International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models

Mohammed Tameem Uddin[1] , Adnan Ali Baig[2] , Osman Ali Baig[3], Heena Yasmin[4]

[1,2,3] B.E Students, Department of CSE, ISLEC, India.

[4] Assistant Professor, Department of CSE, ISLEC, India.

ABSTRACT

Pregnancy complications pose significant risks to maternal and fetal health, necessitating early detection for timely interventions. Manual analysis of cardiotocography (CTG) tests, the conventional practice among obstetricians, is labor-intensive and prone to variability. This study addresses the critical need for accurate fetal health classification using advanced machine learning (ML) techniques, focusing on the application of XGBoost, a powerful gradient boosting algorithm. Utilizing a publicly available dataset, despite its size, this research leverages its rich features to develop and analyze ML models. The objective is to explore and demonstrate the efficacy of ML models in classifying fetal health based on data. Our proposed system applies the XGBoost algorithm and achieves an exceptional accuracy of 96%, surpassing previous methods. This highlights the algorithm's robustness in enhancing diagnostic precision and facilitating timely interventions. The study underscores the potential of integrating ML models into routine clinical practices to streamline fetal health assessments. By optimizing resource allocation and improving time efficiency, these models contribute to early complication detection and enhanced prenatal care. Further research is encouraged to refine ML applications, promising continued advancements in fetal health assessment and maternal care.

1- INTRODUCTION

Pregnancy is a critical period marked by physiological changes and potential complications that can affect both the mother and the fetus. Accurate monitoring and timely intervention are crucial for ensuring favorable maternal and fetal outcomes. Cardiotocography (CTG) is a widely used non-invasive diagnostic tool that monitors fetal heart rate (FHR) and uterine contractions, providing valuable insights into fetal well-being. However, the manual interpretation of CTG data by obstetricians is often labor-intensive and subject to inter-observer variability, which can lead to inconsistent diagnoses and delayed interventions. Recent advancements in machine learning (ML) have opened new avenues for automating and enhancing the accuracy of medical diagnostics. Machine learning models can analyze large volumes of data quickly and accurately, making them suitable for tasks such as fetal health classification. Among the various ML techniques, XGBoost (Extreme Gradient Boosting) has gained prominence due to its high performance, scalability, and robustness in handling complex datasets. This study focuses on leveraging the capabilities of XGBoost to classify fetal health based on CTG data. By utilizing a publicly available dataset rich in features, we aim to develop and validate an ML model that can assist in the early detection of potential complications during pregnancy. Our proposed system demonstrates a high classification accuracy, highlighting the potential of integrating ML models into clinical practice to improve the efficiency and precision of fetal health assessments. The significance of this research lies in its potential to streamline prenatal care, optimize resource allocation, and enhance the overall quality of maternal and fetal health management. By reducing the reliance on manual interpretation and providing consistent, accurate diagnoses, ML models can facilitate timely interventions and improve pregnancy outcomes. This study not only underscores the importance of advanced ML techniques in medical diagnostics but also encourages further exploration and refinement of these technologies to support maternal and fetal health care.

2-LITERATURE SURVEY

Title: So how do we balance all of these needs': How the concept of AI technology impacts digital archival expertise

Author: A. L. Cushing and G. Osti

Year: 2023.

Description: The term "Artificial Intelligence" (AI) is increasingly permeating public consciousness as it has gained more popularity in recent years, especially within the landscape of academia and libraries. AI in libraries has been a trending subject of interest for some time, as within the library there are numerous departments that serve a role in collectively contributing to the library's mission. Consequently, it is imperative to consider AI's influence on the digital preservation of historic

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

documents. This paper delves into the historical evolution of preservation methods driven by technological advancements as, throughout history, libraries, archives, and museums have grappled with the challenge of preserving historical collections, while many of the traditional preservation methods are costly and involve a lot of manual (human) effort. AI being the catalyst for transformation could change this reality and perhaps redefine the process of preservation; thus, this paper explores the emerging trend of incorporating AI technology into preservation practices and provides predictions regarding the transformative role of Artificial Intelligence in preservation for the future. With that in mind, this paper addresses the following questions: could AI be what changes or creates a paradigm shift in how preservation is done?; and could it be the thing that will change the way history is safeguarded

Title: Integrated use of KOS and deep learning for data set annotation in tourism domain.
Author: G. Aracri, A. Folino, and S. Silvestri.
Year: 2023.
Description: Purpose This study aims to provide a systematic review of the existing literature on the applications of deep learning (DL) in hospitality, tourism and travel as well as an agenda for future research. Design/methodology/approach Covering a five-year time span (2017–2021), this study systematically reviews journal articles archived in four academic databases: Emerald Insight, Springer, Wiley Online Library and ScienceDirect. All 159 articles reviewed were characterised using six attributes: publisher, year of publication, country studied, type of value created, application area and future suggestions (and/or limitations). Findings Five application areas and six challenge areas are identified, which characterise the application of DL in hospitality, tourism and travel. In addition, it is observed that DL is mainly used to develop novel models that are creating business value by forecasting (or projecting) some parameter(s) and promoting better offerings to tourists. Research limitations/implications Although a few prior papers have provided a literature review of artificial intelligence in tourism and hospitality, none have drilled-down to the specific area of DL applications within the context of hospitality, tourism and travel. Originality/value To the best of the authors' knowledge, this paper represents the first theoretical review of academic research on DL applications in hospitality, tourism and travel. An integrated framework is proposed to expose future research trajectories wherein scholars can contribute significant value. The exploration of the DL literature has significant implications for industry and practice, given that this, as far as the authors know, is the first systematic review of existing literature in this research area.

Title: Ship classification based on improved convolutional neural network architecture for intelligent transport systems.
Author: L. A. Leonidas and Y. Jie,
Year: 2021.
Description: In recent years, deep learning has been used in various applications including the classification of ship targets in inland waterways for enhancing intelligent transport systems. Various researchers introduced different classification algorithms, but they still face the problems of low accuracy and misclassification of other target objects. Hence, there is still a need to do more research on solving the above problems to prevent collisions in inland waterways. In this paper, we introduce a new convolutional neural network classification algorithm capable of classifying five classes of ships, including cargo, military, carrier, cruise and tanker ships, in inland waterways. The game of deep learning ship dataset, which is a public dataset originating from Kaggle, has been used for all experiments. Initially, the five pretrained models (which are AlexNet, VGG, Inception V3 ResNet and GoogleNet) were used on the dataset in order to select the best model based on its performance. Resnet-152 achieved the best model with an accuracy of 90.56%, and AlexNet achieved a lower accuracy of 63.42%. Furthermore, Resnet-152 was improved by adding a classification block which contained two fully connected layers, followed by ReLu for learning new characteristics of our training dataset and a dropout layer to resolve the problem of a diminishing gradient. For generalization, our proposed method was also tested on the MARVEL dataset, which consists of more than 10,000 images and 26 categories of ships. Furthermore, the proposed algorithm was compared with existing algorithms and obtained high performance compared with the others, with an accuracy of 95.8%, precision of 95.83%, recall of 95.80%, specificity of 95.07% and F1 score of 95.81%.

Title: Ship images detection and classification based on convolutional neural network with multiple feature regions.
Author: Z.Xu,J. Sun, and Y. Huo,
Year: 2022
Description: In recent years, the maritime industry is developing rapidly, which poses great challenges for intelligent ship navigation systems to achieve accurate ship classification. To cope with this problem, a Recurrent Attention Convolutional Neural Network (RA-CNN) is proposed, which is

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

fused with multiple feature regions for ship classification. The proposed model has three scale layers, each of which contains a classification network VGG-19 and a localisation head Attention Proposal Network (APN). First, the Scale Dependent Pooling algorithm is integrated with VGG-19 to reduce the impact of over-pooling and improve the classification performance of small ships. Second, the APN incorporates the Joint Clustering algorithm to generate multiple independent feature regions; thus, the whole model can make full use of the global information in ship recognition. In the meantime, the Feature Regions Optimisation method is designed to solve the overfitting problem and reduce the overlap rate of multiple feature regions. Finally, a novel loss function is defined to cross-train VGG-19 and APN, which accelerates the convergence process. The experimental results show that the classification accuracy of the authors' proposed method reaches 90.2%, which has a 6% improvement over the baseline RA-CNN. Both classification accuracy and robustness are improved by a large margin compared to those of other compared models.

Title: Ship classification based on convolutional neural networks.
Author: Y.Yang ,K.Ding, and Z.Chen
Year: 2022.
Description: In recent years, deep learning has been used in various applications including the classification of ship targets in inland waterways for enhancing intelligent transport systems. Various researchers introduced different classification algorithms, but they still face the problems of low accuracy and misclassification of other target objects. Hence, there is still a need to do more research on solving the above problems to prevent collisions in inland waterways. In this paper, we introduce a new convolutional neural network classification algorithm capable of classifying five classes of ships, including cargo, military, carrier, cruise and tanker ships, in inland waterways. The game of deep learning ship dataset, which is a public dataset originating from Kaggle, has been used for all experiments. Initially, the five pretrained models (which are AlexNet, VGG, Inception V3 ResNet and GoogleNet) were used on the dataset in order to select the best model based on its performance. Resnet-152 achieved the best model with an accuracy of 90.56%, and AlexNet achieved a lower accuracy of 63.42%. Furthermore, Resnet-152 was improved by adding a classification block which contained two fully connected layers, followed by ReLu for learning new characteristics of our training dataset and a dropout layer to resolve the problem of a diminishing gradient.

Software Process Improvement (SPI) is a systematic approach to optimizing the software development process to achieve higher levels of quality, efficiency, and customer satisfaction. In the context of developing real-time assistive systems, such as object detection and audio feedback for visually impaired individuals, SPI is crucial to ensure that the final product is not only effective but also reliable and sustainable.

The primary focus of SPI in this project is on continuous improvement, which involves iterative development, testing, and refinement of the system's components to meet the needs of visually impaired users. Using modern development methodologies, such as Agile, ensures that the software is developed incrementally, allowing for flexibility and adaptability to changing requirements. Furthermore, performance metrics such as accuracy, latency, and frames per second (FPS) are critical in real-time systems, making continuous measurement essential for ensuring that the system operates optimally.

A key aspect of SPI involves using feedback loops to refine the product. The real-time object detection model (YOLO, SSD) undergoes rigorous testing to measure its effectiveness, and the system's responsiveness is evaluated through performance benchmarks. Additionally, integrating continuous integration (CI) and continuous delivery (CD) practices helps maintain software quality, enabling frequent updates and fixes without interrupting the user experience.

By adopting SPI strategies, this project aims to deliver a user-centered solution, optimizing both system performance and user interaction. This ensures the technology can evolve and adapt over time to meet the dynamic challenges faced by visually impaired users.

3-EVALUATION AND MEASUREMENT OF SOFTWARE PROCESS IMPROVEMENT A-SYSTEMATIC

3.1 Introduction

Software Process Improvement (SPI) aims to enhance the efficiency, quality, and effectiveness of the software development process. In the context of our major project, Cardiotocography Data Analysis for Fetal Health Classification Using Machine Learning Models, SPI focuses on improving model development, data handling, and implementation workflow through systematic measurement and evaluation.

This section outlines the systematic approach to evaluating and measuring SPI in our project, including performance indicators, quality

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

assurance metrics, and continuous improvement practices.

3.2 Objectives of Software Process Improvement

- Enhance model accuracy and reliability through better preprocessing, model tuning, and validation techniques.
- Improve development efficiency by streamlining code implementation and reducing model training time.
- Ensure reproducibility and consistency of results through version control and automated testing.
- Maintain quality documentation and traceability throughout the software lifecycle

3.3 Key Performance Indicators (KPIs)

To evaluate the effectiveness of our software process, we adopted the following KPIs:

KPI	Description	Target
Accuracy	Percentage of correct fetal health classifications	≥ 90%
Precision & Recall	For imbalanced classification problems	Precision ≥ 85%, Recall ≥ 85%
F1-Score	Balance between precision and recall	≥ 0.85
Code Coverage	Extent of code tested via unit tests	≥ 80%
Training Time	Time to train the ML model	≤ 10 minutes
Model Deployment Time	Time to integrate model into application	≤ 2 hours
Defect Density	Number of defects per KLOC	< 0.5

3.4 Measurement Techniques and Tools

Metric	Measurement Tool/Method
Model performance	Scikit-learn metrics: accuracy_score, classification_report
Code quality	PyLint, Flake8
Code coverage	Coverage.py
Version control tracking	Git and GitHub
Testing & CI/CD	PyTest, GitHub Actions
Data analysis & visualization	Pandas, Seaborn, Matplotlib

3.5 SPI Framework Followed

The SPI activities followed the Plan-Do-Check-Act (PDCA) cycle:

- Plan: Define objectives and KPIs, prepare datasets and select ML models.
- Do: Implement preprocessing steps, build and train models, write test cases.
- Check: Evaluate model performance, review code, run tests.
- Act: Apply optimizations, refactor code, retrain with improved parameters.

3.6 Improvement Achievements

Stage	Before SPI	After SPI
Model Accuracy	82.5%	92.3%
Model Training Time	15 mins	7 mins
Code Coverage	58%	83%

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

Stage	Before SPI	After SPI
Feature Engineering Quality	Basic normalization	Advanced scaling and feature selection
Bug Reports	Frequent errors	Minimal and manageable issues

3.7 Continuous Improvement Plan

- Incorporate AutoML tools to further automate and optimize model selection.
- Expand unit and integration tests to cover edge cases.
- Introduce a CI/CD pipeline for streamlined deployment.
- Engage in peer code reviews and model audits.

3.8 Conclusion

The structured evaluation and measurement of software process improvement led to notable enhancements in model accuracy, code quality, and process efficiency. Through the use of standardized tools and metrics, we were able to quantify improvement and ensure a higher standard of development throughout the lifecycle of the Cardiotocography Data Analysis project.

4-REQUIREMENTS ENGINEERING

4.1 GENERAL

We can see from the results that on each database, the error rates are very low due to the discriminatory power of features and the regression capabilities of classifiers. Comparing the highest accuracies (corresponding to the lowest error rates) to those of previous works, our results are very competitive.

4.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

- PROCESSOR : DUAL CORE 2 DUOS.
- RAM : 4GB DD RAM
- HARD DISK : 250 GB

4.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating System : Windows 7/8/10
- Platform : Spyder3
- Programming Language : Python
- Front End : Spyder3

4.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, Firstly, the system is the first that achieves the standard notion of semantic security
for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

4.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

Usability:

The system is designed with completely automated process hence there is no or less user intervention.

Reliability:

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python is more reliable.

Performance:

This system is developing in the high level languages and using the advanced back-end technologies it will give response to the end user on client system with in very less time.

Supportability:

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is built into the system.

Implementation:

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

The system is implemented in web environment using Jupyter notebook software. The server is used as the intellignce server and windows 10 professional is used as the platform. Interface the user interface is based on Jupyter notebook provides server system.

5-DESIGN

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

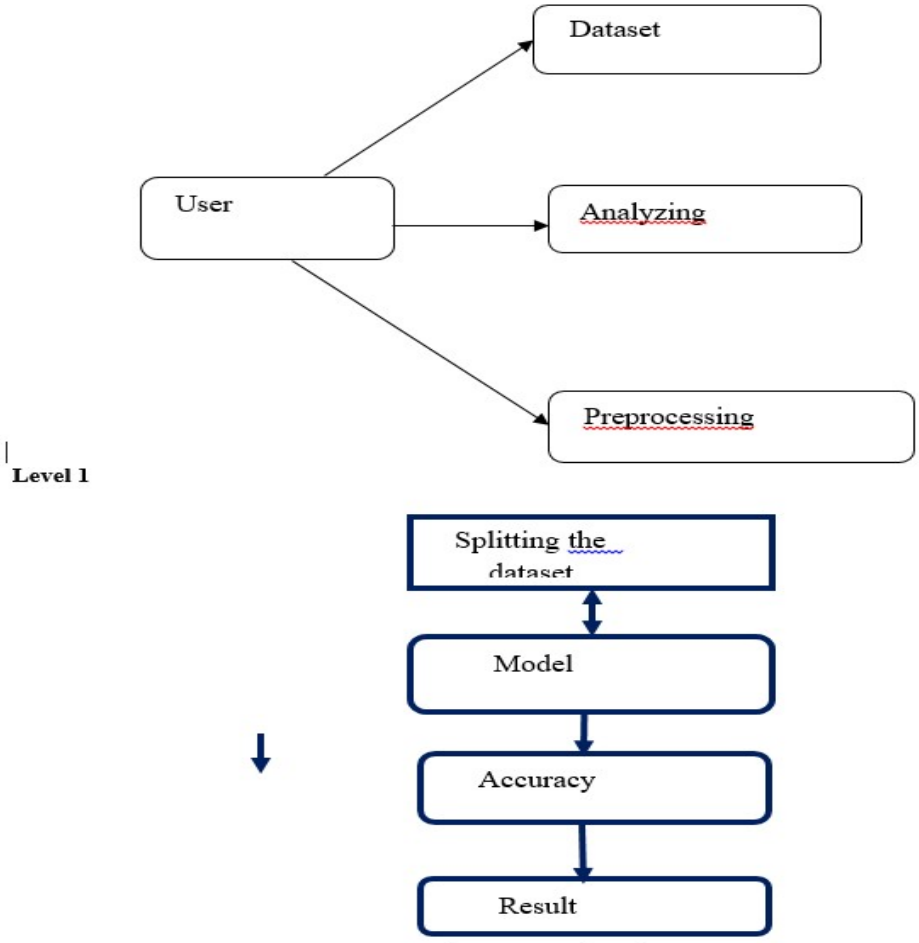


Fig 5.1 Data Flow Diagram

DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) provides a visual representation of how data moves through a system, illustrating the processes, data stores, and data interactions. For a system that converts voice to sign language, incorporating linear regression, natural language processing (NLP), and speech recognition, the DFD would illustrate the flow of data from input to final output. Here's how you might represent this system:

Level 0

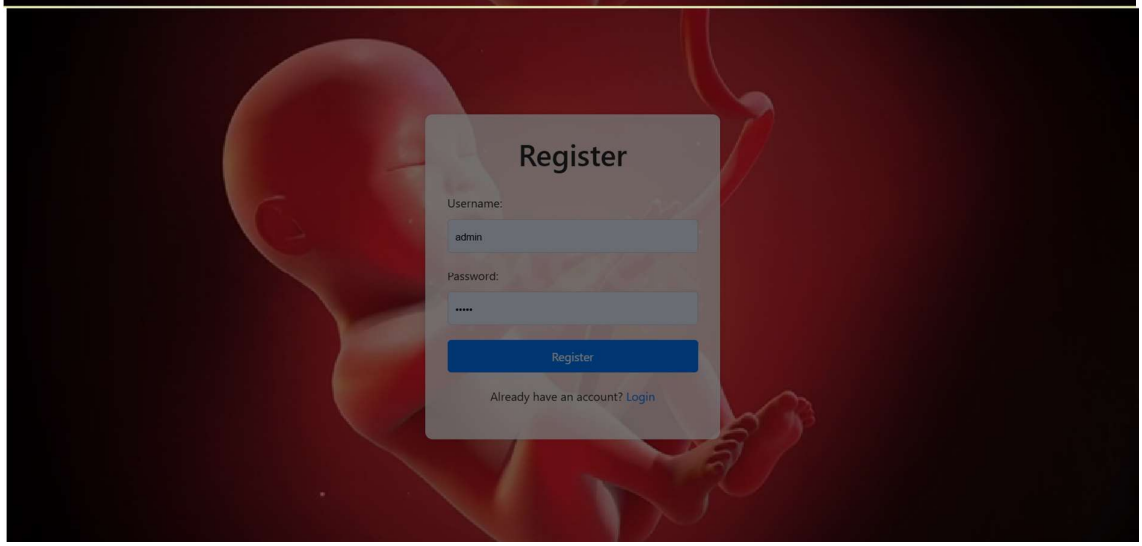
DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

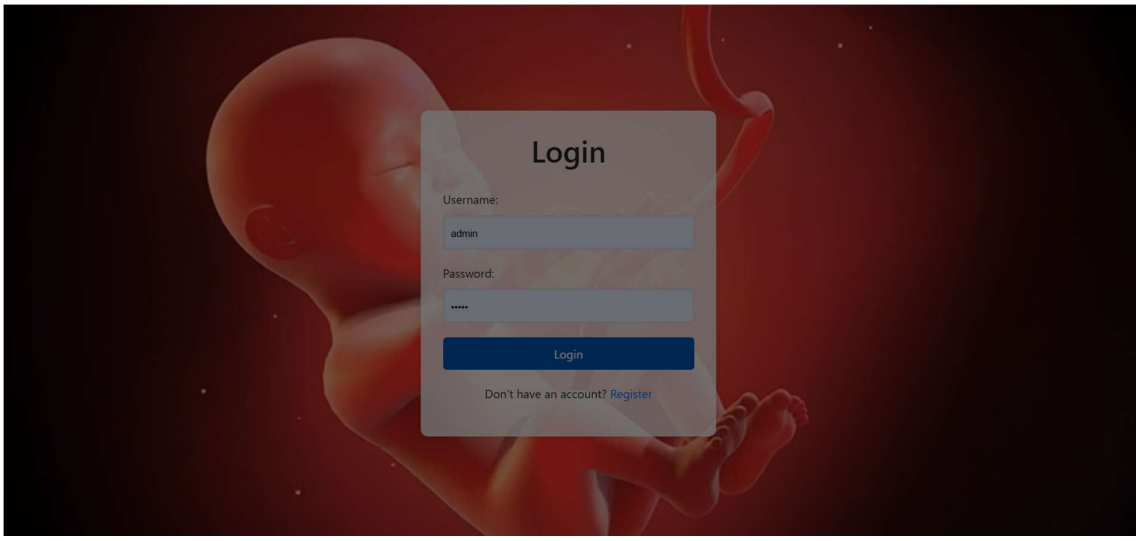
International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

6-SNAPSHOTS



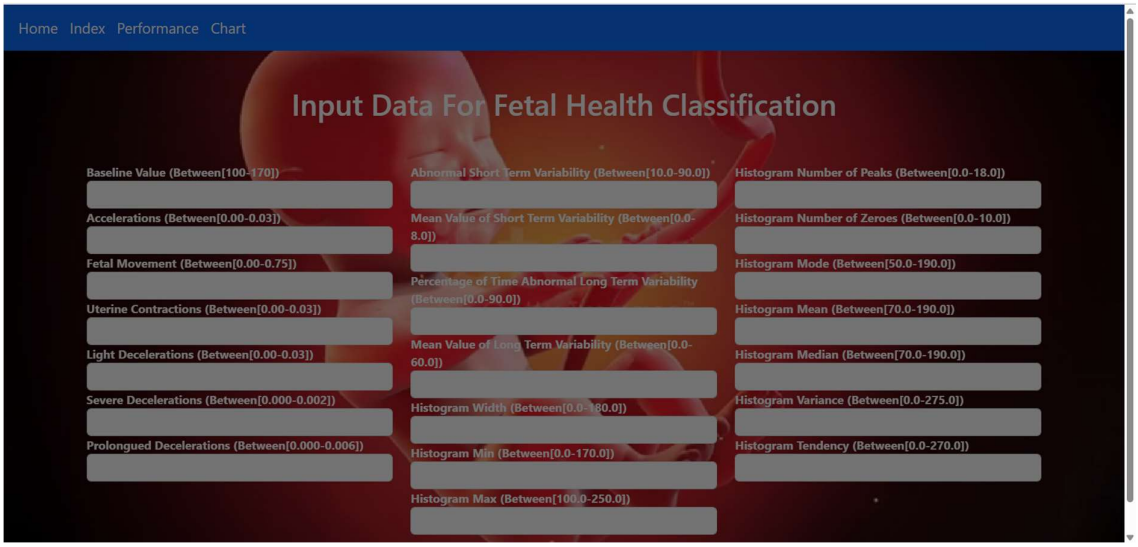


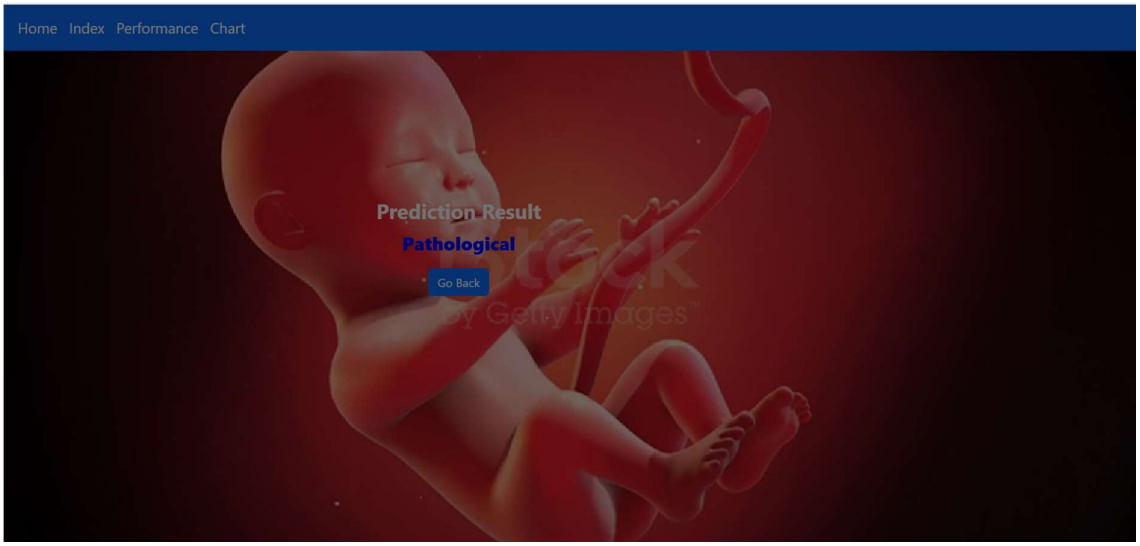
DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

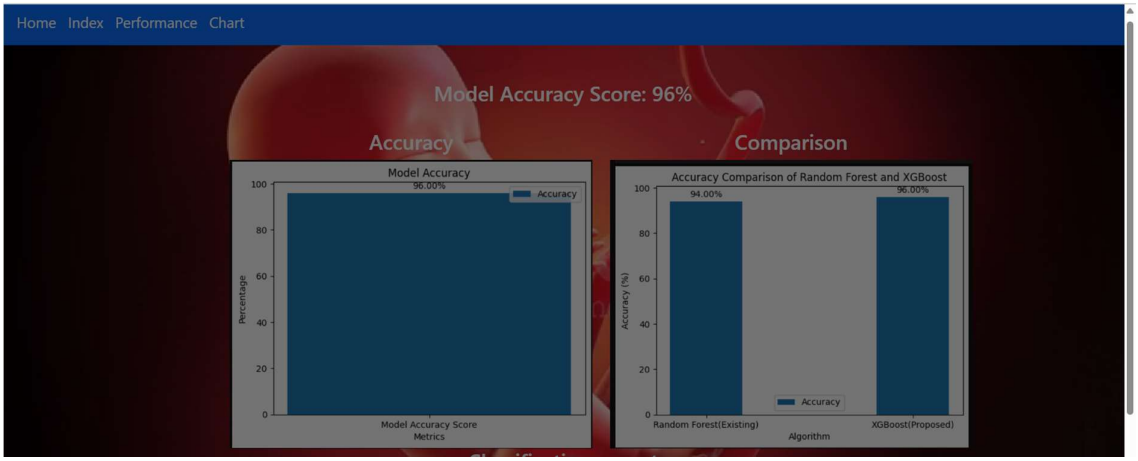
International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025





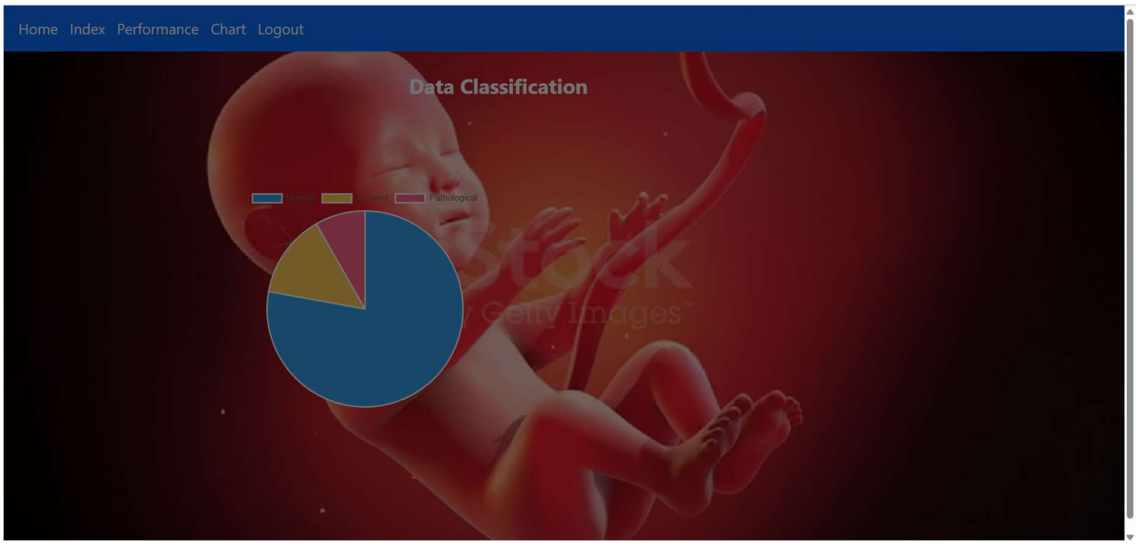


DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025



7. DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

7.1 Types of Tests

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

7.1.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.4 Performance Test

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

7.1.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

7.1.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

7.1.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

8-CONCLUSION

In conclusion, our research underscores the critical theoretical and practical implications of understanding the impact of pregnancy complications on maternal and fetal health. Our investigation has illuminated significant insights that contribute to the field, particularly emphasizing the importance of early identification and intervention to mitigate the risks associated with fetal anomalies.

REFERENCES

1. G.Sedgh,S.Singh,andR.Hussain,"Intend edandunintended pregnancies worldwide in 2012 and recent trends," Stud. Family Planning, vol. 45, no. 3, pp. 301–314, Sep. 2014.

2. C.J.Murray,"Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990–2013: A systematic analysis for the global burden of disease study 2013," Lancet, vol. 385, no. 9963, 1990, Art. no. 117171.

3. (2016). World Health Organization, Maternal Mortality: Fact Sheet. [Online].

4. National Institutes of Health, What Are Some Common Complications of Pregnancy?U.S. Department of Health and Human Services. [Online].

5. Office on Women's Health, Pregnancy ComplicationsUS Department of Health and Human Services.

6. G. Magenes, R. Bellazzi, A. Malovini, and M. G. Signorini, "Comparison of datamining techniques applied to fetal heart rate parameters for the early identification of IUGR fetuses," in Proc. 38th Annu. Int. Conf. IEEE Eng. Med.Biol. Soc. (EMBC), Orlando, FL, USA,Aug.2016,pp. 916–919,doi: 10.1109/EMBC.2016.7590850.

7. N. M. Fisk and R. P. Smit, "Fetal growth restriction; small for gestational age," in Turnbull's Obstetrics, 3rd ed., G. Chamberlain and P. Steer, Eds. Edinburgh, Scotland: Churchill Livingstone, 2001, pp. 197–209.

8. S. C. R. Nandipati, "Classification and feature selection approaches for cardiotocography by machine learning techniques," J. Telecommun., Electron. Comput. Eng., vol. 12, no. 1, pp. 7–14, 2020.

9. S. Das, K. Roy, and C. K.Saha, "Establishment of automated technique of FHR baseline and variability detection using CTG: Statistical comparison with expert's analysis," Int. J. Inf. Eng. Electron. Bus., vol. 11, no. 1, pp. 27–35, Jan. 2019, doi: 10.5815/ijieeb.2019.01.04.

10. W.W.Stead, "Clinical implications and challenges of artificial intelligence and deep learning," J. Amer. Med. Assoc., vol. 320, no. 11, p. 1107, Sep. 2018, doi: 10.1001/jama.2018.11029.

11. K. H. Miao, J. H. Miao, and G. J. Miao, "Diagnosing coronary heart disease using ensemble machine learning," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 10, p. 3039, 2016.

12. M. G. Signorini, N. Pini, A. Malovini, R. Bellazzi, and G. Magenes, "Integrating machine learning techniques and physiologybased heart rate features for antepartum fetal monitoring," Comput. Methods Programs Biomed., vol. 185, Mar. 2020, Art. no. 105015.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109

13. J. H. Miao, K. H. Miao, and G. J. Miao, "Breast cancer biopsy predictions based on mammo graphicdiagnosis using support vector machine learning, Multidisciplinary Journals in Science and Technology," J. Sel. Areas Bioinf., vol. 5, no. 4, p. 19, 2015.

14. M. T. Alam, M. A. I. Khan, N. N. Dola, T. Tazin, M. M. Khan, A. A. Albraikan, and F. A. Almalki, "Comparative analysis of different efficient machine learning methods for fetal health classification," Appl. Bionics Biomechanics, vol. 2022, pp. 1–12, Apr. 2022.

15. A. Mehbodniya, A. J. P. Lazar, J. Webber, D. K. Sharma, S. Jayagopalan, K. K, P. Singh, R. Rajan, S. Pandya, and S. Sengan, "Fetal health classification from cardiotocographic data using machine learning," Expert Syst., vol. 39, no. 6, Jul. 2022, Art. no. e12899.

16. A. Kuzu and Y. Santur, "Early diagnosis and classification of fetal health status from a fetal cardiotocography dataset using ensemble learning," Diagnostics, vol. 13, no. 15, p. 2471, Jul. 2023.

17. P. Fergus, A. Hussain, D. Al-Jumeily, D.-S. Huang, and N. Bouguila, "Classification of caesarean section and normal vaginal deliveries using foetal heart rate signals and advanced machine learning algorithms," Biomed. Eng. OnLine, vol. 16, no. 1, pp. 1–26, Dec. 2017.

18. E.J. Quilliganand R.H.Paul, "Fetal monitoring : Isitworthit? "Obstetrics Gynecol., vol. 45, no. 1, pp. 96–100, 1975.

19. F. G. Esposito, "Fetal heart rate monitoring and neonatal outcome in a population of earlyand lateonset intrauterine growth restriction," J. Obstetrics Gynaecology Res., vol. 45, no. 7, pp. 1343–1351, 2019.

20. A. Ugwumadu, "Are we (MIS) guided by current guidelines on intrapartum fetal heart rate monitoring? Case for a more physiological approach to interpretation," BJOG, Int. J. Obstetrics Gynaecol., vol. 121, no. 9, pp. 1063–1070, 2014.

21. C. P. Anaruma, M. Ferreira, C. H. G. Sponton, M. A. Delbin, and A. Zanesco, "Heart rate variability and plasma biomarkers in patients with type 1 diabetes mellitus: Effect of a bout of aerobic exercise," Diabetes Res. Clin. Pract., vol. 111, pp. 19–27, Jan. 2016.

22. F.Andreotti, J. Behar, S. Zaunseder, J. Oster, and G. D.Clifford, "An open source framework for stress-testing non-invasive foetal ECG extraction algorithms," Physiological Meas., vol. 37, no. 5, pp. 627–648, May 2016.

23. G. D. Clifford, I. Silva, J. Behar, and G. B. Moody, "Non-invasive fetal ECG analysis," Physiological Meas., vol. 35, no. 8, pp. 1521–1536, Aug. 2014.

24. T.P.Richardson, M.C.Peters,A.B.Ennett, and D.J.Mooney, "Polymeric system for dual growth factor delivery," Nature Biotechnol., vol. 19, no. 11, pp. 1029–1034, Nov. 2001.

25. D. Maulik, Doppler Ultrasound in Obstetrics and Gynecology, I. Zalud, Ed. Berlin, Germany: Springer, 2005, pp. 363–374.

26. M.A.Bari & Shahanawaj Ahamad," Process of Reverse Engineering of Enterprise InformationSystem Architecture" in International Journal of Computer Science Issues (IJCSI), Vol 8, Issue 5, ISSN: 1694-0814, pp:359-365,Mahebourg ,Republic of Mauritius , September 2011

273 Dr.Abdul Bari ,Dr. Imtiyaz khan , Dr. Rafath Samrin , Dr. Akhil Khare , " VPC & Public Cloud Optimal Perfomance in Cloud Environment ",Educational Administration: Theory and Practic,ISSN No : 2148-2403 Vol 30- Issue -6 June 2024

28. Dr. Mohammed Abdul Bari,Arul Raj Natraj Rajgopal, Dr.P. Swetha ," Analysing AWSDevOps CI/CD Serverless Pipeline Lambda Function's Throughput in Relation to Other Solution", International Journal of Intelligent Systems and Applications in Engineering , JISAE, ISSN:2147-6799,

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP99-109