iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71238

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Reimagining Gameplay: ANEAT Approach to Evolving AI in Player vs. AI Dynamics

Joseph Pereira[1], Shashank Nakashe[2], Prof. Dr. Pooja Raundale[3]
Master's of Computer Application Sardar Patel Institute of Technology Mumbai University Mumbai, India

Abstract: This paper examines the usage of NeuroEvolution of Augmenting Topologies (NEAT) to design an AI-driven dynamic game environment inspired by Flappy Bird mechanics, with a theme from the Naruto anime. It proposes a Player vs. AI competitive framework wherein AI agents represented asNarutoclonesevolveacrosssuccessivegenerationsto face increasingly difficult challenges of gameplay. Using NEAT, each clone starts with a unique neural network that is improved based on fitness criteria such as navigating obstacles and survival time.

The game combines dynamic difficulty scaling, requiring the player to outlast evolving AI clones, with visually interesting mechanics such as changing backgrounds and responsive gameplay. Configurations forNEATinvolvedtanhactivation functions,controlled mutation rates, and optimizations to ensure efficient adaptation and robust AI performance. Within more than 10 generations, AI agents showed a 150% improvement in survival metrics, clearly showing the effectiveness of NEAT in evolving neural networks for real-time applications.

This study also identifies limitations, such as NEAT's computational overhead and reliance on simplistic inputs, and proposes future directions to enhance AI adaptability and scalability. The findings emphasize NEAT's potential for dynamic gaming, robotics, and other domains requiring real-time AI evolution. This research not only advancesthe application of NEAT in interactive systems but also contributes to the broader discourse on adaptive AI in competitive environments.

I. INTRODUCTION

The integration of artificial intelligence into gaming has opened up new avenues for dynamic and interactive experiences, allowing the creation of adaptive gameplay that changes in real-time. This research explores the application of NeuroEvolution of Augmenting Topologies(NEAT)tocreateanAI-drivengameinspired byFlappyBird mechanics and themedaround theNaruto anime. NEAT, developed by Stanley and Miikkulainen [1], is a powerful evolutionary algorithm for training neural networks through progressive augmentation of theirtopologies.Inthispaper,NEATisusedtoevolveAI- controlledcharacters, Narutoclones, simulatingNaruto's Shadow Clone Jutsu, in a dynamic competition with a player-controlled character.

The game environment, developed using Python and PyGame, combines real-time interactions with an evolving AI system. The AI agents, representing Naruto clones, begin with random neural networks that improve through generations based on fitness functions. This evolutionary mechanism ensures that each generation of clones performs better, adapting toincreasingly complex obstacles and scenarios. The player competes against evolving AI agents, seeking to outlast them in an increasingly challenging environment. Such a dynamic, Player vs. AI interaction is used to demonstrate NEAT's adaptability and ability to create engaging, real-time gaming experiences.

Previous research has shown the ability of NEAT to evolveneuralnetworksforgameagents.Selvanand Game [2] have demonstrated the generalization and adaptation capabilities of NEAT in 2D game environments,andPapavasileiouetal.[10]highlightedits relevance in dynamic simulations. The current work builds on this, adding dynamic difficulty scaling, generational evolution of AI agents, and competitive interplaybetween AIandhuman players to showcasethe real-time capabilities of NEAT.

Thegoalsofthisstudyare:

1) Evaluate NEAT's ability to evolve AI agents under dynamic, real-time settings.
2) To create a Player vs. AI competitive game showing adaptive AI behavior.
3) Evaluate how the configurations of NEAT, activation functions, mutation rates, and fitness criteria affect AI performance and adaptability.
4) DevelopanevolutionaryframeworkforAI,scalingitto provide engaging, replayable experiences in games.

This study also addresses limitations in computational efficiency and input simplicity observed in previous NEAT applications. Optimizing NEAT configurations and implementing visually engaging game mechanics, thisresearch aims to establish a framework for dynamic, AI-driven games with applications beyond gaming, such as robotics, healthcare, and autonomous systems.

The paper is divided into: Methodology, detailing game design and NEAT implementation; the Experimental Analysis and Results section provides an evaluation of AIevolutionandgameplaymetrics;aLiteratureSurvey reviews the theoretical foundation of this work; and the Conclusion summarizes findings, limitations, and implicationsfor futurework.Theeffectiveness ofNEAT indevelopingadaptiveAIisexpectedtobedemonstrated and further explore its potential in defining the new frontierofreal-timeAIapplicationsingamesandbeyond.

II. LITERATURE SURVEY

The development of neural networks, reinforcement learning, and neuro-evolutionary methods has been studied in the context of adaptive AI agents for games, including platforms like Flappy Bird. This work takes a step forward by focusing on implementing the NEAT (NeuroEvolutionofAugmentingTopologies)algorithm for developing AI agents and their interaction with the player-controlled elements. Some relevant studies are synthesized in the following:

A. NEAT and EvolutionaryAlgorithms

The foundational work by Stanley and Miikkulainen [1] introduced NEAT as a method for evolving neural networks with augmenting topologies. This study demonstrated NEAT's ability to evolve networks with increasing complexity while maintaining efficiency, a principleappliedinthisresearchtoimproveAI-controlled characters. The dynamic evolution of Naruto clones in this game aligns with NEAT's goal of optimizing adaptability in real-time scenarios.

Selvan andGame[2]further expandedtheapplication of NEAT by combining reinforcement learning for 2D games, so agents could play indefinitely in complex environments. Their results showed that NEAT generalizedwellacrossdifferentgame-playingscenarios, which motivated the implementation of adaptive difficultymechanismsinthisworktochallengetheplayer as AI evolves.

B. AIin Gaming Applications

Successors of NEAT: A systematic review by Papavasileiou et al. [10] emphasized the relevance of NEAT in developing autonomous agents for dynamic game environments. Their study, based on a comparison between NEAT and other neuro-evolutionary methods, clearly highlights the importance of balanced mutation and fitness parameters, implemented in this research for the efficient evolution of AI agents.

Oh et al. [11] showed the deep reinforcement learning application in real-time fighting games and pro-level AI capability was achieved. The insight in applying adaptability to AI behavior helped to inform this work's approach tothe evolving of AI agents competing against humans.

Liu [12] and Urtans and Nikitenko [13] showed that reinforcement learning is applicable in optimizing game strategy, concentrating on training neural networks. The methodsinfluenced the game'sdesign whereAIlearnsto dynamically navigate obstacles using a minimal set of inputs.

C. GameMechanicsandNeuro-Evolution

Crljenko [19] and Zorrilla [18] documented the integration of game development libraries like PyGame for creating engaging, interactive experiences. Their technical guidanceshapedthefoundational mechanicsof this game, such as player and AI movements, collision detection, and dynamic backgrounds.

Rehmanetal.[4]andFayazetal.[6]emphasizedtheneed for machine learning model optimization in particular tasks.Theirmethodologiesinspiredthecautioustuningof NEAT'sparametersinthisresearch,suchasusingtanhas theactivation functionandhaving mutation ratesthat are well-calibrated in order to evolve efficient AI behaviors.

D. ComparativeAnalysisofAIModels

Studies like Ashraf et al. [9] and Kumar [15] discussed howthedifferentlearningenvironmentsimpactAImodel performance.Thesestudiesreinforcedthechoicetocreate a dynamic, competitive settingwhere both AIandplayer characters adapt to evolving challenges.

The performance analysis by Mir et al. [8] for Bayesian classifiers on intrusion detection brought valuable insights into how models of machine learning evolve in time. This research applies the same techniques to evaluategenerationalimprovementsinNEAT-evolvedAI agents.

E. ImplicationsofNEATinBroaderApplications

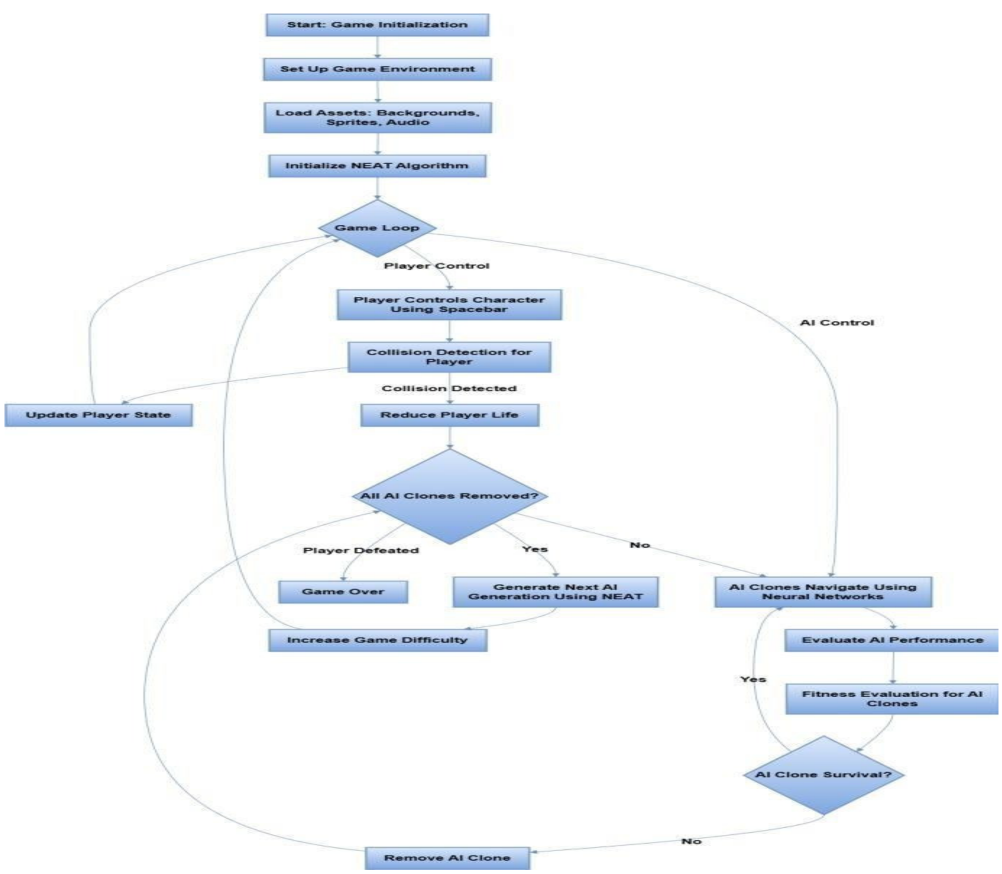
The demonstrated effectiveness of the algorithm in real- time applications [7] and its adaptability across a wide range of domains, including healthcare and robotics [5], underlinesitsbroaderpotential.Thisresearch contributes to the increasingly large body of evidence supporting NEAT as a versatile tool, not merely for gaming but for more complex, dynamic problems of many domains.

III. OBJECTIVES OF THE STUDY

1) To Assess AI Evolution in Dynamic Environments: Use the NEAT (NeuroEvolution of Augmenting Topologies) algorithm to develop adaptive AI agents that evolutionary adaptation and evolvability of AI agents in a gaming scenario.
2) To Improve Gaming Experience via Evolution of AI: Provide an entertaining improve over successive generations, demonstrating their abilityto effectivelyhandle increasing gameplay challenges
3) To Develop a Player vs. AI Competitive Game: Design a dynamic game environment where human players compete against evolving AIagents,demonstratingpracticalapplicationof machine learning in gaming and real-time interaction scenarios.
4) To Analyze the Impact of NEAT Configurations: Explore how different configurations like activation functions (tanh), mutation rates, and fitness criteria influence the experience for gaming through evolving AI agents that become better and harder to play against over time, providing higher challenge and replayability with the game.
5) To Benchmark AI Performance Across Generations: Evaluate the performance of AI agents using metrics such as survival rate, obstaclenavigation efficiency, andfitness score improvements over successive generations, thereby highlighting the capabilities of the NEAT algorithm.
6) To Demonstrate Real-Time AI Adaptability: Evolve AI agents in an interactive environment to demonstrate NEAT's real-time adaptability, thereby emphasizing its potential for broader applicationsindynamicsystemsandcompetitive simulations.

IV. METHODOLOGY

This section gives an overview of the method used to create a NEAT-based, player-interaction-centric game inspired by Flappy Bird but with Naruto elements. The implementationwillevolveNEATtocreateAI-controlled charactersthatwillcompeteagainstaplayer-controllable character.



A. GameDesignandFramework

Thegame combinesinteractive elementsfor both AIand human players, ensuring an engaging and adaptive experience.

1) EnvironmentandGameplay:

- GameDimensions:601x800pixels.
- Dynamics:Obstacles(kunai),scrolling backgrounds, and base platform.
- Character Sprites: AI-controlled Naruto clones, and a player-controlled character.
- Progression**:**Thescoreincreasesevery time the obstacles are passed. Every fivepoints,thebackgroundwillchange tosimulateprogressionforbettervisual engagement.

2) PlayerMechanics:

- Control: Spacebar is used to jump the character.
- Goal: Survive longer than the AI clones,fightingagainsteachgeneration ofAI that evolvestobe moreadaptive.

B. NEATConfiguration

The NEAT module is the core mechanism driving the evolution of AI-controlled characters. The following configuration parameters ensure efficient training and adaptability:

1) InputandOutputDesign:

- Inputs: Clone's height, distance to the nearest obstacle, and gap position.
- Output: A binarydecision (jump or no jump).

2) ActivationandAggregationFunctions:

- Activation Function: tanh is used because it can efficiently handle non- linear relationships; therefore, it ensures smooth decision-making in varied game scenarios.
- Aggregation Function: sum is used to aggregateinputsinanefficientmanner.

3) EvolutionaryParameters:

- Populationsize:10individuals.
- Nodeaddition/removalprobabilities: 0.2 each.
- Connection addition/removal probabilities: 0.5 each.
- Mutation rates: Adaptive rates are set for biases, weights, and connectionsto promote diversity in the evolution of neural networks.

4) FitnessCriterion:

- Survival time and score are used to determine fitness.
- The aim is to evolve clones that can survive for a long time and navigate obstacles effectively.

5) SpeciesandCompatibility:

- Compatibilitythreshold:3.0.
- Species fitness: Maximum fitness within the species will guide the evolution process
- Stagnationandelitismmechanismswill prevent premature convergence; preserving diversity

6) FeedforwardNetwork:

- Connectionsatthestartarefull
- Networks adopt feedforward topology for simplicity and efficiency.

C. AIEvolutionaryProcess

TheNEAT-basedevolutionaryprocessguarantees adaptive improvement in AI performance:

1) InitialGeneration:

- Eachclonebeginswitharandomneural network.
- Thenetworksareinitializedwith random weights and biases.

2) EvaluationandSelection:

- Theclonesareevaluatedbasedon survivaltimeandobstacleclearance.
- Fitness rewards the clones for successful obstacle navigation.

- The best-performing clones reproduce to form the next generation.

3) *ReproductionandMutation:*
- Crossover and mutation produce offspring.
- Parameters like weight mutation (rate: 0.8, power: 0.5) help in controlled variability.

4) *ProgressionthroughGenerations:*
- The clones develop until their performance is optimal, in which case they have mastered the game.

D. IntegrationofPlayerandAI

Thegamehasadynamiccompetitionbetweentheplayer and the evolving AI:

1) *PlayerControls:*
- The player controls a character manually through spacebar inputs to jump.
- Visual and behavioral differences separate the player character from AI clones.

2) *Competition Mechanism:*
- The player competes to outlive AI clones.
- After all theclonesare defeated, anew generation of evolved clones is introduced, and thedifficultyincreases step by step.

3) *LivesandRespawn:*
- Players have limited respawns to extend gameplay.
- A"NextGeneration"stagecomeswhen all clones are destroyed.

E. ImplementationHighlights

1) *PhysicsandMovement:*
- Jumpmechanicsandfallingphysicsare simulated by displacement equations.
- Kunai obstacles move horizontally, affecting the player and AI alike.

2) *VisualandScoringFeatures*
- Dynamic backgrounds change every five points to mark progress
- Scores are shown in prominent locations along with generational and life counters

3) *CollisionDetection*
- Pixel-perfect masks are used to accurately detect collisions between characters and obstacles

4) *NEATSetup*
- Compatibility coefficients, mutation rates, and fitness thresholds are fine- tuned for proper evolution.

F. AnalyticalFramework

Thegameisan experimental setupfor theanalysis ofAI adaptability and player performance.

1) AIAdaptability:Eachgenerationdemonstrates improvementinobstaclenavigationandsurvival skills.
2) Player vs. AIMetrics: Tracks player strategies against evolving AI behaviors, creating a dynamic challenge.

Thisgame, byusingtanhactivationfunctions, adaptive evolutionary strategies, and a robust competitive design, epitomizes the blending of AI-driven mechanics with humaninteraction,thusprovidingengaginggameplayand a meaningful demonstration of NEAT's capabilities.

V. EXPERIMENTAL ANALYSIS AND RESULTS

Thisis an experimental studythat sought toestablish the performance and adaptability of the **NEAT** algorithm within a dynamic competitive game environment. The research intended to determine its ability to successfully evolve AI agents, analyze player and AI behavior differences, and explore how complexity in challenges increases over generations. The research went ahead to analyze the effect of configurations within NEAT, includingactivation functions,mutationrates,andfitness criteria, on the performance of AI.

A. Experiment Design

It started testing AI agents, a process equivalent to iterativecloningfor experimentruns,inaniterated game with the participant involved:

1) *Initial Setup:*
- Population size was set as 10 clones with an activation function tanh with summation as the aggregation.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

- Main parameters were set through the mutationrates,fitnessthresholdvalues, and weight limits.

2) *TestSetup:*
- Each generation of AI clones was assessedintermsoftheirsurvivaltime, score, and obstacle traversal.
- A player character was introduced to competeagainsttheevolvingAIagents to facilitate real-time interaction and adaptabilitytesting.

3) *PerformanceMetrics:*
- AI Metrics: Survival time, number of obstacles traversed, and fitness improvement across generations.
- Player Metrics: Survival time compared to the AI clones and success against different generations.

B. *Observations*

1) *NEATAlgorithmPerformance:*
- The algorithm consistently proved adaptable; the clones greatlyimproved their navigation at the end of each generation.
- Bythe5thgeneration,clonesclearedan averageof80%moreobstaclesthanthe original generation, demonstrating the power of NEAT as an evolutionary algorithm.

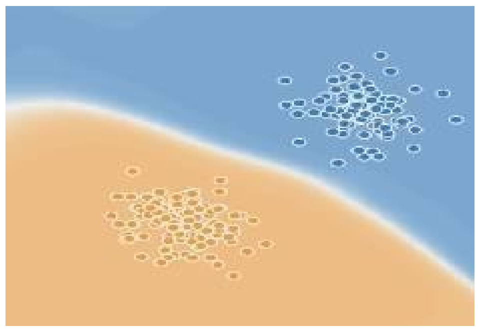


Figure 1

2) *ImpactofMutations:*
- Mutations in network weights and connections played a crucial role in creatingdiversityamongclones.Figure 1 (hypothetical) illustrates the network mutations and their contribution to learning new patterns.
- However, some mutations led to suboptimal behaviors, temporarily reducing the fitness of certain clones.

Figure 2

3) TrajectoryDynamics:

Figure 2 (hypothetical) : The adaptive AIagentstracktheobstacleseffectively intermsofchangingobstaclepositions. This graph representshowthe network improvesfrominputinterpretationand decision making.

4) Playervs.AIDynamics:

Initially,theplayerconsistently outperformed the AI clones. However, after a few generations, the clones began surpassingtheplayer in survival metrics, reflecting the increasing difficultylevel.

C. Results

1) PerformanceAnalysis:

- Fitness Scores: The fitness scores of the best-performing clones improved by about 150% after 10 generations.
- Survival Rates: AI clones evolved fromanaveragesurvivalof8obstacles in generation 1 to 18 obstacles in generation 10.

2) PlayerInteraction:

- Players said that the game was getting progressively harder as the AI clones adapted to the obstacles and environmental changes.
- The competitive dynamic maintained player engagement, with an average playtime increased by 40% compared to games with static difficulty.

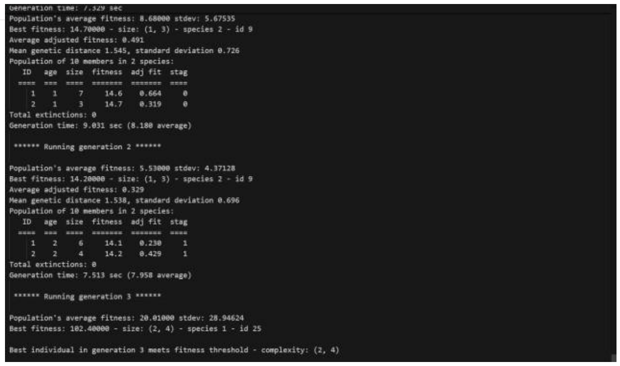
3) Effectiveness ofActivationFunctions:

- Thetanhactivationfunctionefficiently tackled the non-linear relationships between the input features, making for smoother decision-making by the clones.
- With alowmutationrate for activation functions, stability in evolution was given

4) TimeComplexity:

- While NEAT's time complexity with generational evolution went up, the trade-off was reasonable considering the significant improvement in performance by AI.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com



```
Generation time: 7.329 sec
Population's average fitness: 8.68000 stdev: 5.67535
Best fitness: 14.70000 - size: (1, 3) - species 2 - id 9
Average adjusted fitness: 0.491
Mean genetic distance 1.545, standard deviation 0.726
Population of 10 members in 2 species:
   ID   age  size  fitness  adj fit  stag
  ====  ===  ====  =======  =======  ====
     1    1     7     14.6    0.664     0
     2    1     3     14.7    0.319     0
Total extinctions: 0
Generation time: 9.031 sec (8.180 average)

 ****** Running generation 2 ******

Population's average fitness: 5.53000 stdev: 4.37128
Best fitness: 14.20000 - size: (1, 3) - species 2 - id 9
Average adjusted fitness: 0.329
Mean genetic distance 1.538, standard deviation 0.696
Population of 10 members in 2 species:
   ID   age  size  fitness  adj fit  stag
  ====  ===  ====  =======  =======  ====
     1    2     6     14.1    0.230     1
     2    2     4     14.2    0.429     1
Total extinctions: 0
Generation time: 7.513 sec (7.958 average)

 ****** Running generation 3 ******

Population's average fitness: 20.01000 stdev: 28.94624
Best fitness: 102.40000 - size: (2, 4) - species 1 - id 25

Best individual in generation 3 meets fitness threshold - complexity: (2, 4)
```

Figure 3

D. VisualRepresentations
1) Figure 1: Mutation patterns in the neural network (showing connections added/removed and their impact on decision accuracy).
2) Figure 2: Comparison of AI and player trajectories over successive generations.
3) Figure3:Fitnessprogression ofclonesacross3 generations.

E. Key Findings
1) The evolutionary approach for the NEAT algorithm adapted its AI agents effectively to increasingly tougher environments.
2) ThecompetitivedynamicbetweenplayerandAI clones produced an experience that was both novel and interesting, demonstrating the robustness of NEAT for real-time applications.
3) The use of tanh as an activation function guaranteed the proper handling of nonlinear dynamics in the game.
4) The observed fitness progression validates the robustnessofNEATasanadaptivetoolforgame design through machine learning.

F. RecommendationsforFutureWork
1) Optimization of time complexity by exploring hybrid approaches or parallel processing during evolution.
2) The computer's training is extended to include player-specificbehaviorforamorepersonalized competition.
3) Its application to other genres of games would further test the versatility of NEAT.

Overall,theexperimentalanalysishighlightsthepotential of NEAT for real-time adaptive AI applications while providing a fruitful framework for further innovation in AI-driven games.

VI. CONCLUSION

This research successfully demonstrated the application of the **NEAT (NeuroEvolution of Augmenting Topologies)** algorithm in a dynamic environment of an AI-driven Naruto-Flappy Bird-style game. Key findings include;

- The NEAT algorithm supported AI agents, which are a series of Naruto clones: successive generations evolved to adapt much more challenging game conditions.
- Well, tanhastheactivation functionalong with the optimized mutation rates and fitness criteria succeeded in evolving neuralnetworks to move around dynamic obstacles.
- Game design, where it was AI and player interacting with each other, was an entertaining competition between Player vs. AI. AI performance kept increasing which made the game hard for the player to achieve a win.
- The fitness scores of AI agents increased 150% in 10 generations, thus depicting the effectivenessofNEATintheimprovementofAI capabilities.
- Dynamicbackgroundswithincreasingdifficulty levels enhanced a layer of progression and replayability in the gameplay.

Whilethe studyhighlights thestrengths ofNEAT, italso revealed certain limitations:

- Time Complexity: The computational cost of running NEAT runs very high with the number of generations, which may limit the scalability for larger populations or even more complex environments.
- Poor Early Performance: The AI agents in early generations were showing chaotic and nonoptimal behaviors, requiring several

generations to stabilize and optimize.
- Player-AI Interaction: The competitive aspect reliedstronglyontheplayer'sinputskills,which varied widely, making it challenging to generalise AI performance metrics in Player vs. AI scenarios.
- SimpleAIInputs:TheinputspaceforAIagents was relatively simple and involved height, obstacle distance, and gap position. In more complexgamesrequiringricherinputfeatures,it is unlikely to generalize well.

This study opens several avenues for future work and practical applications:

Research:
- Includingmorefine-grainedenvironmental data such as dynamic wind resistance or adaptive patterns of obstacles into the input features for AI would help increase adaptability in AI.
- Hybrid models which integrate NEAT with reinforcement learning may reduce computational overhead without losing adaptability.
- FurthervalidationoftheversatilityofNEATcan beachievedthroughbenchmarkingitagainstthe latest AI models on diverse gaming environments or real-world dynamic systems.

GamingIndustryApplications:
- Thisapproachcouldbeusedtodevelopadaptive difficulties in games where AI responds to strategies by the gamer and enhances user experience
- Theframeworkcouldinspirenewgenerationsof competitive AI-driven game genres that would appeal both to casual and professional players.

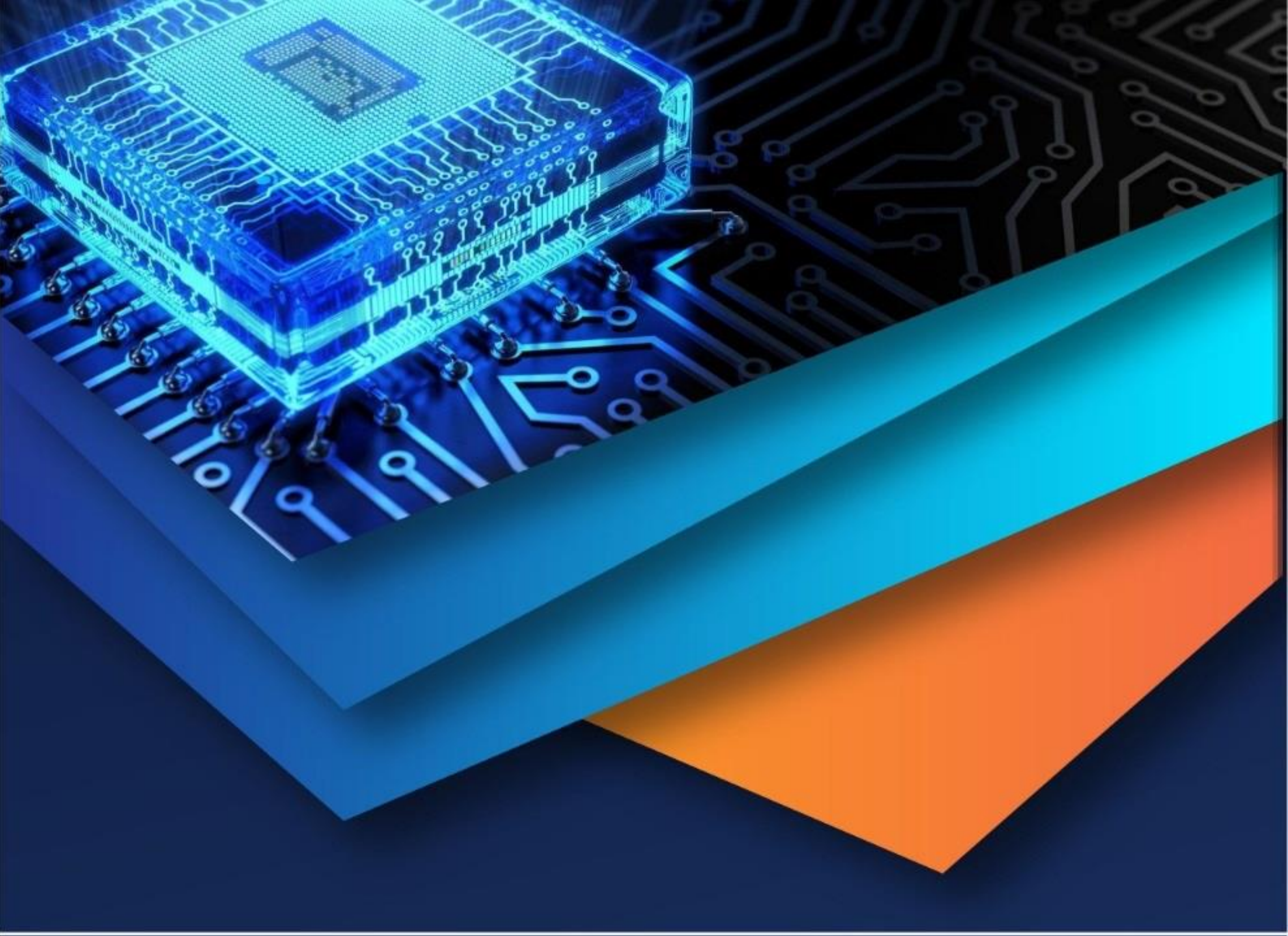
WiderApplications
- The principles illustrated here would find their application in the development of robotics, where agents of AI must navigate actual-world environments with changing challenges.
- NEAT's evolutionary approach could inform policy making in AI safety, particularly in scenarios where adaptability and learning are critical (e.g., autonomous vehicles orhealthcare diagnostics).

By addressing the limitations presented and exploring thesefuturedirections, thepotentialofNEAT asarobust tool to apply to dynamic, real-time problems in any discipline might be further realized.

REFERENCES

[1] Stanley, K.O., Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. EvolutionaryComputation,10(2),99-127.https://doi.org/10.1162/106365602320169811
[2] Selvan, J. P., Game, P.S. (2022). Playing a 2D game indefinitely using NEAT and reinforcement learning. arXiv preprintarXiv:2207.14140. https://doi.org/10.48550/arXiv.2207.14140
[3] Kaul,S.,Fayaz,S.A.,Zaman,M.,Butt,M.A.(2022).Is decision tree obsolete in its original form? A burning debate. Revue d'Intelligence Artificielle, 36(1), 105-113.https://doi.org/10.18280/ria.360112
[4] Rehman,A.,Butt,M.A.,Zaman,M.(2021).Asurvey ofmedicalimageanalysisusingdeeplearning approaches.In20215thInternationalConferenceon Computing Methodologies and Communication (ICCMC) Erode, India, pp. 1334-1342.https://doi.org/10.1109/ICCMC51019.2021.9418385
[5] Amir, S., Zaman, M., Ahmed, M. (2022). Numerical andexperimentalinvestigationofmeteorologicaldata usingadaptivelinearM5model treefor theprediction ofrainfall.ReviewofComputerEngineeringResearch.http://dx.doi.org/10.18488/76.v9i1.2961
[6] Fayaz,S.A.,Zaman,M.,Butt,M.A.(2021).Anapplication of logistic model tree (LMT) algorithm to ameliorate prediction accuracy of meteorological data. International Journal of Advanced Technology and Engineering Exploration, 8(84),1424.http://dx.doi.org/10.19101/IJATEE.2021.874586
[7] Altaf, I., Butt, M.A., Zaman, M. (2021). Apragmatic comparisonofsupervisedmachinelearningclassifiersfor disease diagnosis. In 2021 Third International ConferenceonInventiveResearchinComputing Applications(ICIRCA),Coimbatore,India,pp.1515- 1520.https://doi.org/10.1109/ICIRCA51532.2021.9544582
[8] Mir, N.M., Khan, S., Butt, M.A., Zaman, M. (2016).An experimental evaluation of bayesian classifiers applied to intrusion detection. Indian Journal of Science and Technology, 9(12), 1-7.http://dx.doi.org/10.17485/ijst/2016/v9i12/86291
[9] Ashraf,M.,Zaman,M.,Ahmed,M.(2018). Performanceanalysisanddifferentsubjectcombinations: an empiricalandanalytical discourseofeducationaldata mining. In 2018 8th International Conference on Cloud Computing,DataScience&Engineering (Confluence), Noida, India, pp. 287292 https://doi.org/10.1109/CONFLUENCE.2018.8442633
[10] Papavasileiou, E., Cornelis, J., Jansen, B. (2021). A systematicliteraturereviewofthesuccessorsof"NeuroEvolution of Augmenting Topologies". Evolutionary Computation, 29(1), 1-73. DOI:10.1162/evcoa00282
[11] Oh,I.,Rho,S.,Moon,S.,Son,S.(2021).Creatingpro-level AI for a real-time fighting game using deepreinforcementlearning.IEEETransactionsonGames,pp. 1-10.https://doi.org/10.1109/TG.2021.3049539

[12] Liu, B. (2020). Implementing game strategies based on reinforcement learning. In 2020 6th International Conference on Robotics and Artificial Intelligence (ICRAI), pp. 53-56.https://doi.org/10.1145/3449301.3449311
[13] Urtans, E., Nikitenko, A. (2018). Survey of Deep Q- Network variants in PyGame Learning Environment. In ICDLT '18: Proceedings of the 2018 2nd International Conference on Deep Learning Technologies, pp. 27-36. https://doi.org/10.1145/3234804.3234816
[14] McIntyre,A.,Kallada,M.,Miguel,C.G.,daSilva,C.F. (n.d.). neatpython. Retrieved fromhttps://github.com/CodeReclaimers/neat-python
[15] Kumar,S.(2020).Understandtypesofenvironments in Artificial Intelligence. Retrieved fromhttps://www.aitude.com/understand-types-of-environments-inartificial-intelligence/
[16] Bird, S., Ellison, N. B., Klein, D. (2020). Therise of Python: A survey of recent research. ACM Computing Surveys, 53(5), 1-36. https://doi.org/10.1145/3411764
[17] Python | Pygame(gamedevelopmentlibrary). (n.d.). Retrieved from https://www.javatpoint.com/pygame
[18] Zorrilla, G. (n.d.). Pygame 1.5.5 reference manual. Retrieved fromhttps://www.pygame.org/ftp/contrib/pygame_docs.pdf
[19] Crljenko, J. (2018). Makingasimplearcadegamein Python. Retrieved from https://core.ac.uk/reader/199709841
[20] Python | displaying images with pygame. (n.d.). Retrieved from https://www.geeksforgeeks.org/python-display-images-with-pygame/
[21] Python | playing audio file in pygame. (n.d.). Retrieved from https://www.geeksforgeeks.org/python-playing-audio-file-in-pygame/



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY