

USING GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION IN LEGAL SYSTEM

Syed Sameer Ali^{1*}, Kusum Lata Agarwal²

^{1*}Unitedworld Institute of Technology , Karnavati University, Gandhinagar, 202306020011@karnavatiuniversity.edu.in, 0009-0004-8617-1552

²United world Institute of Technology, Karnavati University, Gandhinagar, kusum@karnavatiuniversity.edu.in, 0000-0002-9492-7653

ABSTRACT

In contrast to the theory of jurisprudence, pragmatic reasoning was developed as a model for legal case analysis and interpretation. Specific legal cases are thought of as having a set of contextual information surrounding it. The combination of optimization techniques namely as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) gathered notable attention in legal systems, where the reduction of court cases is a pressing concern. The reasoning derived from these cases leads to solutions that are closely linked to particular situations. In order to generalize these solutions, machine learning methods are applied. GA helps in feature selection by identifying the most meaningful case attributes, while PSO fine-tunes model parameters to achieve higher accuracy. One of the techniques to induce reasoning from cases is considered 'examples' with a 'classification' in a certain area. In this paper, a genetic algorithm approach to inductive feature learning of cases from a legal database resulting in case-based reasoning taxonomy is presented. The acquired knowledge can then be used to reason from cases which are by their features and classifications 'interesting' and predict the classification of other cases.

KEYWORDS: Artificial Intelligence(AI), Legal Case, Genetic Algorithm, Particle Swarm Optimization.

1. INTRODUCTION

Within AI, case-based reasoning is known as an effective modeling strategy in the legal system. Because law is basically unstructured and intricate, the legal methodology attempts to counter the lack of a formal consultation model by informal processes of legal reasoning and synthesis. In a multilingual country like India, Natural Language Processing (NLP) allows translators to provide real-time translation of legal documents, so that language does not become an issue in administering justice[1]. AI based analytics can identify the reasons for unproductive delays and suggest better resource allocation and policy changes to encounter the negative impacts of absence of sufficient resources or inefficient working practices. There is an opportunity to use AI in the Indian judiciary system, but surveillance and the use of a human component are critical in eliminating biased algorithms, data privacy concerns, and case backlogs[2].

A case is made up of a multi-faceted set of facts and involves a 'set of circumstances and 'result' or classification determined from these circumstances. The paper describes the problem of case classification based on circumstances and offers a solution through case-based reasoning and Genetic Algorithms. Legal cases are viewed as cases with particular contextual components, and they form the components that detail the reasons which are classified into various types. Several countries are employing algorithms to settle civil cases, such as traffic cases and to determine the amount of fines. In the knowledge-based reasoning prototype the system is educated with rules and the user supplies the truth and the system will generate a decision[3]. In the case-based knowledge model the system is based on the experience of the model, its capability to contrast and express reasons from other situations. Case-based reasoning is based on the model's capacity to draw upon its past successful conclusions, and recognize alike behavior beforehand, so that they can be evaluated in the future[4]. In other parts of the world numerous countries have utilized hybrid algorithms of these models to resolve consumer, traffic, divorce cases, etc [5].

Genetic Algorithms(GA) are an empirical solution-search or optimization technique, at first motivated by the Darwinian principle of evolution through (genetic) selection. A GA uses an immensely hypothetical genre of advancement procedure to evolve solutions of given problems. Each GA operates on a populace of constraints, a real number which is a rate of how satisfactory a solution is for a specific problem. The intersection of biology and computational techniques offers valuable insights into algorithm design that could enhance the efficiency of legal systems. Genetic algorithm(GA) and Particle swarm optimization(PSO) were selected because both are powerful optimization techniques that mimic natural processes. GA works like biological evolution, using selection, crossover, and mutation to gradually improve solutions. PSO on the other hand, imitates the collective intelligence of bird flocks or fish schools, where each "particle" acquire knowledge from its own experience and swarm's best experience. Beginning with a randomly generated population of variables, a GA carries out a procedure of fitness-based selection and reunite to produce a successor population, the next generation. As this procedure iterates successive generation evolves and the mean fitness of the variables tends to increase until some conclusion criterion is reached. In this way GA proves and develops as a best solution to the given problem.

The application and design of heuristic-based improvement methods have greatly increased. Because they employ a populace of solutions during their search, the overall solution of a certain problem is more likely to be obtained. Also, it employs only a straightforward scalar measure of performance that does not require or utilize imitative information. Particle swarm optimization approach is one of heuristic-based enhancement methods and that was used with success in

numerous optimization problems. GA and PSO are two of the heuristic optimization methods. They are very similar to their inbuilt parallel nature. Investigation reveal that they have their own merits in solving various optimization problems[6]. In contrast to GA, PSO has some interesting features[7,8]. It contains memory i.e., good solutions are remembered by all the particles; while in GA, past information about the problem is lost when the populace is updated. It has positive cooperation among particles, i.e., particles within the swarm exchange information between themselves. Up to now, PSO has been effectively used for optimizing a variety of continuous nonlinear functions in application[9]. Over the last couple of years, numerous reported contributions have been made regarding the amalgamation of PSO with other heuristic optimization methods. In PSACO algorithm was presented for non-convex optimization problems[10]. Also, in GA has been integrated into PSO as a hybrid method that combines two heuristic enhancement methods in order to perform global optimization on multimodal functions[11]. The primary aim of this research is to present a novel hybrid algorithm to take advantage of the strengths of the two heuristic optimization methods, GA and PSO.

LITERATURE REVIEW

Analysis of the scope of AI technologies in the legal sector in India with a specific focus on understanding how these technologies shift the practice of law is done[12]. The impact which AI has in judicial case management, case investigation, and decision making in Indian Law is studied. When AI-enabled case analysis was done it was observed that the time taken to prepare cases decreased by 40% thereby improving productivity and precision. Potential of biological computations has been studied and it was found that it can lead to development of innovative algorithms. These algorithms help in enhancing resource allocation and decision-making processes within the legal domain[13]. Biological computation and computational biology can be combined to reduce the complexities of legal case management.

The adoption and application of AI by several countries in their public sector is studied[14]. Effort is made to understand how different nations have integrated AI into their legal and administrative systems. Case studies originating from different regions were analyzed to identify patterns and issues in the application of AI. There was a tendency for greater incorporation of AI in the management of justice and in legal processes, but the discrepancies with the legal systems and definitions were also exhibited. Privacy of data and biased algorithms emerged as the limitations suggesting that some restrictions and policies regarding these matters would be required in due course. A case study, focused on China, studied the impact on judicial decision-making using AI. The influence of AI tools on the operational efficiency, clarity, and consistency of the juridical system in question was measured. The qualitative research, looked at the advancements in the AI technology that were applicable to the judicial system in China and the impact of such advancements on decision making and public confidence towards the legal system. The findings proved that indeed, AI contributed to the process improvements and case management consistency[15].

The research involves AI in enhancement of the processing technology and admits the wider access to justice by assisting the Indian judicial system with improvements[16]. In a precise way, in what manner and how greatly AI is a factor in swift court activities are questions of vital importance that needs to be considered. Through the deductive method, data will be collected from cases that qualify and will then be processed meaningfully before the issue of judgement. Accordingly, research and analysis were performed on the cases that are prioritized, along with the trends in legal sector. The main area of research was to determine how technology was behind the functioning of the judicial system. A mixed-method approach was adopted. The records issued by the courts were used to conduct interview of the professionals and then compared with the policy documents to evaluate the impact of technology on case handling. As far as the efficiency in the justice and legal system of the court is concerned, time invested was reduced with several other benefits including lighter administrative loads and the eventual solution along with better rights to the citizens[16]. How the technological transformation has played an analytical part in enabling the integrity of judicial services within the legal sector of India amidst the pandemic and its consequential aftermath is discussed. The objective of the research was to find out how technology contributed to the paperless process of legal dealings during the pandemic even though it was at a nascent stage. Information from court records were checked against the policies in vogue to check how the cases were handled. The introduction of digitalization i.e. filing and virtual hearings has resulted in a massive decrease in the procedural delay rate, thereby enhancing the Indian judiciary efficiency on a continuous basis[17].

It has been established beyond any dispute that meta-heuristic enhancement algorithms work well by best managing various general purpose real-world optimization problems, from automation, wireless networks power systems, job shop arranging to classification and artificial neural network training[18-25]. In pursuit of the global optimum solution, many fitness evaluations are demanded by the meta-heuristic algorithms. This is generally a severe obstacle against the usage of meta-heuristic algorithms on high-computational optimization issues that widely occur in computational fluid dynamic optimization and structural optimization, among numerous others[26,27]. To solve those issues, candidate solutions are commonly tested using high-fidelity numerical analysis methods which can subtract CPU time from several minutes to many hours, or even days as a consequence. In the meta-heuristic paradigm, numerous Swarm Intelligence(SI) methods and types were developed to solve complex optimization problems[28,29]. Efforts have been put to do research in the field of particle swarm and new applications. The strength of the research was to describe elaborative challenges and open issues in the PSO algorithm. Nevertheless, this research failed to address the compatibility of the application of PSO with each approach presented[30,31].

Identifying some problems related to Wireless Sensor Networks(WSNs), authors in proposed PSO and debated the applicability of the algorithm to WSN applications[32]. It also discussed a concise overview of how PSO is customized to solve the node localization, node deployment, data aggregation, and energy-aware clustering problems as optimization issues. The strength of this review lies in briefly summarizing open issues in WSNs. The compatibility of high-speed real time applications with every approach was not taken into account. Research in summarized the work concerning the native PSO modifications and its engineering application in real-world problems[33]. The quick modifications of PSO have

emerged in various examples, such as two-step PSO and PSO-Support Vector Machine((PSO-SVM). The combination of the PSO and the engineering application of PSO with the accepted benchmark algorithm have also produced good results. The benefit of this survey was to introduce fresh diversified variations in PSO and examine the precision of PSO in various fields. The primary flaw of this research is the absence of statistical details regarding the discussed standards PSO and its utilization within diverse given contexts.

Different versions of PSO with respect to swarm utilization, mutation operators, and inertia weight has been discussed[34]. The primary benefit of such an overview was to emphasize the significance of incorporating the various mutation operators and the inertia weight parameter to enhance the performance of PSO. Other potential versions of PSO were not taken into consideration. Systematic review of the development of clustering methods on PSO was done and the outcomes of fast-growing trends in the SI literature, PSO paradigm literature, and PSO-based data clustering methods literature, demonstrating that such methods have a dramatic boost in popularity[35]. This work confirmed that the methods are new and easy to apply and is a build-up on collaboration and communication. This research discussed various application fields of PSO applicable to clustering. Yet, applications to more intricate problems do not exist. Variants of PSO for clustering multidimensional data were also surveyed by[36]. This survey highlighted the efforts of various publications in data dimension reduction. The drawback remains due to non-availability of more applications related to clustering.

Detailed explanation of the PSO algorithm to demonstrate its capability in addressing various types of optimization tasks in chemometrics was done[37]. The significance of the proper selection of PSO meta-parameters has been emphasized here by way of choosing realistic examples in the research topic of variable selection, robust PCA solution estimation, and signal warping. In the origin and history of the PSO algorithm was introduced and a theoretical study of the algorithm was done. The present state of application was examined and research on algorithmic structure, topology, parameter setting, multi-objective optimization, discrete and parallel PSO algorithm, and engineering applications was done. Several researchers have used PSO via string distances to organize the test examples and check their method on a real-world TSL dataset. To pick test cases based on redundancy, binary constraint PSO and its mixed version with local search methods were created[38]. PSO was used with local search to select test cases, aiming for more branch coverage and lower costs[39]. Since the results were promising, PSO was also combined with harmony search, which worked better than NSGA-II[40]. In a measure to check the diversity and diagnosability of a test suite was done and the results got better when they used local search algorithms with PSO to increase requirement coverage while reducing costs[41].

RESEARCH METHODOLOGY

To improve the performance of AI models, a mixed approach combining Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) is utilized. The process starts with data collection, data cleaning, and feature extraction using principal component analysis(PCA). Then a hybrid method incorporates Genetic algorithm(GA) and Particle Swarm Optimization(PSO).GA handle large search spaces performing the use of domain derived or free-formed heuristic information from problem bound domains. In PSO particles are not only driven by their own experiences in moving towards their optimal position but also by swarm's knowledge learnt from its members including its neighbor's best-known position[42].

Together the GA and PSO equilibrium exploration(searching widely) and exploitation (refining good solutions),ensuring that the AI model does not get stuck in poor solutions but still converge towards reliable outcomes. The backlog of cases and complexity of judicial data required methods that could search through massive solution spaces quickly and reliably. GA helps in feature selection by identifying the most meaningful case attributes, while PSO fine-tunes model parameters to achieve higher accuracy. Together they balance exploration (searching widely) and exploitation (refining good solutions), ensuring that that the AI models don't get stuck in poor solutions but still converge toward reliable outcomes. Proposed Methodology is as brought out in Fig. 1.

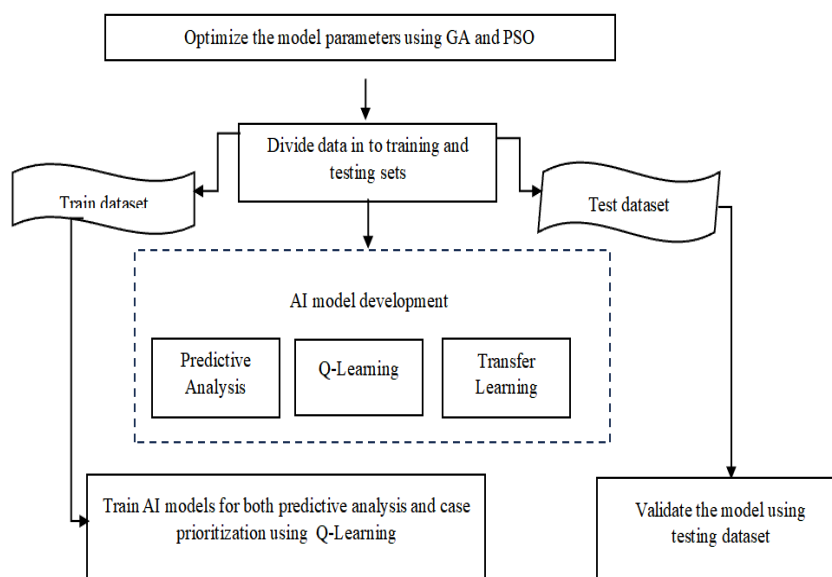


Fig.1. Proposed Methodology

Genetic Algorithm (GA)

Genetic Algorithms uses heuristics to implement search and optimization. The real-world optimization problems are addressed using GA which is based on the hypothesis of evolution[43]. The diverse morphological and behavioral traits which are exhibited by species in their environment helps to adjust their behavior[44]. Over time, they develop complex physical structures that help them thrive in various conditions. The capacity to reproduce best solution is a key indicator of success from an evolutionary standpoint. These compelling factors have led many to advocate for optimization strategies rooted in evolutionary theory[45].

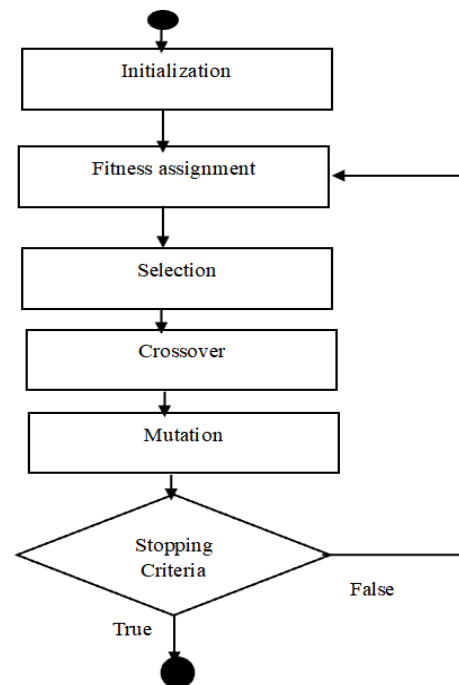


Fig. 2. Working of Genetic Algorithm

In the legal domain, GA is used to select which case features (like court number, district code, type of case, are most important in predicting delays or outcomes. GA is capable of navigating large search spaces and can utilize both domain-specific and general heuristic information from problem-related areas. It has positioned itself as a strong competitor among the best scheduling methods available[46,47]. Fig. 2 illustrates how the genetic algorithm operates. By simulating the natural evolution process, GA enhances the selection of features. It involves repeatedly choosing the most effective features for predicting case outcomes with minimal error during the processing of judicial cases.

Steps involved:

Initialization: Make a group of individuals called P, which includes P₁, P₂, ..., P_m. Each individual is a binary string that shows which features are chosen.

Fitness Evaluation: Calculate how good each individual is using the formula:

$$F(P_i) = 1/(1 + E(P_i))$$

Where E(P_i) is the mistake rate of the model when using the chosen features.

Selection: Choose individuals based on how fit they are, using methods like roulette wheel selection.

Crossover: Combine two individuals to make new ones with a chance of P_c.

Mutation: Change some parts of the individuals randomly with a chance of P_m to keep things varied.

Particle Swarm Optimization (PSO)

Particle Swarm optimization (PSO) is an algorithmic optimization way imagined by the social behaviors observed in environment, such as the flocking of birds and the schooling of fish. In a swarm, multiple particles move through a search space to gather potential solutions. This knowledge gathered by their own experiences while seeking optimal positions combined with the collective knowledge of all the pieces in the swarm try to find the best-known positions of their neighbors. The parameters like learning rate, dropout, or iterations of ML models are optimized so that issues like case urgency, fraud detection, or guilty/not guilty decisions are predicted accurately.

PSO is easy to use, has low computational cost and is versatile. It helps solve problems in the fields of machine learning, engineering design, robotics, and finance[48]. One of the benefits of PSO is how well it balances exploration with strategy improvement. This balance facilitates movement through intricate search spaces in a timely manner[49]. PSO mimics social behaviors to optimize model hyperparameters, aiming to identify the best parameters more efficiently. This approach improves the correctness and execution of the model predicting and prioritizing cases. Fig. 3 illustrates the geometric representation of the PSO algorithm[50].

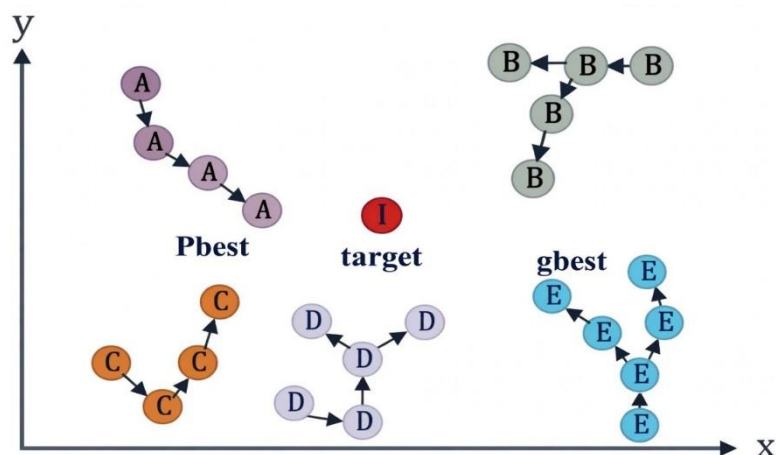


Fig.3. Geometrical illustration of PSO algorithm

PSO is an approach based on the conduct of social groups, such as in nature[51]. Each agent in this approach relies on two kinds of information when they need to make a decision. The first is their own history[52]. That is, they have already experimented with various options and know which of them was the best so far and how good that was. The second information is from other agents[53]. This implies they have an idea of the decisions others around them have made useful and how well those decisions worked. In the PSO system, every agent makes a conclusion based on their own experience and what other agents have discovered. The system begins with a populace of random solutions. Every solution, referred to as a particle or agent, is assigned a random velocity and traverses the problem space. Each agent hold memory about its best history position and the outcome of that position. There is also a best performer within the group, who is mentioned to as the global best, and who holds the best result out of all the agents.

Steps involved in PSO:

Initialize Particles:

Set starting positions $X(0)$ and speeds $V(0)$ for each particle.

Velocity Update:

$$V(t+1) = w \cdot V(t) + c1 \cdot r1 \cdot (P_{best} - X(t)) + c2 \cdot r2 \cdot (G_{best} - X(t))$$

Here, w is the inertia weight, $c1$ and $c2$ are numbers that control how much the particle looks at its own best and the group's best, $r1$ and $r2$ are random numbers, P_{best} is the best position the particle has ever been in, and G_{best} is the best position found by all particles.

Position Update:

$$X(t+1) = X(t) + V(t+1)$$

The hybrid approach iterates by alternating between GA and PSO until convergence, ensuring optimal feature selection and model parameter tuning.

Dataset Description

The dataset used in this research originates from the research conducted by[54]. It includes case records available on the e-courts platform managed by Indian Government. The database is from 2010 to 2018 and encompasses cases from over 7000 districts and interim trial courts, involving more than 20,000 judges[54]. Data from website can be downloaded, which contain more than 77 million cases[54].

Key information contains the name of the presiding judge, the nature of the case, and its intended outcome. Details about the judge's gender, term, and the relevant section and action for criminal charges are presented separately. Also, it contains unique case ID serving as index. Certain data, such as the dates of first, subsequent and final listings, purpose names and disposition, were excluded from the study due to their unavailability at the time of filing. The Table 1 brings out the feature set of the legal database studied.

Table 1. Features incorporated in analysis

Column Name	Description
date of filing	Date when the case was filed
date of decision	Date when the case reached a decision
Court-no.	Unique court number of every court in a district
Number sections ipc	Count of sections implicated in criminal cases
State Code	Unique state code
Dist code	Unique district code of each district within a state
Judge position	The judge's position according to the current hierarchy
Female_judge_filing	Gender of presiding judge in the case

Female judge decision	Gender of presiding judge who rendered the decision
Female adv pet	Gender of the attorney representing the petitioner
Female adv def	Gender of advocate representing the defendant
Type-name	The category of case
Act	Legislation under which case is initiated

The feature set (like date of filing, court number, district code, type of case, judge details, IPC sections) acts as the foundation of the analysis. Each feature provides context that helps the AI understand:

- Where the case is filed (State/District codes)
- What type of case it is (Civil, Criminal, etc.)
- Who is involved (judge, lawyers, gender of petitioners/defendants)
- How Complex it might be (number of sections involved, whether bailable)
- By feeding these features into PCA, GA and PSO, the system can extract only the most important elements that drive case delays or outcomes.

Hyperparameters

Hyperparameters are the parameters that affect the way of learning algorithm grows. They are tuned to optimize the performance of the model overall. Hyperparameters are like the “settings” of the AI model that control how it learns. The following parameters were carefully chosen and optimized using GA and PSO.

Learning Rate: controls how big a step the model takes in learning.

Batch size: figure of samples processed at a time

Epochs: figure of times the model sees the full dataset

Activation: helps handle non-linear patterns

Optimizer: adjusts learning speed dynamically

PSO Particles & Iterations: define how many solutions are tested and refined

These hyperparameters ensure that the models are stable, efficient and accurate for legal predictions as brought out in Table 2.

Table 2. Detailed information of hyperparameter used

Parameter	Value
Batch Size	32
PSO particles	50
Number of epochs	100
Optimiser	Adam
Activation function	ReLU
PSO Iterations	100
Weight decay	0.01

System Configuration:-It consists of

Processor(CPU): It is advisable to utilize processor such as Intel Xeon or AMD Ryzen thread dipper with minimum of 16 physical cores.

RAM(Memory): 32 GB or more DDR4 RAM is sufficient to fit large datasets into memory.

GPU (Graphics Processing Unit): GPU such as NVIDIA RTX 3090 or better, a Tesla V100.

Operating System: The ideal options are Ubuntu 20.04 LTS or windows 10 since they support widely used frameworks.

Storage(SSD): 1TB SSD assists in rapid data access and delivers high read and write speeds.

CUDA Version: Installing CUDA 11 is necessary to utilize CUDA NVIDIA GPUs.

The following were used in this research.

Python It is flexible enough to support different programming paradigms[55].

Python libraries: Some of the key libraries used are Scikit, Seaborn and Pandas.

Google Colab: It enables users to execute, write, and share python code in a collaborative manner.

Research Findings

The GA optimization is a multiple step process which tells how it is trying to find an optimum solution. In the graph (Fig. 4.), performance metric which assesses the quality of solutions is denoted on y-axis and the iteration number is on x-axis. The green line shows how the performance metric changes over time showing how the algorithm uses the solution space.

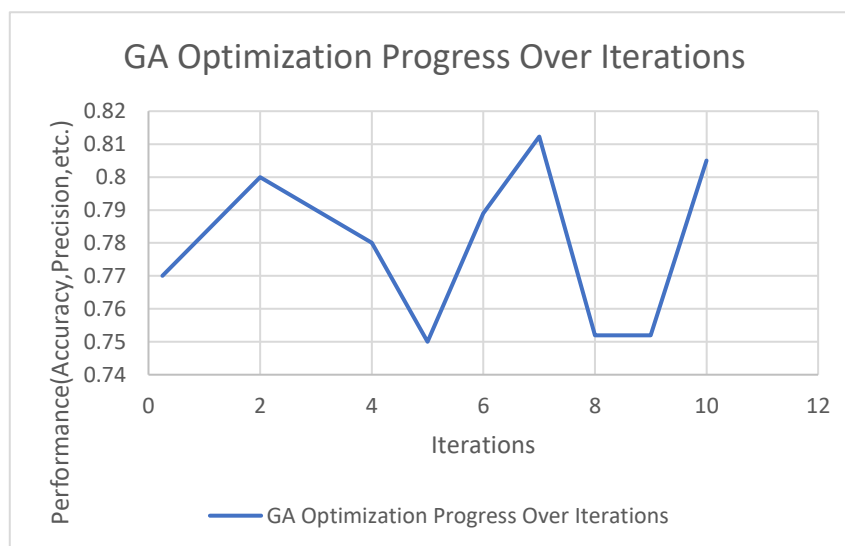


Fig. 4. Using GA Optimization

Before the 4th iteration, it is clear that significant improvement of the performance metric which peaks at around 0.82. At this point in the performance, we can say that the genetic algorithm(GA) has achieved reasonably good solution by pick any elite individuals and their traits through crossover and then merging them, after this point the performance peak, the performance metric decreases slightly ranging from 0.75 to 0.82 over considerable number of iterations. This is a common pattern of genetic algorithms, when they try to maintain equilibrium between search and optimization. The performed iterations indicate algorithm runs back and forth attempting to improve its search strategy to not succumbs to an unsatisfactory solution too early. Later on, the performance metric increases again, leveling out at 0.81. This suggests the last change in measure indicates that the GA has reached near optimal solution which is achieved by the GA using evolutionary principles to increase search effectiveness and performance[56].

Optimization using PSO

The evolution of the PSO algorithm between iterations is indicative of its dynamic pursuit of an optimal solution. This metaheuristic optimization technique attempts to identify the optimal solution to problem by relentlessly updating candidate solutions depending on their individual performance as well as their performance collectively[57].

Fig. 5 is a graph where the performance measure is plotted on the y-axis, which reflects the quality of solutions, while x-axis indicates the iteration number, showing the progressive improvement in solutions as the algorithm proceeds. The blue line in graph represents how the performance measure changes during the optimization process.

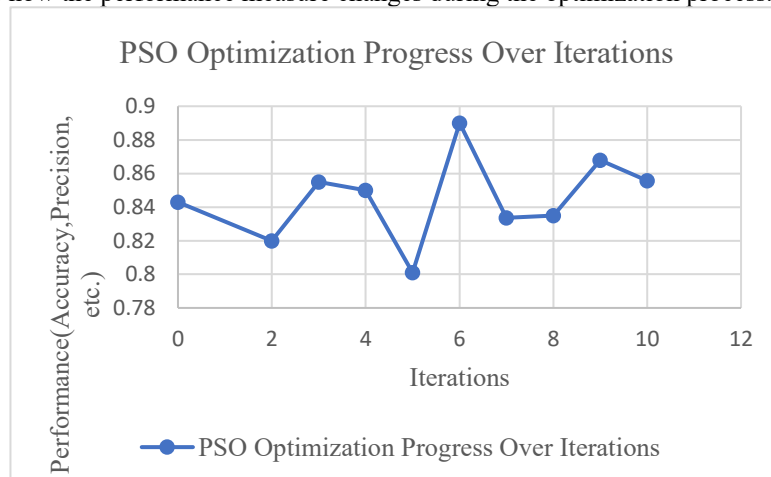


Fig. 5. Outcome of PSO

At the early stages, the performance metric starts at 0.845 and sways, which indicates the exploratory behavior of the PSO algorithm. These sways emphasize the dynamic movement of particle positions in an attempt to find better-performing solutions. A significant landmark is experienced at around the 6th iteration, where the performance measure leaps to approximately 0.886. This sudden spike implies that the algorithm has found a much more optimal solution. This highlighting of PSO's capacity to rapidly target worthwhile regions of the search space. After the peak at iteration 6, the performance measure slowly decreases and converges to 0.85. This shows that algorithm is entering the exploitation phase, refining its search to converge to near-optimal solution. The convergence of the metric suggests that further iterations lead to diminishing improvements, which means convergence to a solution that optimizes performance against feasibility. For PSO, the optimized factor is the Learning Rate(LR) of the classifier.

PSO+GA Optimization

The optimization process of two metaheuristic algorithms, PSO and GA, was tracked for multiple iterations. The y-axis shows the performance measure, which is probably measuring the solution quality being optimized, and the x-axis represents the iteration number, which is an indicator of the optimization process's advancement.

Code developed is given below in Fig.6 & Fig.7 -

```
def load_and_preprocess_data(file_path):
    """Load and preprocess the court cases dataset."""
    # Read the data
    df = pd.read_csv(file_path)

    print("Original target values:")
    print(df['disp_name'].value_counts().sort_index())

    # Convert continuous target to categorical
    bins = [-np.inf, -1.5, -0.5, 0.5, 1.5, np.inf]
    labels = ['VL', 'L', 'M', 'H', 'VH']
    df['disp_name'] = pd.cut(df['disp_name'], bins=bins, labels=labels)

    print("\nAfter binning:")
    print(df['disp_name'].value_counts().sort_index())

    # Separate features and target
    X = df.drop('disp_name', axis=1)
    y = df['disp_name']
```

Fig.6. Code developed

```
# Initialize and run optimizer
optimizer = HybridOptimizer(
    X=X,
    y=y,
    label_encoder=label_encoder,
    n_features=len(X.columns),
    population_size=50,
    n_generations=20
)

# Run optimization
best_model, selected_features, final_accuracy = optimizer.optimize()
```

Fig.7. Code developed

RESULT

The red line in graph represents the combined performance of the PSO and GA methods, offering insights into their collaborative optimization strategy. In the early iterations performance metric begins at around 0.89, highlighting the exploration phase of the combined optimization effort. During this stage, the algorithms work together to explore the solution specializing PSO's particle-based exploration alongside GA's evolutionary techniques to ensure a varied and thorough search. A significant milestone is just reached before the 4th iteration, where the performance metric jumps to 0.92 as depicted in Fig. 8. This notable increase suggests the combine PSO and GA approach has discovered a better solution at this point, demonstrating the effective collaboration between the two methods in rapidly converging on promising areas of the solution space.

After reaching its highest point, the performance metric goes up and down. It first drops to about 0.87 and then goes back up to around 0.91. These ups and downs show that the system is constantly balancing between exploring new options and making the best choices it has found so far. The changes mean the system can avoid getting stuck in a less-than-ideal solution, making sure the process doesn't settle too early on a poor result. Later on, performance metric starts to go down again and settles around 0.87. This show the combined PSO and GA method has found a solution that is very close to the best possible one, and its doing this efficiently without using too much computing power.

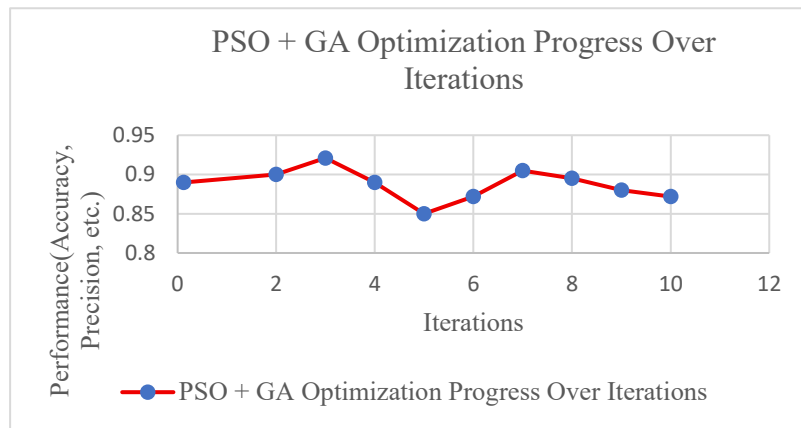


Fig. 8. Optimization using PSO+GA

DISCUSSION

The graph reflects how efficient is to combine PSO and GA approaches, applying their respective strengths to find their way through the solution space. The fluctuations and subsequent convergence observed reflect the strength of this hybrid approach in avoiding local optima, maximizing performance and finding high-quality solutions. Figure 6. highlights the usefulness of combining these two algorithms for addressing large optimization problems, pointing to their ability to enhance decision and solution quality in the overall methodology.

PSO Graph: Shows performance rising quickly (exploration), stabilizing after the 6th iteration, and converging around 0.85. It proves PSO can find good solutions fast.

GA Graph: Shows up and downs (peaks and valleys), reflecting genetic diversity. It slowly converges around 0.81, showing GA's strength in avoiding premature convergence.

Combined GA+PSO Graph: Shows the best of both, fast exploration plus careful refinement converging around 0.87-0.91 After the application of ML, the pending case reduction summary result achieved is as shown in Table 3.

Table 3. Pending Case Reduction Summary

Index	Metric	Value
0	Total Cases	900.0
1	Pending Cases Before ML	213.0
2	Pending Cases After ML	122.0
3	Reduction Achieved	91.0
4	Reduction(%)	42.72
5	Optimization Rate (Adaptive)	42.86

CONCLUSION & FUTURE WORK

The common as well as specific features across cases are used for their classification. This method is implemented in a computer program named Genetic Algorithm for Induction (GAINC) from cases and is said to be generative as opposed to deductive or statistical reasoning. Both the PSO and GA techniques work well. The PSO method found potential solutions in the 6th iteration, while the GA method demonstrated an investigation and exploitation process at the same point. Thus, GA performs feature optimization accuracy rises from 0.89 to 0.93 and PSO performs parameter optimization accuracy rises from 0.89 to 0.93. The combination PSO-GA hybrid approach was a kind of way to facilitate this optimization process which in turn had given the renewable and efficient system of the resource allocation. Hybrid optimization (GA+PSO) ensures better feature selection and hyperparameter tuning, reducing the chances of overfitting and enhancing processing efficiency. This makes the algorithms highly relevant in legal workflows and improving prioritization of urgent cases, reducing noise in case data, and enhancing predictive accuracy for judicial decisions.

The special aspect of this approach is the blended nature, where high-level optimization methods such as particle swarm optimization and genetic algorithms are combined with AI models to make the court system more effective. Although other research uses AI in the courts, but the combination of both of this approach enhance the performance and resource allocation in case management. Utilizing both methodologies in tandem is preferable for constructing the models because it ensures that the predictions made by the AI not only happen to be correct but also versatile with respect to the various forms of legal data, which is something the old system has difficulty with. Future work will include case prioritization using Q-learning and comparative analysis using performance metrics.

Competing Interest-The authors declare that they do not have any competing interest.

Funding Information-This research received no external funding.

Author Contribution-All authors contributed, critically reviewed and approved the final manuscript.

Data Availability Statement-Not applicable

Research involving Human and/or Animals-Not applicable.

Informed Consent-Not applicable.

REFERENCES

1. Arno R. Lodder and John Zeleznikow. "Artificial Intelligence and Online Dispute Resolution in Mohammed Abdel Wahab, Ethan Katsh and Daniel Rainey(ed.)," Online Dispute Resolution Theory and Practice, 2013, pp 73-94, Hague :Eleven International Publishing.
2. Bhupatiraju, S., Chen, D. L., & Joshi, S. "The Promise of Machine Learning for the Courts of India" National Law School of India Review, 2020, 33(2).
3. Jeremy Barnett and Philip Treleven. "Algorithmic Dispute Resolution-The Automation of Professional Dispute Resolution Using AI and Blockchain Technologies" The computer Journal, 61 (3):2017,299-408.
4. M.A. Panduro, C.A. Brizuela, L.I. Balderas, D.A. Acosta , "A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays." Progress in Electromagnetics Research B,13, 2009, pp.171-186.
5. O. Uysal, S. Bulkan , "Comparison of genetic algorithm and particle swarm optimization for bicriteria permutation flowshop scheduling problem." International Journal of Computational Intelligence Research, 4(2),2008, pp.159-175.
6. P.S. Shelokar, P.Siarry, V.K. Iayaraman, B.D. Kulkarni, "Particle Swarm and ant colony algorithms hybridized for improved continuous optimization." Applied Mathematics and Computation,188, 2007, pp.129-142.
7. Scharf, M. "TextFrame: Cosmopolitanism and Non-Exclusively Anglophone Poetries", Doctoral dissertation, City University of New York,2019.
8. Suzanne Van Arsdale. "User Protections in Online Dispute Resolution" Harvard Negotiation Law Review, 2015, 21(107).
9. X. Yang,J. Yuan,H.Mao, "A modified particle swarm optimizer with dynamic adaptation." Applied Mathematics and Computation,189, 2007, pp.1205-1213.
10. X.H. Shi, Y.C. Liang, H.P. Lee, C. Lu, L.M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm." Information Processing Letters 93, 2005, pp.255-261.
11. Y. Kao, E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions." Applied Soft Computing, 2008, pp.849-857.
12. Chelly Dagdia, Z., Avdeyev, P., & Bayzid, M. S. , "Biological computation and computational biology : survey, challenges, and discussion, Artificial Intelligence Review." Springer Science and Business Media LLC, 2021, <https://doi.org/10.1007/s10462-020-09951-1>.
13. Tiwari, "S. K. Exploring the Impact of Artificial Intelligence in the context of Legal Profession in India." Algorithms, 12(3). Volume 12 Issue 3,2024.
14. Alam S, Dobbie G, Koh YS, Riddle P, Rehman SU, "Research on particle swarm optimization-based clustering: a systematic review of literature and techniques." Swarm Evol Comput 17:1-13, 2024.
15. Atabekov, A., "Artificial intelligence in contemporary societies: legal status and definition, implementation in public sector across various countries." Social Sciences, 12(3), 178, 2022
16. Bernardino HS, Barbosa HJ,Fonseca LG, "Surrogate-assisted clonal selection algorithms for expensive optimization problems." Evol Intel 4(2):81-97, 2011.
17. Bhowmick, Aditi, Paul Novosad, Sam Asher, Elliott Ash, Bilal Siddiqi, Christoph Goessmann, Daniel Chen, and Tanaya Devi. "In-group bias in the Indian judiciary: Evidence from 5.5 million criminal cases. In Proceedings of the 4th AM SIGCAS Conference on Computing and Sustainable Societies,2021, pp.47-47."
18. Bui, Xuan-Nam, Hoang Nguyen, Yosoon Choi, Trung Nguyen-Thoi, Jian Zhou, and Jie Dou. "Prediction of slope failure in open-pit mines using a novel hybrid artificial intelligence model based on decision tree and evolution algorithm." Scientific reports 10, no. 1: 9939, 2020.
19. Chen CH, Liu TK, Chou JH, "A novel crowding genetic algorithm and its applications to manufacturing robots." IEEE Trans Ind Inf 10(3):1705-1716, 2014.
20. Churnbumroong S,Cang S,Yu H, "Genetic algorithm-based classifiers fusion for multisensory activity recognition of elderly people." IEEE J Biomed Health Inform 19(1):282-289, 2014.
21. Correia, D., "An industrial application of test selection using test suite diagnosability.In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia,2019, pp. 1214-1216."
22. De Souza, L. S. ; Prudencio, R.B. ; Barros , F.D.A.; Aranha , "E.H.D.S. Search based constrained test case selection using execution effort." Expert Syst. Appl.40,4887-4896, 2013.
23. De Souza, L.S.; Prudencio, R.B.; Barros , "F.D.A. A hybrid binary multi-objective particle swarm optimization with local search for test case selection." In Proceedings of the Brazilian conference on intelligent systems, Sao Paulo, Brazil, 2014, pp . 414-419.
24. De Souza, L.S.; Prudencio, R.B.C.; De Barros , "F.A. A hybrid particle swarm optimization and harmony search algorithm approach for multi-objective test case selection, J. Braz." Comput. Soc, 21,2015, 1-20.
25. Del Valle Y, Venayagamoorthy GK,Mohagheghi S,Hernandez JC,Harley RG, "Particle swarm optimization: basic concepts,variants and applications in power systems." IEEE Trans Evol Comput 12(2):171-195,2008.
26. Elhalid, Osama Burak, Zaynelabdin Alm Alhelal, and Samer Hassan. "Exploring the Fundamentals of Python Programming: A comprehensive guide for beginners." International Journal of Computer & Information Sciences/International Journal of Computer and Information Sciences (2023).
27. Esmine AA, Coelho RA, Matwin S , "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data." Artif Intell Rev 44(1):23-45, 2015.
28. Houssein, Essam H., Ahmed G. Gad, Kashif Hussain, and Ponnuthurai Nagarathnam Suganthan. "Major advances in particle swarm optimization: theory, analysis, and application." Swarm and Evolutionary Computation 63: 100868, 2021.

29. Imran M, Hashim R, Khalid NEA, "An overview of particle swarm optimization variants." *Procedia Eng* 53:491-496, 2013.
30. J. Kennedy, R.C. Eberhart, "Swarm Intelligence, Morgan Kaufmann, San Francisco.", 2021.
31. Ji B, Song X, Sciberras E, Cao W, Hu Y, Pickert V, "Multiobjective design optimization of igbt power modules considering power cycling and thermal cycling." *IEEE Trans Power Electron*, 30(5):2493-2504, 2014.
32. Jin Y, Olhofer M, Sendhoff B, "A framework for evolutionary optimization with approximate fitness functions." *IEEE Trans Evol Comput* 6(5):481-494, 2002.
33. Jin Y, Sendhoff B, "A systems approach to evolutionary multiobjective structural optimization and beyond." *IEEE Comput Intell Mag* 4(3):62-76, 2009.
34. Juang CF, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design." *IEEE Trans Syst Man Cybernet Part B(Cybern)* 34(2):997-1006, 2004.
35. Kennedy, J., R. C. Eberhart, and Y. Shi (2001), *Swarm Intelligence*, Elsevier, New York
36. Kothari V, Anuradha J, Shah S, Mittal P, "A survey on particle swarm optimization in feature selection. In : International conference on computing and communication systems." Springer, pp 192-201, 2011.
37. Kulkarni RV, Venayagamoorthy GK, "Particle swarm optimization in wireless-sensor networks: a brief survey." *IEEE Trans syst Man Cybern Part C(Applications and Reviews)* 41(2):262-267, 2011.
38. Marini F, Walczak B, "Particle swarm optimization(pso). A tutorial." 2015, *Chemom Intell Lab Syst* 149:153-165.
39. National Judicial Data Grid, (2020, December 30). "Pendency of court cases in India." https://njdg.ecourts.gov.in/njdgnew/?p=main/pend_dashboard.
40. Nguyen S, Zhang M, Johnston M, Tan KC, "Automatic programming via iterated local search for dynamic job shop scheduling." *IEEE Trans Cybern* 45(1):1-14, 2014.
41. Pannu, A. S. "Using genetic algorithms to inductively reason with cases in the legal domain(1995). *ACM Digital Library*." <https://doi.org/10.1145/222092.222232>.
42. Paryati, Paryati. "Determination Of Mineral Mining Exploration Area With Fuzzy Using Genetic Algorithm To Reduce Mining Risk." (2022): 11-11.
43. Pawan, YVR Naga, Kolla Bhanu Prakash, Subrata Chowdhury, and Yu-Chen Hu. "Particle swarm optimization performance improvement using deep learning techniques." *Multimedia Tools and Applications* 81, no. 19 (2022): 27949-27968.
44. Piotrowski, Adam P., Jaroslaw J. Napiorkowski, and Agnieszka E. Piotrowska. "Population size in particle swarm optimization." *Swarm and Evolutionary Computation* 58 (2020): 100718.
45. Poli R, Kennedy J, Blackwell T, "Particle swarm optimization." *Swarm Intell* 1(1):33-57, 2007.
46. Prabhavathi, N., & Durai, K. "Role Of Artificial Intelligence in Access to Justice and Justice Delivery in India." *Library Progress International*, 44(3), 12328-12337, 2024.
47. Saxena, P. "Technological Innovations in India's Legal Sector for Access to Justice During and Post Pandemic." *Law and Development Review*, (0), 2024.
48. Sohail, Ayesha. "Genetic algorithms in the fields of artificial intelligence and data sciences." *Annals of Data Science* 10, no. 4 : 1007-1018, 2023.
49. Tahura, U. S., & Selvadurai, N. "The Use of Artificial Intelligence in Judicial Decision-Making: The Example of China." *Int'l JL Ethics Tech.*, 1, 2022.
50. Wang D, Tan D, Liu L, "Particle swarm optimization algorithm :an overview." *Soft Comput* 22(2):387-408, 2018.
51. Wang, Yu-Cheng, and Toly Chen. "Adapted techniques of explainable artificial intelligence for explaining genetic algorithms on the example of job scheduling." *Expert Systems with Applications* 237: 121369, 2024.
52. Wu TY, Lin CH, "Low-sar path discovery by particle swarm optimization algorithm in wireless body area networks." *IEEE Sens J* 15(2):928-936, 2014.
53. Yoon Y, Kim YH, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks." *IEEE Trans Cybern* 43(5):1473-1483, 2023.
54. Development Data Lab. 2021. <https://www.devdatalab.org/judicial-data>.
55. Zhou Z, Ong YS, Lim MH, Lee BS, "Memetic algorithm using multi surrogates for computationally expensive optimization problems." *Soft Comput* 11(10):957-971, 2007.
56. Zivkovic, Miodrag, Venkatachalam K, Nebojsa Bacanin, Aleksandar Djordjevic, Milos Antonijevic, Ivana Strumberger, and Tarik A. Rashid. "Hybrid genetic algorithm and machine learning method for covid-19 cases prediction." In *Proceedings of international conference on sustainable expert systems: ICSES 2020*, pp. 169-184. Springer Singapore, 2021.
57. Lin TL, Horng SJ, Kao TW, Chen YH, Run RS, Chen RJ, Lai JL, Kuo IH, "An efficient job-sjop scheduling algorithm based on particle swarm optimization." *Expert Syst Appl* 37(3):2629-2636, 2010.