AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2026)
www.ajmimc.com

Cloud-Enabled AI Framework for Real-Time AI-Driven Dual-Target Decision Support System in Emergency Medical Services

P. Vijay Goud[1*], Palsa Srinivas[2], Badhavath Sitharam[2], Ragiri Sheshu[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghanpur, Ghatkesar, 501301, Telangana, India.

*Correspondence: P. Vijay Goud (panjala.vijay123@gmail.com)

ABSTRACT

The rapid increase in hypertension and diabetes cases has created a strong demand for advanced healthcare systems that support early diagnosis and effective clinical decision-making. Managing these chronic conditions requires continuous evaluation of multiple patient attributes, which becomes difficult when performed manually. Traditional approaches rely heavily on basic statistical techniques and human interpretation, making them inefficient for handling large-scale data, identifying complex patterns, and providing real-time predictions. As a result, such systems often lack accuracy, scalability, and reliability, particularly in distributed healthcare environments. A key challenge lies in their inability to handle imbalanced datasets, perform multi-condition prediction, and enable seamless remote communication between systems. To address these limitations, this work proposes a real-time decision support system powered by Artificial Intelligence (AI) using a dual client–server architecture. The server handles data preprocessing, model training, and prediction using Machine Learning (ML) algorithms such as Complement Naive Bayes (CNB), Multinomial Naive Bayes (MNB), Perceptron, and a Tao Tree Classifier (TTC). Preprocessing methods include Label Encoding and K-Means Synthetic Minority Oversampling Technique (KMeans-SMOTE) to manage categorical data and class imbalance. A Flask-based Application Programming Interface (API) using Hypertext Transfer Protocol (HTTP) enables efficient communication between the client and server. The client system allows users to upload datasets, which are processed remotely to predict blood pressure categories and diabetes status. Lightning Memory-Mapped Database (LMDB) is used for secure and efficient data management. The proposed system ensures accurate multi-target prediction, real-time accessibility, and seamless device communication, ultimately improving healthcare services, reducing manual effort, and supporting better clinical decisions.

Keywords: Hypertension Prediction, Diabetes Detection, Machine Learning (ML), Client–Server Architecture, LMDB, Flask Framework

1. INTRODUCTION

This study emphasizes the need for a thorough evaluation of clinical digital solutions powered by Artificial Intelligence (AI) across multiple healthcare environments, including home care, outpatient services, and ambulance-based emergency systems, to determine their impact on patient outcomes [1]. The proposed approach involves systematic data collection from diverse stages of care, ranging from home monitoring to intensive care units, incorporating both clinical and socio-economic attributes. These include factors such as living conditions, social status, age, gender, medical history, current health status, and various physiological parameters, all of which are essential for assessing their influence on patient outcomes. Despite the increasing adoption of AI in healthcare, only a limited number of system providers have critically evaluated their solutions in real-world scenarios such as home environments, community healthcare, and ambulance services [2]. While AI-driven systems offer promising advancements, they also introduce significant concerns, including accountability for errors, the risk of bias amplification, data privacy and security issues, and the lack of transparency in clinical decision-making processes.

This work addresses these challenges by implementing robust data protection mechanisms and developing a transparent AI framework capable of continuous learning, bias detection, and algorithmic improvement. Furthermore, AI-based systems provide substantial benefits in real-time Emergency Medical Services (EMS) coordination by optimizing ambulance dispatch, identifying the fastest routes, and minimizing delays in reaching patients. These systems also enhance coordination with hospitals by directing patients to the most appropriate facilities based on their medical conditions, thereby improving patient outcomes and reducing hospital overcrowding. In addition, AI systems dynamically adapt to changing scenarios, providing real-time updates and recommendations to EMS teams, which improves resource utilization and reduces the cognitive burden on dispatchers and medical personnel [3].

However, the implementation of real-time AI-driven decision support in EMS also presents several challenges [4], including technical integration with existing infrastructure such as GPS tracking, communication systems, and hospital databases. Moreover, user acceptance remains a critical factor, as EMS professionals must trust and effectively utilize AI-generated recommendations, which requires proper training and system adoption. Ethical concerns related to transparency, explainability, and accountability further highlight the need for strong regulatory frameworks and oversight to ensure safe and reliable deployment of AI in healthcare systems [5].

2. RELATED WORK

2.1 AI-Driven Healthcare Assistance and Workflow Optimization

Schmidt Batista, et al. [6] analysed the impact of AI-powered medical assistants on healthcare operations through a comprehensive review of existing studies and real-world implementations. Their findings indicated that AI systems significantly improved clinical productivity by enhancing documentation accuracy, optimizing triage processes, and reducing administrative workload. The integration with EHRs enabled real-time data capture, structured record management, and automated clinical decision support, which reduced physician burnout and improved patient throughput. Additionally, these systems supported predictive analytics for early risk identification. However, the study also highlighted critical challenges, including algorithmic bias affecting fairness in clinical decisions, data governance and privacy concerns, and resistance from healthcare professionals due to trust and usability issues.

Zota, et al. [11] explored the role of AI in improving workflow efficiency and information accessibility in healthcare systems. The study demonstrated that AI-based systems streamlined hospital operations by automating repetitive administrative tasks, organizing large volumes of medical data, and enhancing inter-departmental communication. Their approach enabled faster retrieval of patient information and improved coordination between healthcare units, ultimately supporting timely decision-making. Furthermore, the study emphasized the importance of integrating AI solutions with legacy healthcare infrastructures to ensure seamless data flow. Despite these advantages, challenges such as lack of standardized data formats, interoperability issues, and system compatibility constraints were identified as major barriers to effective implementation.

2.2 Intelligent Decision Support and Collaborative Frameworks

Lujak, et al. [7] presented a multi-agent DSS for distributed EMS coordination, focusing on patients requiring urgent cardiac interventions. The system employed a three-level optimization framework that dynamically assigned patients to ambulances and specialized hospitals equipped for procedures such as angioplasty. By incorporating an auction-based algorithm, the model efficiently solved the resource allocation problem, ensuring optimal utilization of available emergency resources. The system was evaluated using simulation scenarios, which demonstrated significant improvements in coordination efficiency, reduced patient waiting times, and minimized delays in critical care delivery. The study also highlighted the adaptability of the framework in handling dynamic and uncertain emergency environments.

Tang, et al. [8] introduced a MC framework leveraging LLMs to enable intelligent and coordinated decision-making in healthcare environments. The study demonstrated that LLMs could perform advanced reasoning tasks, including zero-shot and context-aware learning, allowing them to interpret complex medical scenarios without extensive task-specific training. The framework facilitated collaboration among multiple agents, enabling distributed problem-solving in time-critical situations such as EMS operations. This approach improved decision accuracy and responsiveness under high-pressure conditions. However, the authors pointed out significant limitations, including concerns over model reliability, lack of regulatory standards, potential hallucination issues, and challenges in deploying such systems in real-world clinical environments where accuracy and accountability are critical.

2.3 Healthcare Data Integration and System Efficiency

Miller, et al. [9] highlighted the importance of integrating data from multiple healthcare sources to enhance trauma care and patient monitoring. Their approach involved linking heterogeneous datasets from ambulance services, emergency departments, and hospitals to create a unified and continuous patient information flow from pre-hospital care to final treatment. This integration improved situational awareness for medical professionals, enabled more accurate clinical decisions, and supported comprehensive patient tracking across different stages of care. Additionally, the study demonstrated how integrated data systems could be used to evaluate and optimize trauma care performance at a systemic level.

Elfahim, et al. [10] examined EMS as a complex and resource-constrained system facing increasing demand and operational challenges. The study emphasized the growing adoption of AI techniques to enhance decision-making, optimize resource allocation, and improve overall system performance. AI models were shown to provide predictive insights, enabling proactive management of emergency scenarios. However, the research identified several limitations, including lack of model explainability, limited availability of high-quality data, and practical challenges in integrating AI technologies into existing EMS infrastructures. These issues hinder the scalability and widespread adoption of AI-driven EMS solutions.

Nickel, et al. [12] addressed EMS planning as a multi-objective optimization problem that balances cost efficiency with service quality. Their approach incorporated multiple performance metrics, including response time, service coverage, system preparedness, and operational costs, to provide a realistic and flexible planning framework. The study highlighted inherent trade-offs between minimizing costs and maintaining high-quality emergency response services. By applying multi-objective optimization techniques, the model supported better decision-making for resource allocation and strategic planning in EMS systems, ensuring a balance between efficiency and effectiveness.

3. PROPOSED SYSTEM

The proposed system architecture integrates data preprocessing, Machine Learning (ML) model development, evaluation, and distributed deployment into a unified framework that enables real-time prediction and secure communication, as illustrated in Fig. 1. The architecture is structured into three main layers: the admin/server hub, the communication layer, and the client/respondent system, each handling specific functionalities. Initially, the admin module performs dataset ingestion, preprocessing, and transformation using techniques such as cleaning, encoding, and balancing to ensure high-quality input for model training, while data storage and fast retrieval are supported using Lightning Memory-Mapped Database (LMDB). In the model building phase, multiple ML algorithms including CNB, MNB, Perceptron, and the proposed TTC are trained and evaluated using metrics calculators and visualization tools to identify the most effective model.

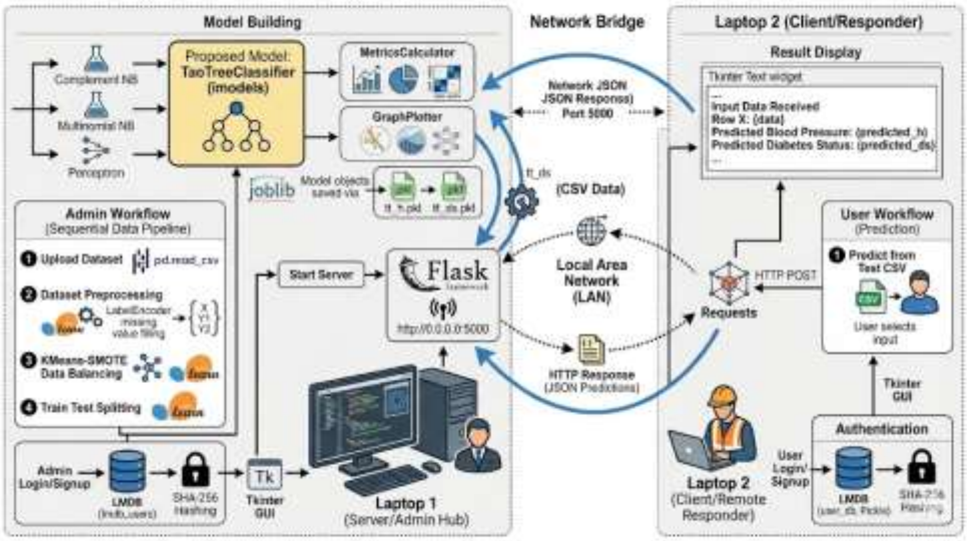

Fig. 1: Proposed system architecture

The selected model is then serialized using joblib and deployed on a Flask server, which acts as the central processing unit for handling prediction requests. Communication between server and client is established over Local Area Network (LAN) using HyperText Transfer Protocol (HTTP), with data exchanged in JavaScript Object Notation (JSON) format for efficient and structured transmission. The client system interacts through a Tkinter-based GUI, allowing users to input data and receive predictions in real time. Additionally, authentication mechanisms using Secure Hash Algorithm 256 (SHA-256) ensure secure access and protect user credentials. The modular design of the system separates preprocessing, model computation, deployment, and user interaction layers, enhancing scalability, maintainability, and flexibility. Overall, the architecture ensures efficient data flow, accurate predictions, secure communication, and seamless integration of ML within a distributed environment.

1. User Authentication The system begins with a secure authentication process where authorized administrators register and log in through a graphical interface. User credentials are protected using encrypted password storage to prevent unauthorized access. This step ensures that only permitted users can manage data, train models, and start prediction services within the system.

2. Dataset Upload After successful login, the administrator uploads the medical dataset in CSV format. The uploaded data is immediately displayed in the interface, allowing verification of attributes and records. This step ensures transparency and correctness before initiating any analytical operations.

3. Data Preprocessing The uploaded medical data undergoes preprocessing to improve quality and consistency. Missing values are handled, and categorical attributes are converted into numerical representations. These operations prepare the data for efficient machine learning model training and reduce the risk of biased predictions.

4. Target Identification Two critical health outcomes blood pressure category and diabetes status—are identified as prediction targets. The dataset is separated into input features and output labels for each health condition. This dual-target approach enables simultaneous analysis of multiple medical risks.

5. Class Imbalance Handling To address uneven class distributions commonly found in healthcare datasets, a clustering-based oversampling technique is applied. K Means-SMOTE generates synthetic samples for minority classes while preserving data structure. This step significantly improves model learning stability and predictive fairness.

6. Train–Test Data Splitting The balanced dataset is divided into training and testing subsets using an 80:20 ratio. The training set is used to build machine learning models, while the testing set evaluates performance. This separation ensures unbiased assessment and prevents overfitting.

7. Model Building and Training Multiple baseline classification algorithms are trained independently for each health condition. These models learn patterns from the training data and serve as reference methods for performance comparison. Training multiple models helps identify strengths and limitations of traditional classifiers.

8. Performance Evaluation Each trained model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Comparative graphs are generated to visually analyze performance differences. This evaluation helps in selecting the most reliable model for deployment.

9. Model Storage and Reuse Trained models are stored locally to avoid repeated training in future sessions. This reduces computational overhead and enables faster system execution. Stored models ensure consistency across multiple prediction cycles.

10. Real-Time Prediction Service A web-based service is activated to handle real-time prediction requests from external clients. New patient data is processed through the trained models to generate immediate health risk predictions. This step supports timely decision-making during medical emergencies.

3.1 TTC

The TTC model constructs a hierarchical tree structure where each internal node represents a decision based on a feature and each leaf node corresponds to a class label. At each split, the algorithm selects the feature that maximizes information gain or minimizes impurity, ensuring optimal separation of classes. Once the tree is built, pruning techniques are applied to remove redundant or weak branches, simplifying the final structure and preventing overfitting as shown in Fig. 2. The TTC further incorporates probabilistic refinement through a built-in optimization layer that adjusts predictions slightly to enhance stability and accuracy. This makes it particularly effective in providing transparent, interpretable, and reliable predictions for medical decision-making.

Step 1: Input Feature Representation The process begins by representing each instance as a numerical feature vector containing relevant attributes. These features serve as the basis for learning decision rules. Proper representation ensures that meaningful patterns can be extracted during tree construction.

Step 2: Root Node Initialization All training samples are initially assigned to a single root node. This node represents the complete dataset before any partitioning is performed. It serves as the starting point for recursive tree growth.

AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2026)
www.ajmimc.com

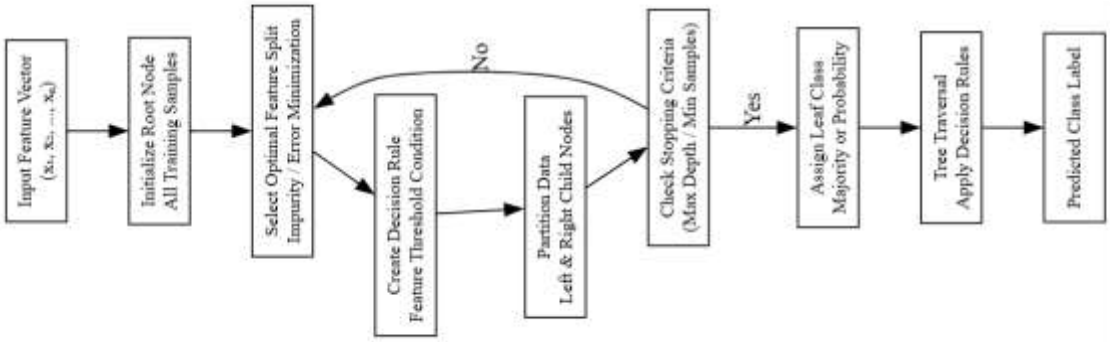

Fig. 2: Internal working of TTC.

Step 3: Optimal Feature Selection At each stage, the algorithm evaluates all candidate features to identify the best splitting attribute. The selection is based on minimizing classification error or impurity. This step ensures that each split improves class separation.

Step 4: Decision Rule Construction A decision rule is created using the selected feature and an appropriate threshold. This rule defines how data is divided into different branches. It forms the core decision logic of the tree.

Step 5: Data Partitioning The dataset is divided into child nodes based on the decision rule. Each child node contains a subset of samples sharing similar feature characteristics. This hierarchical partitioning allows the model to capture complex patterns.

Step 6: Stopping Criteria Evaluation, the algorithm checks predefined stopping conditions such as maximum tree depth or minimum number of samples. If the conditions are not met, further splitting is performed. This prevents excessive tree growth and overfitting.

Step 7: Leaf Node Assignment When stopping criteria are satisfied, the node is converted into a leaf node. A class label is assigned based on majority voting or probability estimation. This label represents the final decision for that branch.

Step 8: Tree Traversal for Prediction For prediction, new instances traverse the tree by following the learned decision rules from the root to a leaf. Each decision guides the instance to the appropriate branch. This traversal process is efficient and interpretable.

Step 9: Final Output Generation The class label assigned at the leaf node is returned as the final prediction. This output represents the model’s decision based on hierarchical rule evaluation. The result supports accurate and explainable decision-making.

Advantages

- Able to capture complex, non-linear relationships between features, making it suitable for real-world medical data.
- Provides interpretable decision rules, allowing predictions to be easily understood and explained.
- Handles both numerical and categorical attributes effectively without extensive feature transformation.
- Offers stable performance on balanced datasets by learning hierarchical decision structures.
- Supports multi-class classification naturally through tree-based decision paths.

AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2026)
www.ajmimc.com

- Requires minimal parameter tuning compared to many advanced machine learning models.

4. RESULTS DESCRIPTION

Fig. 3 shows the Tao-Tree classifier for binary diabetes prediction. Rows represent True Class (bottom to top: Y, N), and columns represent Predicted Class (left to right: Y, N). The diagonal shows correct predictions: True Y → Predicted Y: 951 (bright yellow), True N → Predicted N: 867 (lime green). Off-diagonal errors are minimal: only 8 cases of Y misclassified as N, and 10 cases of N misclassified as Y. The color scale ranges from dark purple (~200) to bright yellow (~800), highlighting the model's outstanding accuracy with just 18 total misclassifications.

Fig. 4 illustrates the prediction process carried out on Laptop 2 (client) within the client–server architecture, where the user uploads a test dataset in comma-separated values (CSV) format for analysis. The figure depicts how the client system transmits the input data to the server using HTTP and receives the predicted results in real time through the API. It highlights the processing of multiple input instances, where each row of patient data is evaluated to generate predictions for blood pressure category and diabetes status. The output demonstrates the effectiveness of the deployed ML models in handling multi-attribute medical data and returning structured prediction results.

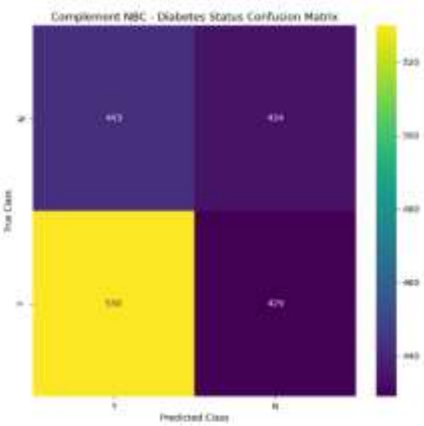

Fig. 3: Confusion matrices obtained with hypertension data using Proposed Tao-tree model.

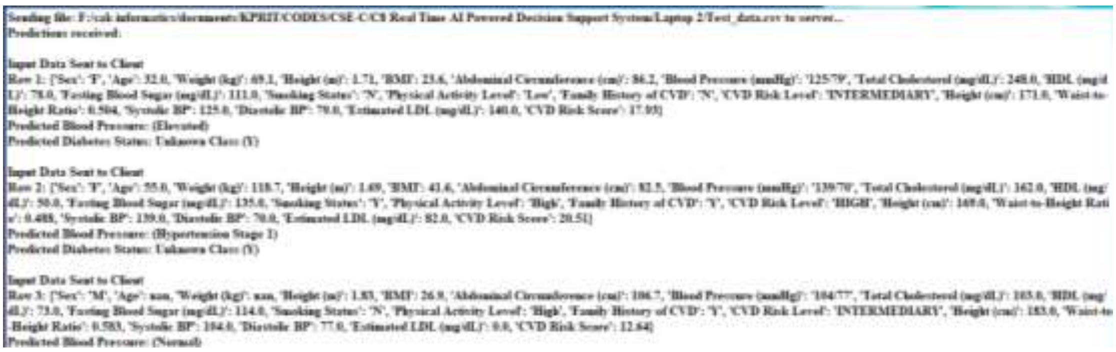

Fig. 4: Predicting Test Data in Laptop 2 (Client)

The comparative analysis as shown in table 1 & 2, evaluates the performance of various classification algorithms applied for hypertension prediction. The CNB Classifier achieved a modest accuracy of 39.07%, showing limited capability in learning complex feature interactions. Similarly, the MNB model performed slightly better with an accuracy of 46.63%, indicating marginal improvement in classification precision and recall. The Perceptron algorithm, while effective in some linear classification tasks, recorded a lower performance with 38.93% accuracy, reflecting challenges in adapting to nonlinear patterns within the dataset. In contrast, the proposed TTC demonstrated exceptional performance, achieving an impressive 98.38% accuracy, along with consistently high precision, recall, and F1-score values.

Table. 1: comparative analysis for emergency medical services.

Algorithm	Accuracy	Precision	Recall	F1-Score
Complement NBC - Hypertension	39.072	40.905	38.618	28.971
Multinomial NBC - Hypertension	46.638	49.577	46.352	44.165
Perceptron - Hypertension	38.937	37.860	38.788	35.175
TTC - Hypertension	98.386	98.378	98.386	98.377

Table 2: Classifier Performance Comparison for Diabetes Status

Algorithm	Accuracy	Precision	Recall	F1-Score
Complement NBC – Diabetes Status	52.5	52.4	52.4	52.4
Multinomial NBC – Diabetes Status	52.6	52.4	52.4	52.4
Perceptron – Diabetes Status	48.0	51.5	50.1	34.3
TTC – Diabetes Status	99.0	99.0	99.0	99.0

5. CONCLUSION

The study successfully demonstrates the effectiveness of an intelligent, data-driven approach for supporting medical decision-making in emergency scenarios. By integrating advanced preprocessing techniques, class imbalance handling, and multiple classification models, the system is able to accurately analyse patient health data and generate reliable predictions for critical conditions such as blood pressure category and diabetes status. The comparative evaluation highlights the strengths and limitations of different algorithms, providing a clear understanding of their suitability for healthcare applications. The inclusion of a tree-based learning approach further enhances prediction stability and interpretability, making the outcomes more transparent for practical use. The system’s ability to deliver real-time predictions through an interactive interface and web-based service emphasizes its applicability in time-sensitive medical environments. Overall, the work confirms that the adoption of artificial intelligence can significantly improve diagnostic support, reduce response time, and contribute to more informed and effective healthcare decision-making.

REFERENCES

[1] O’Cathain, A., Webber, E., Nicholl, J., Munro, J., & Knowles, E. NHS Direct: consistency of triage outcomes. Emergency Medicine Journal, 20, 289-292.
[2] Kuttan, N., Pundkar, A., Gadkari, C., Patel, A., & Kumar, A. (2025). Transforming emergency medicine with artificial intelligence: From triage to clinical decision support. Multidisciplinary Reviews.

[3] Masoud, M.P., Nejad, M.K., Darebaghi, H., Chavoshi, M., & Farahani, M. (2018). The Decision Support System and Conventional Method of Telephone Triage by Nurses in Emergency Medical Services: A Comparative Investigation. Int. J. E Bus. Res., 14, 77-88.

[4] Bashiri, A., Alizadeh Savareh, B., & Ghazisaeedi, M. (2019). Promotion of prehospital emergency care through clinical decision support systems: opportunities and challenges. Clinical and Experimental Emergency Medicine, 6, 288-296.

[5] Kanani, S., Mahdavi, I., Ziaie, N., & Rahimpour Cami, B. (2024). Optimization of referral system for providing medical services to cardiac patients with cardiogenic shock manifestation under uncertainty. Scientia Iranica.

[6] Schmidt Batista, A. (2025). The Transforming Clinical Practice: The Role of AI-Powered Medical Assistants in Enhancing Healthcare Efficiency and Decision-Making. Journal of Next-Generation Research 5.0, 1(4). https://doi.org/10.70792/jngr5.0.v1i4.121

[7] Lujak, M., Billhardt, H. & Ossowski, S. Distributed coordination of emergency medical service for angioplasty patients. Ann Math Artif Intell 78, 73–100 (2016). https://doi.org/10.1007/s10472-016-9507-9

[8] Tang, X.; Zou, A.; Zhang, Z.; Zhao, Y.; Zhang, X.; Cohan, A.; Gerstein, M.B. MedAgents: Large Language Models as Collaborators for Zero-shot Medical Reasoning. *arXiv* 2023, arXiv:2311.10537.

[9] Miller, M.; Jorm, L.; Partyka, C.; Burns, B.; Habig, K.; Oh, C.; Immens, S.; Ballard, N.; Gallego, B. Identifying prehospital trauma patients from ambulance patient care records; comparing two methods using linked data in New South Wales, Australia. *Injury* 2024, *55*, 111570

[10] Elfahim, O.; Edjinedja, K.L.; Cossus, J.; Youssfi, M.; Barakat, O.; Desmettre, T. A Systematic Literature Review of Artificial Intelligence in Prehospital Emergency Care. *Big Data Cogn. Comput.* 2025, *9*, 219. https://doi.org/10.3390/bdcc9090219

[11] Zota, R. D., Cîmpeanu, I. A., & Lungu, M. A. (2025). Exploring AI in Healthcare Systems: A Study of Medical Applications and a Proposal for a Smart Clinical Assistant. *Electronics*, *14*(18), 3727. https://doi.org/10.3390/electronics14183727

[12] Nickel, S.; Reuter-Oppermann, M.; Saldanha-Da-Gama, F. Ambulance location under stochastic demand: A sampling approach. *Oper. Res. Health Care* 2016, *8*, 24–32.

[13] Neira-Rodado, D.; Paz-Roa, J.C.; Escobar, J.W. Emergency Medical Services Strategic Design: A Comprehensive Multiobjective Approach to Ensure System Sustainability and Quality. *Smart Cities* 2025, *8*, 52. https://doi.org/10.3390/smartcities8020052