PAPER NAME

Automatic cardiac Arrhythmia classification.docx

AUTHOR

Deepak Mustare

WORD COUNT

4866 Words

CHARACTER COUNT

28381 Characters

PAGE COUNT

37 Pages

FILE SIZE

2.9MB

SUBMISSION DATE

May 31, 2023 1:04 PM GMT+5:30

REPORT DATE

May 31, 2023 1:05 PM GMT+5:30

● 28% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 20% Internet database
- Crossref database
- 8% Submitted Works database
- 5% Publications database
- Crossref Posted Content database

A PROJECT REPORT ON

"Automatic cardiac Arrhythmia classification using SVM"

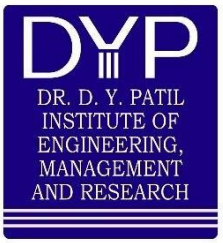
SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Siddhesh Chavan	71922488J
Nikita Sonar	71922843B
Deepak Mustare	72036761B
Ankita Thorat	72036943G



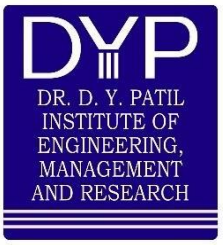
DEPARTMENT OF COMPUTER ENGINEERING

DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT & RESEARCH

AKURDI, PUNE 411044

SAVITRIBAI PHULE PUNE UNIVERSITY

2022 -2023



CERTIFICATE

This is to certify that the project report entitles

"Automatic cardiac Arrhythmia classification using SVM"

Submitted by

Siddhesh Chavan	71922488J
Nikita Sonar	71922843B
Deepak Mustare	72036761B
Ankita Thorat	72036943G

are bonafide students of this institute and the work has been carried out by him/her under the supervision of **Prof. Pooja Mishra** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Prof. Pooja Mishra
Guide
Computer Engineering

Prof. P. P. Shevatekar
Head of the Department
ComputerEngineering

Dr. Anupama V. Patil
Principal,
Dr. D. Y. Patil Institute of Engineering, Management & Research Akurdi Pune – 41

Place : Pune
Date :

External Examiner's Name:
Sign:

ACKNOWLEDGEMENT

This is a great pleasure and immense satisfaction to express our deepest sense of gratitude and thanks to everyone who has directly or indirectly helped us in completing our project work successfully.

We express my gratitude towards Project Guide **Prof. Pooja Mishra** and **Mrs. P. P. Shevatekar**, Head of Department of Computer Engineering, Dr. D Y Patil Institute of Engineering, Management and Research, Akurdi who guided and encouraged us in completing the seminar work in scheduled time. We would like to thanks our Principal Dr. A. V. Patil, for allowing us to pursue our project in this institute.

Finally, we would like to thank our friends who have directly or indirectly helped us in our project work.

Siddhesh Chavan
Nikita Sonar
Deepak Mustare
Ankita Thorat
(B.E. COMPUTER ENG.)

ABSTRACT

ECG plays a vital role in diagnosis of many such cardiac disorders. Some arrhythmias including ventricular fibrillation and premature ventricular contraction can be fatal if not dealt on time Ventricular and atrial tachycardias are some of the most common clinical cardiac arrhythmias. Therefore, accurate automated arrhythmia detection systems for various types of arrhythmias are required in clinical settings. Recently, deep learning has been actively used for arrhythmia diagnosis, resulting in significant performance improvements. In this study, we classify Cardiac Arrythmia using Support-Vector machine. Cardiac arrhythmias occur in a short duration; therefore, it is difficult to distinguish them using the human eye directly. Cardiac arrhythmia is associated with abnormal electrical activities of the heart, which can be reflected by altered characteristics of electrocardiogram (ECG). Due to the simplicity and noninvasive nature, the ECG has been widely used for detecting arrhythmias and there is an urgent need for automatic ECG detection. We used the Support Vector Machine (SVM) classifier and the best result we chose had a global accuracy greater 80%, while still obtaining relatively high Sensitivities and Positive Predictive Value and low False Positive Rates. Thus, this method presents a noble, accurate, and reliable method for classification of cardiac arrhythmia beats.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
HTTP	Hyper Text Transfer Protocol
MQTT	Message Query Telemetry Transport

LIST OF FIGURES

1. INTRODUCTION

1.1. OVERVIEW

Different cardiologists hold varying perspectives on the differentiation of arrhythmias. Consequently, there is a need for precise automated systems capable of detecting various types of arrhythmias in clinical environments. Recently, deep learning has emerged as a prominent approach for diagnosing arrhythmias, leading to notable advancements in performance. However, challenges persist when it comes to implementing these systems practically. One key challenge involves extracting distinctive features specific to arrhythmias from electrocardiogram (ECG) signals using deep learning algorithms. ECG signals exhibit unique morphological and temporal characteristics, resulting in diverse ECG morphologies for a single arrhythmia across different patients. Consequently, extracting distinctive ECG features for specific cardiac conditions using deep learning poses a significant challenge.

By successfully completing this project, a reliable and user-friendly system is created that can aid in the classification of cardiac arrhythmias using an SVM algorithm. The Tkinter GUI enhances the usability and accessibility of the system, allowing healthcare professionals or users to input data and obtain real-time predictions conveniently.

1.2. MOTIVATION

The motivation behind the project stems from the need to improve the accuracy and efficiency of diagnosing cardiac arrhythmias. Cardiac arrhythmias are abnormal heart rhythms that can have serious health implications and may require timely intervention for proper treatment. However, accurately classifying different types of arrhythmias can be challenging for medical professionals due to the complexity and variability of these conditions.

Overall, the project's motivation lies in leveraging machine learning techniques, specifically SVM, and GUI development to create an automated classification system that improves the accuracy, efficiency, and consistency of diagnosing cardiac arrhythmias. By addressing these aspects, the project aims to enhance patient care, support medical professionals, and potentially save lives through timely and accurate arrhythmia identification.

1.3. PROBLEM DEFINATION

The project aims to develop a system that can accurately classify different types of cardiac arrhythmias based on input data, using a Support Vector Machine (SVM) algorithm. The system will be integrated with a graphical user interface (GUI) built using Tkinter, allowing users to conveniently input data and receive real-time predictions.

By addressing these aspects, the project aims to provide a reliable and user-friendly system for accurate classification of cardiac arrhythmias using SVM. The integration of Tkinter GUI enhances the usability and accessibility of the system, allowing healthcare professionals to conveniently input data and receive real-time predictions, aiding in the diagnosis and treatment of cardiac arrhythmias.

1.4. LIMITATIONS

- Users must have knowledge of English.
- Users must have all required software to run the application.
- It requires internet connection.
- Dependency on Data Quality:
- Limited Generalization
- Interpretable Feature Extraction
- Complexity of Arrhythmia Patterns

It is important to acknowledge these limitations and conduct thorough analysis, validation, and continuous improvement of the system to mitigate these challenges and ensure its safe and effective use in a clinical setting.

1.5. METHODOLOGIES OF PROBLEM SOLVING

To approach problem-solving we followed these methodologies:

1. Define the Problem: Clearly understand the problem statement and the goals of the project. Identify the specific requirements and constraints, such as the expected accuracy of classification, the types of arrhythmias to be classified, and the features to be used.

2. Gather Information: Gather relevant information about cardiac arrhythmia, SVM classification, and GUI development using Tkinter. Understand the basics of ECG data, arrhythmia classification methods, and the SVM algorithm. Explore the functionalities and capabilities of Tkinter for building GUI applications.

3. Break down the Problem: Break down the problem into smaller, manageable tasks. Identify the steps involved, such as data preprocessing, SVM model training, GUI design, and result display.

4. Design the Solution: Design a solution by determining the appropriate data preprocessing techniques, SVM model parameters, and GUI layout. Decide on the input fields, buttons, and result display mechanisms in the Tkinter GUI. Consider the user experience and ease of use.

5. Implement the Solution: Implement the solution by writing the code. Develop the data preprocessing functions, SVM classification algorithms, and Tkinter GUI components based on the defined design. Test each component independently to ensure they work as expected.

6. Integrate and Test: Integrate the individual components, such as data preprocessing, SVM classification, and Tkinter GUI, to create a complete system. Test the integrated system thoroughly to verify its functionality, accuracy, and performance. Write unit tests to validate the correctness of the implementation.

7. Debug and Refine: Identify and fix any bugs or issues that arise during testing. Refine the code and algorithms to improve the accuracy and usability of the classification system. Consider feedback from users or stakeholders to enhance the GUI interface and overall user experience.

8. Validate and Evaluate: Validate the classification system by comparing its results with known ground truth or expert diagnoses. Evaluate the system's performance using appropriate metrics, such as accuracy, precision, recall, or F1 score. Assess the usability and effectiveness of the Tkinter GUI based on user feedback.

9. Document and Communicate: Document the entire process, including the problem statement, methodologies, implementation details, and evaluation results. Create clear and concise documentation for future reference. Communicate the findings, limitations, and recommendations to stakeholders or peers involved in the project.

10. Iterate and Improve: Based on the evaluation and feedback, iterate on the solution to improve its performance, accuracy, and user experience. Consider incorporating additional features or techniques to enhance the classification system. Continuously refine and update the system as needed.

2. LITERATUREN SURVEY

Sr. No	Paper Title	Journal Name	Authors & Publication Date	Methodology
1	A Telemetry System for the Study of Spontaneous Cardiac Arrhythmias	IEEE	Dennis L. Rollins, Cheryl R. Killingsworth, Gregory P. Walcott, R. Kyle Justice	To study arrhythmias in animal models, an eightchannel implantable radio telemetry system has been developed to record continuously cardiac electrograms over a period of weeks to months, with maintenance restricted to changing batteries.
2	Comparison of Arrhythmia Prevalence in NUVANT Mobile Cardiac Telemetry System Patients in the US and India.	IEEE	Jonathan M. Engel, Niranjan Chakravarthy, Grant Nosbush, Matt Merkert, Richard Fogoros M.D., Abhi Chavan	Analysis of a sample of 2231 US and 1053 Indian NUVANT patients has revealed interesting statistics on the prevalence of various cardiac arrhythmias in the patient populations of the two nations. The population is non-randomized and consists of US patients where often a traditional Holter Monitor study was negative and self-pay Indian patients.
3	Classifying Normal Sinus Rhythm and Cardiac Arrhythmias in ECG Signals Using Statistical Features in Temporal Domain	IEEE	Mavera Mazharbhatt, Usman Akram, Shoab A. Khan	Any morphological abnormality or atypical group of conditions in a cardiac rhythm indicates a typical class of arrhythmias. ECG plays a vital role in diagnosis of many such cardiac disorders. Some arrhythmias including ventricular fibrillation and premature ventricular contraction can be fatal if not dealt on time.

4	Cardiac propagation Pattern Mapping with Vector Field for Helping Tachyarrhythmias Diagnosis with Clinical Tridimensional Electro Anatomical Mapping Tools	IEEE	Corentin Dallet, Caroline Roney, Ruairidh Martin, Takeshi Kitamura, Stephane Puyo, Josselin Duchateau, Carole DumasPomier, Gwladys Ravon	Invasive electro-anatomical mapping for an accurate diagnosis using electrograms (EGMs) acquired with intra-cardiac catheters and localized on the surface mesh of the studied cavities and non-invasive electrocardiographic imaging (ECGi) for a global view of the arrhythmia, with EGMs mathematically reconstructed from body surface electrocardiograms using 3-D cardio-thoracic surface meshes obtained from CT-scans
5	A Wearable Auto-Patient Adaptive ECG Processor for Shockable Cardiac Arrhythmia	IEEE	Syed Muhammad Abubakar, Muhammad Rizwan Khan, Wala Saadeh, Muhammad Awais Bin Altaf.	A non-machine learning patient-specific shockable cardiac arrhythmia-(SCA) classification processor based on single lead electrocardiogram (ECG)is presented. The proposed SCA detection processor integrates a hardware efficient reduced-set-of-five (RSF5) Computer Engineering department, DYPIEMR,‘22-’23 14 feature extraction engine to extract SCA and non-SCA, self-adaptive patient-specific threshold engine for the peak and interval detection from the ECG, and simplified decision logic to discriminate the arrhythmia in real-time

3. SOFTWARE REQUIREMENT SEPCIFICATION

3.1. ASSUMPTIONS AND DEPENDENCIES

Input: Input as textual dataset

Domain: Machine Learning

3.2. FUNCTIONAL REQUIREMENTS

Proposed system consists of 4 modules:

- Feature point extraction: Feature points of each image gets detected.
- Feature correspondence matching: Matching of selected feature points across various image frames.
- Point estimation: Position estimation and vision system orientation during navigation.
- Position refinement: Location estimate based, accurate location derivation.

3.3. EXTERNAL INTERFACE REQUIREMENTS

3.3.1. USER INTERFACES

Application of Tkinter Framework to build a User-friendly UI.

3.3.2. HARDWARE INSTERFACES

- RAM: 8 GB (As we are using Machine Learning Algorithm and Various High-Level Libraries)
- Laptop RAM: minimum 8 GB.
- Hard Disk: 40 GB.
- Processor: Intel i5 Processor (Spyder ide environment is to be used and data loading should be fast hence fast processor is required)
- IDE: Spyder
- Coding Language: Python Version 3.5 (Highly specified Programming Language for Machine Learning because of availability of High-Performance Libraries.)
- Operating System: Windows 11 (Latest Operating System that supports all type of installation and development Environment)

3.3.3. SOFTWARE INTERFACES

- Operating System: Windows 11
- IDE: Spyder
- Programming Language: Python

3.4. NONFUNCTIONAL REQUIREMENTS

3.4.1. PERFORMANCE REQUIEMENTS

- The performance of the functions and every module must be well.
- The overall performance of the software will enable the users to work efficiently.
- Performance of encryption of data should be fast.
- Performance of the providing virtual environment should be fast Safety Requirement.
- The application is designed in modules where errors can be detected. This makes it easier to install and update new functionality if required.

3.4.2. SAFETY REQUIREMENTS

The application is designed in modules where errors can be detected and fixed easily.

This makes it easier to install and update new functionality if required.

3.4.3. SOFTWARE QUALITY ATTRIBUTE:

Our software has many quality attributes that are given below:

Adaptability: This software is adaptable by all users.

Availability: This software is freely available to all users. The availability of the software is easy for everyone.

Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

Reliability: The performance of the software is better which will increase the reliability of the Software.

User Friendliness: Since, the software is a GUI application; the output generated is much user friendly in its behavior.

Integrity: Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.

Security: Users are authenticated using many security phases so reliable security is provided.

Testability: The software will be tested considering all the aspects.

3.5. SYSTEM REQUIREMENTS

3.5.1 DATABASE REQUIREMENTS

- Browser for SQLite (DBMS) is a high quality, visual, open-source tool to create, design, and edit database files compatible with SQLite.

- DBMS is for users and developers who want to create, search, and edit databases.

- DBMS uses a familiar spreadsheet-like interface, and complicated SQL commands do not have to be learned.

- Controls and wizards are available for users to:
    - Create and compact database files
    - Create, define, modify, and delete tables
    - Create, define, and delete indexes o Browse, edit, add, and delete records o Search records o Import and export records as text
    - Import and export tables from/to CSV files
    - Import and export databases from/to SQL dump files
    - Issue SQL queries and inspect the results

3.5.2 SOFTWARE REQUIREMENTS

Anaconda Navigator: Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

- Operating System : Windows 11
- IDE : Spyder
- Programming Language : Python

3.5.3 HARDWARE REQUIREMENTS

RAM: 8 GB (As we are using Machine Learning Algorithm and Various High-Level Libraries)

- Laptop RAM: minimum 8 GB.
- Hard Disk: 40 GB.
- Processor: Intel i5 Processor (Spyder ide environment is to be used and data loading should be fast hence fast processor is required)
- IDE: Spyder
- Coding Language: Python Version 3.5 (Highly specified Programming Language for Machine Learning because of availability of High-Performance Libraries)
- Operating System: Windows 11 (Latest Operating System that supports all type of installation and development Environment)

3.6. ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

SDLC Models stands for Software Development Life Cycle Models. In this article, we explore the most widely used SDLC methodologies such as Agile. Each software development life cycle model starts with the analysis, in which the Also, here are defined the technologies used in the project, team load. One of the basic notions of the software development process is SDLC models which stands for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. There is no one single SDLC model. They are divided into main groups, each with its features and weaknesses.

1. Requirement Analysis - Requirement Analysis is the most important and necessary stage in SDLC. The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry. Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage. Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

2. System Design - The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

3. Implementation - In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers must follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

4. Testing - After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage. During this stage, unit testing, integration testing, system testing, acceptance testing is done.

5. Deployment - Once the software is certified, and no bugs or errors are stated, then it is deployed. Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment. After the software is deployed, then its maintenance begins.

6. Maintenance - Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time. This procedure where the care is taken for the developed product is known as maintenance.

4. SYSTEM DESIGN

4.1. SYSTEM ARCHICTURE

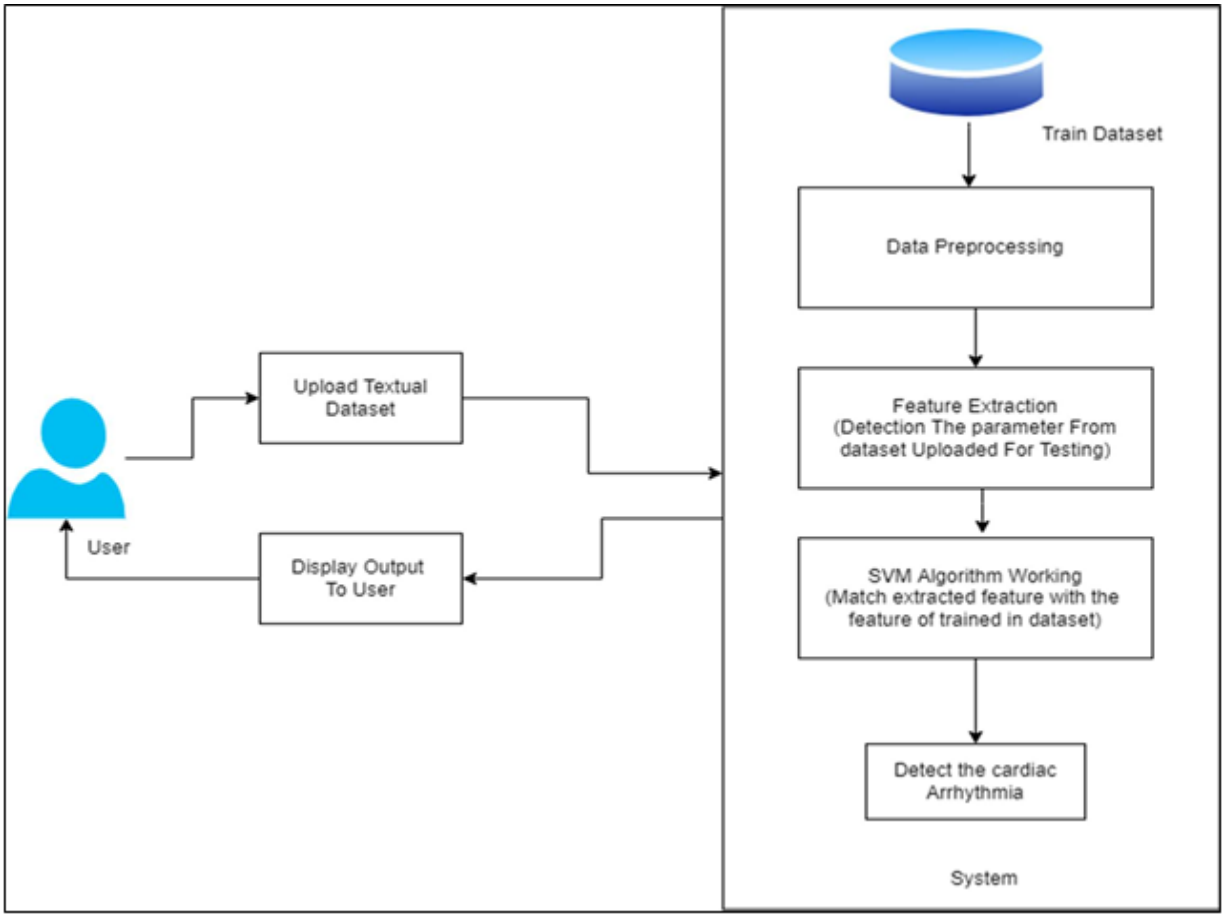
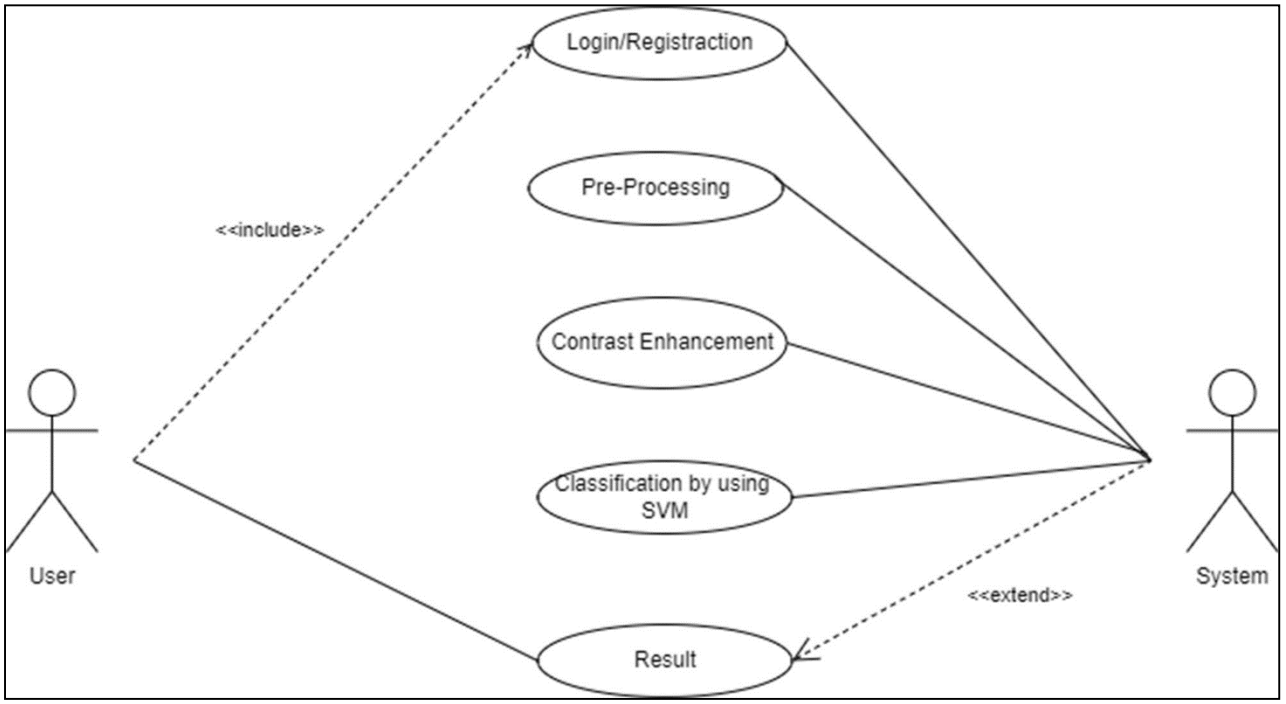


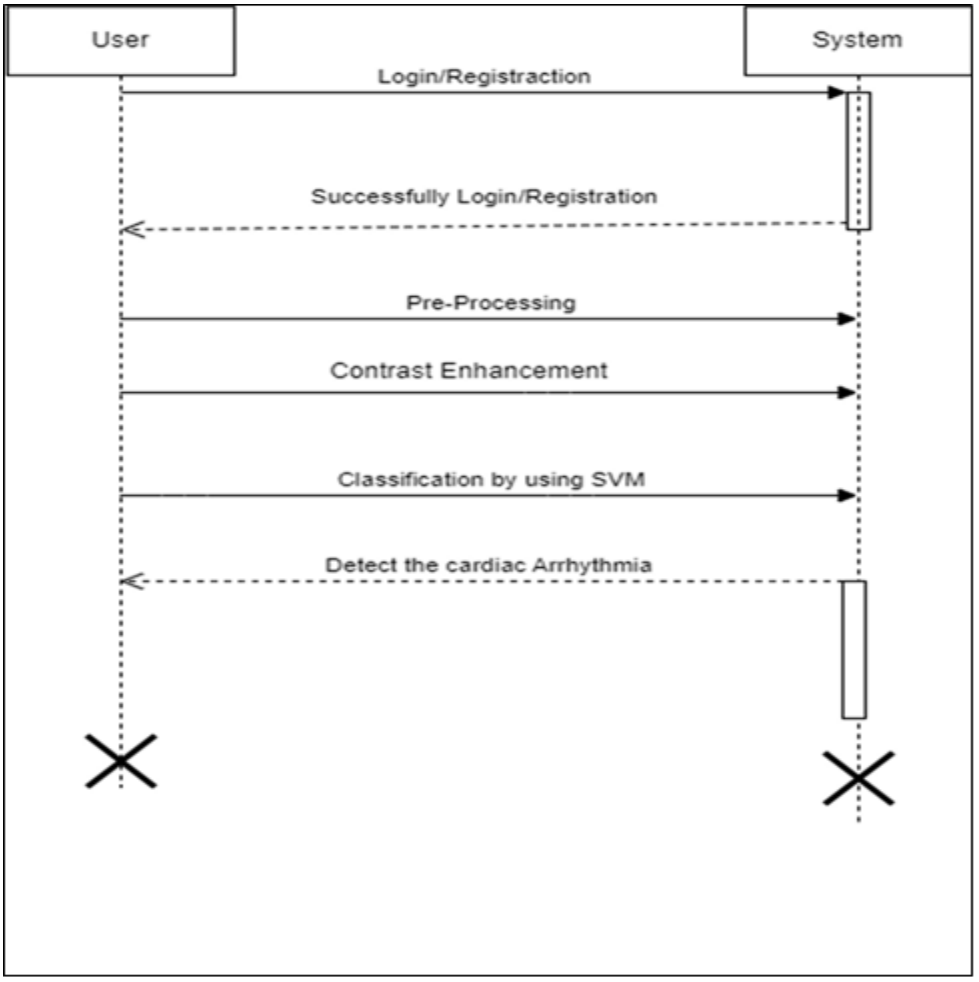
Fig. 1. System Architecture for Automatic Cardiac Arrythmia using SVM

4.2. UML DIAGRAMS

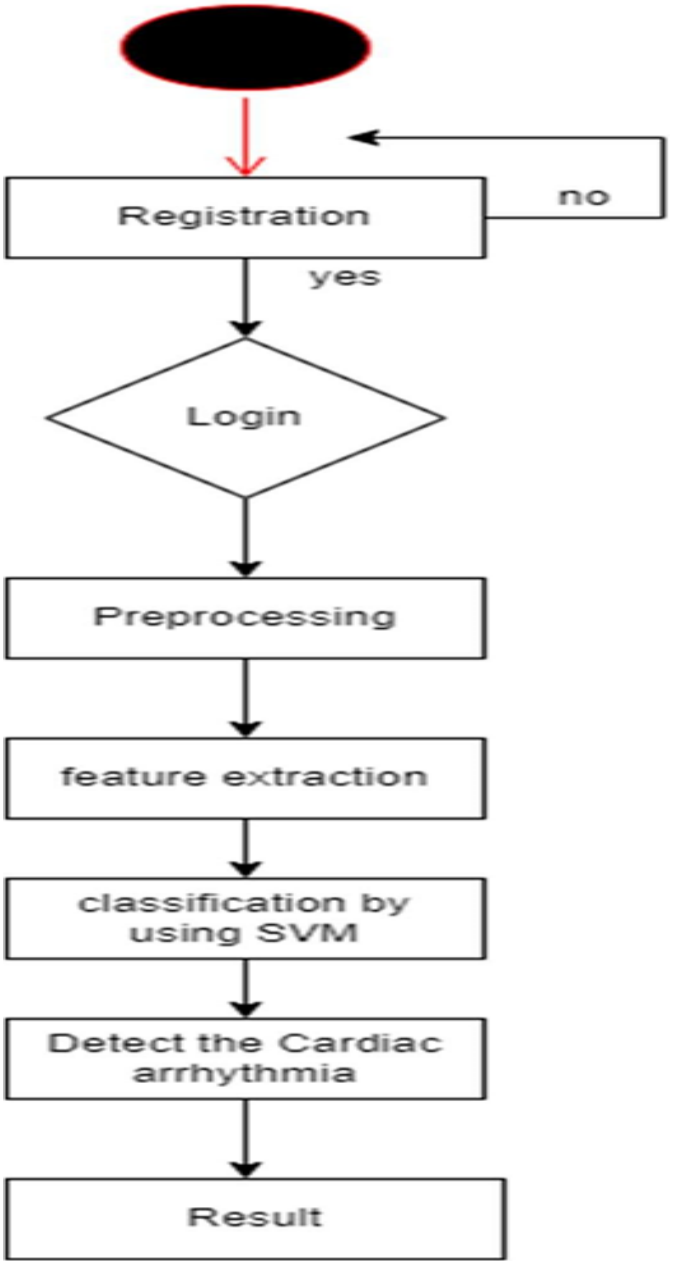
4.2.1. USE CASE DIAGRAM



4.1.1. SEQUENCE DIAGRAM



4.1.2. ACTIVITY DIAGRAM



5. OTHER SPECIFICATION

5.1 ADVANTAGES

The timely diagnosis of cardiac arrhythmias, determined by irregular and fast heart rate, may help lower the risk of strokes. Electrocardiogram signals have been widely used to identify arrhythmias due to their non-invasive approach.

5.2 LIMITATIONS

Lack of blood flow can damage the brain, heart and other organs. If irregular heartbeats are frequent or chronic, they can be very serious. Troublesome arrhythmias can be made worse, or can be caused by a weak or damaged heart. Uncontrolled rapid arrhythmias can result in heart weakness over time.

5.3 APPLICATIONS

Heart arrhythmia treatment may include medications, catheter procedures, implanted devices or surgery to control or eliminate fast, slow or irregular heartbeats. A heart healthy lifestyle can help prevent heart damage that can trigger certain heart arrhythmias.

6. PROJECT IMPLIMENTATION

6.1 OVERVIEW OF PROJECT MODULES

1. Data Collection: Obtain a dataset of cardiac arrhythmia cases, which may include electrocardiogram (ECG) recordings or other relevant data.

2. Data Preprocessing: Prepare the collected data for training and classification. This step may involve cleaning the data, removing noise, and addressing missing values or outliers. Additionally, feature extraction techniques can be applied to derive meaningful features from the raw data, such as extracting heart rate variability metrics or frequency domain features.

3. Feature Selection: Choose a subset of relevant features from the preprocessed data. This step helps in reducing the dimensionality of the data and removing any redundant or irrelevant features that may hinder the classification performance.

4. Feature Scaling: Normalize or scale the selected features to ensure that they have a similar range or distribution. Common techniques include z-score normalization or min-max scaling, which help prevent certain features from dominating the SVM training process due to their larger magnitudes.

5.Training/Test Split: Split the preprocessed and scaled data into training and testing sets. The training set is used to train the SVM model, while the testing set is used to evaluate its performance.

6. SVM Model Training: Utilize the training set to train an SVM classifier. SVMs aim to find an optimal hyperplane that separates the different classes of cardiac arrhythmia in the feature space. The choice of the kernel function (e.g., linear, polynomial, radial basis function) and tuning of hyperparameters (e.g., regularization parameter C, kernel parameters) can significantly impact the model's performance.

7. Model Evaluation: Assess the performance of the trained SVM model using the testing set. Common evaluation metrics for classification tasks include accuracy, precision, recall, F1 score, and receiver operating characteristic (ROC) curve analysis. These metrics provide insights into how well the model can classify different types of arrhythmias.

8. Model Optimization: Fine-tune the SVM model by adjusting hyperparameters or exploring different kernel functions to optimize its performance. Techniques like grid search or random search can be used to systematically search through the hyperparameter space and find the best combination.

9. Prediction: Apply the trained SVM model to classify new, unseen cases of cardiac arrhythmia. The model takes the relevant features of an input sample and predicts the corresponding arrhythmia class based on the learned decision boundary.

6.2 TOOLS AND TECHNOLOGIES USED

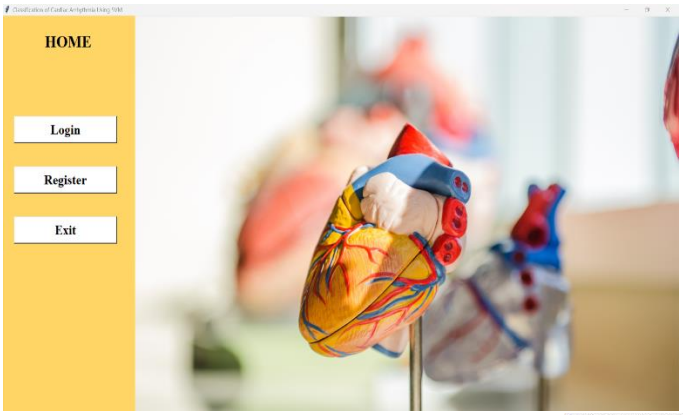
1. Programming Language: Python is a popular choice due to its extensive libraries for machine learning and data analysis.
2. Machine Learning Libraries: Various libraries in Python provide SVM implementations and other relevant functions. The most commonly used library for SVM is scikit-learn, which offers robust SVM models with different kernel functions and hyperparameter tuning options.
3. Data Analysis and Manipulation: Libraries such as NumPy and Pandas are commonly used for data manipulation, preprocessing, and feature engineering. They provide efficient data structures and functions to handle numerical computations and data handling tasks.
4. Data Visualization: Matplotlib and Seaborn are widely used libraries for data visualization in Python. They help create plots, charts, and graphs to visualize the data distribution, feature correlations, and classification results.
5. Feature Selection and Dimensionality Reduction: Scikit-learn provides various methods for feature selection and dimensionality reduction, such as Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA).
6. Cross-Validation and Model Evaluation: Scikit-learn includes modules for cross-validation, evaluation metrics, and model selection. Functions like cross_val_score, GridSearchCV, and classification_report can be utilized to perform cross-validation, hyperparameter tuning, and evaluate the model's performance.
7. Model Deployment: Tkinter, popular Python web frameworks, can be used to deploy the trained SVM model as a web application or API, allowing users to make predictions on new data.
8. Integrated Development Environment (IDE): IDEs such as PyCharm, Jupyter Notebook, or Visual Studio Code provide a development environment with code editing, debugging, and interactive execution capabilities, making the development process more efficient.
9. Data Storage: Depending on the size and nature of the dataset, various data storage solutions can be employed, including CSV files, databases (e.g., SQLite, MySQL, or PostgreSQL), or cloud-based storage services like Amazon S3 or Google Cloud Storage.

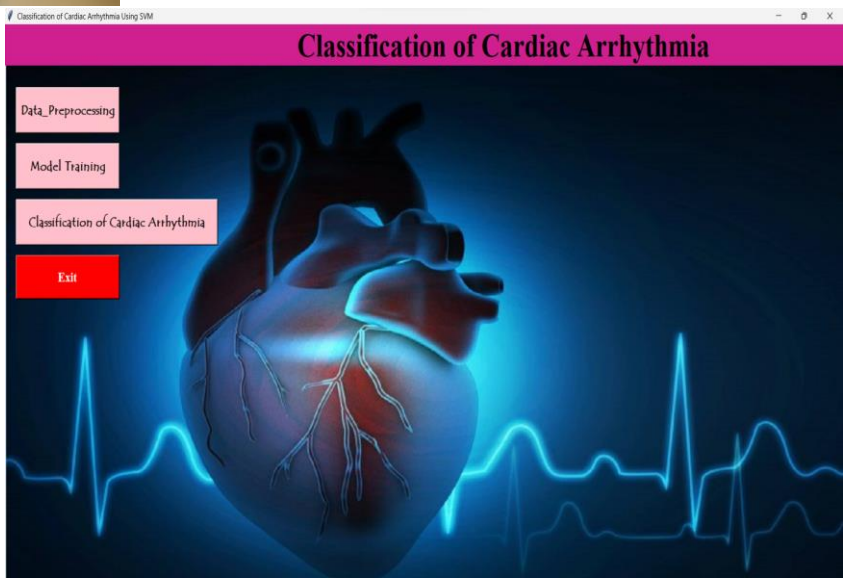
.

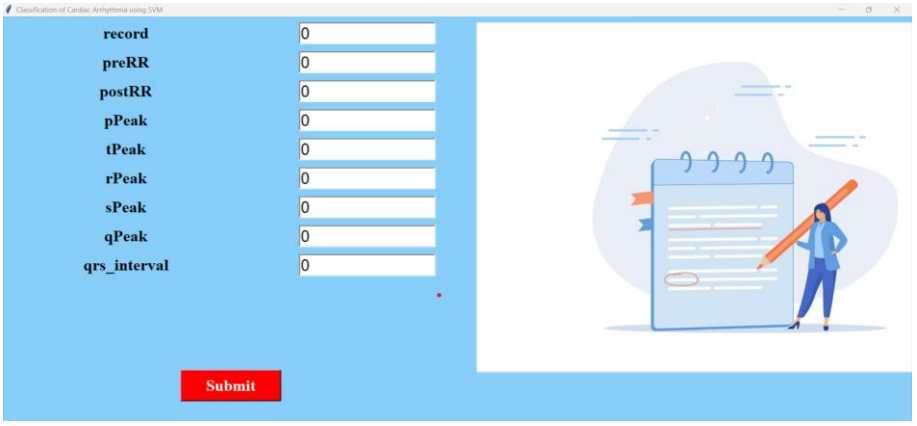
6.3 ALGORITHM DETAILS

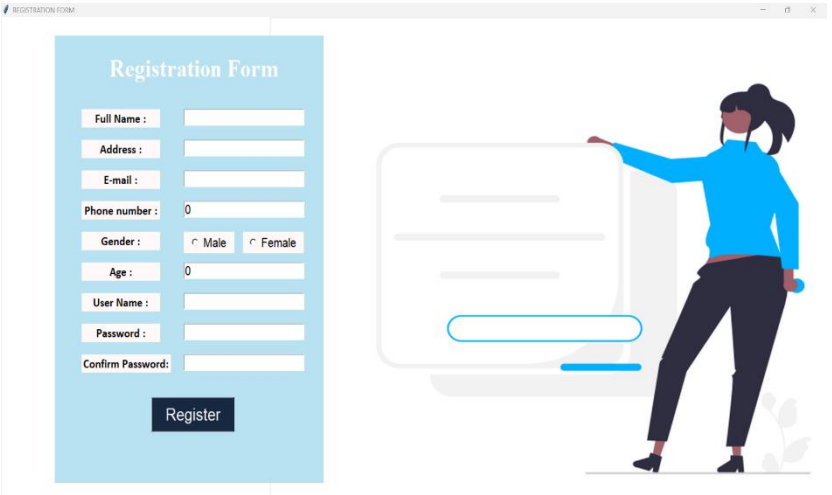
1. Import the necessary libraries for data preprocessing, SVM classification, and Tkinter GUI development.
2. Load the dataset containing cardiac arrhythmia data.
3. Perform data preprocessing tasks such as cleaning, filtering, and feature extraction on the dataset.
4. Split the preprocessed dataset into training and testing sets.
5. Implement the SVM model using a suitable kernel function (e.g., linear, polynomial, radial basis function) and train it using the training set.
6. Create a Tkinter GUI window with appropriate widgets for user interaction.
7. Design and place input fields and options in the GUI to allow users to enter or select relevant parameters for classification (e.g., heart rate, intervals, or ECG waveforms).
8. Create a button widget in the GUI to trigger the classification process.
9. Define a function that retrieves the user input from the GUI, preprocesses it according to the training data preprocessing steps, and feeds it to the trained SVM model for prediction.
10. Display the predicted arrhythmia class in the GUI, either through a label or a message box.
11. Test the GUI by running the application and verifying that the classification process and result display are functioning correctly.
12. Refine and enhance the GUI design and functionality based on user feedback and testing.
13. Optionally, implement additional features such as visualizations or data summaries to provide more insights into the classification results.
14. Iterate on the design and implementation to improve usability, accuracy, and user experience.
15. Deploy the final GUI application for users to classify cardiac arrhythmia cases using the SVM model.

7. RESULT OUTCOMES









8. CONCLUSIONS

CONCLUSION

In conclusion, the project "Classification of Cardiac Arrhythmia using SVM with Tkinter GUI" aims to develop an accurate and user-friendly system for classifying different types of cardiac arrhythmias. By integrating a Support Vector Machine (SVM) algorithm with a Tkinter graphical user interface (GUI), the project addresses the need for efficient arrhythmia diagnosis and classification.

The project's objectives include improving the accuracy of arrhythmia classification, enhancing clinical workflow, and providing decision support to healthcare professionals. By leveraging SVM, a powerful machine learning algorithm, the system aims to achieve high accuracy in identifying and differentiating between various arrhythmia types.

FUTURE WORK

The current project focuses on classifying a specific set of arrhythmia types. Future work can involve expanding the classification system to cover a broader range of arrhythmias, including rare or atypical ones. Extending the project to incorporate real-time ECG signal acquisition and analysis would enable continuous monitoring of arrhythmias. This could involve connecting the system to live ECG devices or integrating it with a telemetry system for remote monitoring. Providing real-time feedback and alerts to healthcare professionals would aid in timely interventions.

The proposed model can be useful for long-term ECG monitoring using single-lead wearable devices in clinical settings in the future.

APPENDIX A

References:

[1] B. de Luna, P. Coumel, and J. F. Leclercq, "Ambulatory sudden cardiac death: Mechanisms of production of fatal arrhythmia on the basis of data from 157 cases," Amer. Heart J., vol. 117, pp. 151–159, 1989.

[2] J. F. Leclercq, P. Coumel, P. Maison-Blanche, B. Cauchemez, M. Zimmer- mann, F. Chouty, and R. Slama, "Mise en evidence des mecanismes d´eterminants de la mort subite," Arch Mal Coeur, vol. 79, pp. 1024–1033, 1986.

[3] H. A. Fozzard, "Electromechanical dissociation and its possible role in suddencardiac death," J. Amer. Coll. Cardiol., vol. 5, pp. 31B–34B, 1985.

[4] K. Dawson, A. S. Leon, and H. L. Taylor, "Effect of pentobarbital anesthesia on vulnerability to ventricular fibrillation," Amer. J. Physiol., vol. 239, pp.H427–H431, 1980.

[5] T. L. Wenger, F. E. Harrell Jr., K. K. Brown, S. Lederman, and H. C. Strauss,"Ventricular fibrillation following canine coronary artery reperfusion: Different outcomes with pentobarbital and alpha-chloralose," Can. J. Physiol. Phar-macol., vol. 62, pp. 224–228, 1984.

[6] F. H. De Jonge, E. A. Bokkers, W. G. Schouten, and F. A. Helmond, "Rearingpiglets in a poor environment: Developmental aspects of social stress in pigs," Physiol. Behavior, vol. 60, pp. 389–396, 1996.

[7] W. G. Schouten and V. M. Wiegant, "Individual responses to acute and chronicstress in pigs," Acta Physiologica Scandinavica. Supplementum, vol. 640, pp. 88–91, 1997.

[8] M. Roelke, H. Garan, B. A. McGovern, and J. N. Ruskin, "Analysis of the

initiation of spontaneous monomorphic ventricular tachycardia by stored in- tracardiac electrograms," J. Amer. Coll. Cardiol., vol. 23, pp. 117–122, 1994.

[9] S. B. Knisley and W. M. Smith, "Experimental techniques for investigatingcardiac electrical activity and response to electrical stimuli," Proc. IEEE, vol.
84, pp. 417–427, Mar. 199

APPENDIX C

PLAGIARISM REPORT

57% Overall Similarity

Top sources found in the following databases:

- 45% Internet database
- Crossref database
- 36% Submitted Works database
- 22% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	irjmets.com Internet	11%
2	D.L. Rollins, C.R. Killingsworth, G.P. Walcott, R.K. Justice, R.E. Ideker, ... Crossref	5%
3	ijraset.com Internet	5%
4	researchgate.net Internet	3%
5	m.mu.edu.sa Internet	2%
6	cumminscollege on 2022-05-31 Submitted works	2%
7	coursehero.com Internet	2%
8	kclpure.kcl.ac.uk Internet	1%

9	Yun Kwan Kim, Minji Lee, Hee Seok Song, Seong-Whan Lee. "Automati... Crossref	1%
10	Shri Guru Gobind Singhji Institute of Engineering and Technology on 20... Submitted works	1%
11	pubmed.ncbi.nlm.nih.gov Internet	1%
12	Savitribai Phule Pune University on 2018-12-05 Submitted works	1%
13	semanticscholar.org Internet	<1%
14	marshfieldclinic.org Internet	<1%
15	Manipal University on 2023-05-25 Submitted works	<1%
16	docs.anaconda.com Internet	<1%
17	Ravinder Yadav, Himanika Chugh, Vandit Jain, Prasenjit Baneriee. "Ind... Crossref	<1%
18	doaj.org Internet	<1%
19	infona.pl Internet	<1%
20	ijirset.com Internet	<1%

21	Symbiosis International University on 2021-05-19 Submitted works	<1%
22	ijies.org Internet	<1%
23	vimeo.com Internet	<1%
24	New Zealand School of Acupuncture on 2021-12-05 Submitted works	<1%
25	researchid.co Internet	<1%
26	cumminscollege on 2022-05-20 Submitted works	<1%
27	pubfacts.com Internet	<1%
28	hal.univ-cotedazur.fr Internet	<1%
29	Abdullah Gul University on 2023-05-20 Submitted works	<1%
30	Middle East College of Information Technology on 2023-03-22 Submitted works	<1%
31	Savitribai Phule Pune University on 2016-02-09 Submitted works	<1%
32	Cranfield University on 2022-12-15 Submitted works	<1%

33	Jonathan M. Engel, Niranjan Chakravarthy, Grant Nosbush, Matt Merke... Crossref	<1%
34	hindawi.com Internet	<1%
35	IUBH - Internationale Hochschule Bad Honnef-Bonn on 2023-05-29 Submitted works	<1%
36	Allam Jaya Prakash, Samit Ari. "A system for automatic cardiac arrhyt... Crossref	<1%
37	Roehampton University on 2012-04-26 Submitted works	<1%
38	researchr.org Internet	<1%
39	University of Westminster on 2023-05-11 Submitted works	<1%
40	cse.anits.edu.in Internet	<1%
41	Manipal University on 2023-05-29 Submitted works	<1%
42	Modern Education Society's College of Engineering, Pune on 2016-07-19 Submitted works	<1%
43	Yeditepe University on 2022-06-05 Submitted works	<1%
44	iGroup on 2017-05-25 Submitted works	<1%

45	CSU, San Jose State University on 2023-04-08 Submitted works	<1%
46	Universiti Malaysia Pahang on 2021-01-17 Submitted works	<1%
47	ABV-Indian Institute of Information Technology and Management Gwal... Submitted works	<1%
48	Kaplan Professional on 2023-05-13 Submitted works	<1%
49	Aston University on 2023-04-04 Submitted works	<1%
50	Savitribai Phule Pune University on 2018-11-26 Submitted works	<1%
51	University of Bradford on 2012-09-17 Submitted works	<1%
52	University of Westminster on 2023-04-26 Submitted works	<1%
53	idoc.pub Internet	<1%
54	amrita.edu Internet	<1%
55	University of Bradford on 2023-02-16 Submitted works	<1%
56	ebsu on 2023-05-27 Submitted works	<1%

57 University of Melbourne on 2015-06-09

Submitted works

<1%