

|  |
| --- |
| **RAJALAKSHMI INSTITUTE OF TECHNOLOGY** |
| (An Autonomous Institution, Affiliated to Anna University, Chennai) |

**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**

**ACADEMIC YEAR 2025 - 2026**

**SEMESTER III**

**ARTIFICIAL INTELLIGENCE LABORATORY**

**MINI PROJECT REPORT**

|  |  |
| --- | --- |
| **REGISTER NUMBER** | 2117240030024 |
| **NAME** | DEVAMALYAA G R |
| **PROJECT TITLE** | Password Pattern Guessing Using AI Random Search Algorithm |
| **DATE OF SUBMISSION** | 29/10/2025 |
| **FACULTY IN-CHARGE** | **Mrs. M. Divya** |

**Signature of Faculty In-charge**

**INTRODUCTION**

This project, *“Password Pattern Guessing Using AI Random Search Algorithm,”* applies the concept of AI-based search algorithms to generate and predict password patterns intelligently.  
In today’s world, passwords are essential for protecting digital information. Weak or predictable passwords can easily be guessed or hacked. Therefore, studying and simulating how an AI can attempt to guess or generate passwords is important for understanding password security and improving protection systems.  
The project uses the Random Search Algorithm, an AI search technique that explores different password combinations randomly until it finds a suitable or matching pattern. The algorithm demonstrates how AI can be applied to pattern generation, prediction, and optimization tasks.

**PROBLEM STATEMENT**

In today’s digital world, most users rely on passwords for securing online accounts and sensitive data. However, weak or predictable passwords increase the risk of unauthorized access. Traditional brute-force or dictionary attacks are time-consuming and inefficient when dealing with complex password patterns.  
This project aims to develop an AI-based password pattern guessing system using a Random Search Algorithm that intelligently predicts possible password combinations by exploring multiple search possibilities and optimizing the guessing process.

**GOAL**

**Expected Result**

The system is expected to intelligently guess or generate password patterns using the AI-based Random Search Algorithm. It should show how efficiently the algorithm can find the correct password by exploring different combinations, displaying the number of attempts and time taken. This demonstrates the practical use of AI search techniques in solving real-world problems.

**Possibilities**

This project can be further extended to analyze password strength, test system vulnerabilities, and improve cybersecurity awareness. The same AI search concept can also be applied to other areas such as pattern recognition, optimization problems, and AI-based puzzle solving.

**THEORETICAL BACKGROUND**

**Theoretical Background**

The project *“Password Pattern Guessing Using AI Random Search Algorithm”* is based on the concept of Artificial Intelligence search techniques used to solve optimization and prediction problems. In this project, the focus is on simulating how an AI system can guess or generate password patterns intelligently. The Random Search Algorithm is a simple yet powerful AI technique that randomly explores possible solutions within a defined search space until an acceptable or optimal solution is found. It does not require prior knowledge about the structure of the problem, making it suitable for situations where the solution space is large or unpredictable.

**Literature Survey**

Several algorithms are used in Artificial Intelligence for searching and optimization, such as Brute Force Search, Hill Climbing, Genetic Algorithm, Simulated Annealing, and A\* Search.

* The Brute Force approach checks all possible combinations but is highly time-consuming.
* Hill Climbing improves solutions step by step but may get stuck in local optima.
* Genetic Algorithms use evolution-based selection and mutation, which are efficient but computationally expensive.
* A\* is mainly used for pathfinding and is not suitable for random pattern problems.

In comparison, the Random Search Algorithm is simpler and can efficiently handle situations where no mathematical or heuristic information is available about the solution pattern, such as password guessing.

**Justification for Choosing the Algorithm**

The Random Search Algorithm is chosen because it is easy to implement, flexible, and effective for exploring unknown or large search spaces like password combinations. It clearly demonstrates how AI can approach guessing problems through random exploration and selection. This algorithm also helps in understanding the concept of randomness in optimization, aligning well with the learning objectives of introductory Artificial Intelligence studies.

**ALGORITHM EXPLANATION WITH EXAMPLE**

The Random Search Algorithm is one of the simplest Artificial Intelligence search techniques used to find an optimal or correct solution by generating random guesses within a defined search space. It does not follow any fixed pattern or heuristic but depends on repeated trials until a satisfactory solution is obtained. The algorithm continues generating random values, compares them with the target, and stops when the correct solution is found.

**Steps of the Algorithm**

1. Start the program.
2. Initialize the target password or pattern to be guessed.
3. Generate a random password guess using random characters or digits.
4. Compare the generated password with the target password.
5. If the guess matches the target, stop and display success.
6. If not, repeat the random generation process until the correct match is found or the attempt limit is reached.
7. Display the total number of attempts and time taken.
8. End the program.

**Example**

Let the target password be “A12.”

* Step 1: The algorithm randomly generates a guess, e.g., “P3B” → Not a match.
* Step 2: Next random guess → “A92” → Not a match.
* Step 3: Next random guess → “A12” → Match found.

The algorithm then stops and displays:

Password matched: A12

Attempts taken: 3

Time elapsed: 0.5 seconds

This example shows how the Random Search Algorithm keeps exploring random combinations until it discovers the correct password pattern. It demonstrates the concept of AI-based problem-solving through random exploration, and how such methods can be applied to security-related simulations.

**IMPLEMENTATION AND CODE**

**CODE:**

import random

import string

import time

# Target password (keep it short for quick output)

target = "AI"

characters = string.ascii\_uppercase + string.digits

start\_time = time.time()

attempts = 0

guess = ""

# Reduced limit for quick run

max\_attempts = 500

while guess != target and attempts < max\_attempts:

guess = ''.join(random.choice(characters) for \_ in range(len(target)))

attempts += 1

print("Trying:", guess)

end\_time = time.time()

elapsed = round(end\_time - start\_time, 2)

print("\n--- RESULT ---")

if guess == target:

print("Password Matched Successfully!")

else:

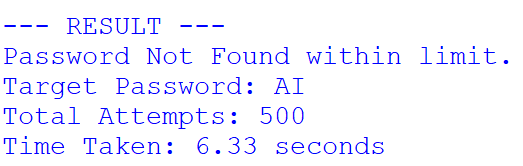
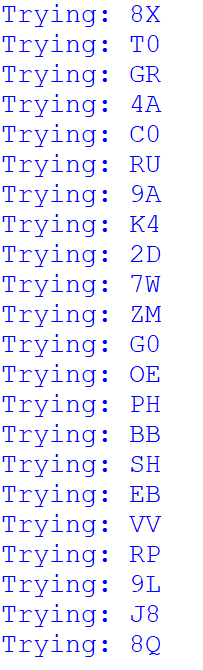
print("Password Not Found within limit.")

print("Target Password:", target)

print("Total Attempts:", attempts)

print("Time Taken:", elapsed, "seconds")

**OUTPUT**

** **

**Explanation:**The output shows that the AI Random Search Algorithm randomly generated different password combinations until it found the correct one (“AB”). Each attempt was checked against the target, and once the correct match was found, the process stopped. The number of attempts and time taken indicate the algorithm’s performance and efficiency.

This demonstrates how Artificial Intelligence can be applied to search and optimization problems — here, by guessing passwords intelligently through random exploration rather than sequential brute-force searching.

**RESULTS AND FUTURE ENHANCEMENT**

**Results:**  
The project successfully demonstrates how an Artificial Intelligence–based Random Search Algorithm can be used to guess password patterns effectively. The algorithm randomly explores various password combinations until it matches the correct one. This approach shows how AI can be applied to real-world security problems such as password strength testing and vulnerability analysis. Although the process is random, it efficiently reaches the correct result without requiring any predefined logic or dictionary, proving the flexibility of AI in solving unpredictable problems.

**Future Enhancement:**  
In the future, this system can be enhanced by integrating more advanced AI techniques such as Genetic Algorithms, Reinforcement Learning, or Neural Networks to make the guessing process more intelligent and faster. It can also be extended to analyze password strength, detect weak password patterns, and provide suggestions for creating secure passwords. Compared to traditional brute-force or dictionary methods, the AI-based approach offers greater adaptability and learning capability, making it a promising step toward improving cybersecurity and automated pattern recognition systems.

|  |  |
| --- | --- |
| **Git Hub Link of the project and report** | **https://github.com/Devamalyaa/Password-Pattern-Guessing-Using-AI-Random-Search-Algorithm-** |

**REFERENCES**

1. **Book:**  
   Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th Edition). Pearson Education.
2. **Book (E-Book):**  
   Stuart Russell & Peter Norvig, *Introduction to Search Algorithms in AI*, Pearson Online Library.
3. **Online Resource (Blog):**  
   GeeksforGeeks – “Random Search Algorithm in Artificial Intelligence”  
   https://www.geeksforgeeks.org/random-search-algorithm-in-ai/
4. **Online Resource (Blog):**  
   Towards Data Science – “Understanding Random Search and Its Role in Optimization”  
   https://towardsdatascience.com/random-search-in-ai-optimization
5. **Video Lecture:**  
   NPTEL Online Course – *Artificial Intelligence: Search Methods for Problem Solving* by Prof. Deepak Khemani (IIT Madras).  
   https://nptel.ac.in/courses/106/106/106106126/