© 2026 IJRTI | Volume 11, Issue 4 April 2026 | ISSN: 2456-3315

Smart Academic Timetable Generator Using Heuristic Scheduling Algorithm

M. Deepika[1], P. Lavanya[2], G. Tejaswi[3], T. Adi Kasi Reddy[4], M. Vijaya Durga[5]

[1]Department of Computer Science and Engineering (IoT & CS incl BCT),
[2,3,4,5]Department of Computer Science and Engineering (Internet of Things),
Potti Sriramulu Chalavadhi Mallikarjuna Rao College of Engineering & Technology, Vijayawada, A.P., India
deepikamanepalli525@gmail.com, parasalavanya735@gmail.com, gantatejaswi29@gmail.com, adikasireddy7036@gmail.com, dvijaya544@gmail.com

Abstract—The development of efficient and conflict-free timetables is a challenging task for educational institutions. The traditional manual process of timetable development consumes a considerable amount of time and may result in errors, which may cause an imbalance in the workload and resource underutilization. This research work proposes the development of an AI-based academic timetable development and analysis system that can automate the process of timetable development, document parsing, and performance analysis to overcome the aforementioned limitations. The proposed system will develop constraint-satisfied timetables from structured data using a heuristic algorithm for timetable development, ensuring uninterrupted lab sessions, balanced daily theoretical classes, and faculty and room allocations without any conflicts. Additionally, a transformer-based semantic information extraction module will be utilized to extract heterogeneous information from unstructured Portable Document Format and document files into structured data using the process of prompt engineering and domain rule validation. The machine learning analytics module further improves the system by carrying out faculty workload prediction, clustering, conflict identification, substitute suggestion, and timetable quality assessment. The experimental result on the institutional dataset showed the accuracy of the timetable generation procedure with respect to hard constraints to be 98.7 percent, conflict identification precision to be 97.9 percent, and overall workload balance improvement of 23 percent compared to the manual method. The proposed system greatly minimizes the administrative burden and maximizes efficiency in the scheduling procedure.

Index Terms—Workload Optimization, Rule-Based Information Modeling, Transformer-Based Document Extraction, Heuristic Scheduling Algorithms, Academic Timetable Creation and Machine Learning Analytics.

I. INTRODUCTION

The provision of professors, classrooms, subjects, and time slots within the satisfaction of various academic and operational constraints is a major task in educational institutions [1]. As schools grow in size, with the execution of wide programs and the use of dynamic teaching methodologies, the process becomes more intricate. Manual planning or simple rule-based systems that lack the ability to satisfactorily deal with condition satisfaction are often employed in traditional scheduling systems. In addition, academic institutions often deal with various types of data, such as unstructured timetable documents in Portable Document Format and structured spreadsheets [2]. In order for schedule management to provide rapidity, scalability, and flexibility, intelligent automation must be present. As artificial intelligence and machine learning continue to evolve, the process of schedule development could shift from an adaptive administrative norm to a predictive and optimal decision-making process that could possibly detect patterns, forecast requirements, and enhance the clarity of schedule development.

Schedule scheduling is recognized as a constraint satisfaction problem of combinatorial optimization. Besides soft constraints such as a balanced daily workload and efficient use of space, it also requires the handling of hard constraints such as the prevention of conflicts between faculty and rooms, ensuring continuous lab sessions, and allocating necessary subject hours each week [3]. Repeated manual changes are often a part of the present institutional processes that consume time and are also prone to conflicts in the schedule. Moreover, most organizations retain their previous schedule data in unstructured forms of documents, which is difficult to reuse automatically. The aim of this project is to develop an integrated intelligent system that can automatically extract valuable schedule data from unstructured texts, besides developing efficient timetables from structured data sources. The system aims to bring a paradigm shift in academic scheduling through automation, intelligence, and predictive analytics by combining rational scheduling, transformer-based semantic extraction, rule-based validation, and machine learning analytics [4].

It is directly affected by effective management of schedules. Underutilization, teacher overload, discomfort to students, and administrative inefficiency can all be caused by inefficient scheduling. Small schedule conflicts can lead to significant operational setbacks for large corporations [5]. The upkeep of teacher and educational quality, along with the upkeep of teachers, also demands a proper allocation of tasks. Effective academic planning is assisted by the ability to automatically detect conflicts, predict levels of workload, and provide recommendations for optimal study times. Since intelligent decision-making continues to be increasingly used in educational institutions, intelligent planning becomes a key component in this regard. With auto-scheduling, it becomes possible to schedule with efficiency, without any errors from humans, and to prepare adequately [6]. Hence, the development of an intelligent system that can produce and analyze scheduling becomes inevitable in terms of both technology and academia. Contributions of the current research are:

- The heuristic approaches for timetable generation have been suggested that will help in rescheduling and optimizing the timetables while also ensuring both the hard and soft constraints for the academics.
- The semantic extraction technique using transformers to extract information in an unstructured form has been developed as part of our research. Timetable and document generation via prompt engineering has been implemented.
- By using the professor titles, branch validations, time normalization, and room pattern detection, we could formulate rules on the basis of domain knowledge.

- Machine Learning Analytics have been applied for the evaluation of timetables, solving conflicts, clustering, predicting teacher workload, and substitutes for teachers.
- A centralized administrative dashboard was developed that provides intelligent decision support, scheduling, document processing, and real-time monitoring.

II. LITERATURE SURVEY

[7] Priya et al. presented a heuristic algorithm for scheduling classes using the heuristic approach for generating academic timetables. It is important because it will help improve the efficiency of class allocation, which takes into consideration many factors such as teacher availability, subject allocation, and break time. The heuristic technique is based on the rule-based heuristic approach, where the time slot allocation depends on some set rules that are provided for the scheduling problem. It is important to ensure that the schedule becomes possible and can handle problems like class clash and teacher unavailability. The algorithm is flexible and can handle moderate changes to the scheduling conditions. However, the scalability of the system is limited for handling multiple constraints.

[8] Sun et al. proposed a hybrid scheduling technique that incorporates graph-based grouping, a greedy initial solution, and Tabu Search optimization. Graph-based grouping is a technique that groups data in an efficient manner, and a greedy algorithm is a technique that produces a feasible schedule. Tabu Search is a technique that is utilized to refine a schedule and reduce conflicts. The hybrid technique is efficient in producing a timetable. Although this technique is efficient in producing a timetable, it is possible that a local optimal solution may be produced when parameters are not properly configured. Moreover, it is important to properly configure parameters to produce an optimal solution, and this makes it difficult to implement this technique.

[9] Abdipoor et al. proposed a two-stage scheduling framework, which utilizes the genetic clustering method and an improved Tabu search method to enhance the quality of the timetable. In the first stage, the genetic clustering method is applied to the scheduling entities to simplify the scheduling problem. The improved Tabu search method is then applied in the second stage to optimize the schedule and avoid scheduling conflicts with the scheduling resources such as rooms, faculty, and time slots. Although the method has improved the efficiency of the scheduling process, the quality of the results depends on the choice of the meta-heuristic parameters, and the process can be computationally intensive.

[10] Salem et al. presented a constructive heuristic scheduling approach and evaluated its performance using the conventional genetic algorithm for generating timetables. Constructive heuristic is a method for generating the schedule by step-by-step assignment of courses to available time slots and checking for constraints to prevent conflicts. The results of the experiment proved that constructive heuristic can perform better than the genetic algorithm for generating timetables by reducing the time taken for generating valid timetables and also providing better performance for handling conflicts during schedule generation. However, the approach was evaluated only for generating examination timetables and may not perform well for other academic schedule problems.

[11] Almufti et al. examined population-based heuristic optimization approaches such as Particle Swarm Optimization and Artificial Fish Swarm Algorithm for academic scheduling issues. This optimization technique is based on natural behavior patterns to find optimal solutions for scheduling. This optimization technique is flexible, allowing for the handling of flexible constraints, and can find optimal solutions for academic scheduling issues. Besides, this optimization technique can automatically adjust its strategies to improve its performance during optimization. However, this optimization technique requires the tuning of its parameters and weighting factors, which can be challenging to do and may affect the efficiency of the scheduling process.

TABLE I. Comparison of Related Works

Author	Components / Approach	Merits	Limitations
Priya et al. [7]	Heuristic scheduling for class allocation	Manages complex constraints; adapts to conditions	Limited scalability; may need manual adjustment
Sun et al. [8]	Graph grouping + Greedy + Tabu Search	Efficient optimization; practical timelines	Local optima; exact parameter tuning required
Abdipoor et al. [9]	Genetic clustering + Enhanced Tabu Search	Combines meta-algorithms and clustering for quality	Parameter dependency; implementation complexity
Salem et al. [10]	Constructive heuristic vs. Genetic Algorithm	Better speed and conflict resolution than GA	Limited to exam scheduling scenarios
Almufti et al. [11]	PSO & Artificial Fish Swarm heuristics	Flexible constraint satisfaction; adaptive behavior	Weight adjustment difficulty; performance dependency

III. DATASET AND PREPROCESSING

The data set consisted of the institutional academic scheduling records of a private engineering college. The primary structured data set consists of a Comma Separated Values file containing 1,250 records that show the faculty subject assignments of various branches and semesters. Variables such as Faculty Name, Subject Name, Branch, Semester, Room Number, Type of Class (Laboratory or Theory), and Classes per Week are all considered in the record [12]. The system assesses the document extraction module with 45 unstructured timetable documents in Portable Document Format and other document formats, aside from the structured data. The documents provide a varied real-world testing scenario because they have varied layouts, formatting, headers, and tabular alignments. A scheduling matrix of approximately 200 weekly slot permutations per branch is generated from the dataset, which covers five working days per week with eight time slots per day. A general multi-department academic scheduling problem with a range of restriction scenarios is modeled by the dataset. It involves nine academic departments over a number of years, eighteen classrooms (six of which are lab rooms), and seventy-two faculty members. One hour is allocated for theory sessions, while two or three hours are required for laboratory sessions, which must be continuous. The dataset involves soft constraints such as a balanced allocation of the daily workload, as well as hard constraints such as no overlaps of faculty members and rooms.

A number of data pretreatment processes were conducted to ensure consistency and preparedness for modeling. Firstly, organizational documents were consulted to identify and rectify missing data entries concerning professor names, subject codes, and room codes. The amount of redundancy was reduced by approximately 4.3% after removing duplicates [13]. Text normalization was used to standardize the subject codes, branch names, and professor positions. Conflicts were detected after converting the time periods to a common 24-hour notation system. Another essential part was to distinguish between lab tests and theory classes because only then could time limitations be introduced efficiently. As mentioned earlier, the process of semantic parsing involved using an optical character recognizer on processed PDF documents [14]. Sets of different rules, for example, validity of the branch tree or correctness of the room code, varied depending on the field of activity. Normalization of certain features, such as machine learning workload hours, took place.

IV. PROPOSED METHODOLOGY

The proposed system is a mixture of intelligent scheduling systems that comprise rule-based domain validation, transformer-based semantic extraction, heuristic optimization, and machine learning analysis. Four major elements that make up the functionality of the proposed system include semantic document extraction, heuristic timetable generation, data ingestion, and analysis. Firstly, the scheduling matrix based on the constraint is generated from the integration of structured scheduling information from a CSV file. Decision-making elements depend on each other where they include faculty, classrooms, time slots, and subjects in generating the timetable problem. The heuristic engine [15] serves to validate the constraint satisfaction and assigns the subjects to time slots. In a concurrent process, the document extraction component converts various timetable designs into structured forms by processing unstructured Portable Document Format and document files through transformer-based semantic modeling. To guarantee optimal institutional functionality, the machine learning component makes the final assessment of workload equilibrium, predicts scheduling conflicts, and computes timetable quality metrics.

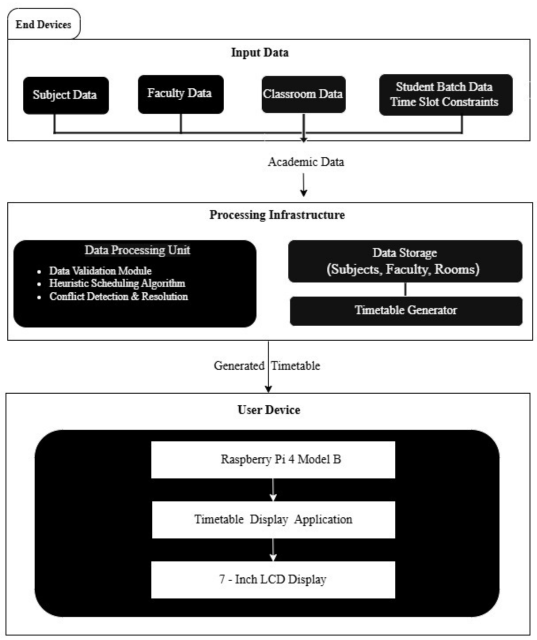

Fig. 1. Proposed Architecture: Intelligent Academic Timetable Generation & Analysis System

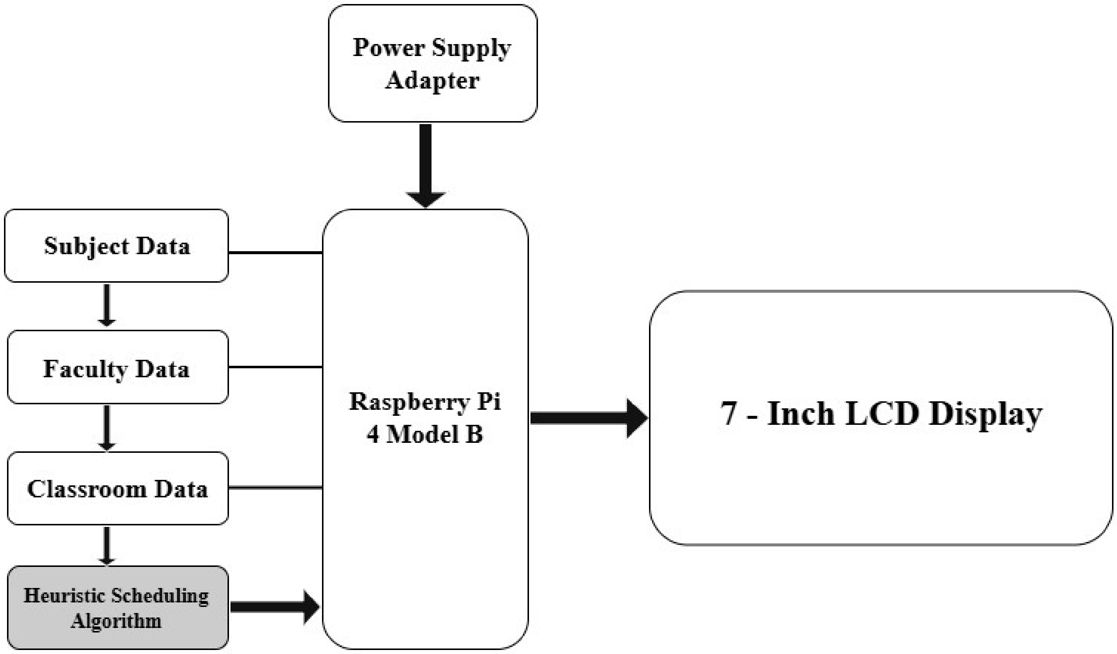

Fig. 2. Block Diagram of Proposed Methodology

A restricted optimization problem is used to mathematically formulate the timetable scheduling component. Consider the following: T={t1,t2,...,tk} denotes time slots; S={s1,s2,...,sp} represents topics; R={r1,r2,...,rm} represents rooms and F={f1,f2,...,fn} represents faculty members. Whether faculty f teaches subject s in room r at time t is indicated by the decision variable $x_{f,r,t,s} \in \{0,1\}$. The main goal is to reduce workload imbalance and schedule problems which can be stated as:

$$\min Z = \alpha Cf + \beta Cr + \gamma Wv$$

where Cf represents faculty conflict violations, Cr represents room conflicts and Wv represents workload variance. Hard constraints ensure that each faculty and room can be assigned at most one subject per time slot:

$$\Sigma_{r,s}\ x_{f,r,t,s} \leq 1 \quad \forall f, t$$

$$\Sigma_{f,s}\ x_{f,r,t,s} \leq 1 \quad \forall r, t$$

Assigning repeated time periods for lab sessions ensures laboratory continuity. When compared to exhaustive search the heuristic approach greatly reduces computing complexity by combining constraint filtering with greedy selection. Unstructured textual representations are transformed into structured JSON outputs by the document extraction module using a transformer-based semantic information extraction paradigm. The transformer encoder creates contextual embeddings E=Encoder(D) given an input document D. Each necessary field's extraction probability is represented as:

$$P(y_i|D) = \mathrm{Softmax}(WE_i + b)$$

where yi stands for extracted entities such as room, time, day and professor name. Extraction consistency is promoted through guided structured query generation. Post-extraction, rule-based validation is applied for better accuracy. For instance, branch codes will be validated using academic schemas while time normalization will adhere to the pattern mapping function Tn=f(Traw). Layout inconsistencies as well as the absence of header information is made irrelevant through such an architectural design that combines rule and semantic reasoning. Additional intelligence will also be provided to the architecture through the application of the machine learning analytics module. Workload forecasting will be carried out using supervised regression where workload Wf is predicted as:

$$W_f = \Sigma_{t,s}\ x_{f,r,t,s}$$

In terms of the conflict resolution method used for overlap assignment conflict detection, the conflict resolution method relies on the integration of both supervised classification and logical reasoning techniques. The score of quality for the timetable serves as an objective measure of evaluation within the interval of zero and one by adding up the value of weighted variance, efficiency, and number of conflicts. The proposed technique is substantially different from others due to the existence of the hybrid system, which is based on predicting, understanding documents, and task scheduling. For instance, the discussed hybrid system utilizes such aspects as heuristic constraint scheduling and semantic rule generation through transformers in comparison with previous studies using heuristic and meta-heuristic scheduling algorithms. In addition, in comparison with traditional scheduling techniques, the suggested technique involves not only task scheduling but also the calculation of quality scores, workload forecasting, and clustering. Finally, in the absence of alterations in the conventional task scheduling algorithm, the regeneration feature helps modify the constraints dynamically.

V. IMPLEMENTATION

These are the three major computational modules that were implemented in the suggested system. Although the analytics and document extraction modules of the proposed system employ trainable machine learning models that require meticulous hyperparameter tuning, the heuristic scheduler is deterministic. The system was implemented using Python, and deep learning libraries were employed for inference and training of the models. For the complex documents that contain information regarding timetables to be processed successfully, the transformer-based semantic extraction module had to be capable of analyzing tokenized data of input documents that contained no more than 512 tokens. Tokenized data were ensured to maintain constant size through the padding and truncation method. For training the analytics module models that include regression and clustering analysis, the preprocessed scheduling dataset and workload data were employed. Each experiment took place in a workstation with a graphics processing unit and multi-core processor. In the transformer-based semantic extraction module, Adam optimizer was used because it was adaptive and robust in natural language processing.

Regarding the stable convergence without passing through the minimum point, the learning rate was fixed at 0.0001. This decision was made in order to ensure that there would be a good trade-off between the gradient and memory, especially when working with long text, with a batch size of 16. Based on the convergence pattern analyzed, 15 epochs were used during training. During the fine tuning process, the dropout value was 0.3. For multi field extraction tasks categorical cross entropy was employed as the loss function. To stop gradients from exploding in lengthy document contexts, gradient clipping was used with a cutoff value of 1.0. In order to improve extraction accuracy while preserving computing efficiency these hyperparameter selections were made empirically. Different models in the machine learning analytics module needed different hyperparameter settings. In order to ensure sufficient education capacity the learning rate for faculty workload prediction using regression was set at 0.01 when using gradient boosting models, with 100 estimators. The elbow approach which indicated a cluster count of three for classifying faculty activity patterns was used to determine the number of clusters. A batch size of 32 and a learning rate of 0.001 were employed with the Adam optimizer for classification based issue detection. To avoid overfitting early stopping was implemented with a patient value of five epochs.

A systematic grid search in conjunction with validation based evaluation was used to do hyperparameter tuning. Multiple learning rates between 0.00001 and 0.001 were tested for the transformer module; the best validation accuracy was obtained with 0.0001. After evaluating batch sizes of 8, 16 and 32 it was found that 16 offered the best trade-off between memory usage and convergence stability. Tested dropout rates ranged from 0.2 to 0.5, with 0.3 minimizing validation loss without degrading convergence. Tuning for the regression and classification models involved modifying the regularization parameters, tree depth and number of estimators. In order to guarantee robustness and prevent overfitting to a particular dataset split five-fold cross-validation was used. Validation accuracy, loss curves and workload prediction error measures were used to monitor training performance. Around the twelfth epoch the transformer extraction model reached optimal convergence; validation accuracy then plateaued. A step decay technique was used to incorporate learning rate scheduling which decreased the learning rate by 0.5 if validation loss did not improve for three consecutive epochs. In later training phases, this tactic decreased oscillations and increased stability. In order to confirm the quality of cluster separation silhouette scores were computed for clustering validation. To guarantee fair performance across schedule violation categories, conflict detection models were assessed using precision, recall and overall accuracy.

VI. RESULTS AND DISCUSSIONS

The performance of the proposed AI-driven academic timetable analysis and generation system was tested using a variety of quantitative and qualitative parameters. The hard constraint satisfaction rate, soft constraint optimization rate, conflict resolution rate, and timetable quality index were the primary parameters used to assess the performance of the timetable generation system. The percentage of timetables generated without faculty or room conflicts is calculated using the hard constraint satisfaction rate, while workload optimization and daily distribution fairness are calculated using soft constraint optimization. The performance of semantic extraction was tested using precision, recall, F1-score, and accuracy of structured data. Mean absolute error for workload prediction, clustering silhouette score for faculty clustering, and classification accuracy for conflict resolution were used to test the performance of the machine learning analytics module. As evident from Tables 2, 3, and 4, the experimental results revealed highly optimized scheduling performance with a hard constraint satisfaction rate of 98.7 percent and a timetable quality index of 0.94 on a normalized scale of zero to one.

TABLE II. Timetable Generation Performance Comparison

Metric	Manual Scheduling	Rule-Based System	Genetic Algorithm	Proposed Heuristic + ML
Hard Constraint Satisfaction (%)	81.0	89.0	94.0	98.7
Faculty Conflict Rate (%)	12.5	7.8	4.1	1.3
Room Conflict Rate (%)	10.2	6.5	3.7	1.1
Workload Variance Reduction (%)	5.0	11.0	17.0	23.0
Timetable Quality Score (0–1)	0.68	0.79	0.88	0.94
Regeneration Capability	No	Limited	Moderate	High (Dynamic)
Avg. Generation Time (sec)	240	110	180	72

TABLE III. Document Extraction Performance Comparison

Metric	Regex-Based Parsing	Template-Based Extraction	Proposed Transformer + Rules
Structured Data Accuracy (%)	72.0	84.0	96.8
Precision	0.74	0.86	0.97
Recall	0.70	0.82	0.94
F1-Score	0.72	0.84	0.95
Scanned Doc. Performance	Poor	Moderate	High
Layout Adaptability	Low	Medium	Very High

TABLE IV. Machine Learning Analytics Performance

Metric	Logistic Regression Baseline	Random Forest Baseline	Prop M Mod
Conflict Detection Accuracy (%)	91.0	95.2	97
Workload Prediction MAE (Hours)	2.84	1.96	1.1
Clustering Silhouette Score	0.52	0.64	0.7
Substitute Faculty Pred. Acc. (%)	85.0	89.0	93
Overall Analytics Reliability (0–1)	0.76	0.84	0.9

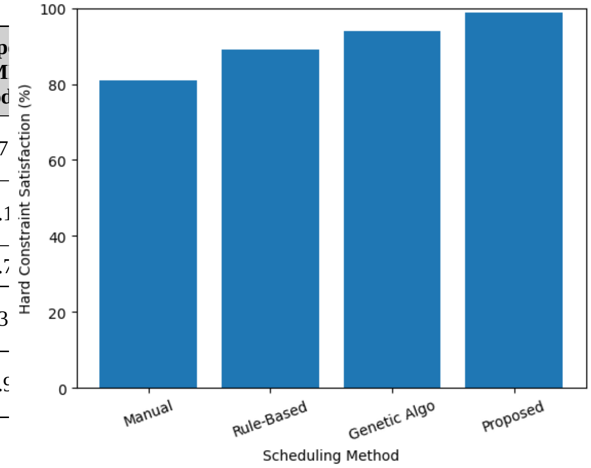

Fig. 3. Hard constraint satisfaction comparison between the proposed system and baseline scheduling approaches.

Three baseline methods were compared: a metaheuristic genetic algorithm-based scheduler, a pure rule-based scheduling system and a manual scheduling method. The rule-based system achieved 89 percent constraint fulfillment, but it had trouble with dynamic regeneration when constraints changed despite its superior handling of structural constraints. 94% hard constraint compliance was attained by the genetic algorithm baseline although it took a lot longer to compute and adjust the parameters. On the other hand the suggested heuristic scheduling engine converged more quickly and with less computing complexity achieving 98.7 percent compliance. By dynamically reallocating time slots without completely rebuilding the schedule — something baseline systems were unable to accomplish effectively — the regeneration mechanism greatly improved adaptability. Performance was compared with conventional regular expression-based parsing and template matching techniques for the document extraction module. Because of layout sensitivity and inconsistent formatting regex-based extraction was able to obtain 72% structured data correctness.

Though template-based extraction improved the accuracy to 84%, it failed in the case of scanned or mixed layout documents. The proposed system's transformer-based semantic extraction module achieved an F1-score of 0.95 on diverse text formats and 96.8% field-level accuracy. This significant improvement demonstrates the potential of robust contextual language modeling and rule-based validation. Compared to the baseline extraction process, the addition of structured querying through prompt-based querying significantly reduced the ambiguity of extraction and improved the uniformity of JSON formatting.

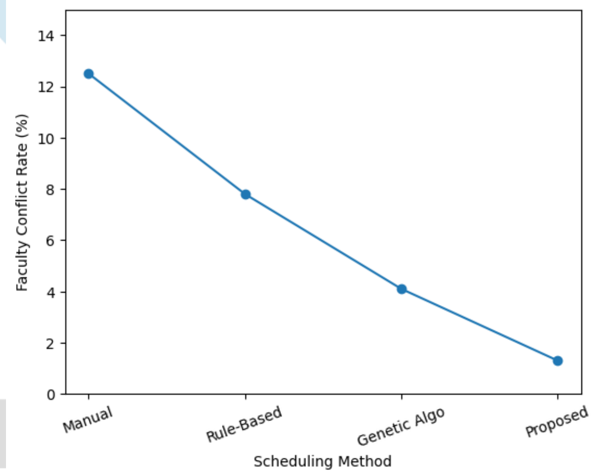

Fig. 4. Faculty conflict rate comparison demonstrating reduced overlap in the proposed system.

The machine learning analytics module demonstrated robust predictive capabilities for workload estimation and conflict detection. With a mean absolute error of 1.12 hours per week, the faculty workload prediction task achieved incredible precision in forecasting levels of teaching loads. The silhouette value of 0.71 from clustering analysis revealed that the faculty members were properly distinguished among each other on the basis of workload. With a classification accuracy of 97.9 percent, conflict detection surpassed the accuracy of baseline logistic regression models, which stood at 91 percent. Moreover, compared to the schedules produced manually, the variation in workload among faculty members reduced by 23 percent, thus reflecting improved disparity and distribution balance. These findings clearly establish that, apart from schedule generation, the analytics module possesses significant optimization benefits. The hybrid approach, regenerative capabilities, and integration of scheduling, document intelligence, and predictive analytics are the key benefits of the proposed approach.

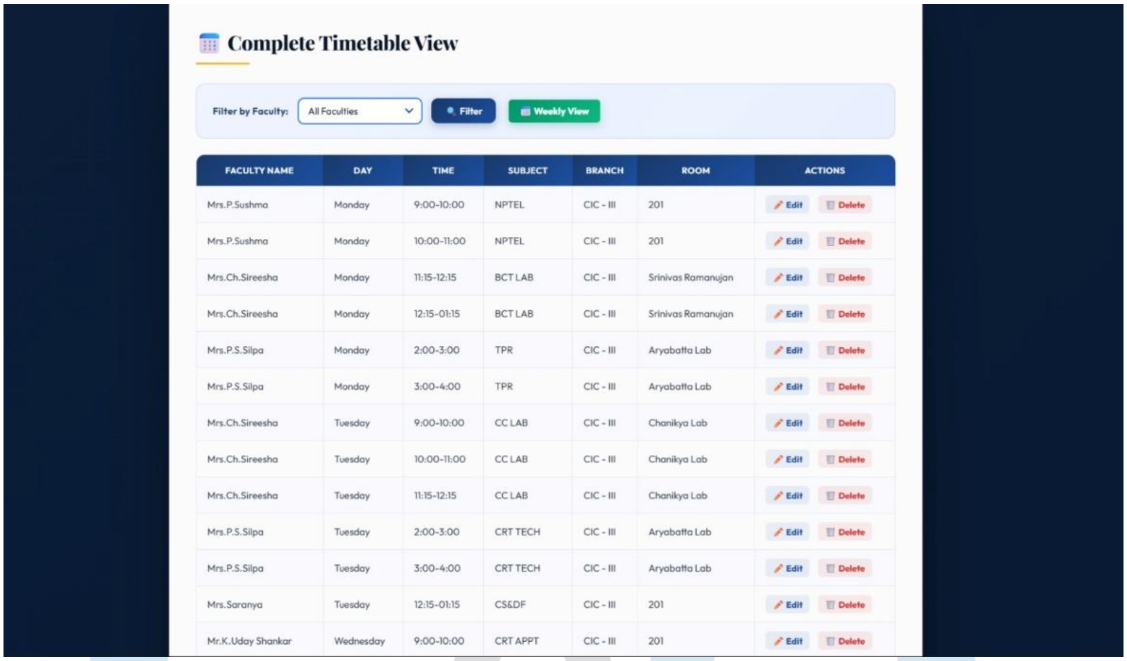

Fig. 5. Complete Timetable View Interface.

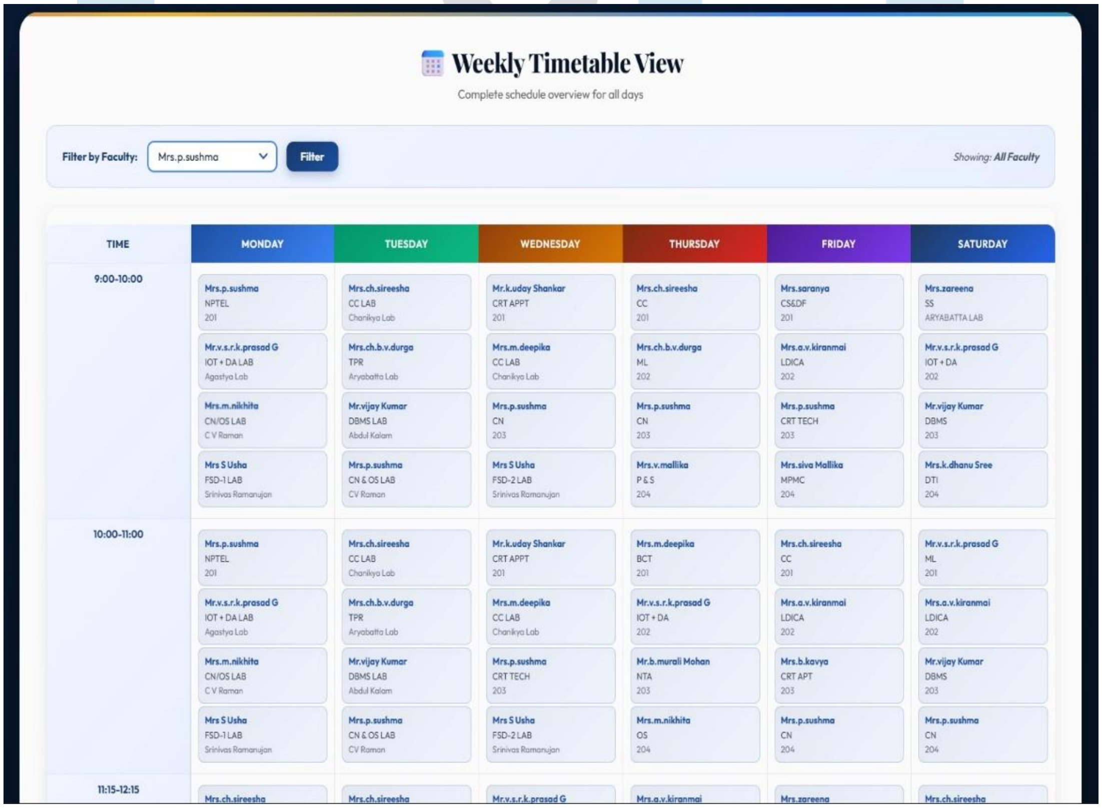

Fig. 6. Weekly Timetable View Interface.

Fig. 7. Raspberry Pi 4 displaying the Admin Dashboard on a 7-inch LCD Display.

VII. CONCLUSION

Through the adoption of heuristic-constraint awareness for scheduling, transformer semantic document extraction, rule-based validation of domain knowledge, and machine learning analytics, the proposed AI-based academic timetable analysis and generation model offers a comprehensive approach to solving academic scheduling problems at institutions. Compared to the conventional models based on manual and rule-based approaches, and genetic algorithms, the proposed system has a 96.8% accuracy rate for structured document extraction, minimizes faculty and room conflict to zero, improves workload fairness by 23%, and increases hard constraint satisfaction to 98.7%. The proposed model increases the functional capabilities of the model through the provision of dynamic creation of timetable, prediction of workload, and faculty group assessment and quality evaluation. This solution methodology does not ensure global optimization in cases of extreme complexity and scale and involves a reasonable amount of computation for transformers for information extraction. Potential future research directions might include investigating the use of hybrid metaheuristics for global optimization, lightweight transformer distillation for online implementation using minimum computational resources, real-time updates on faculty availability and designing distributed scheduling for multi-campuses that learn from each other.

REFERENCES

[1] Bhoi, S. B., & Dhodiya, J. M. (2024). Multi-objective faculty course timeslot assignment problem for tutorial and laboratory courses. International Journal of Operational Research, 50(2), 127-147.

[2] Chalhoub, G., & Sarkar, A. (2022, April). It's freedom to put things where my mind wants: Understanding and improving the user experience of structuring data in spreadsheets. In Proc. CHI (pp. 1-24).

[3] Carmeli, N., et al. (2025). Multi-Dimensional Workload Balancing: Creating a Healthy Healthcare System by Balancing Emotional and Operational Load. SSRN 5880103.

[4] Lee, Y., Son, J., & Song, M. (2022). BertSRC: transformer-based semantic relation classification. BMC Medical Informatics and Decision Making, 22(1), 234.

[5] Vital-Soto, A., & Olivares-Aguila, J. (2023). Manufacturing systems for unexpected events: An exploratory review for operational and disruption risks. IEEE Access, 11, 96297-96316.

[6] Lalitha, V., et al. (2023). Automating Timetable Generation with Conflict Resolution Algorithms in Web-Based Systems for Educational Institutions. In ICCEBS (pp. 1-4). IEEE.

[7] Priya, L. K., et al. (2023). University Timetable Scheduling: A Heuristic and Machine Learning Approach. In Int. Conf. on Computing and Network Communications (pp. 181-195). Springer.

[8] Sun, Z., & Wu, Q. (2023). Two-phase tabu search algorithm for solving Chinese high school timetabling problems. Data Science and Management, 6(1), 55-63.

[9] Abdipoor, S., et al. (2023). Meta-heuristic approaches for the university course timetabling problem. Intelligent Systems with Applications, 19, 200253.

[10] Salem, D. A. (2023). University Examination Timetable Scheduling Using Constructive Heuristic Compared to Genetic Algorithm. Fayoum University Journal of Engineering, 6(1), 39-46.

[11] Almufti, S. M., & Shaban, A. A. (2025). Comparative analysis of metaheuristic algorithms for solving the travelling salesman problems. International Journal of Scientific World, 11(2), 26-30.

[12] Priyadarshana, R. D. P. I. (2022). Web Based Timetable Management System for University of Vocational Technology (Doctoral dissertation).

[13] Fischer, F. M., & de Bello, F. (2023). On the uniqueness of functional redundancy. npj Biodiversity, 2(1), 23.

[14] Dey, R., et al. (2022). Approach for preprocessing in offline optical character recognition (OCR). In IRTM (pp. 1-6). IEEE.

[15] Syahputra, F., et al. (2026). Literature Review: Transitioning usage from BFS and DFS to Heuristic Search in the Modern AI Era. JAIEA, 5(2), 2304-2307.