# Beyond the Tracks: Evolving Rail Networks Using Evolutionary Genetic Algorithm

Adrin K Jigy[#1], Nandaja P[#2], Sonu S[#3] , Tijo K Abraham[#4]
*Computer Science Engineering, College of Engineering*
*Kallooppara*
*Kerala, India*
*adrinkjigy@gmail.com , nandajap622@gmail.com,*
*sonuspillai8@gmail.com, tijoabraham2002@gmail.com,*

***Abstract*—The Indian railway system, one of the world's largest and oldest, faces multifaceted challenges ranging from operational inefficiencies to safety concerns. In response, this paper proposes an AI-enabled track optimization system designed to revolutionize the operational landscape of the Indian Railways. Leveraging artificial intelligence, specifically genetic algorithms, the system aims to meticulously optimize train tracks, mitigate delays, streamline schedules, and prevent potential collisions. The proposed system adopts a systematic approach comprising encoding, fitness function development, genetic operators implementation, collision detection mechanisms, simulation, and ongoing evaluation. Through the encoding phase, intricate details of train schedules and routes are represented within the genetic algorithm population. A carefully crafted fitness function evaluates solutions against objectives such as minimizing delays, optimizing throughput, and ensuring safety. Genetic operators including selection, crossover, and mutation facilitate the generation of improved candidate solutions. Sophisticated collision detection mechanisms are integrated to swiftly identify and prevent potential train collisions. Simulation and evaluation processes comprehensively assess the system's performance considering various factors like train speeds, track capacities, and signaling systems. Through iterative genetic algorithm processes, the system progressively converges towards optimal solutions, systematically reducing delays and enhancing efficiency. Continuous monitoring and adaptation mechanisms enable dynamic evolution, ensuring sustained efficiency improvement. Anticipated benefits encompass significant improvements in punctuality, reduced overcrowding, enhanced efficiency for both freight and passenger services, and cost optimization. Furthermore, the paper delves into three distinct modules: AI-enabled timetable generation, train delay prediction utilizing random forest regression, and self-driving train simulation using NEAT Python. Each module contributes uniquely to the overarching goal of transforming the Indian railway system into a global benchmark of reliability, safety, and efficiency. The proposed AI-enabled track optimization system holds the potential to redefine the Indian Railways' operational paradigm, paving the way for enhanced reliability, safety, and efficiency on a monumental scale.**

***Keywords*—NEAT, AI.***

## I. INTRODUCTION

The Indian railway system stands as a testament to the nation's rich history and enduring spirit of connectivity. Spanning vast distances and traversing diverse landscapes, it serves as a lifeline for millions, facilitating transportation of goods and people with unparalleled scale and reach. However, beneath its iconic stature lies a network grappling with multifaceted challenges that threaten its efficiency, safety, and sustainability. The Indian Railways, while a symbol of national pride, faces persistent issues of overcrowding, financial losses, delays, and safety concerns. The sheer volume of trains crisscrossing the vast expanse of the subcontinent presents a logistical puzzle of monumental proportions. Timetable inaccuracies, operational inefficiencies, and organizational errors exacerbate these challenges, leading to suboptimal service delivery and occasional catastrophic incidents. In recent years, the urgency to address these challenges has grown exponentially. As India embraces rapid urbanization and economic growth, the demand for efficient, reliable, and safe transportation infrastructure has never been more pressing.The need for transformative interventions within the Indian railway system has become imperative, calling for innovative solutions that can revolutionize its operational landscape. In response to this clarion call, this paper proposes an AI-enabled track optimization system poised to usher in a new era of efficiency and safety for the Indian Railways. By harnessing the power of artificial intelligence, specifically genetic algorithms, this system endeavours to meticulously optimize train tracks, mitigate delays, streamline schedules, and avert potential collisions. Through a comprehensive approach encompassing encoding, fitness function development, genetic operators implementation, collision detection mechanisms, simulation, and ongoing evaluation, the system aims to address the myriad challenges plaguing the Indian railway system. The transformative potential of this AI-enabled track optimization system extends far beyond mere operational improvements. It envisions the Indian Railways not only as the largest but also as the preeminent railway system globally, setting new benchmarks in reliability, safety, and efficiency. With its ability to optimize resources, enhance punctuality, and ensure passenger safety, this system holds the promise of transforming the Indian railway system into a model of excellence for the world to emulate. The primary focus lies in addressing the critical challenges of overcrowding, financial losses, delays, unfavourable timetables, and safety concerns within the existing railway network. By employing cutting-edge technology, specifically genetic algorithms, the system aims to optimize train tracks, ensuring minimal delays, efficient schedules, and the prevention of collisions.

## II. Problem statement

The major issue faced by the Indian railway is overcrowding and yet it is running at financial loss. The absence of track optimization is to blame for this issue, which causes delays, unfavourable train timetables, and safety issues. Due to a lack of technology and organizational errors, there have been a number of catastrophic instances with numerous trains colliding in recent years. To address this issue, we propose the implementation of an AI-enabled system that utilizes genetic algorithms to optimize train tracks, minimizing delays, optimizing schedules, and avoiding collisions. The aim is to enhance the effectiveness of the Indian railway network while taking limitations like station capacity, track maintenance schedules, and safety standards into account. y optimizing train tracks, minimizing delays, streamlining schedules, and proactively preventing collisions. The overarching goal is to elevate the efficiency and efficacy of the Indian railway network while meticulously considering constraints such as station capacity, track maintenance schedules, and stringent safety standards. By harnessing the power of advanced artificial intelligence and genetic algorithms, our solution aspires not only to transform the Indian railway system into the largest but also the most advanced and safest railway network globally. Through the prevention of delays, meticulous timetable optimization, and a steadfast commitment to safety, this system promises substantial benefits for both the Indian Railways and its passengers, heralding a new era of punctuality, reduced overcrowding, enhanced efficiency, and cost optimization .

## III. Objective

As the AI-enabled track optimization system envisioned for the Indian Railway, the primary objective is to revolutionize the current railway operations by addressing the critical issues of overcrowding, financial losses, delays, and safety concerns. The overarching goal is to enhance the efficiency of the Indian railway network while adhering to constraints such as station capacity, track maintenance schedules, and safety standards. The proposed system aims to achieve this by employing advanced technologies, particularly genetic algorithms, to optimize train tracks effectively. The key objectives include the development of a robust encoding scheme to represent train schedules, routes, and relevant factors as individuals in the genetic algorithm population. This encoding will capture essential information such as departure times, routes, and speeds, laying the foundation for comprehensive optimization. A crucial aspect of the system is the formulation of a fitness function that meticulously evaluates solutions based on defined objectives. These objectives encompass minimizing delays, maximizing train throughput, optimizing travel time, and ensuring safety by avoiding collisions. The assignment of higher fitness values to solutions aligning with these objectives will drive the evolution of more efficient and secure railway systems. The genetic operators, including selection, crossover, and mutation, play a pivotal role in generating new candidate solutions. Through these mechanisms, the system favors the survival of the fittest solutions, combining genetic information from successful parents to create potentially improved offspring. The introduction of mutation injects an element of exploration, facilitating the discovery of novel solutions. To address safety concerns, the incorporation of collision detection mechanisms is imperative. Efficient algorithms and data structures are employed to identify potential collisions between trains and trigger preventive actions, mitigating the risk of catastrophic incidents. The subsequent simulation and evaluation processes simulate the railway system using generated solutions, considering factors such as train speeds, track capacities, station dwell times, and signalling systems. This comprehensive assessment aims to accurately gauge the impact of different routes and schedules on efficiency, collision avoidance, and overall system performance. Through iterative refinement, the genetic algorithm process converges towards optimal solutions, continually reducing delays and improving efficiency over multiple iterations. Continuous monitoring and adaptation are essential components of the system, allowing for real-time adjustments based on performance feedback. Regular updates to the genetic algorithm with new data and parameter adjustments ensure adaptability to changing conditions, contributing to a continuous improvement cycle. The anticipated benefits of this transformative initiative include significant improvements in punctuality, reduced overcrowding, enhanced efficiency for both freight and passenger services, and cost optimization for Indian Railways, ultimately positioning it as not only the biggest but also the best railway system globally.

## IV. Proposed System

The Proposed System is divided into 3 modules based on functionality.

### *A. Train delay prediction using random forest regression*

The "Train-Delay-Prediction" project focuses on analyzing historic train data to gain insights and develop a predictive model for predicting train delays. By leveraging machine learning algorithms like Random Forest Regressor and Gradient Boosting Regressor, the project aims to predict whether a train will be delayed based on various characteristics. The objective of the project is to perform analysis of the historic train data to gain valuable insights and build a predictive model to predict whether a train will be delayed or not given a set of train characteristics. The objective of the predictive model is to build a model to predict whether a train will be delayed or not based on certain characteristics of the train. Such a model may help both passengers as well as railways to predict future delays and minimize them. The model is trained with a dataset which contains train data including train names, station names, timings and historical data about train delays. The output shows whether the train is delayed or not depends on the time; if the train is late by 15 minutes or more then it is declared as delayed.

Technologies Used:
1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn
5. Sklearn
6. Random Forest Regressor from Sklearn
7. Gradient Boosting Regressor from Sklearn

a. Random Forest Regression

Random Forest Regression is a machine learning algorithm used for regression tasks, which involves predicting a continuous value based on input features. It is an ensemble learning method, meaning it combines the predictions of multiple individual models (trees) to improve the overall accuracy and robustness of the model.

Here's how the Random Forest Regression algorithm works:

1. Random Forest Overview: A Random Forest is made up of a collection of decision trees. Each tree in the forest is trained independently on a random subset of the data (bootstrap sample) and a random subset of the features. This randomness helps to ensure that each tree in the forest is different from the others.

2. Decision Trees: Each decision tree in the Random Forest makes a prediction by recursively splitting the data into subsets based on the feature values. The splits are chosen to minimize the variance of the target variable (i.e., reduce the error in prediction). The final prediction of the tree is the average (for regression) of the target values of the training instances in the leaf node.

3. Combining Predictions: In a Random Forest, the predictions of all the trees are combined to make the final prediction. For regression tasks, this is typically done by averaging the predictions of all the trees in the forest.

4. Advantages of Random Forest Regression:

- Handles large datasets with higher dimensionality well.
- Can handle missing values in the dataset.
- Reduces overfitting compared to a single decision tree.
- Provides feature importances, which can be useful for understanding the impact of different features on the prediction.

Overall, Random Forest Regression is a powerful and versatile algorithm that is widely used for regression tasks in various fields, including finance, healthcare, and transportation.

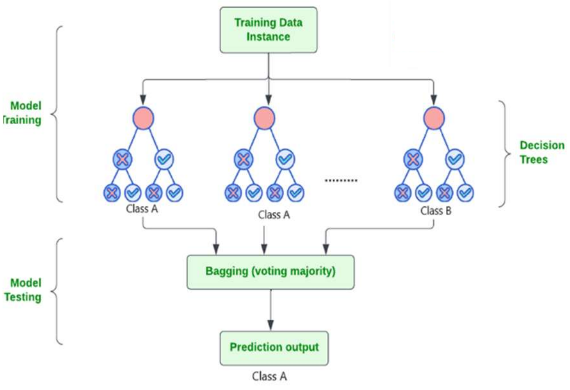

*Figure 1 Random forest regression*

*b.* Advantages of Random Forest Regressor from Sklearn*:*

- High Accuracy: Random forests typically have higher accuracy compared to single decision trees. They reduce overfitting by averaging multiple trees.
- Robustness: They are less likely to overfit on the training data compared to other models. They handle outliers and noisy data well.
- Feature Importance: Random forests provide a feature importance score, which helps in understanding the relative importance of different features in making predictions.
- Efficiency: They can handle large datasets with higher dimensionality efficiently. The training process can be parallelized, making it faster.
- Versatility: Random forests can be used for both regression and classification tasks. They also work well with both numerical and categorical features.
- Reduced Variance: By averaging multiple trees, random forests reduce the variance of the model, making them more stable and less sensitive to noise in the data.

c. Advantages of Gradient Boosting Regressor from Sklearn:

- .High Predictive Power: Gradient Boosting Regressor typically provides higher predictive power compared to other algorithms due to its ensemble nature.
- .Handles Non-linear Relationships: It can capture complex non-linear relationships in the data, making it suitable for a wide range of regression problems.
- Robustness to Outliers: Gradient Boosting is robust to outliers in the data, as it builds trees sequentially, focusing on reducing errors from previous trees.
- Feature Importance: Similar to Random Forests, Gradient Boosting provides feature importance scores, helping in feature selection and understanding the data.
- Less Preprocessing Required: It can handle missing data and does not require extensive preprocessing of the data such as normalization or scaling.
- Flexibility: Gradient Boosting can be used for both regression and classification tasks, making it a versatile algorithm.

In the train delay prediction project, Random Forest Regression model is implemented using the RandomForestRegressor class from the sklearn.ensemble module. Here's a breakdown of how it is implemented:

- .Loading Data: The loadData function reads the data from CSV files (trains.csv and stations.csv) using pandas. It preprocesses the data by dropping unnecessary columns and encoding categorical variables.
- Preprocessing: The preprocessing function splits the data into features (X) and target (Y) variables. It further splits the data into training and testing sets using train_test_split function from sklearn.model_selection.

- Training the Random Forest Regressor: The rfg function creates an instance of RandomForestRegressor. It fits the model on the training data using the fit method.
- Accepting User Input: The accept_data function takes user input for month, day, scheduled departure time, distance, arrival delay, train code, origin station code, destination station code, and day of the week.
- Making Predictions: The prediction function prepares the input vector for the trained model based on the user input. It then uses the trained RandomForestRegressor model to predict whether the train will be delayed or not based on the input data.
- Displaying Prediction: The main function is the entry point of the application. It uses Streamlit to create a web interface where users can select the machine learning model (in this case, only the Random Forest Regressor is available) and input their data.

The main use of the  project is to provide a predictive model for estimating train delays, which can be valuable for both railways and passengers. By analyzing historical train data and considering various factors such as departure time, railway, and station, the model can forecast the likelihood of a train being delayed by 15 minutes or more. This information can help railways to optimize their scheduling and operations, potentially reducing delays and improving overall efficiency. For passengers, knowing the likelihood of a delay can aid in planning travel itineraries and making informed decisions. Overall, the project aims to enhance the efficiency and reliability of train travel by leveraging machine learning algorithms to predict and mitigate train delays.

### *B. Self driving train using NEAT python*

This module simulates a self-driving train using Neuro Evolution of Augmenting Topologies (NEAT) in Python. The core logic for the train's behaviour and NEAT implementation is independent of any specific graphics library. This allows for flexibility in designing the visual representation of the train and its environment using any content creation platform.

1. NEAT and Artificial Neural Networks (ANNs)

NEAT is an evolutionary algorithm used to train artificial neural networks. It works by creating a population of neural networks with varying structures and connections. These networks are then evaluated based on their performance in a specific task (in this case, controlling the train). The networks with the best performance are then used to breed a new generation, with the goal of continually improving the population's overall performance.

2. Project Components:

Train: The train object represents the physical entity within the simulation. It has attributes like position, speed, and direction.

Environment: This defines the world where the train operates. It can include elements like tracks, stations, and obstacles. The environment representation needs to be compatible with the input format of the neural network. (Details on this format will be covered in section.

.NEAT Implementation: This core module utilizes the neat-python library to manage the population of neural networks, their training, and selection for breeding.

Fitness Function: This function determines the performance of a specific neural network during evaluation. In this case, a good fitness function might consider factors like staying on track, reaching the destination efficiently, and avoiding obstacles.

3. Input and Output for the Neural Network

The neural network used in this project takes in information about the train's environment and outputs control signals. Here's a breakdown of a possible structure:

Input:

- Train position (relative to the track)
- Distance to the next station
- Presence of obstacles ahead (binary values for different directions)
- Additional environmental features can be included based on your specific simulation needs.

Output:

- Train acceleration/deceleration
- Change in direction (optional, depending on track complexity)

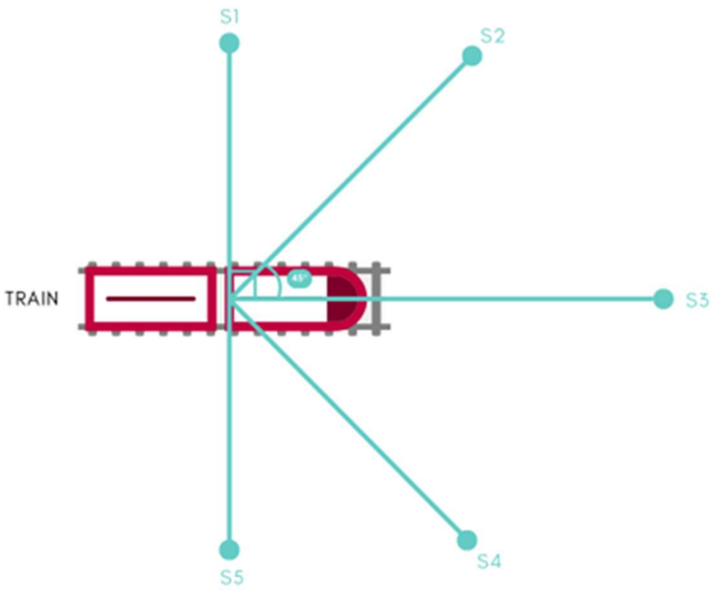

*Figure 2 Train model with sensors*

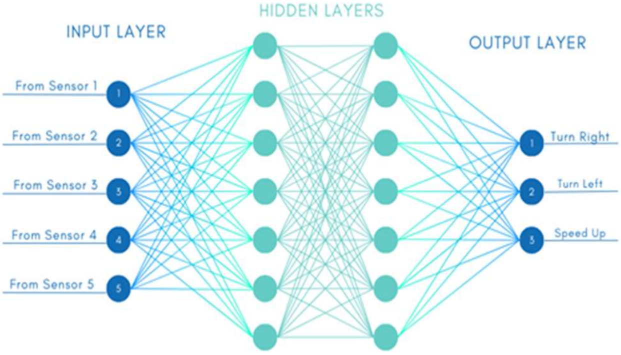

*Figure 3 Artificial neural network*

4. Training Process

- Initialization: Create a population of neural networks with random topologies.
- Evaluation: For each network in the population: Simulate the train's movement for a certain duration using the network's outputs as control signals. Calculate the fitness score based on the defined fitness function.
- Selection: Choose the top-performing networks (based on fitness) for breeding.
- Breeding: Apply NEAT's genetic algorithms to create a new generation of networks with improved characteristics inherited from the previous generation.
- Repeat: Go back to step 2 and continue training for a set number of generations or until a desired performance level is achieved.

5. Visualization

While the core logic is independent of UI visuals, creating a user interface (UI) to represent the simulation can be beneficial. This UI could be built using any graphics library or framework. It would typically show the train moving on the pre-designed map, potentially highlighting relevant information like the current fitness score or generation number.

This project demonstrates the application of NEAT to train a self-driving train in a simulated environment. The modular design allows for customization of the environment and the neural network's input/output structure. This project serves as a foundation for exploring further complexities in train behavior and environment design.

### *C. Train Timetable Optimization Using Genetic Algorithm*

The train timetable optimization aims to enhance efficiency, safety, and service quality within the Indian railway network. The approach involves encoding train schedules and routes into genetic algorithm populations, defining a fitness function to evaluate solutions based on objectives like minimizing delays, optimizing throughput, and ensuring safety. Genetic operators such as selection, crossover, and mutation are applied to generate improved solutions.

The key objectives include the development of a robust encoding scheme to represent train schedules, routes, and relevant factors as individuals in the genetic algorithm population. This encoding will capture essential information such as departure times, routes, and speeds, laying the foundation for comprehensive optimization. A crucial aspect of the system is the formulation of a fitness function that meticulously evaluates solutions based on defined objectives. These objectives encompass minimizing delays, maximizing train throughput, optimizing travel time, and ensuring safety by avoiding collisions. The assignment of higher fitness values to solutions aligning with these objectives will drive the evolution of more efficient and secure railway systems.

The genetic operators, including selection, crossover, and mutation, play a pivotal role in generating new candidate solutions. Through these mechanisms, the system favours the survival of the fittest solutions, combining genetic information from successful parents to create potentially improved offspring. The introduction of mutation injects an element of exploration, facilitating the discovery of novel solutions.

The Genetic algorithm implemented as follows;

1. Population**:** The `Population` class represents a collection of schedules. Each schedule within the population is an individual solution to the optimization problem. The size of the population is specified by the `POPULATION_SIZE` constant.

2. Schedule: The `Schedule` class represents a single timetable solution. It contains a list of classes, each representing a scheduled train journey. The fitness of each schedule is calculated based on conflicts such as platform capacity violations and overlapping schedules.

3. Genetic Algorithm*:*

Initialization: The initial population is created with random schedules generated using the `initialize` method of the `Schedule` class.

Selection: The `Tournament Selection` method is used to select schedules from the population for crossover. This method randomly selects a subset of schedules (determined by `TOURNAMENT_SELECTION_SIZE`) and selects the best schedule from this subset based on fitness.

Crossover: The `crossover population` method selects elite schedules directly, then iteratively selects two parent schedules from the population using tournament selection. These selected parent schedules undergo crossover to produce new schedules with characteristics from both parents.

Mutation: After crossover, the resulting population undergoes mutation with a probability determined by the `MUTATION_RATE`. Mutation introduces random changes to individual schedules to maintain diversity and explore new regions of the search space.

4. Fitness Function: The fitness function is defined within the `Schedule` class. It evaluates each schedule based on conflicts such as platform capacity violations and overlapping schedules. The fitness is calculated as the reciprocal of the total number of conflicts plus one, ensuring that higher fitness values correspond to better schedules.

5. Elitism: The top performing schedules (determined by `NUMB_OF_ELITE_SCHEDULES`) are preserved in each generation without undergoing crossover or mutation. This ensures that the best solutions found so far are retained in subsequent generations.

Overall, the genetic algorithm iteratively evolves a population of schedules over multiple generations, with the aim of finding optimal or near-optimal solutions to the train timetable optimization problem

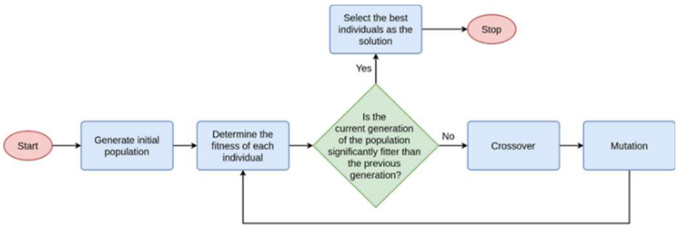
Select the best individuals as the solution
Stop
Yes
Start
Generate initial population
Determine the fitness of each individual
Is the current generation of the population significantly fitter than the previous generation?
No
Crossover
Mutation

*Figure 4 Genetic Algorithm*

## V. Conclusions

In conclusion, this paper proposes an AI-enabled track optimization system designed to revolutionize the operational landscape of the Indian Railways. Leveraging genetic algorithms and artificial intelligence, the system aims to optimize train tracks, mitigate delays, streamline schedules, and prevent potential collisions. The systematic approach adopted by the system, including encoding, fitness function development, genetic operators implementation, collision detection mechanisms, simulation, and ongoing evaluation, ensures comprehensive optimization and performance improvement. Through iterative processes, the system progressively converges towards optimal solutions, reducing delays and enhancing efficiency. Continuous monitoring and adaptation mechanisms enable dynamic evolution, ensuring sustained efficiency improvement. Anticipated benefits encompass significant improvements in punctuality, reduced overcrowding, enhanced efficiency for both freight and passenger services, and cost optimization. The proposed system holds the potential to redefine the Indian Railways' operational paradigm, paving the way for enhanced reliability, safety, and efficiency on a monumental scale.

## Acknowledgment

We would like to place on record my sincere gratitude to our project guide, Mrs. Shilpa Radhakrishnan, Assistant Professor, Computer Science and Engineering, College of Engineering Kallooppara for the guidance and mentorship throughout the project.

## References

[1] Y. Ji, W. Zheng, H. Dong and P. Gao, "Train delays prediction based on feature selection and random forest," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294653. keywords: {Delays;Feature extraction;Predictive models;Prediction algorithms;Rail transportation;Correlation;Random forests}

[2] P. Hu, J. Zhang and N. Li, "Research on Flight Delay Prediction Based on Random Forest," 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Changsha, China, 2021, pp. 506-509, doi: 10.1109/ICCASIT53235.2021.9633476. keywords: {Analytical models;Atmospheric modeling;Weather forecasting;Mean square error methods;Forestry;Predictive models;Airports;random forest;regression model;prediction model;flight delay;data analysis},

[3] S. D. Immanuel and U. K. Chakraborty, "Genetic Algorithm: An Approach on Optimization," 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2019, pp. 701-708, doi: 10.1109/ICCES45898.2019.9002372.

[4] keywords: {Sociology;Statistics;Genetic algorithms;Biological cells;Optimization;Wheels;Next generation networking;Genetic Algorithm;Travelling Salesman;Grammar Induction;Minimal Vertex Cover;Optimization;Evolutionary Algorithms},

[5] D. Long, "Prediction of Urban Rail Road Network Scale on Account of Genetic Algorithm," 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2022, pp. 1179-1182, doi: 10.1109/IPEC54454.2022.9777389. keywords: {Rails;Fault tolerance;Statistical analysis;Roads;Fault tolerant systems;Transportation;Predictive models;Genetic Algorithm;Urban Rail;Traffic Network;Network Scale Prediction},

[6] Z. Dong, Y. Zhao and Z. Chen, "Support vector machine for channel prediction in high-speed railway communication systems," 2018 IEEE MTT-S International Wireless Symposium (IWS), Chengdu, China, 2018, pp. 1-3, doi: 10.1109/IEEE-IWS.2018.8400912. keywords: {Support vector machines;Predictive models;Genetic algorithms;Prediction algorithms;Kernel;Railway communication;Channel prediction;genetic algorithm;high-speed railway communication;support vector machine},

[7] Z. Faris and I. A. Murdas, "A Comparative Study between Artificial Neural Networks and Optical Neural Networks," 2022 4th International Conference on Current Research in Engineering and Science Applications (ICCRESA), Baghdad, Iraq, 2022, pp. 143-147, doi: 10.1109/ICCRESA57091.2022.10352460. keywords: {Integrated optics;Optical propagation;Artificial neural networks;Optical computing;Optical fiber networks;Optical imaging;Nonlinear optics;Artificial intelligence;Artificial Neural Network;Optical Neural Network (ONN);Machine learning(ML);KMNIST dataset;Backpropagation Algorithm},

[8] D. Liu, D. Jinling and C. Xiaohua, "A Genetic Algorithm Based on a New Fitness Function for Constrained Optimization Problem," 2011 Seventh International Conference on Computational Intelligence and Security, Sanya, China, 2011, pp. 6-9, doi: 10.1109/CIS.2011.10. keywords: {Optimization;Evolutionarycomputation;Genetic algorithms;Algorithm design and analysis;Educational institutions;Convergence;Electronicmail;Constrained optimization;genetic algorithm;fitness function;NFFM(New fitness function method)},

.