 Real-Time Heart Arrhythmia Detection Using Apache Spark Structured Streaming Journal of Healthcare Engineering vol 2021 (1) p 6624829

[3] Alarsan F I & Younes M 2019 Analysis and classification of heart diseases using heartbeat features and machine learning algorithms Journal of big data vol 6 (1) pp 1-15

[4] Carnevale L et al. 2017 Heart disorder detection with menard algorithm on apache spark Service-Oriented and Cloud Computing: 6th IFIP WG 2.14 European Conference ESOCC 2017 (Oslo: Springer International Publishing) p 6

[5] Alharbi A et al. 2021 Real-Time System Prediction for Heart Rate Using Deep Learning and Stream Processing Platforms Complexity vol 2021 (1) p 5535734

[6] Nguyen-Tang T & Arora R 2024 On sample-efficient offline reinforcement learning: Data diversity posterior sampling and beyond Advances in neural information processing systems vol 36 Feb 13

[7] Qiu Y Hui Y Zhao P Cai CH Dai B Dou J Bhattacharya S & Yu J 2024 A novel image expression-driven modeling strategy for coke quality prediction in the smart cokemaking process Energy vol 294 May 1 p 130866

[8] Arora S Rani R & Saxena N 2024 A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery Mar 19 e1536

[9] Fernando DW & Komninos N 2024 FeSAD ransomware detection framework with machine learning using adaption to concept drift Computers & Security Feb 1 vol 137 p 103629

[10] Ali Abdu NA & Basulaim KO 2024 Machine learning in concept drift detection using statistical measures International Journal of Computers and Applications May 3 vol 46 (5) pp 281-291