International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(2): 943-957

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i02.07999

Semantic Tag Guided Genetic Programming for Context Aware Optimization

Kumarakrishnan S[1*], V Prasanna Venkatesan[1], Geetha S[1], Madusudanan J[2]

[1]Department of Banking Technology, Pondicherry University, Puducherry, India, [2]Department of Artificial Intelligence and Data Science, Sri Manakula Vinayagar Engineering College, Puducherry, India. *Corresponding Author's Email: skumarakrishnan@smvec.ac.in

Abstract

Context aware systems strive to make more applications intelligent and responsive based on personalised and effective resolutions. It is known that the environment is dynamic and uncertain; therefore, context aware systems require highly dynamic systems to manipulate and function. A genetic algorithm is an evolutionary-based algorithm that operates on a fitness function, crossover and mutation within different populations. An increase in population size reduces the performance of genetic algorithms, leading to bloating, non-convergence and decreased efficiency. A semantic-tag-based genetic algorithm processes the entire population with meaningful tags, which help the algorithm's functions categorise parents and produce the fittest offspring, thereby improving system performance. Therefore, this paper proposes semantic-tag-based genetic programming for context aware systems, which assigns semantic tags to contexts and applies a genetic algorithm. The proposed work was simulated in the Python environment using the Banking dataset from Kaggle, which showed 91.86% accuracy. Redundant data processing, a lightweight framework, an accurate system with semantic tags, crossovers and mutations in the genetic algorithm for context aware systems is achieved through the proposed work. Performance metrics such as confusion matrix, accuracy, precision, recall and region of convergence are experimented with and the results forecast the improvement in the proposed work in comparison with existing work.

Keywords: Context Aware Systems, Genetic Algorithm, Genetic Programming, Innovation, Optimization, Semantic Tags.

Introduction

According to the statistical report from Allied Market Research, the ubiquitous market will attain a Compound Annual Growth Rate (CAGR) of 19.4% from 2024 to 2032, meaning the project will be worth $3.6 billion in 2032 (1). The growth is mainly due to the impact in areas such as wearable computing, smart homes, smart cities and healthcare. Context aware systems produce an immense amount of data — nearly 73.1 zettabytes worldwide —mainly by enabling input and output devices to perform operations based on personalized experiences and data-driven approaches using efficient algorithms. Context aware systems encompass the entities that are the information used to characterize the situation, sense the information from the surroundings, interpret the data and respond based on the current situation (2). Thus, context aware systems are more intelligent and personalized, producing seamless information. As context aware systems are human-centric, smart and require a paradigm shift from conventional approaches, they must operate under uncertain conditions and with a genuine understanding of the instances present in the environment. Therefore, the system needs to detect or sense novel conditions, genuinely understand uncertainty, modify the system based on these understandings and provide experiences beyond fixed responses. The requirements that context aware systems need to improve are anticipation, proactivity, reduced friction, decreased cognitive load, delivery of relevant information, efficient resource utilization, improved resilience, robustness and fault tolerance. The very prominent Evolutionary algorithm that addresses issues in gradient descent optimization is the Genetic Algorithm (GA), which uses a genetic approach to optimize problems under nonlinear conditions (3). The core idea behind this algorithm is the "survival of the Fittest, where the offspring are reproduced from the fittest parents (4). This is an efficient algorithm that has the characteristics of a population-based search, does not require gradient or derivative

(Received 03rd September 2025; Accepted 06th March 2026; Published 14th April 2026)

information, is flexible for a wide variety of multi-modal and multi-objective problems and operates in the global optimization paradigm. A captivating Genetic algorithm within artificial intelligence that applies the core principle to overcome the drawbacks of back propagation techniques uses differentiable models that are difficult to compute. Therefore, this work aims to make the context aware system perform under nonlinear and uncertain conditions using a semantic-tag-based evolutionary genetic algorithm in a hybrid approach to make the system more personalized.

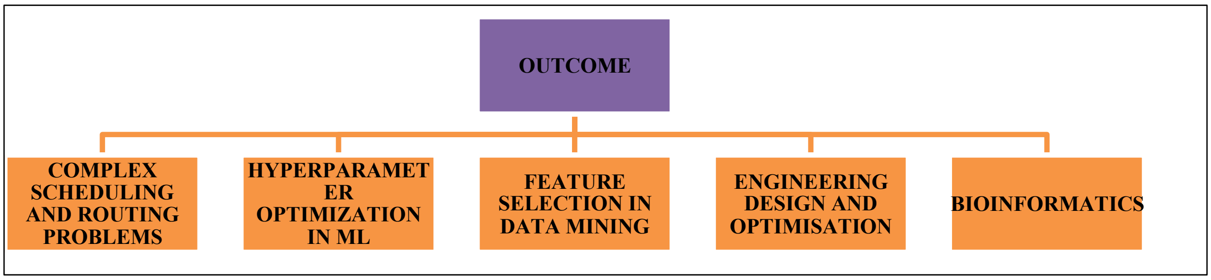

Figure 1: Outcomes of GA from Survey

Survey on Genetic Algorithms

The biological concepts like population, chromosome, gene and fitness function concepts that are used in the Genetic algorithm are applied along with the operators like selection, crossover and mutation, which are used to obtain the optimization problems that find wide application in the area of neural networks of almost all the domains. The advantages of using GA are that it can handle complex data, nonlinear relationships between input and output, help to obtain solutions for non-differentiable problems, is parallelizable and is robust to local minima, with the limitations of slow convergence, requires efficient parameter tuning and obtaining a fitness function for complex and nonlinear problems (5). GA's core mechanism is the Schema Theorem and parallelism, which provides the mathematical functioning of schemas in chromosomes in parallel using the binary strings of representation of time and it finds wide applications in areas like numerical and combinatorial itemization, rule-based system design, robot design and control and modelling of social and biological systems (6). The authors explore the steps of GA techniques, encoding schemes, initialization methods, fitness functions and genetic operators and provide information on their variants, such as generational, steady-state and micro-GA, as well as their hybrid approaches with neural networks (7).

The core concept of GA helps with feature selection, hyper parameter optimization and model architecture search and rule discovery helps with AI techniques such as machine learning (ML) and deep neural networks; thus, this algorithm synergises with data science as well (8). There are many techniques involved in the each step of the GA algorithm is addressed, like binary, tree, real, permutation etc., for encoding techniques, diverse and semi-optimized population initiation methods, fitness function modes, Rouellete Wheel, tournament, rank and stochastic universal sampling based selection criteria, choosing of cross over operators like one and two point, uniform, arithmetic, cycle and order, bit-flip, swap, inversion, insertion and Gaussian based mutation operators, generational and steady-state replacement strategies (9). The application of GA in machine learning is reviewed from various perspectives: as an ML tool; ML applications covering subset selection, learning classifier systems, neural network training, etc.; ML contexts; academic contexts; and considerations for ML practitioners (10). Engineering domain-based application of the GA algorithm focuses on the structural, design, control and resource allocation optimization of engineering systems to address complex optimization factors (11). Figure 1 provides the GA algorithm applications based on the survey.

Survey on Context Aware Algorithms

Context aware systems process an input or sensed entity and respond based on the algorithm used. Therefore, context awareness helps humankind through ubiquitous technology by enabling decision-making, providing feedback, addressing challenges, facilitating interfaces and interactions and addressing ethical considerations (12). Various approaches to context in intelligent systems, such as sense context, acquire context, model context, use context and architectural context, are important for understanding present systems (13). Context awareness is a kind of intelligence that computers possess for real-world situations, following four basic taxonomies such as proximate selection, automatic contextual configuration, contextual information, commands and context-triggered actions (14). In the aim to review various concepts of context aware protocols for IoT, such as sensing, modelling, inferring, dissemination and retrieval and commented on the challenges (15). The synergy between communication technology and context aware techniques is discussed, summarizing the communication barriers, their different forms, dimensions, challenges and applications (16).

The survey of context aware mobile computing provides good insight into the system in the present, past and future, followed by its limitations, such as being less reliable and accurate, lacking fine-grained optimization and having limited security and privacy, as well as diverse information failing to support reasoning and modelling of complex systems (17). The infrastructure of the context aware system mainly depends on low-level entity acquisition, information abstraction, network distribution, application adaptation logic and user interface adaptation (18). Engineering practices for context aware systems include architecture, methodology, software engineering processes, design, development, deployment and maintenance of context aware systems and applications (19). Based on the survey insights, Figure 2 presents the challenges of context aware systems that need to be addressed. Table 1 provides a detailed survey of the GA algorithm applied to context aware systems over the years of research. The table includes fields for the research conducted, the method used and the work's inference.

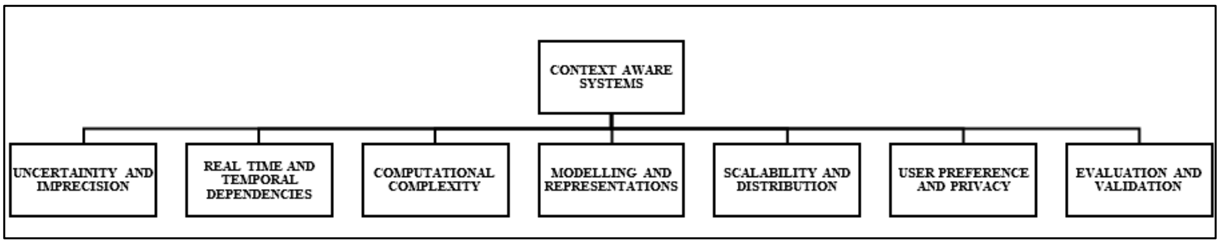

Figure 2: Challenges of Context Aware Systems

Table 1: Survey on Research Combining GA and Context Aware Systems

Year	Field	Method	Inference	Reference
2010	E-Learning	Adaptive learning using GA using the students' contexts	Instead of static content delivery, the GA helps to generate and optimise the learning ways based on the student's context, which results in more adaptive learning	(20)
2018	Recommendation systems	Spatio-temporal context	Based on the queries "when and where, GA algorithms use spatio-temporal contexts that produce the pertinent and actionable recommendations	(21)
2013	Web services	Optimal service compositions	A context space model based on the queries that employs the GA to address the specific entities of the user to produce highly relevant and performant solutions for the user.	(22)
2020	Recommendation systems	Context weighting and data sparsity	Obtains the contextual factors that intelligently weigh the context info and data sparsity for effective recommendations using GA	(23)
2020	Smart cities	Energy efficiency in fog computing	The GA algorithm is used to balance energy efficiency and adaptability to develop context aware, responsive urban infrastructures in fog computing environments.	(24)
2013	Multiple web services	Automatic integration using GA	To provide quality of service GA GA-based context aware systems are used to assemble multiple web services.	(25)
2024	Movie recommender system	Integration of a content-based system	Incorporating the content-based system with the context aware system, where the optimal solution is obtained using the GA algorithm for a personalized entertainment experience.	(26)
2007	Pervasive computing	Task distribution	Based on the optimization of GA using the contexts based on energy, resources and performances, task is allocated constantly based on the real-world actions.	(27)

2017	Grammar based	Rules mining association	Automatic generation of useful data from the contexts using the GA algorithm based on the formal grammars	(28)
2005	Object recognition systems	Optimisation for object identification	GA used to adjust the parameter values to obtain the exact identification of the object using the contextual information.	(29)

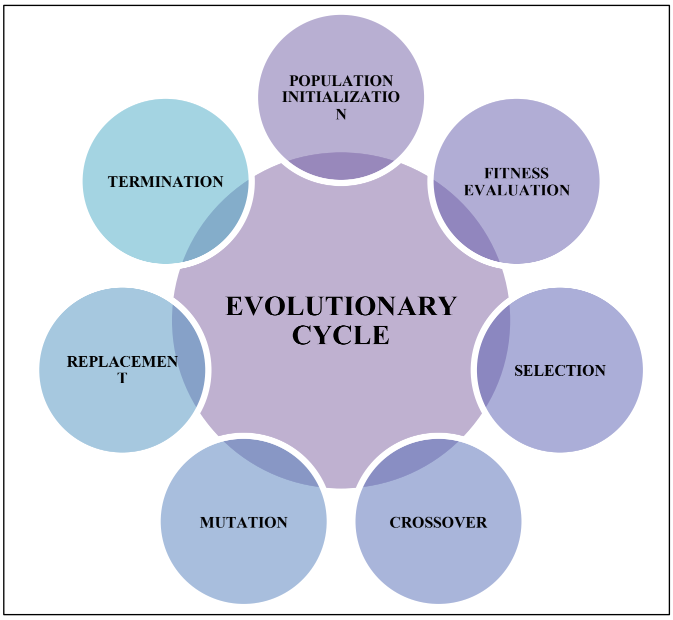

Figure 3: Processes in Genetic Algorithm

Genetic Algorithm

By emulating the process of natural evolution, a powerful and robust technique named "Genetic Algorithm" is designed for optimisation and searching processes to obtain high-quality solutions. The goal is to evolve the population from the current gene type through crossover and mutation, producing high-quality offspring selected by the fitness function.

Figure 3 shows the process of the genetic algorithm, framed in terms of natural evolution. The following steps explain the function of each process in detail:

Initialization: It is the mere random fixation of an initial population number (e.g., a Binary string, a real number, or a permutation) that represents the encoding information of potential solutions.

Fitness Function: The problem-specific fitness function f(x) will be designed so that individuals/chromosomes/genes are selected for crossover. Therefore, this fitness function helps in the formation of the desired quality output. Generally, the fitness function aims to maximise or minimise a function. Common characteristics of fitness functions include non-negativity, scalability, discriminability, computability and weighted functions.

Selection: The Mating pool is created to reproduce fitter individuals and form the population for the current state. The various selection methods are listed below:

a) Selection based on Fitness function -Roulette Wheel Selection or Proportional selection.
b) Fittest individual chosen from the subset of random selection -Tournament method.
c) Ranking proportional to fitness and selection based on rank that overcomes the very fit genes -rank selection method.
d) Prioritising the very good solutions that are obtained from the previous generation and following the same elitist method.

Crossover: This is the recombination of individuals from the mating pool, selected by a specific method and combined using the crossover operator. The crossover rate is the parameter that specifies the probability that parts of the genetic material from both parents are crossed over. The types of crossovers that happen are discussed below:

a) Swapping of genetic material between the parents based on the random crossover point, called single-point crossover.

b) Swapping of segments between the two parents is called the two–point crossover method.
c) Independent exchange of genes based on a certain probability is called as uniform crossover.
d) The exceptional point is that the offspring merely copy the parents' genes where the crossover does not occur.

Mutation: To introduce diversity within the population, it is necessary to involve mutation, where, in the usual case, it is the mere flipping of a bit; thus, one or more genes are randomly varied on the chromosomes to prevent premature convergence to local minima.

Replacement: A new population is formed by replacing the old parents with the latest, capable, fittest offspring, which are selected based on the fitness function for the evaluation of the fittest parent genes.

Termination: Based on a satisfactory level or a fixed computational budget, the convergence of the population is determined over the number of generations.

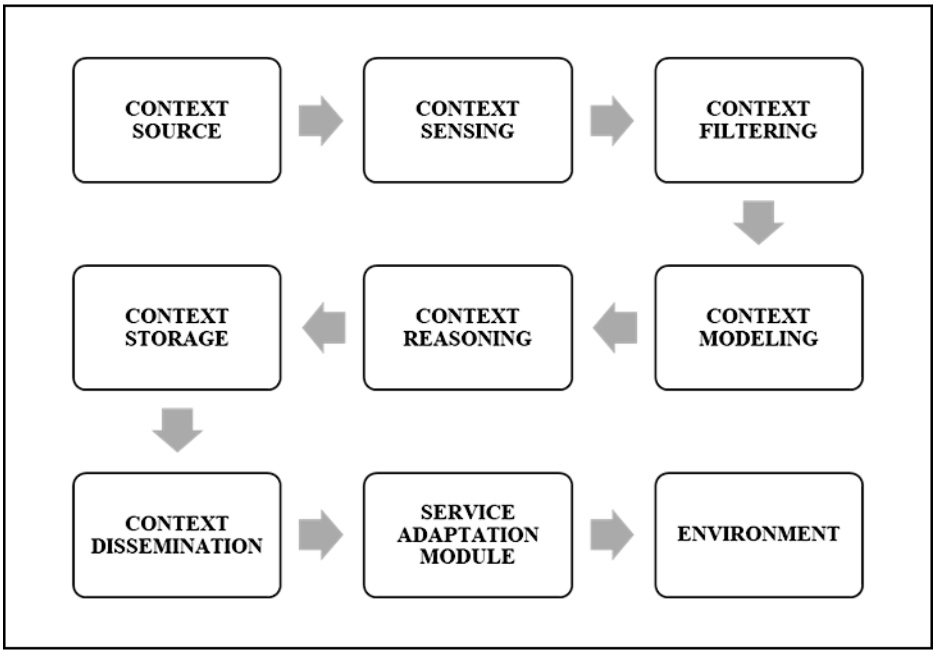

Figure 4: Process in Context Aware Systems

Context Aware Systems

Given the uncertain situations and dynamic environments, the context aware system must detect changes in the state of ongoing activities and process them. The main objective of the context aware system is to interface in a meaningful way using the collected and processed contexts. Figure 4 shows the
Functionality of a context aware system.

Context sources: Context sources are the inputs to the system that help in the study and they can be physical, software, or human sources. Sensors are the main devices for collecting information. Sensor types like physical sensor (RFID, cameras, mic, GPS, etc.), logical sensor (calendars, digital profiles, weather forecasts, etc.) and user-specific sensors (wearable devices, person sensor)

Context Sensing: The sensors collect 'n' number of data from a single entity, which are of heterogeneous types and in different data formats. Data are collected continuously by sensors and APIs retrieve information and query databases. It acts as the acquisition module between the real environment and context management systems.

Context Filtering: The raw data obtained after acquisition is subjected to noise reduction, data fusion, normalisation and aggregation and missing values are handled in the context filtering process.

Context Modelling: For effective dissemination and reasoning, context modelling uses mechanisms such as key-value pairs, ontologies, object-oriented models, relational models and context trees to convert preprocessed information into a formal, machine-learnable form.

Context Reasoning: Here, the abstraction of contexts from labelled information is performed using rule-based reasoning, machine learning, probabilistic reasoning and case-based reasoning.

Context Storage: It is necessary for context aware systems to store collected contexts, past pattern recordings for future analysis and queries for various modules.

Context Dissemination: The availability of context reasoning is achieved using the publish-subscribe model, querying interfaces and event-driven architectures.

Service Adaptation Module: Context awareness is manifested through content adaptation, user interface adaptation, functional adaptations, proactive actions and quality of service adaptations to provide a personalised experience.

Environment: Based on the adaptive services provided above, the entity context takes action and receives feedback from the context sources.

Methodology

Lightweight Genetic Programming with Semantic Tagging

The conventional genetic algorithm focuses more on the structure and syntax of the information provided than on the meaning of the context (30). The resulting issues include bloat—program size growing without converging on increased fitness; semantically meaningless offspring being reproduced due to random recombination, leading to an inefficient search; and the absence of meaningful diversity converging at local minima. To address these issues, the semantic level was also added to the Genetic algorithm, which describes the functionality and purposes of the system more than conventional GA methods do (31). The semantic information is added as "tags" that represent the behavioural patterns of the system's input and output (32). These tags can be obtained either through human specification or through model learning based on the input and output characteristics. The lightweight of Semantic tagging in Genetic algorithms is obtained through the following operations,

a) Semantic crossover reduces the arbitrary selection of crossover points by selecting the pair by semantical compatibility.
b) Semantic mutation helps to achieve a similar functional effect by replacing the random selection of a node or a tree.
c) The redundant information is reduced by semantics that neglects the redundant data, which makes the system less bloated.
d) Targeted search can be obtained through a semantic distinction that improves the diversity of the system.

Semantic Tagging-based Genetic Algorithm for Context Aware Systems

To address the uncertainty, complexity and inherent dynamism of context aware systems, a semantic-tag-based genetic programming method is incorporated into the context aware system, which evolves to find efficient solutions through its adaptive logic and semantic tags. The following functions that take place in the proposed framework are discussed in Figure 5.

Semantic Tags for Entities in Context Aware System: In the input semantics, the tags should describe the type of the fragment, such as location, time, or user activity (e.g., AT_HOME is tagged as LOCATION_UPDATE). Based on the genetic algorithm model's processing of the input function, the adaptive actions are reflected in the output semantics. (e.g. EATING_AT_HOME is tagged as ACTIVITY INFERENCE: DINNER_COMPLETED based on the location and time).

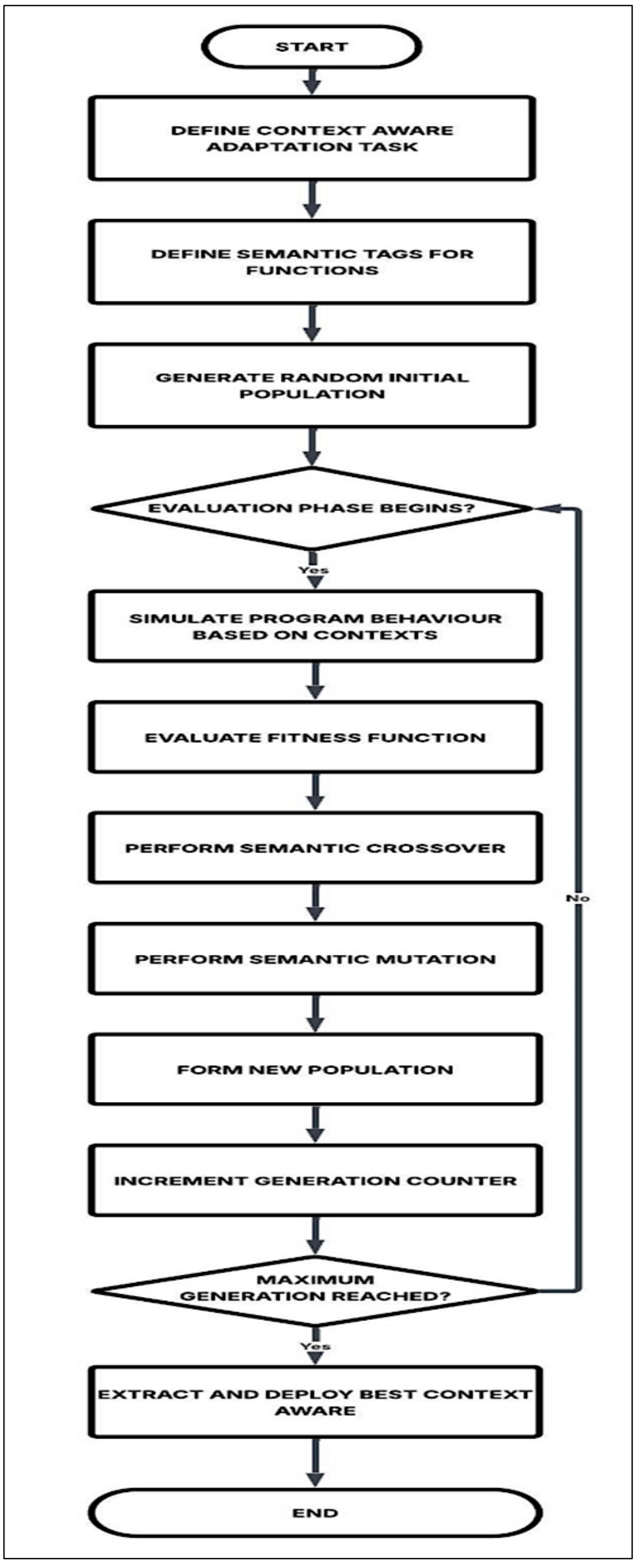

Figure 5: Flowchart of the CA based Genetic Programming

Mechanism

a) The programming representation of the above input semantics is rich vector embeddings. (e.g., LOCATION_TIME (LAT, LOG, HOUR, MIN), TAG – SPATIAL_TEMPORAL CHECK). These labels are converted to vector embeddings based on the number of tags identified.

b) Based on the above tagging, the groupings of the context entities are done; therefore, the crossover takes place corresponding to the semantic tags that improve the validity of the inference produced.

c) In the mutation part, actions like mutation of similar information about the same thing can be done, rather than irrelevant mutation. (e.g., LOCATION_TIME mutated with GPS_INFO)

d) The fitness function has no direct relation with the semantic tags, but it helps the fitness function to evaluate and produce the output with adaptation quality and user satisfaction.

Thus the incorporation of semantic tagging based genetic algorithm makes the context aware systems to dynamically adapts to the situation where the semantic tags changes based on the real -time supporting systems, the relevant services can be delivered based on the user's current context, provides highly customizable services based on the semantic tags like user profile, their preferences, or by habits, etc. Decision-making is enhanced by optimal actions and service configurations derived from user and environmental context using GA. Thus, the semantic tagging handles structured context, whereas genetic algorithms provide adaptive optimization, enabling characteristics such as smart, personalized and scalable actions in context aware systems.

Figure 5 is the flowchart of the proposed work, where in the initial steps, the context aware adaptation task, semantic loss for function and generation of random initial populations are initiated. Once the phase begins, the genetic simulation starts operating, during which the fitness function is evaluated. After the fitness function evaluation, based on its value, crossover and mutation are performed using the semantic tags of the context awareness. After this, a new population is created and the generation counter is incremented. If the phase is converged, deployment of the best context aware system is done.

The fitness function is calculated using the formula in Equation [1], where the predictive performance (α), contextual adaptability (β) and model simplicity (γ) are used for weighing the fitness function. In Equation [1], the primary objective is given to the predictive correctness of high priority, the contextual relevance is aligned based on the semantic contexts and lower importance is given to the penalty weight to avoid the over-penalization.

$$\mathrm{F(p)} = \alpha.\,\mathrm{Accuracy(p)} + \beta.\,\mathrm{Contextual\ relevance} - \mathrm{(p)}\gamma.\,\mathrm{complexity(p)} \qquad [1]$$

Where, F(p) is the fitness score. α – weight based on prediction accuracy, β- weight based on contextual relevance, γ- weight based on penalty.

The semantic relevance when selecting or mutating individuals is given by Equation [2] for the semantic tags for the contexts s_i, where n is the number of tags used.

$$\mathrm{rel(p,c)} = \sum_{i=1}^{n} w_i.\,\delta(s_i, p) \qquad [2]$$

Where, s is the semantic tag for that context, w_i is the weight of the relevance score, which helps in context aware modifications. $\delta(s_i, p)$ Is the indicator function. It will be one if s_i is encoded in p, or else zero.

The crossover between the individuals is given by Equation [3], where γ is the scaling bias factor, rel(p,c) is the relevance score of the individual p in the context c,

$$P_{crossover(p_1,p_2|c)} = \gamma.\left(\frac{\mathrm{rel}\ (p_1,c)+\mathrm{rel}\ (p_2,c)}{2}\right) \qquad [3]$$

Using Equation [3], the context aware fitness function is evaluated by Fitness (p, c), which is the base fitness value of the individual p in the context c. Equation [4] helps to handle the heterogeneous contexts for the 'p' population of individuals that normalise the fitness of prelative to the population,

$$n\mathrm{ormFitness(p,c)} = \frac{\mathrm{Fitness(p,c)}}{\mathrm{fitness\ (p',c)}} \qquad [4]$$

Notation and Primitives

a. Population at generation t: $P^{(t)} = \{p_1^{(t)}, \dots, p_N^{(t)}\}$

b. Context c: Has semantic tags $S(c) = \{s_1, \dots, s_n\}$ with non-negative weights $w_1, \dots, w_n$.

c. Indicator$\delta(s_i, p)$: Value is 1 if tag s_i is encoded/present in the individualp; Otherwise, it is 0.

d. Base fitness: $\text{Fitness}(p,c) \in R_{\geq 0}$

Context aware Selection Pressure

The selection score $\phi(p|c)$ by Equation [5],

$$\phi(p|c) = \lambda \cdot \text{normFitness}(p,c) + (1-\lambda) \cdot \ \text{rel}(p,c) \qquad [5]$$

Where $\lambda \in [0,1]$ and rel(p, c)is the normalised relevance by Equation [6], where λ is the trade-off balancing fitness and semantic relevance,

$$\text{rel}(p,c) = \left\{\frac{\text{rel}(p,c)}{q \in P(t)\ \text{rel}(q,c)}, \text{if maxrel} > 0, 0, \text{otherwise.}\right\} \qquad [6]$$

The roulette-wheel (proportional) selection probability by Equation [7], where Pr is the context aware selection of the individual p and q,

$$\Pr\{p \text{ selected as a parent} \mid c\} = \frac{\phi(p|c)}{\sum_{q \in P(t)} \phi(q|c)} \qquad [7]$$

Context aware Crossover and Expected Offspring Count Semantic Mutation:

The **s**emantic deficit d at the position iIds by Equation [8], where m_i is the semantic tag importance weight for tag s_i,

$$d_i(p,c) = m_i(c) \cdot \big(1 - \delta(s_i, p)\big) \qquad [8]$$

The biased mutation probability towards missing semantics p_{mut} by Equation [9], where p_m is the base mutation rate. ρIs the bias parameter controlling the semantic guided mutation, L is the length of the individual and ϵIs the small positive constant,

$$p_{mut}(i|p,c) = \frac{d_i(p,c)}{\sum_{j=1}^{L} d_j(p,c) + \epsilon} \cdot p_m + (1-\rho) \cdot \frac{1}{L} \cdot p_m \qquad [9]$$

Population-Level Dynamics (Replicator-Mutator Form):

The context aware reproductive fitness f(p|c) by Equation [10],

$$f(p|c) = \lambda \cdot \text{normFitness}(p,c) + (1-\lambda) \cdot \text{rel}(p,c) \qquad [10]$$

The expected frequency update (Markov–replicator equation) by Equation [11], $\sum_p \pi^{(t)}(p)$Is the summation of the frequency probability of individual p at generation t, $T_c(p \to q)$ context-dependent transition probability from p to q,

$$\pi^{(t+1)}(q) = \frac{1}{Z^{(t)}} \sum_p \pi^{(t)}(p) f(p|c) T_c(p \to q) \qquad [11]$$

Where $Z^{(t)} = \sum_p \pi^{(t)}(p) f(p|c)$ is the normalisation constant.

Schema Theorem (Context aware Lower Bound)

Let H Be a schema (pattern) specifying o(H)Fixed positions and use d(H)Or its defining length. Suppose crossover points are sampled uniformly over. $L-1$Cut sites and are semantic-aligned with probability $a_c(H|c)$ (i.e., a cut that preserves H's specified tags). Then the probability that the crossover does not disrupt H is by Equation [12],

$$P^{xo}_{\neg disrupt}(H|c) = 1 - P^{avg}_{crossover} \cdot \big(1 - a_c(H|c)\big) \cdot \frac{d(H)}{L-1} \qquad [12]$$

Where $P^{avg}_{crossover}$Is the average of Equation [3] over mating pairs.

With per-locus guided mutation, the probability of mutation is preserved. H is by Equation [13],

$$P^{mut}_{\neg disrupt}(H|c) = \prod_{i \in \text{fixed}(H)} \big(1 - p_{mut}(i|p,c)\big) \approx 1 - \sum_{i \in \text{fixed}(H)} p_{mut}(i|p,c) \qquad [13]$$

Let, m(H, t)Be the number of instances of the schema H at generation t and $\underline{f}^{(t)}$ the population-mean of f. Then the expected schema growth obeys the context aware schema theorem by Equation [14], where E(.)

is the expectation parameter, $P^{xo}_{\neg disrupt}$ is the probability due to crossover distortion and $P^{mut}_{\neg disrupt}(H|c)$ is the probability of disruption due to mutation,

$$E[m(H,t+1)] \geq m(H,t) \cdot \frac{f(H|c)}{\underline{f}^{(t)}} \cdot P^{xo}_{\neg disrupt}(H|c) \cdot P^{mut}_{\neg disrupt}(H|c) \quad [14]$$

Where, f(H|c)Is the mean f(p|c) over strings in H.

Semantic alignment $a_c(H|c)$Nd deficit aware mutation raises the preservation factors, tightening the bound.

Elitism and P_{BEST}:

The best individual

$p^* = \ \text{argmax}_{p \in P^{(t)}} \text{Fitness}(p,c).$

The best fitness is non-decreasing:

$F^{(t+1)}_{BEST} \geq F^{(t)}_{BEST}$

The best-evolved individual, P_{BEST}, is updated as follows by Equation [15],

$$P^{(t+1)}_{BEST} = \left\{ p^*, \text{if Fitness}(p^*,c) > \text{Fitness}\left(P^{(t)}_{BEST},c\right), P^{(t)}_{BEST}, \text{otherwise.} \right\} \quad [15]$$

One Generation Expectation

Let $\kappa^{(t)}(p)$The expected total offspring count of p after selection, crossover and mutation by Equation [16],

$$k^{(t)}(p) = 2M.\Pr \Pr\{p \text{ is picked as a parent}\}.$$
$$(E[p].E[P^{XO}_{disrupt}.P^{mut}_{disrupt}|p] + (1 - E[P_{crossover]}).E[P^{mut}_{disrupt}\ |p]] \quad [16]$$

This expression shows how relevance, fitness, semantic alignment and mutation guidance jointly shape reproductive success.

The algorithm for the proposed work is provided below, where the initial parameters used for simulation purposes are explained. Originally, the initial population is set for N random programs, where the semantic node tags are assigned to all the functions. Initially set the best evolved context has zero. Evaluate the fitness function and if the current fitness value is greater than the best fitness value, store it as the new nest fitness value and the probabilities are also taken as the best fitness function. Using this fitness function, mating pools are created and they are subjected to the semantic crossover and semantic mutation and they add the population by returning the best evolved context and the program carries out the same operation for 'N' random programs.

Algorithm

F -SET OF INPUT TAGS
T- SET OF OUTPUT TAGS
N-POPULATION SIZE
G_{MAX}- MAX GENERATIONS
P_C CROSSOVER PROB
P_M – MUTATION PROB
C_T-SET OF FITNESS EVALUATED CONTEXTS
P_{BEST}-BEST EVOLVED CONTEXT

Initialization:

a) CREATE INITIAL POPULATION $P^{(0)}$ FOR N RANDOM PROGRAMS
b) ASSIGN SEM_NODE TAGS FOR ALL THE FUNCTIONS
c) P_{BEST}=0

Evolutionary Phase:

For I=1: N
- a) EVALUATE FITNESS FUNCTION, F(i)=FITNESS(Pi, C_T)
- b) if F(i) >F_{BEST}
- c) F_{BEST}=F(i)
- d) P_{BEST}=Pi
- e) end
- f) CREATE MATING POOL SELECTING N PARENTS
- g) $P_{X'}$, $P_{Y'}$ =SEMANTIC CROSSOVER (P_X, N_X, P_Y, N_Y)
- h) $P_{Y'}$=SEMANTIC MUTATION ($P_{Y'}$, N_{MY}, $S_{REPLACE}$)
- i) ADD $P_{X'}$
- j) RETURN P_{BEST}
- k) End

Results and Discussion

The proposed framework of semantic tagging based on genetic programming for the context aware systems is simulated using the parameters listed in Table 2 with their corresponding values. The maximization method is used for selecting the best context aware system. The dataset based on the Bank domain is used for Kaggle, where it contains 4521 entries and 18 columns of age, job, marital, education, default, balance, unnamed, housing, loan, contact, day, month, duration, campaign, passed days after campaign pdays, previous, poutcome and the target variable y estimated as the term deposit. The data split is 75% for training and 25% taken for testing with the random state of 10.

Table 2: Simulation Parameters

Parameter	Value
Population Size	100
No. of generation	20
Selection	Tournament of Size 3
Crossover Rate	0.80
Mutation Rate	0.20
Elitism Rate	0.10
Terminal Set	Semantic Tags
Function Set	If and, Or, > , <, =,+,-
Fitness Function Weights	α= 0.7, β=0.25, γ=0.05
c criteria	Maximization

The proposed work was simulated in Python 3.8.20 using Streamline and Tensor Flow and the end classification was done using the Random Forest method to overcome the overfitting, where the GP is used for feature optimization and classification. Figure 6 is the confusion matrix of the proposed work.

It is seen that the work classifies 982 semantic tags successfully as true negatives (TN) and 38 semantic tags as true positives (TP). There are 87 semantic tags represented as false negatives (FN), which may be due to the misinterpretation of the semantic tags for the crossover and mutation process and 24 false positives (FP). Figure 7 represents the Receiver Operating Characteristics to define the ability of the proposed work for the classifier system. It is seen that the proposed work shows more than 0.5; the proposed work has not shown any saturation, but the rate increase is identified. This proves the proposed work efficiency, where the context aware system chooses the best prediction model. Due to the imbalance in the dataset, the minority classes are underrepresented, which is the reason for the AUC of 0.68 and the majority classes show dominance with high accuracy.

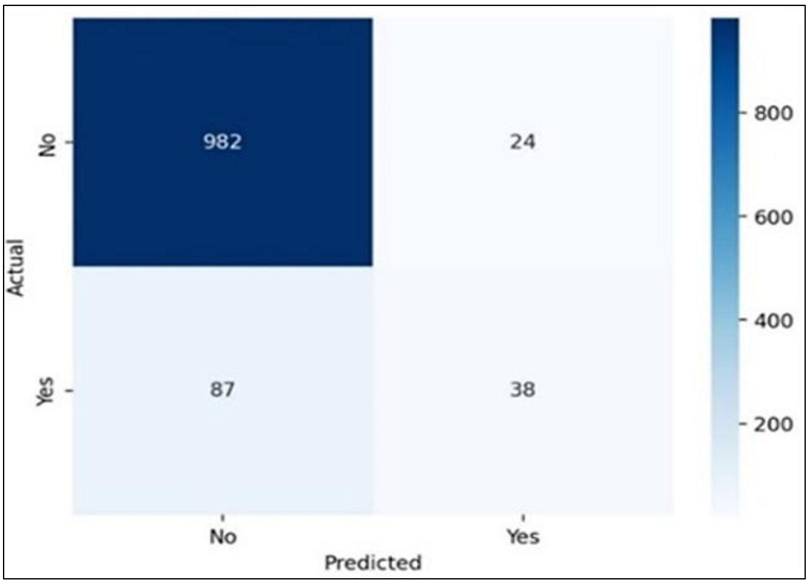

Figure 6: Confusion Matrix of the Proposed Work

Table 3: Classification Report

Metrics	Decision	Rerecall	F1-Score	Support
00	0.9186	0.9761	0.9465	1006
11	0.6129	0 .304	0.4064	125
Accuracy	0 .9019	0.9019	0.9019	0.9019
Macro Avg	0.7658	0.6401	0.6765	1131
Weighted Avg	0.8848	0.9019	0.8868	1131

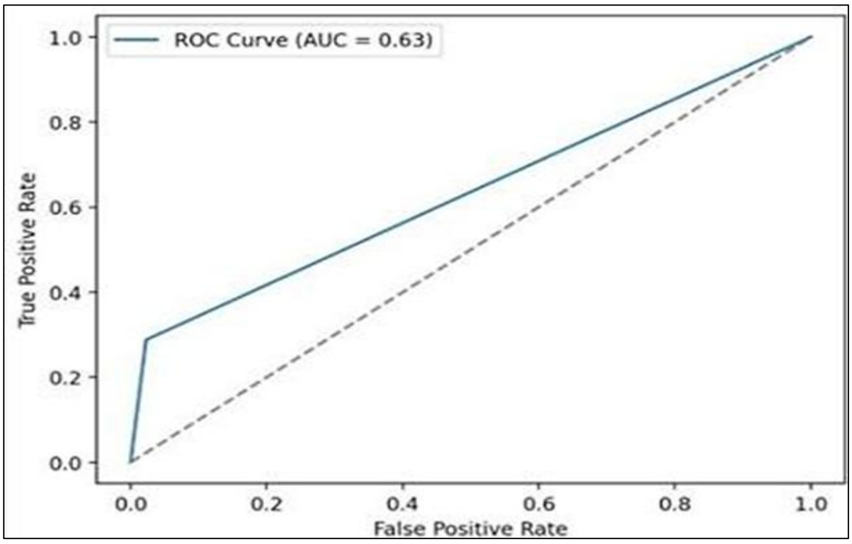

Figure 7: ROC for the Proposed Work

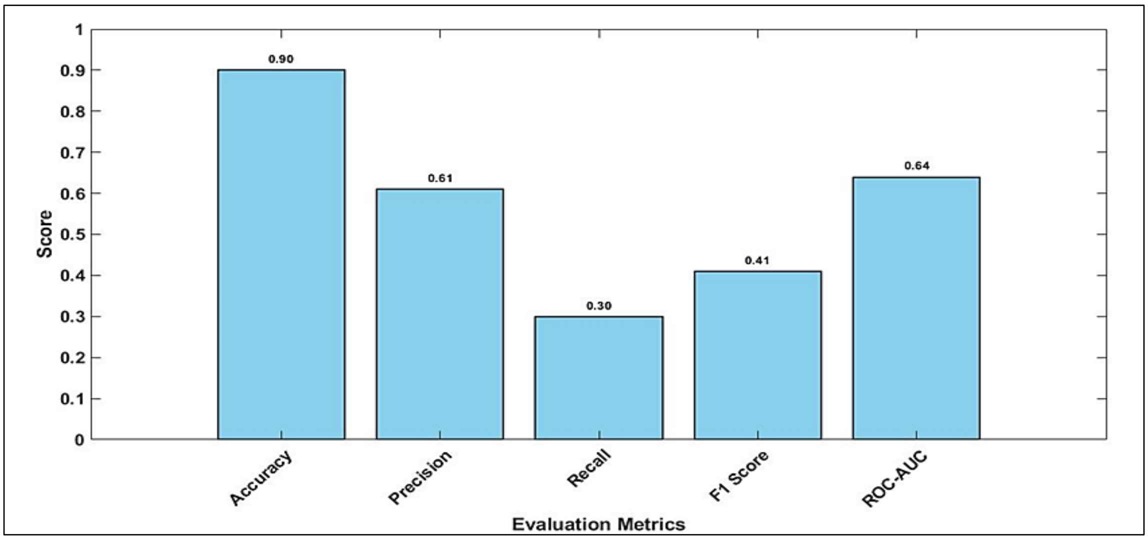

Figure 8: Evaluation Metrics for the Proposed Work

Figure 8 is the evaluation metric plot for the proposed work based on the values of the confusion matrix. The accuracy of the work obtained 91.86% with moderate precision and recall value. The F1 score stood 41.64% that required improvement in binary 1 classification. This lower F1 score is mainly due to the high imbalance in the dataset used.

The proposed work is compared with the existing context aware systems. The proposed work outperforms the conventional system by 4% and it is noted that the precision of the proposed work shows nearly 30% improvement compared to traditional context aware systems shown in Figure 9 (33). This helps to find the robustness of the proposed work by using the semantic tags based crossovers and mutations in the genetic algorithm for the context aware systems. The binary classification trade-off for the proposed work is also shown 16% efficient compared to the context aware systems only (34). Table 3 classification report discusses the performance metrics for the proposed work.

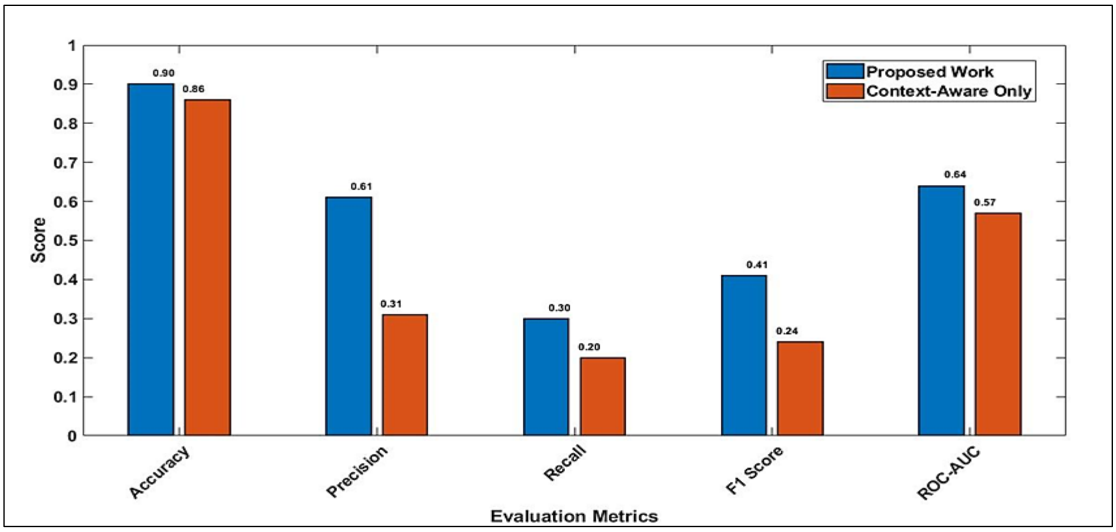

Figure 9: Comparison of the Proposed Work with the Existing Context aware Only System

Conclusion

The proposed work of using lightweight semantic tagging using genetic programming for context aware systems is described and the outputs obtained using Python are presented. It is seen that the proposed work shown 91.86% accuracy with 68% of AUC. The semantic tags used for contexts efficiently uses genetic algorithm to identify the functions based on the tags and process them. Semantic tags provided the meaning information that was actually reflected in the output. Semantic tags not only improved the accuracy of the system by providing lightweight to the whole genetic programming that reduces the processing of redundant data. The extension of this work is planned to apply variants of genetic programming using semantic tags for context aware systems.

Abbreviations

AI: Artificial Intelligence, AUC: Area Under the Curve, CAS: Context aware Systems, GA: Genetic Algorithm, GP: Genetic Programming, ML: Machine Learning, ROC: Receiver Operating Characteristic.

Acknowledgement

The Authors like to thank Pondicherry University for the technical support and experimental validation are done using the datasets from open access Kaggle.

Author Contributions

Kumarakrishnan S: conceptualization, research design, simulation of the results, V Prasanna Venkatesan: supervision, technical validation, Geetha S: formal analysis, manuscript editing, Madusudanan J: proofreading, validation of the work.

Conflict of Interest

There is no conflict of interest for this publication.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Declaration of Artificial Intelligence (AI) Assistance

This work declares that no generative AI tools and AI assited technologies are used for writing process.

Ethics Approval

The research utilized only publicly available datasets and no involvement of private sources.

Funding

None.

References

1. Papneja H, Yadav N. Self-disclosure to conversational AI: a literature review, emergent framework and directions for future research. Pers Ubiquit Comput. 2025; 29:119–151. https://doi.org/10.1007/s00779-024-01823-7
2. Hong JY, Suh EH, Kim SJ. Context aware systems: A literature review and classification. Expert Syst Appl. 2009;36(4):8509-22. https://www.researchgate.net/publication/220219749_Context aware_systems_A_literature_review_and_classification

3. Holland JH. Genetic algorithms. Sci Am. 1992;267(1):66-73. http://dx.doi.org/10.1038/scientificamerican0792-66
4. Floreano D, Dürr P, Mattiussi C. Neuroevolution: From architectures to learning. Evol Intell. 2008;1:47-62. doi: 10.1007/s12065-007-0002-4
5. Lambora A, Gupta K, Chopra K. Genetic Algorithm- A Literature Review. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon*)*, Faridabad, India. 2019:380-384. doi: 10.1109/COMITCon.2019.8862255.
6. Forrest S. Genetic algorithms. ACM Comput Surv. 1996;28(1):77-80. https://www.sci.brooklyn.cuny.edu/~chipp/cis32/papers/forrest96genetic.pdf
7. Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present and future. Multimed Tools Appl. 2021;80:8091-126. doi: 10.1007/s11042-020-10139-6
8. Haldurai L, Madhubala T, Rajalakshmi R. A study on genetic algorithm and its applications. Int J Comput Sci Eng. 2016;4(10):139-43. https://www.researchgate.net/publication/309770246_A_Study_on_Genetic_Algorithm_and_its_Applications
9. Sohail A. Genetic algorithms in the fields of artificial intelligence and data sciences. Ann Data Sci. 2023;10(4):1007-18. doi:10.1007/s40745-021-00354-9
10. Tabassum M, Mathew K. A genetic algorithm analysis towards optimisation solutions. Int J Digit Inf Wirel Commun. 2014;4(1):124-42. doi:10.17781/P001091
11. Shapiro J. Genetic algorithms in machine learning. Advanced Course on Artificial Intelligence. Berlin: Springer; 1999: 146-68. https://doi.org/10.1007/3-540-44673-7_7
12. Jiang N, Liu X, Liu H, Lim ETK, Tan CW, Gu J. Beyond AI-powered context aware services: the role of human–AI collaboration. Ind Manag Data Syst. 2023;123(11):2771-802. doi:10.1108/imds-03-2022-0152
13. Coppola P, Della Mea V, Di Gaspero L, Lomuscio R, Mischis D, Mizzaro S, *et al.* AI techniques in a context aware ubiquitous environment. Pervasive Computing: Innovations in Intelligent Multimedia and Applications. 2010: 157-80. https://doi.org/10.1007/978-1-84882-599-4_8
14. Aziz H, Afzal S. Critical Insights into Context aware Systems: A Literature Review. IJCR. 2024 Dec;19:1(1). https://ijcres.in/index.php/ijcr/article/view/18
15. Schilit B, Adams N, Want R. Context aware computing applications. First Workshop on Mobile Computing Systems and Applications. IEEE; 1994: 85-90. doi: 10.1109/WMCSA.1994.16
16. Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: A survey. IEEE Commun Surv Tutor. 2013;16(1):414-54. doi:10.1109/surv.2013.042313.00197
17. Schilit BN, Hilbert DM, Trevor J. Context aware communication. IEEE Wirel Commun. 2002;9(5):46-54. https://www.researchgate.net/publication/3435880_Context aware_communication
18. Chen, Guanling and Kotz, David, A Survey of Context aware Mobile Computing Research. Computer Science Technical Report TR2000-381. 2000. https://digitalcommons.dartmouth.edu/cs_tr/177
19. Li X, Eckert M, Martinez JF, Rubio G. Context Aware Middleware Architectures: Survey and Challenges. Sensors (Basel). 2015 Aug 20;15(8):20570-607. doi: 10.3390/s150820570. PMID: 26307988; PMCID: PMC4570438.
20. Alegre U, Augusto JC, Clark T. Engineering context aware systems and applications: A survey. J Syst Softw. 2016;117:55-83. https://doi.org/10.1016/j.jss.2016.02.010
21. Bhaskar M, Das MM, Chithralekha T, Sivasatya S. Genetic algorithm-based adaptive learning scheme generation for context aware e-learning. Int J Comput Sci Eng. 2010;2(4):1271-9. https://www.researchgate.net/publication/49620441_Genetic_Algorithm_Based_Adaptive_Learning_Scheme_Generation_For_Context_Aware_E-Learning
22. Linda S, Bharadwaj KK. A genetic algorithm approach to context aware recommendations based on spatio-temporal aspects. Integrated Intelligent Computing, Communication and Security. Singapore: Springer; 2018: 59-70. https://doi.org/10.1007/978-981-10-8797-4_7
23. Zhang Z, Zheng S, Li W, Tan Y, Wu Z, Tan W. Genetic algorithm for context aware service composition based on context space model. 20th International Conference on Web Services. IEEE; 2013: 605-6. doi:10.1109/ICWS.2013.94
24. Linda S, Minz S, Bharadwaj KK. Effective context aware recommendations based on context weighting using genetic algorithm and alleviating data sparsity. Appl Artif Intell. 2020;34(10):730-53. https://doi.org/10.1080/08839514.2020.1775011
25. Reddy KHK, Luhach AK, Pradhan B, Dash JK, Roy DS. A genetic algorithm for energy-efficient fog layer resource management in context aware smart cities. Sustain Cities Soc. 2020;63:102428. doi: 10.1016/j.scs.2020.102428
26. Yuan Y, Zhang X, Sun W, Cao Z, Wang H. Optimal web service composition based on context awareness and genetic algorithm. International Conference on Information Science and Cloud Computing Companion. IEEE. 2013: 660-7. doi: 10.1109/ISCC-C.2013.98.
27. Abdolmaleki A, Rezvani MH. An optimal context aware content-based movie recommender system using genetic algorithm: a case study on MovieLens dataset. J Exp Theor Artif Intell. 2024;36(8):1485-511. doi:10.1080/0952813X.2022.2153279
28. Pawar P, Mei H, van Beijnum BJ, van Halteren A. Context aware task assignment in ubiquitous computing environment—a genetic algorithm-based approach. IEEE Congress on Evolutionary

Computation. IEEE; 2007: 2695-702.
doi: 10.1109/CEC.2007.4424811.
29. Luna JM, Pechenizkiy M, Del Jesus MJ, Ventura S. Mining context aware association rules using grammar-based genetic programming. IEEE Trans Cybern. 2017;48(11):3030-44.
doi: 10.1109/TCYB.2017.2750919
30. Nam MY, Rhee PK. Adaptive object recognition using context aware genetic algorithm under dynamic environment. International Conference on Pattern Recognition and Image Analysis. Berlin: Springer; 2005: 257-67.
https://doi.org/10.1007/11552499_29
31. Kattan A, Ong YS. Surrogate genetic programming: A semantic aware evolutionary search. Inf Sci. 2015;296:345-59.
doi:10.1016/j.ins.2014.10.053
32. Vanneschi L, Castelli M, Silva S. A survey of semantic methods in genetic programming. Genet Program Evolv Mach. 2014;15:195-214.
doi: 10.1007/s10710-013-9210-0
33. Graff M, Tellez ES, Escalante HJ, Miranda-Jiménez S. Semantic genetic programming for sentiment analysis. Results of the Numerical and Evolutionary Optimization Workshop NEO 2015. Springer;2017:43-65.
https://doi.org/10.1007/978-3-319-44003-3_2
34. Sarwat M, Bao J, Chow CY, Levandoski JJ, Magdy A, Mokbel MF. Context awareness in Mobile Systems. Data Management in Pervasive Systems. Springer Nature Link. 2015.
ISSN: 2197-9723.
doi: 10.1007/978-3-319-20062-0_13

How to Cite: Kumarakrishnan S, Venkatesan VP, Geetha S, Madusudanan J. Semantic Tag Guided Genetic Programming for Context Aware Optimization. Int Res J Multidiscip Scope. 2026; 7(2): 943-957.
DOI: 10.47857/irjms.2026.v07i02.07999