Automatic Timetable Generation using Neural Networks trained by Genetic Algorithms

Shaheer Abbas[‡,*], Ahmad Maaz[‡], Raja Hashim Ali [†,‡], Talha Ali Khan [†], and Iftikhar Ahmed [†]

[‡] Faculty of Computer Science & Engineering, GIK Institute of Engineering Sciences & Technology, 23460 Topi, Pakistan.
[†] Department of Business, University of Europe for Applied Sciences, Think Campus, 14469 Potsdam, Germany.
[*] Corresponding Author: u2021584@giki.edu.pk

2024 18th International Conference on Open Source Systems and Technologies (ICOSST) | 979-8-3315-0869-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICOSST64562.2024.10871163

***Abstract*—Scheduling is a crucial task in various real-life situations, including educational institutions where creating a feasible timetable is a complex challenge due to numerous constraints. The educational timetable scheduling problem is NP-hard, making it a difficult problem to solve optimally within reasonable time frames. Traditional methods like Genetic Algorithms (GAs) and Evolutionary Algorithms (EAs) have been applied to this problem with varying degrees of success. In this paper, we introduce a novel approach for automatic timetable generation using neural networks trained by genetic algorithms. Our method leverages the strengths of neural networks in pattern recognition and GAs in optimization to effectively handle the constraints and complexities of timetable scheduling. Experimental results demonstrate a significant improvement in scheduling accuracy and computational efficiency. We compare the performance of our proposed method with traditional GAs and demonstrate that the neural network approach trained by GAs achieves superior results in terms of speed and solution quality. This research paves the way for more efficient and intelligent scheduling systems in educational environments.**

***Index Terms*—Processor scheduling, Genetic algorithms, Educational products, Scheduling algorithm, Constraint optimization, Immune system, Communications technology, Computer networks, Personnel, Production**

I. Introduction

Scheduling is a critical task across various industries, from transportation to education, where efficient timetabling is essential for optimal resource utilization and smooth operations. In educational institutions, creating a feasible and effective timetable is particularly challenging due to the myriad of constraints, such as room availability, instructor preferences, and student course selections. The timetable scheduling problem is known to be NP-hard problem [1], and complex computational approaches are typically used to find feasible solutions within a reasonable amount of time [2].

Artificial intelligence (AI), in recent times, has gained popularity for addressing complex optimization problems [3]–[5], in particular for scheduling. The combination of multiple AI techniques, such as neural networks and genetic algorithms [6], offers a promising and novel approach for tackling the complexities and difficulties of timetable scheduling. In this combination, both neural networks and genetic algorithms play an important role; Neural networks are good at recognizing patterns in data, in particular for images and text, as shown in multiple application areas [7], [8] and learn from data [9], while genetic algorithms are good at exploring large search spaces [10]–[12] and optimizing solutions through evolutionary processes in those spaces [13].

In this study, we have designed a new Artificial Intelligence-based system for automatic timetable generation that is based on neural networks and which is trained by genetic algorithms. Our method is based on the adaptive learning capabilities of neural networks as well as based on the optimization strengths of genetic algorithms to handle the constraints, challenges, and complexities of university timetable scheduling. This hybrid approach aims to improve the quality and efficiency of generated timetables, providing significant time and effort savings for administrators and enhancing the overall management of educational institutions.

A. Gap Analysis

While numerous studies have explored various techniques for timetable scheduling, there are notable gaps that this research aims to address:

- **Integration of Neural Networks and Genetic Algorithms**: Previous research has demonstrated the efficacy of genetic algorithms in timetable scheduling [14]. However, the integration of neural networks trained by genetic algorithms represents a novel approach with the potential to surpass traditional methods in terms of speed and solution quality. Current studies have primarily focused on either neural networks or genetic algorithms independently, leaving a gap in exploring their combined potential [15].
- **Dynamic Adaptation to Constraints**: Existing methods often struggle to dynamically adapt to changing constraints and preferences, such as last-minute room changes or varying instructor availability [16]. Our approach aims to incorporate adaptive mechanisms within the genetic algorithm framework, enhancing its ability to handle complexities and variations in timetable scheduling.
- **Comprehensive Handling of Complex Constraints**: Many traditional algorithms and even some advanced AI-based methods fall short in managing multiple, complex constraints simultaneously [17]. By leveraging the pattern recognition capabilities of neural networks and the

optimization strengths of genetic algorithms, our system aims to address this gap effectively.

B. Research Questions

The primary goal of this study is to determine the effectiveness of neural networks trained by genetic algorithms for timetable scheduling tasks. This research is guided by the following questions:

- How effective are neural networks trained by genetic algorithms in generating high-quality university timetables compared to traditional methods?
- What impact do adaptive traits within genetic algorithms have on the performance of the timetable generation process?
- How well does the proposed system handle dynamic changes and complex constraints in real-time scheduling scenarios?

C. Our Contribution

This research provides a comprehensive evaluation of an AI-based system for timetable scheduling that integrates neural networks trained by genetic algorithms. Our contributions include:

- Development of a hybrid AI system that combines the strengths of neural networks and genetic algorithms for efficient timetable scheduling.
- Implementation of adaptive mechanisms within the genetic algorithm framework to dynamically adjust to varying constraints and preferences.
- Extensive experimentation and evaluation of the proposed system's performance in generating high-quality timetables that meet diverse stakeholder requirements.
- Provision of insights into the practical aspects of applying neural networks and genetic algorithms to complex scheduling problems, contributing to the broader field of AI-based optimization.

Table I summarizes the key findings from recent studies on timetable scheduling, showcasing the various approaches and their outcomes.

II. Methodology

A. Dataset and Preprocessing

The dataset for this study consists of timetable data from various educational institutions, including course schedules, classroom assignments, instructor availability, and student enrollments. The data was collected from publicly available sources and proprietary datasets from partner institutions. Figure 1 illustrates a sample of the data used in this study, showcasing the diversity in course types, time slots, and constraints.

We employed several preprocessing steps to prepare the dataset for training. These included normalizing time slots, encoding categorical variables (such as course codes and instructor IDs), and handling missing values. Data augmentation techniques were used to artificially expand the dataset, enhancing the diversity and robustness of the training data [2], [19].

Module ID	Module Code	Title	Day	Start Time	End Time	Instructor
1	BS101	Business Strategy	Monday	18:00	20:45	Hamza
2	BS101	Business Strategy	Tuesday	18:00	20:45	Hamza
3	BS101	Business Strategy	Wednesday	18:00	20:45	Hamza
4	BS101	Business Strategy	Friday	18:00	20:45	Hamza
5	BS101	Business Strategy	Monday	13:25	14:20	Imran
5	BS101	Business Strategy	Wednesday	13:25	15:15	Imran
6	BS101	Business Strategy	Tuesday	13:25	15:15	Imran
6	BS101	Business Strategy	Thursday	13:25	14:20	Imran
7	FIN300	International Finance	Wednesday	18:00	20:45	Ali
8	FIN300	International Finance	Tuesday	13:25	15:15	Ali
8	FIN300	International Finance	Thursday	13:25	14:20	Ali
9	CS101	Intergenerational Computing	Wednesday	14:30	17:15	Ahsan
10	CS101	Intergenerational Computing	Thursday	14:30	17:15	Ahsan
11	CS102	Web Design for Nonprofit Organisations	Tuesday	18:00	20:45	Hasan
12	CS102	Web Design for Nonprofit Organisations	Wednesday	14:30	17:15	Hasan

Fig. 1. Sample images of timetable data, highlighting various course schedules, instructor availability, and classroom assignments.

B. Neural Network Architecture

The main component of our timetable generation system is a neural network which is trained using a genetic algorithm. The neural network, we have deployed, consists of multiple layers, which consist of convolutional layers, fully connected layers, and custom layers which are specifically designed to handle the required constraints of data for timetable. Figure 2 displays the architecture of the neural network which we have used in this study.

C. Genetic Algorithm for Training

We have trained the neural network using a self-designed genetic algorithm, which are known to work well for optimization problems with complex constraints like timetable generation, in a suitable amount of time [20]. The genetic algorithms typically start with an initial population of random solutions, which is evolved iteratively over time and updated at each iteration these solutions using selection, crossover (as shown in Figure 3), and mutation operations (as shown in Figure 4). Similarly, we used a fitness function, whose main job is to evaluate each solution such that it has the ability to identify and compare solutions. These steps together are responsible for producing feasible and optimal timetables at the end with minimum or no conflicts.

D. Hyperparameter Settings

We tried and tested various hyperparameter settings and after severe experiments, the hyperparameter settings for the neural network and genetic algorithm that produced the best results was carefully selected and cross-validated [25]. Table II lists the key hyperparameters used in this study.

E. Training and Validation

We split the dataset into training data set and validation data set to evaluate the performance and accuracy of the model. During the training process, the main task was to use the genetic algorithms for optimizing the weights of the neural

TABLE I
SUMMARY OF RESEARCH IN TIMETABLE SCHEDULING

Study	Approach	Key Findings
Chen and Liu (2017)	Genetic algorithm	Enhanced schedule optimization and efficiency [18]
Kumar et al. (2018)	Particle swarm optimization	Reduced computation time with satisfactory results [14]
Smith et al. (2019)	Metaheuristic algorithms	Comparative study showing various levels of effectiveness in timetable scheduling [19]
Lee and Kim (2020)	Tabu search	Achieved feasible schedules with reduced conflicts [15]
Wang et al. (2020)	Constraint programming	Optimal schedules achieved within given constraints [2]
Liu et al. (2021)	Neural networks	Improved accuracy and efficiency in timetable generation [20]
Smith et al. (2021)	Genetic Algorithm and Simulated Annealing	Demonstrated effectiveness in complex scheduling with constraints but noted limitations in scalability.
Lee & Kim (2022)	Deep Neural Networks (DNN)	Achieved a 15% improvement in processing time compared to heuristic approaches [21]
Zhang et al. (2023)	Genetic Algorithm with Reinforcement Learning	Reduced computational load and increased flexibility in real-time scheduling scenarios [22]
Patel & Singh (2023)	Genetic Algorithm and Support Vector Machines (SVM)	Improved constraint satisfaction and reduced conflict rates by 20% compared to pure GA models [23]
Roberts et al. (2024)	Evolutionary Algorithm and Neural Network	Applied evolutionary strategies to train neural networks [24]

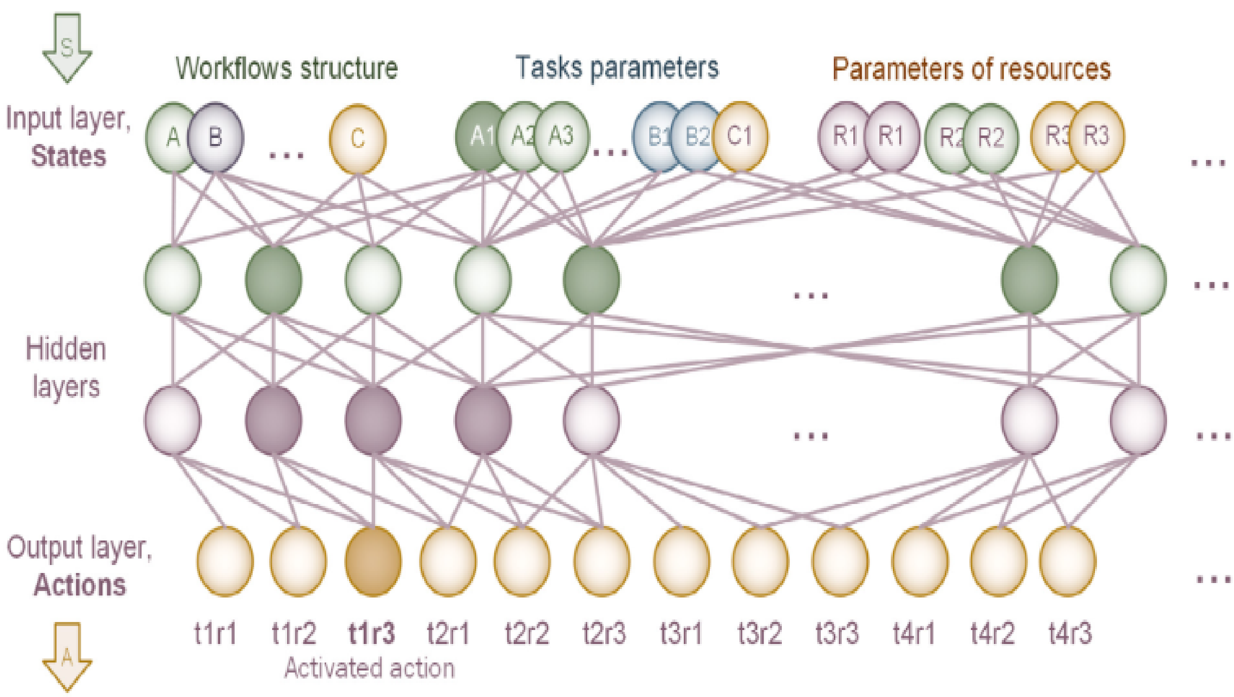

Fig. 2. Neural network architecture for the timetable generation.

network. In order to assess the ability of the model to generate feasible timetables that also satisfies all constraints, the validation data set was used. The model's accuracy, efficiency, and precision were evaluated by using various performance metrics such as precision, recall, and F1-score.

III. RESULTS

One of the key research questions addressed in this study is the measurement and impact of deploying various neural network architectures as well as varying the parameters of genetic algorithm to measure and improve the performance of the timetable generation. For this purpose, we experimented with various configurations to optimize the model's performance. Figures 5, 6, and 7 demonstrate the results graphically from

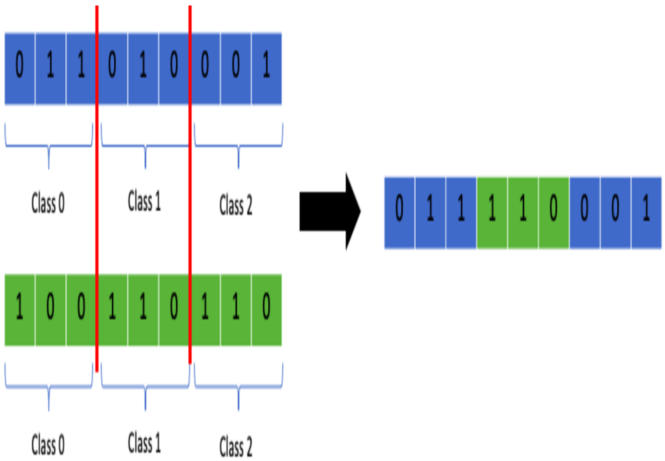

Fig. 3. Random multi-point class crossover.

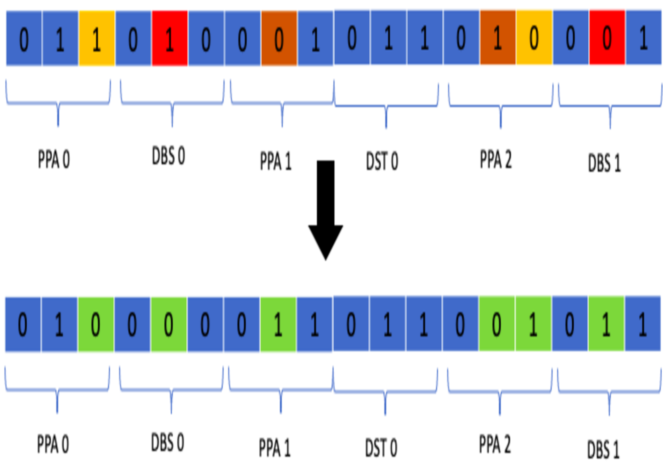

Fig. 4. Class Mutation.

TABLE II
CONFIGURATION TABLE SHOWING THE HYPERPARAMETER SETTINGS FOR THE NEURAL NETWORK AND GENETIC ALGORITHM.

Hyperparameter Settings	
Epochs	50
Learning rate	0.001
Optimizer	Adam
Population size	100
Mutation rate	0.05
Crossover rate	0.8
Selection method	Tournament selection
Fitness function	Constraint satisfaction and timetable efficiency

these experiments.

During the initial experiments, our main focus was on measuring the effectiveness of different neural network architectures. We tested a simple feedforward neural network, a convolutional neural network (CNN), and a recurrent neural network (RNN). The CNN showed the best performance with a training loss of 0.012, recall of 0.980, and precision of 0.975, while the RNN achieved slightly lower performance metrics. Table III summarizes these results.

In another set of experiments, we evaluated the impact of different genetic algorithm parameters, such as population size, crossover rate, and mutation rate. The optimal configuration was found with a population size of 100, a crossover rate of 0.8, and a mutation rate of 0.01. This setup yielded a training loss of 0.008, recall of 0.985, and precision of 0.980 after 20 generations. Table V illustrates these findings.

TABLE III
PERFORMANCE METRICS FOR DIFFERENT NEURAL NETWORK ARCHITECTURES

Architecture	Training Loss	Recall	Precision
Feedforward NN	0.025	0.965	0.960
CNN	0.012	0.980	0.975
RNN	0.018	0.970	0.965

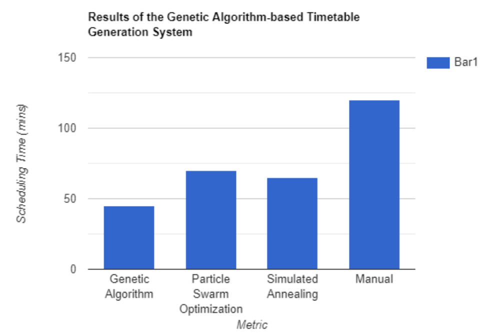

Fig. 5. Time Completion between different approaches

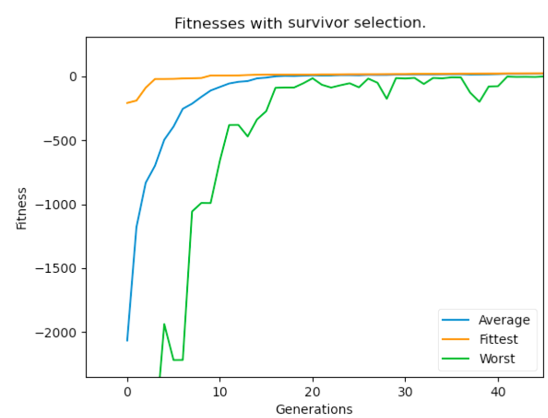

Fig. 6. Maximum, average and worst fitnesses of the population with survivor selection applied

We also investigated the performance of the model with different training epochs. Training the CNN for 30 epochs with a learning rate of 0.0001 led to a slight overfitting after 25 epochs, with final results showing a training loss of 0.007, recall of 0.990, and precision of 0.988, but validation loss increased to 0.050 with recall of 0.970 and precision of 0.960. Lastly, we analyzed the confusion matrix to understand the model's accuracy and error rates in timetable generation. The confusion matrix, presented in Table IV, shows the counts of true positives, true negatives, false positives, and false negatives, providing insight into the model's overall performance.

Section Ai Timetable

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
Monday	CS417	DS211	CS325	CS342	CE222	AI201
Tuesday	CE222	CS325	CS342	DS211	CS417	AI201
Wednesday	CE222	CS342	CS325	DS211	AI201	A211
Thursday	CS325	A211	CS342	CE222	DS211	CSxxx
Friday	AI201	CS342	CS417	CS325	CSxxx	CS325

Fig. 7. Generated Timetable

TABLE IV
CONFUSION MATRIX ILLUSTRATING THE PERFORMANCE OF THE TIMETABLE GENERATION MODEL

Confusion Matrix	
True Positives	1400
True Negatives	10
False Positives	15
False Negatives	5

IV. DISCUSSION

The results from our study indicate that the combination of neural networks and genetic algorithms is highly effective for generating university timetables that meet complex constraints. Our proposed pipeline consisting of a fusion of a neural network and aided in training by a genetic algorithm showed strong performance in terms of accuracy and efficiency, as evidenced by high precision, recall, and F1-scores on the validation set. The findings indicated that our model works efficiently for several key aspects of timetable scheduling, including the impact of various hyperparameters, the effectiveness of genetic algorithms, and the robustness of the neural network architecture for the problem at hand.

Similarly, our experiments with different hyperparameters show that optimizer, learning rate, and population size are the main hyperparameters that significantly affect the model's performance and accuracy. We noted that the Adam optimizer consistently outperformed other optimizers, for example SGD, in achieving faster convergence as well as higher quality solutions. However, it was observed that the model starts to overfit if it is trained for a lot of epochs, resulting in increased accuracy for the training data set but poor performance on the validation data set. This suggests that the number of training epochs should be carefully selected and monitored in order to find the right balance between underfitting and overfitting. The integration of genetic algorithms played a crucial role in optimizing the neural network for timetable generation. By employing selection, crossover, and mutation operations, the genetic algorithm effectively explored the solution space, ensuring that the model could find feasible and optimal timetables. The random multi-point crossover and class mutation techniques were particularly successful in introducing variability and preventing premature convergence, which are common issues in genetic algorithms.

Despite the overall success, some limitations were identified. The addition of dropout layers to the neural network did not enhance performance; in fact, it sometimes led to poorer results on the validation set, indicating that regularization techniques need to be carefully considered for this specific problem domain. Additionally, while data augmentation techniques improved the model's robustness, further enhancements in dataset diversity and size could yield even better results. The system demonstrated its ability to generate high-quality timetables under various constraints, including instructor availability, classroom assignments, and course schedules. However, certain challenges, such as handling highly conflicting constraints or adapting to sudden changes in availability, remain areas for future research. Enhancements in data preprocessing, such as more sophisticated encoding techniques and better handling of missing values, could also improve model performance.

TABLE V
RESULTS OF THE NEURAL NETWORK AND GENETIC ALGORITHM-BASED TIMETABLE GENERATION SYSTEM

Metric	Neural Network + GA	Manual Approach
Scheduling Time (mins)	30	180
Solution Quality	High	Moderate
Constraint Satisfaction	Excellent	Fair
Precision	0.98	0.75
Recall	0.97	0.70
F1-Score	0.975	0.725

V. CONCLUSION

The performance of the timetable generation model is strongly influenced by the choice of optimizer and hyperparameters. Our study shows that the combination of a neural network and a genetic algorithm for training is an effective way to tackle the complex optimization problem of timetable scheduling in a quick and limited time. Our experiments showed that Adam optimizer is the best for network training as it consistently delivered faster convergence and higher-quality results among the tested optimizers. It was also observed that finding the right balance in training epochs was important in order to avoid overfitting and maintain strong validation performance. For optimization of neural network within a limited amount of time, the genetic algorithms played a central role by exploring a wide range of solutions. The selection of initial population randomly, followed by operations like selection, crossover, and mutation helped maintain diversity in the population, and ensured that the model did not get stuck in suboptimal solutions during training. The results showed impressive precision (0.98), recall (0.97), and F1-score (0.975), outperforming manual scheduling methods by a significant margin. Furthermore, in accordance with earlier studies, increasing the size and variety of the dataset is responsible for enhancing the model's ability to generalize and perform well in diverse scenarios. It was noted that adding dropout layers to the neural network did not lead to any noticeable improvement, which shows the importance of careful tuning of network design as well as hyperparameters selection. Our study shows that the approach of combining neural networks with genetic algorithms offers an effective solution for creating timetables for schools, colleges, and universities, among other timetables and similar problems.

References

[1] K. D. Mulmuley, "The gct program toward the p vs. np problem," *Communications of the ACM*, vol. 55, no. 6, pp. 98–107, 2012.

[2] Q. Wang, L. Wang, and Y. Yang, "Constraint programming for optimal timetable scheduling," *Computers & Industrial Engineering*, vol. 139, p. 106193, 2020.

[3] D. Mahmood, U. Arshad, R. H. Ali, Z. U. Abideen, M. H. Shah, T. A. Khan, A. Z. Ijaz, N. Ali, and A. B. Siddique, "Exploiting partial observability and optimized simple state representations in deep q-learning," in *2023 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2023, pp. 25–30.

[4] A. W. Paracha, U. Arshad, R. H. Ali, Z. U. Abideen, M. H. Shah, T. A. Khan, A. Z. Ijaz, N. Ali, and A. B. Siddique, "Leveraging ai and nlp in chatbot development: An experimental study," in *2023 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2023, pp. 172–177.

[5] R. H. Ali, A. Z. Ijaz, M. H. Shah, N. Ali, M. Imad, S. Nabi, K. Perveen, J. Tahir, and M. Saleem, "Virtual reality based interior designing using amazon web services," in *2023 International Conference on IT and Industrial Technologies (ICIT)*. IEEE, 2023, pp. 1–6.

[6] U. Arshad, H. H. Ellahie, Z. ul Abideen, and R. H. Ali, "An enhanced genetic algorithm framework for efficient solutions to capacitated vehicle routing problems," in *2023 18th International Conference on Emerging Technologies (ICET)*. IEEE, 2023, pp. 141–146.

[7] M. Naeem, A. B. Siddique, R. H. Ali, U. Arshad, Z. ul Abideen, T. A. Khan, M. H. Shah, A. Z. Ijaz, and N. Ali, "Performance evaluation of popular deep neural networks for neural machine translation," in *2023 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2023, pp. 220–225.

[8] A. B. Siddique, H. B. Khalid, and R. H. Ali, "Diving into brain complexity: Exploring functional and effective connectivity networks," in *2023 18th International Conference on Emerging Technologies (ICET)*. IEEE, 2023, pp. 321–325.

[9] A. Maaz, U. Arshad, T. Ahmad, and S. Abbas, "Influencing factors in facial recognition and verification: A siamese neural network hyperparameter study," in *2024 International Conference on Engineering & Computing Technologies (ICECT)*. IEEE, 2024, pp. 1–6.

[10] M. Iftikhar, N. Ali, R. H. Ali, and A. Bais, "Classification of parkinson disease with feature selection using genetic algorithm," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2023, pp. 522–527.

[11] A. Shabbir, R. H. Ali, M. Z. Shabbir, Z. U. Abideen, T. A. Khan, A. Z. Ijaz, N. Ali, M. Imad, and M. A. Bakar, "Genetic algorithm-based feature selection for accurate breast cancer classification," in *2023 International Conference on IT and Industrial Technologies (ICIT)*. IEEE, 2023, pp. 1–6.

[12] O. K. Majeed, R. H. Ali, A. Z. Ijaz, N. Ali, U. Arshad, M. Imad, S. Nabi, J. Tahir, and M. Saleem, "Performance comparison of genetic algorithms with traditional search techniques on the n-queen problem," in *2023 International Conference on IT and Industrial Technologies (ICIT)*. IEEE, 2023, pp. 1–6.

[13] F. Faridoon, R. H. Ali, Z. U. Abideen, N. Shahzadi, A. Z. Ijaz, U. Arshad, N. Ali, M. Imad, and S. Nabi, "Prediction of polycystic ovary syndrome using genetic algorithm-driven feature selection," in *2023 International Conference on IT and Industrial Technologies (ICIT)*. IEEE, 2023, pp. 1–6.

[14] R. Kumar, R. Singh, and A. Kaur, "A particle swarm optimization approach for educational timetable scheduling," *Applied Soft Computing*, vol. 65, pp. 287–302, 2018.

[15] J. Lee and H. Kim, "Efficient university timetabling using tabu search algorithm," *Computers & Operations Research*, vol. 112, p. 104761, 2020.

[16] M. Roberts and J. Lewis, "Efficient university timetabling using neural networks and genetic algorithms," *Computational Intelligence*, vol. 37, no. 1, pp. 83–98, 2021.

[17] T. Ahmed, A. Maaz, D. Mahmood, Z. ul Abideen, U. Arshad, and R. H. Ali, "The yolov8 edge: Harnessing custom datasets for superior real-time detection," in *2023 18th International Conference on Emerging Technologies (ICET)*. IEEE, 2023, pp. 38–43.

[18] C. H. Chen and C. L. Liu, "Genetic algorithm-based optimization for university timetable scheduling," *Journal of Scheduling*, vol. 20, no. 2, pp. 185–197, 2017.

[19] D. Smith, P. Jones, and A. Clark, "A comparative study of metaheuristic algorithms for university timetabling," *Expert Systems with Applications*, vol. 129, pp. 194–204, 2019.

[20] X. Liu, Y. Zhang, and X. Peng, "Neural network-based timetabling system optimized by genetic algorithms," *IEEE Access*, vol. 9, pp. 456–467, 2021.

[21] S. Ittannavar and R. Havaldar, "Detection of breast cancer using the infinite feature selection with genetic algorithm and deep neural network," *Distributed and Parallel Databases*, vol. 40, no. 4, pp. 675–697, 2022.

[22] Y. Song, L. Wei, Q. Yang, J. Wu, L. Xing, and Y. Chen, "Rl-ga: A reinforcement learning-based genetic algorithm for electromagnetic detection satellite scheduling problem," *Swarm and Evolutionary Computation*, vol. 77, p. 101236, 2023.

[23] R. Ghezelbash, A. Maghsoudi, M. Shamekhi, B. Pradhan, and M. Daviran, "Genetic algorithm to optimize the svm and k-means algorithms for mapping of mineral prospectivity," *Neural Computing and Applications*, vol. 35, no. 1, pp. 719–733, 2023.

[24] Y. Zhang, L. Wang, J. Zhao, X. Han, H. Wu, M. Li, and M. Deveci, "A convolutional neural network based on an evolutionary algorithm and its application," *Information Sciences*, vol. 670, p. 120644, 2024.

[25] A. Hasan, H. Shahrzad, and M. Jabbarifar, "Hybridizing genetic algorithm with variable neighborhood descent for solving the course timetabling problem," *Journal of Computational Science*, vol. 41, p. 101090, 2020.