# Probabilistic Random Graph Generations with Feedback Loops


By :-

DEVANSH SRIVASTAVA (2005309)

MONJIMA MAJUMDAR (2005314)

SANDEEP S. SAHOO (2005328)

SAPTANGSHU KAVIRAJ (2005330)

SHIBASISH KAR (2005335)

SAKSHYA VARDHAN SINGH (2005752)

# INTRODUCTION

This project work has been done to utilize the randomness that exists in the systems of real world and to emulate it in such manners that there will be a chance of attaining a secure and endless possibility of networking. There can be future works regarding this project that may be influenced by this work which will lead to more and more algorithms being devised.

## So what have we tried to achieve ?

It is to produce a randomly generated graph that will have certain factors which can be used by us to simulate any circumstance.

# TRUE RANDOMNESS

In nature we cannot say that there exists a state of true randomness . Surely within atomic particles we might be able to find it but in a macroscopic level it is not defined. Any event that occurs will have a chance to occur again in a certain time interval , but we might not be able to track the interval due to limited technologies.

So what we tried to emulate was a state of pseudo randomness where the chance of the event occurring tends towards -> 0.

# Erdos–Rényi Model

In graph theory, the Erdos–Rényi model is related to generation of random graphs. It is a fundamental model in random graph theory.

- ➔ **Concept of Random Graphs**

  Focuses on properties of graphs formed by random processes.
  Analyzes how likely a graph is to exhibit certain characteristics.

- ➔ **The G(n,p) Model**

  n: Number of nodes in the graph.
  p: Probability of any two nodes being connected by an edge.
  Each edge is independent and has the same probability

- ➔ **Significance in Graph Theory**

  Provides insights into the emergence of network properties.
  Fundamental for theoretical research in network science.

The Erdős–Rényi theorem and the G(n,p) model are crucial for understanding the behavior of large-scale networks and have numerous applications in various scientific fields.

# LITERATURE REVIEW

| SUBDOMAIN | PAPER NAME | TOOLS USED | REMARKS ON SUB DOMAIN |
|---|---|---|---|
| TRUE RANDOMNESS | Nathalie Bochard, Florent Bernard, Viktor Fischer, Boyan Valtchanov, "True-Randomness and Pseudo-Randomness in Ring Oscillator-Based True Random Number Generators", *International Journal of Reconfigurable Computing*, vol. 2010 | Ring Oscillators and Timing Jitter<br>XOR gate<br>Cypress USB interface device CY7C6B013A-100AXC | We discussed the properties of the true randomness and how they can be generated in a pseudo manner.<br>Among all the experiments we got to the final conclusion that there exists a state of pseudo randomness that can be achieved with some equipments. |
| | Fei Yu, Lixiang Li, Qiang Tang, Shuo Cai, Yun Song, Quan Xu, "A Survey on True Random Number Generators Based on Chaos", *Discrete Dynamics in Nature and Society*, vol. 2019 | Thermal noise on resistors capacitors , phase jitter of oscillating signals,Boolean Chaotic Oscillator Jitter Booster Circuit , Chaotic Oscillator | |
| | Teh, J.S., Samsudin, A., Al-Mazrooie, M. *et al.* GPUs and chaos: a new true random number generator.<br>*Nonlinear Dyn* **82**, 1913–1922 (2015) | CMOS Boolean chaotic oscillator, On-Semiconductor<br>0.35µm | |

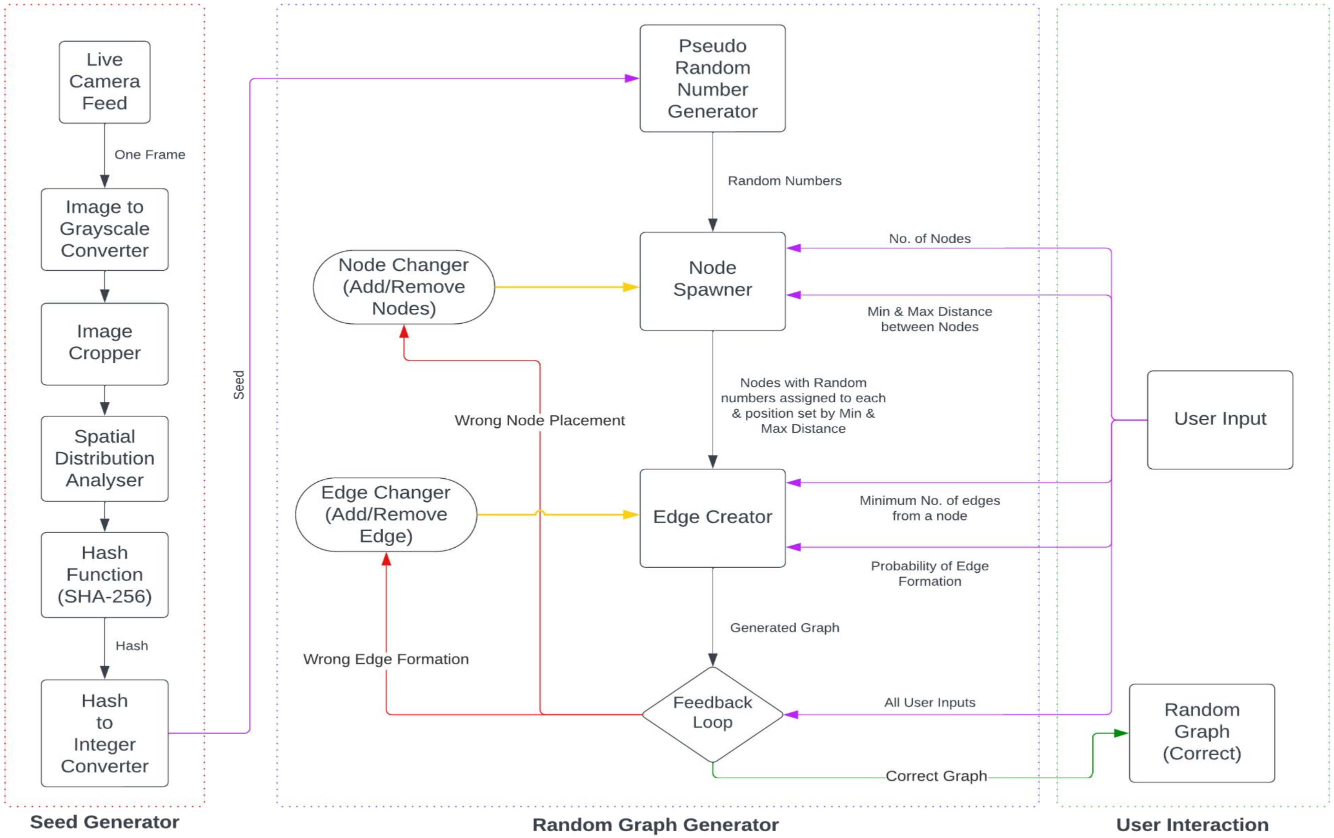
# LITERATURE REVIEW

<table>
<tr><th>SUBDOMAIN</th><th>PAPER NAME</th><th>TOOLS USED</th><th>REMARKS ON SUB DOMAIN</th></tr>
<tr><td rowspan="3">RANDOM GRAPH GENERATION</td><td>Aiello, William, Fan Chung, and Linyuan Lu. "A random graph model for massive graphs." Proceedings of the thirty-second annual ACM symposium on Theory of computing. 2000.</td><td>JGAP library, Motif generators , Noise generators,etc.</td><td rowspan="3">Random graph generation is a fundamental subdomain in graph theory, extensively used for modeling and analyzing complex network systems across various fields.<br><br>Random graph theory provides insights into network resilience and the behavior of networks under different conditions, making it a vital tool in both theoretical research and practical applications.</td></tr>
<tr><td>Zhang, H., Kiranyaz, S. & Gabbouj, M. Outlier edge detection using random graph generation models and applications. J Big Data 4, 11 (2017).</td><td>Erdős- Rényi random graph generation model.</td></tr>
<tr><td>Stéphane Bressan, Alfredo Cuzzocrea, Panagiotis Karras, Xuesong Lu, Sadegh Heyrani Nobari,An effective and efficient parallel approach for random graph generation over GPUs,Journal of Parallel and Distributed Computing, Volume 73, Issue 3, 2013.</td><td>Gilbert's model , ER , ZER and PreZER , etc.</td></tr>
</table>

# LITERATURE REVIEW

| SUBDOMAIN | PAPER NAME | TOOLS USED | REMARKS ON SUB DOMAIN |
|---|---|---|---|
| NETWORK SECURITY | H. Lin, Z. Yan, Y. Chen and L. Zhang, "A Survey on Network Security-Related Data Collection Technologies," in *IEEE Access*, vol. 6, pp. 18345-18365, 2018. | Intrusion detection system model,Network Protocol Analyzer, KDD99 data sets. | Network security, when modeled using probabilistic random graph generations with feedback loops, offers a sophisticated method to analyze and predict network behaviors under security threats<br><br>They are particularly useful in understanding the resilience of network security systems, factoring in uncertainties and aiding in the development of effective security strategies. |
| | Smith, J. A., & Jones, M. B. (2021). "Cybersecurity Threats in IoT Networks: A Machine Learning Approach," Journal of Network Security, vol. 2021. | TensorFlow, Keras, PyTorch, NS3 OMNeT++ , Arduino, Raspberry Pi. | |
| | Chen, F., & Luo, X. (2021). "Evaluating the Effectiveness of Attack Graph Analysis in Network Security," *International Journal of Network Management*, vol. 2021. | NS3 for network simulation<br><br>MulVAL,SeaMonster,Gephi,PyTorch | |

# PROPOSED MODEL



# RANDOM SEED GENERATOR

Randomness in Graph Theory

- **Overview of Graph Theory**

  Randomness critical for network structures.
  Shapes how nodes (vertices) connect with edges.

- **Pseudo-random vs. True Randomness**

  Traditional models use PRNGs - algorithmically generated, limited unpredictability.
  PRNGs: fast, reproducible but deterministic.

- **Limitation of PRNGs**

  Deterministic nature: Same seed results in identical number sequences.
  Issue in scenarios needing high unpredictability (e.g., cryptography).

# RANDOM SEED GENERATOR

True Randomness in Graph Theory

- **Defining True Randomness**
  Originates from inherently unpredictable physical events.
  No set pattern, non-reproducible, genuinely random.
- **Advantages over PRNGs**
  Better mirrors the unpredictability in real-world networks.
  Vital for secure, complex systems simulations.
- **Example - Traffic Footage for Randomness**
  Captures random movements in traffic as a randomness source.
  Movement influenced by various factors (weather, time, environmental changes).

# RANDOM SEED GENERATOR

Implementing True Randomness

➔ **Process Overview**

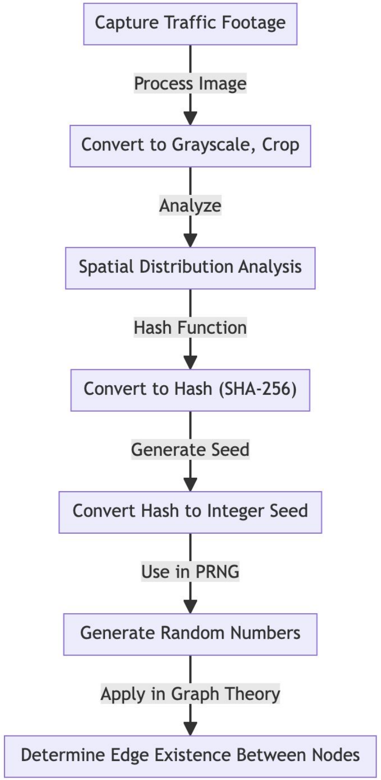
Traffic movements captured, transformed to grayscale, cropped for chaos focus.
Image frames processed for pixel data extraction.

➔ **From Chaos to Numbers**

Spatial distribution analysis leads to hash generation (using SHA-256).
Hash transformed into an integer seed for PRNG.

➔ **Unpredictability in Edge Formation**

True randomness dictates edge placements in graph.
Leads to diverse and complex graph structures, akin to natural systems.



# RANDOM SEED GENERATOR

Applications and Challenges

➔ **Impact on Graph Properties**
Influences key network characteristics: degree distribution, clustering, path lengths.
More realistic modeling of social and biological networks.

➔ **Suitability for Complex Systems**
Ideal for systems where interactions are not fully understood or random.
Adds depth in modeling technological and social network evolution.

➔ **Computational Complexity**
Process more resource-intensive than using PRNGs.
Needs sophisticated algorithms to balance randomness with graph parameters.

# NODE SPAWNER

- Graphs are generated based on connectivity probabilities without regard for the 'distance' between nodes.

- The concept of distance in graph theory can be interpreted in various ways depending on the context: physical distance in geographical networks, functional distance in biological networks, or similarity distance in social networks.

- Deciding on the number of nodes is like determining the scale of a map; more nodes mean a more detailed map, while fewer nodes could represent a more high-level overview.

# NODE SPAWNER

- There's a trade-off between the granularity of the network representation and the computational feasibility of working with the graph.
- In our model, firstly the number of nodes are determined by the user, Each node is then assigned a random number from the numbers from the random number generator.
- The node spawner takes in the number of nodes and arranges them according to the minimum and maximum given distance so that the edges can be formed accordingly.

# EDGE CREATOR

- The probability of edge formation is a fundamental aspect that dictates how these connections are established in a random graph. In simple terms, this probability is like the chance that any two people in a room might know each other.

- In our model, we take the nodes identified with random numbers, if the number surpasses a certain threshold value which is constant, an edge is formed, otherwise not.

- The probability of edge formation must be considered alongside other parameters like the number of nodes, minimum edges, and spatial constraints.

# EDGE CREATOR

- In our model, we check each node for the minimum number of edges, if it does not match the value, then an edge is again added, which will again be determined by the threshold value and the closeness to other nodes.

- For example, in a graph with a high number of nodes, a high minimum edge requirement could lead to a very dense network

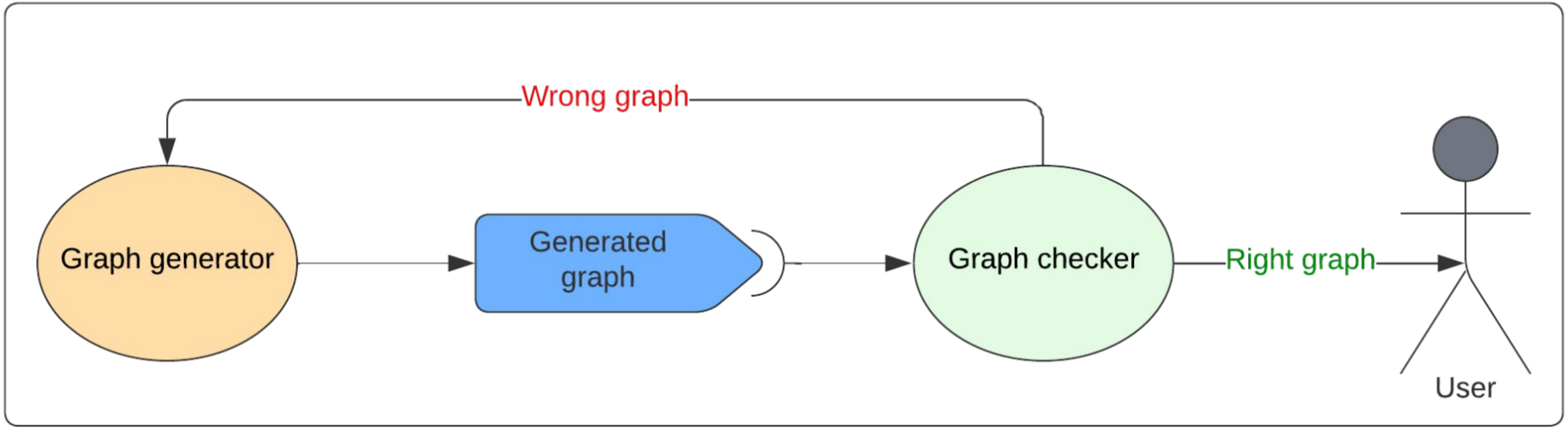
# FEEDBACK LOOP

- Feedback loops in complex systems influence future operations, either amplifying (positive) or dampening (negative) effects, crucial for stability and adaptability.
- In computational models like random graph generation, feedback loops refine algorithms, ensuring accuracy and alignment with specified criteria in simulations and optimization problems.
- The random graph model integrates iterative feedback loops to analyze and adjust the generated graph based on parameters, such as node count, edge probability, and spatial constraints.
- The iterative refinement process involves multiple cycles of graph generation, evaluation, and adjustments for a well-defined representation of the specified model.

# FEEDBACK LOOP

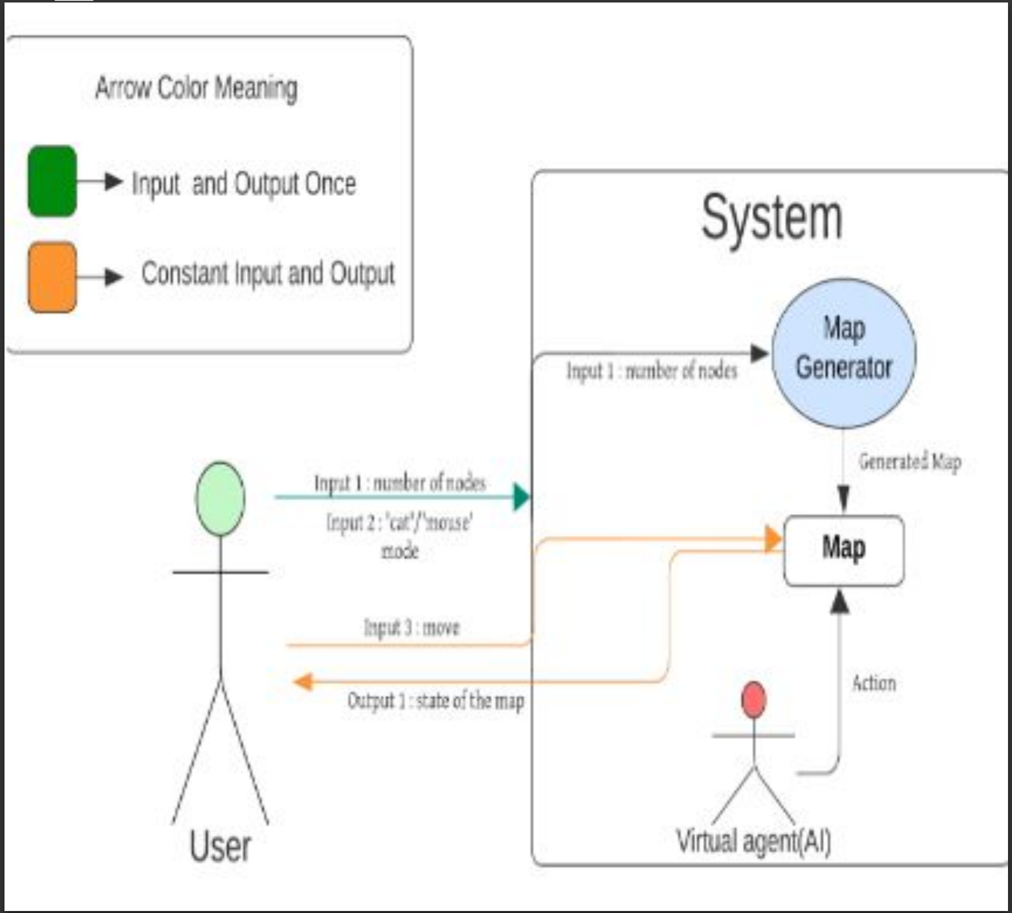
- Implementing feedback loops demands sophisticated algorithmic design capable of simultaneously evaluating multiple graph characteristics.
- Computational complexity is addressed through optimization, parallel computing, and heuristic methods, crucial for handling large graphs.
- Balancing conflicting parameters requires prioritization mechanisms in the adjustment process.
- True randomness introduces complexity, necessitating a delicate balance between unpredictability and adhering to structured graph parameters..

# FEEDBACK LOOP



# IMPLEMENTATION

The scope of this system is to build a game wherein the graphs are generated via the random algorithm discussed in the report.The graphs are generated with complete randomness and by ensuring that the probability of both the agents of the game namely the ‘cat’ and the ‘mouse’ have a probability of winning nearly equal to half.




# WORKING OF THE SYSTEM

**The system is developed so that the player will give the first two inputs once which are as follows:**

**a) number of nodes**
**b) ‘cat’/’mouse’ mode.**

**Thereafter the input and output to the player will be displayed on the UI where it will be constantly updated.**

# CONCLUSION

We have found a way of harnessing the chaos in the nature and are able to generate a state of randomness using which we have generated a series of graph that can be generated multiple number of times and in multiple number of ways but what remains is to utilize these graphs in order to achieve something.

Since the area of network computing is increasing and there is a rise in quantum computing we hope that this project would be used as a reference in these fields and we also would like to contribute something to the field of networking.

# FUTURE WORKS

1. To complete the game and make it properly functional.

2. To create an encryption standard using the graphs generated.

3. To create newer networks using this model.

# THANK YOU