# Chapter 1

# Introduction

In the multifaceted realm of network theory and complex systems, the generation and analysis of random graphs have been pivotal. These models serve as a linchpin in unravelling the intricate web of connectivity and interdependence that form the backbone of such systems. Among these, the Erdős–Rényi (ER) model, a brainchild of Paul Erdős and Alfréd Rényi from the late 1950s, has been instrumental. By positing that the formation of nodes and edges in a graph adheres to certain probabilistic rules, this model has shed light on the phase transitions and structural nuances inherent in random graphs. However, its elegance in simplicity belies a critical limitation: the inability to encapsulate the multifarious traits evident in real-world networks, such as clustering, community structures, and, most notably, feedback loops.

This project embarks on a novel trajectory, extending the traditional paradigm of random graph generation to encompass the concept of feedback loops. These loops, emblematic of self-regulation and mutual influence, are ubiquitous across natural, technological, and social systems. They are the keystones in regulatory processes, maintaining homeostasis, and are integral in technological realms for adaptive control and fault tolerance. In the sphere of social networks, they underpin the propagation of information and the genesis of communities. Our empirical investigations reveal that introducing feedback loops imparts a temporal dimension to the evolution of networks. Nodes, under the influence of their historical states and the states of their interconnected peers, display adaptive behaviours. This leads to a dynamic reconfiguration of the network structure, with clusters emerging, dissolving, and re-emerging, challenging the static nature of conventional random graph models and aligning more closely with the fluid dynamics observed in real-life systems.

By weaving feedback loops into the fabric of probabilistic graphs, we enhance the Erdős–Rényi model, infusing it with a degree of realism previously unattainable. This augmentation is not merely theoretical; it is palpable in a myriad of networks, from ecological systems to the labyrinthine corridors of social media. In these arenas, the nodes and edges, enriched by the feedback loops' interconnected elements, evolve over time. They are sculpted by their antecedent states and connections, culminating in complex structures that defy the oversimplifications of standard ER models.

The cornerstone of this project is to delve into the "Probabilistic Random Graph Generations with Feedback Loops" framework. Our goal is to bridge the chasm between theoretical abstractions and the multifaceted complexity characteristic of real-world networks. To navigate this exploration, we pose several pivotal research questions. These include inquiries into how feedback loops temporally influence the topology of random graphs, their implications on network robustness and vulnerability, and the potential application of these insights in real-world scenarios ranging from epidemiological models to information dissemination and infrastructure resilience. We hypothesize that the incorporation of feedback loops will yield network structures that more accurately mirror the dynamic and adaptive essence of real-world systems.

This endeavour strives to render a more authentic portrayal of the networks that permeate our daily lives. Our report meticulously details the theoretical underpinnings of random graph models and feedback loops, alongside a thorough review of pertinent literature to contextualize our research within the current scholarly landscape. We outline an intricate methodology for crafting probabilistic random graphs with feedback loops, followed by an exhaustive analysis of our experimental outcomes. The discussion and interpretation of our findings illuminate the transformative effects of feedback mechanisms on both the structure and dynamics of networks. Additionally, we explore the practical applications of this enhanced random graph model, highlighting its potential impact across various fields, from epidemiology and sociology to infrastructure design and information dissemination.

Through this exploration, we aspire to deepen the understanding of the intricate interplay between network structures and feedback mechanisms, focusing on the emergent properties of complex systems. This endeavour not only sheds light on the nuanced dynamics of networks but also significantly advances the field of network science, enhancing our comprehension of the intricate tapestry of the world around us.

The main objective of this project is to explore the framework of "Probabilistic Random Graph Generations with Feedback Loops" as a means to bridge the gap between theoretical models and the complexity of real-world networks. In order to guide our exploration regarding random graphs using feedback loops, several key research questions arise, for example, the ways in which the topology of the random graphs are influenced over time by the feedback loops, the implications of feedback loops on network robustness and vulnerability, the ways in which these insights can be applied to real-world scenarios such as epidemiological methods, information diffusion or infrastructure resilience and so on. We hypothesize that the introduction of feedback loops will result in network structures that better mimic the dynamic and adaptive nature of real-world systems.

It aims to provide a more accurate representation of the networks we encounter in our daily lives. In this project report, there is a vivid description about the theoretical foundations of random graph models and feedback loops, reviewing relevant literature to position our research in the current landscape. There is also a detailed methodology employed for the creation of probabilistic random graphs with feedback loops, followed by a comprehensive analysis of the results of our experiments. Discussions and interpretations of our findings will lead to insights into the transformative impact of feedback mechanisms on the network structure and dynamics. Finally, there is also a mention about the practical applications of this extended random graph model and the implications for various fields, from epidemiology and sociology to infrastructure design and information dissemination. While our project findings made significant strides, there still remain avenues for further exploration. The impact of varying feedback loop strengths, the influence of different network topologies, and the interplay between multiple feedback mechanisms warrant deeper investigation. Moreover, extending the application of our enriched random graph model to diverse fields, such as ecological systems and economic networks, holds promise for uncovering universal principles governing complex systems.

Thus, by the help of this exploration, we aim to contribute to the understanding of the interplay between network structure and feedback mechanisms, focussing on the intricate and often surprising emergent properties of complex systems at a deeper level. This also has the potential to contribute significantly to the advancement of network science and its profound impact on our comprehension of the world around us.

# Chapter 2

# Literature Review

## 2.1 True Randomness

We might be able to find true randomness if we would have a larger scale of dimensions for measurement but since we are unable to do so then we can only find a state of pseudo randomness that will have the probability of the event occurring tending towards 0. So we set out to find such experiments that would demonstrate the state of pseudo randomness and we made these following observations : -

1) In the research paper [1] the authors had used ring oscillators and timing jitters for generating a series of random numbers by using the ring oscillator to trace its position and determine a random number accordingly.

2) In the paper [2] & [3] they have used a thermal noise on resistors and capacitors that can detect the heat produced by those equipment and at different times the heat produced were varying in nature so they used this factor to produce their random number.

From this we thought that in the real world the chaos occurs mostly in the traffic , so we thought of analysing the traffic to get our randomness.

## 2.2 Random Graph Generation

Generation of graph randomly is a tedious task because of the complex nature of putting nodes and making the edges between them and we have to make the graph on certain variables and with constraints. So we searched for ways as discussed by the [4] report that would be beneficial and an already existing model was there in [5] so we took the ER model and used it for generation of the graph.

The Erdos-Renyi (ER) model for generating graphs takes in two variables the 'number of nodes' and the 'probability of any two nodes being connected with an edge'. In the ER model the graphs being generated were not up to our mark so we had to alter the algorithm to suit our purpose and thus we added a few variables of our own. They were the ‘maximum and minimum distance between the nodes’ and the ‘probability for the edge formation’. In the [6] report there has been talks of another model similar to ER which was the ‘Gilbert’s model’ which had similar graph generation strategies.

## 2.3 Network Security

Network security is a topic that has been there since the advent of the internet and is rapidly growing as new threats and the ways to counter them emerge. So we had taken [7] as our base work for thinking what had been done and what new could be added to the system and this part of our work will continue till a further time because it has not been solidified yet. In the [9] report the network simulation has been done and we were able to trace that the packets being sent over the network had a standard encryption and then we thought about making our own encryption standard. This is a part of our work that will span till a further time.

# Chapter 3

# Problem Statement and Requirement Specifications

We propose generating graphs by harnessing the randomness that is present in nature and by using a pre-made model of ER method we can obtain our goals of creating these graphs.
Requirements Specifications

Software :- i) Python
ii) Python Editor

Libraries Required :- i) Pygame
ii) Spritesheet
iii) Game Math

## 3.1 Project Planning

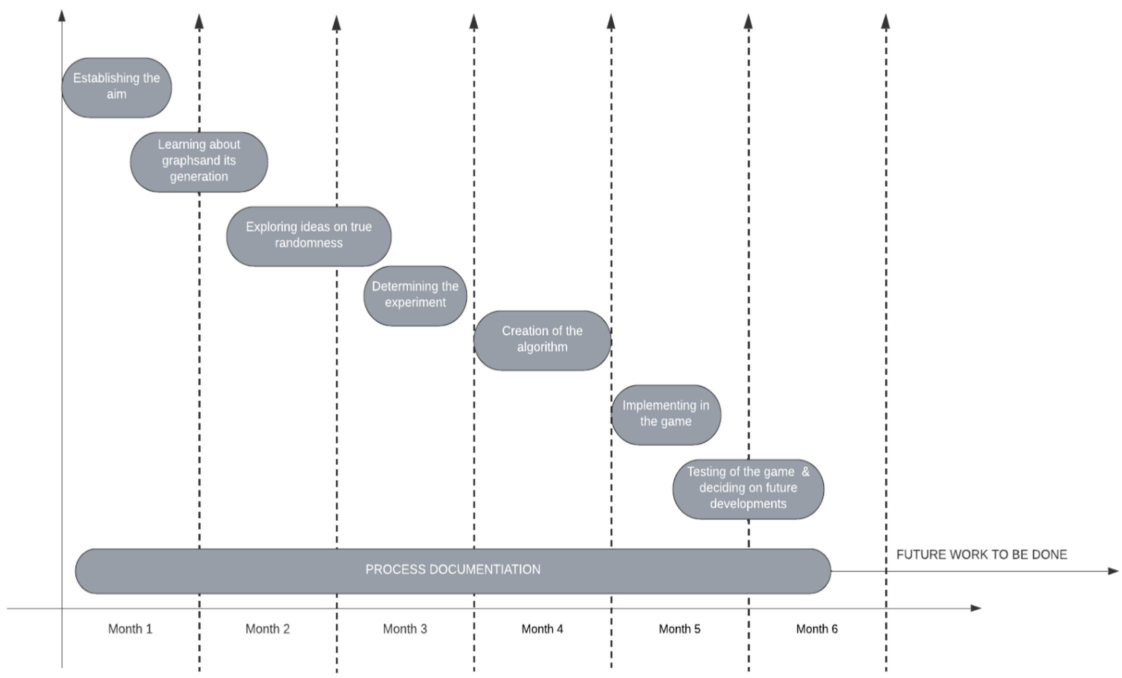


Figure 3.1.1: GANTT CHART

## 3.2 Project Analysis

A) Game Perspective

The users when entering the game will be asked how many nodes they want and upon giving a number the map is generated according to the algorithm discussed in . Then the player is given a choice to either play as the 'cat' or the 'mouse' where the conditions are different for both the entities.

B) 'Cat' Perspective

In the 'cat' perspective the player's main objective is to catch the 'mouse' entity which is an AI that works on [2] algorithm. The 'cat' entity can move to any adjacent node on the map. The map will be generated in such a way that the probability of the 'mouse' winning will be close to 0.5.

C) 'Mouse' Perspective

In the 'mouse' perspective the player's main objective is to run away from the 'cat' entity till a set number of moves after which there will be a doorway on one of the nodes in the map , upon reaching which the 'mouse' will be declared the winner. In this mode the 'cat' entity will use [2] algorithm for catching the 'mouse' entity.

The restriction on this entity is that it can traverse the adjacent nodes but it cannot go back to the node where it came from i.e it can't go back to the node from where it reached the present node.

D) Game System

The system is developed so that the player will give the first two inputs once which are as follows: a) number of nodes and b) 'cat'/'mouse' mode. Thereafter the input and output to the player will be displayed on the UI where it will be constantly updated.

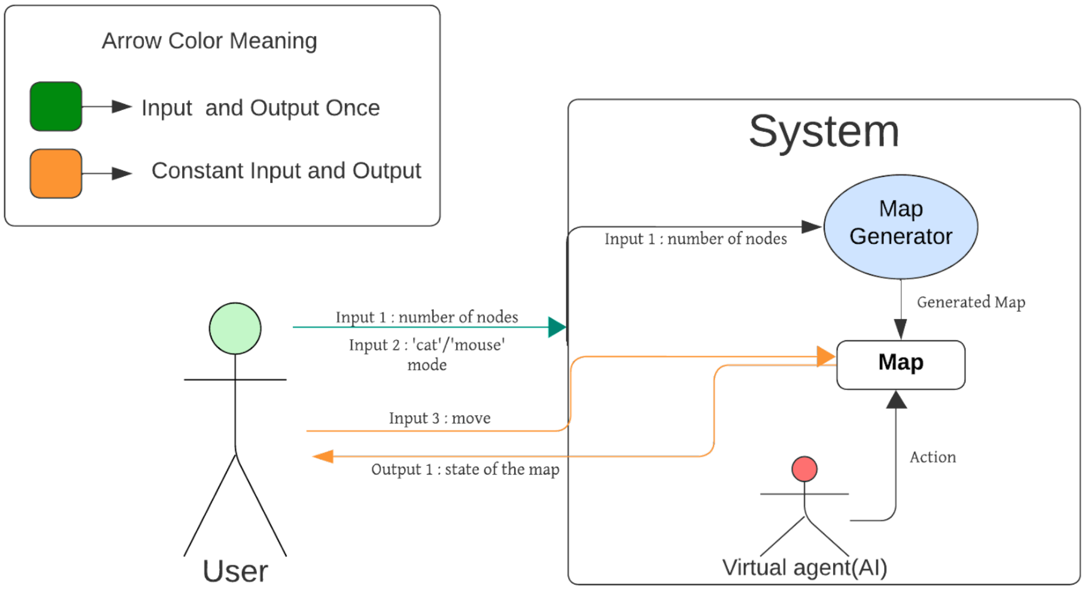


Figure 3.2.1: INTERACTION OF USER WITH SYSTEM

The game will continuously update the environment according to the player's decision and the AI's decision and show the updated map in every instance to the user.

## 3.3 System Design

### 3.3.1 Design constraint

Predictability: The AI controlling the cat or mouse should be somewhat predictable to allow strategic gameplay.

Adaptability: AI should adapt to player strategies, offering a dynamic challenge.

Balanced Difficulty: Ensure the game is neither too easy for the cat nor too hard for the mouse, maintaining a fair challenge for both.

Movement Mechanics: Define clear rules for how the cat and mouse can move within the game environment.

## 3.3.2 Block Diagram

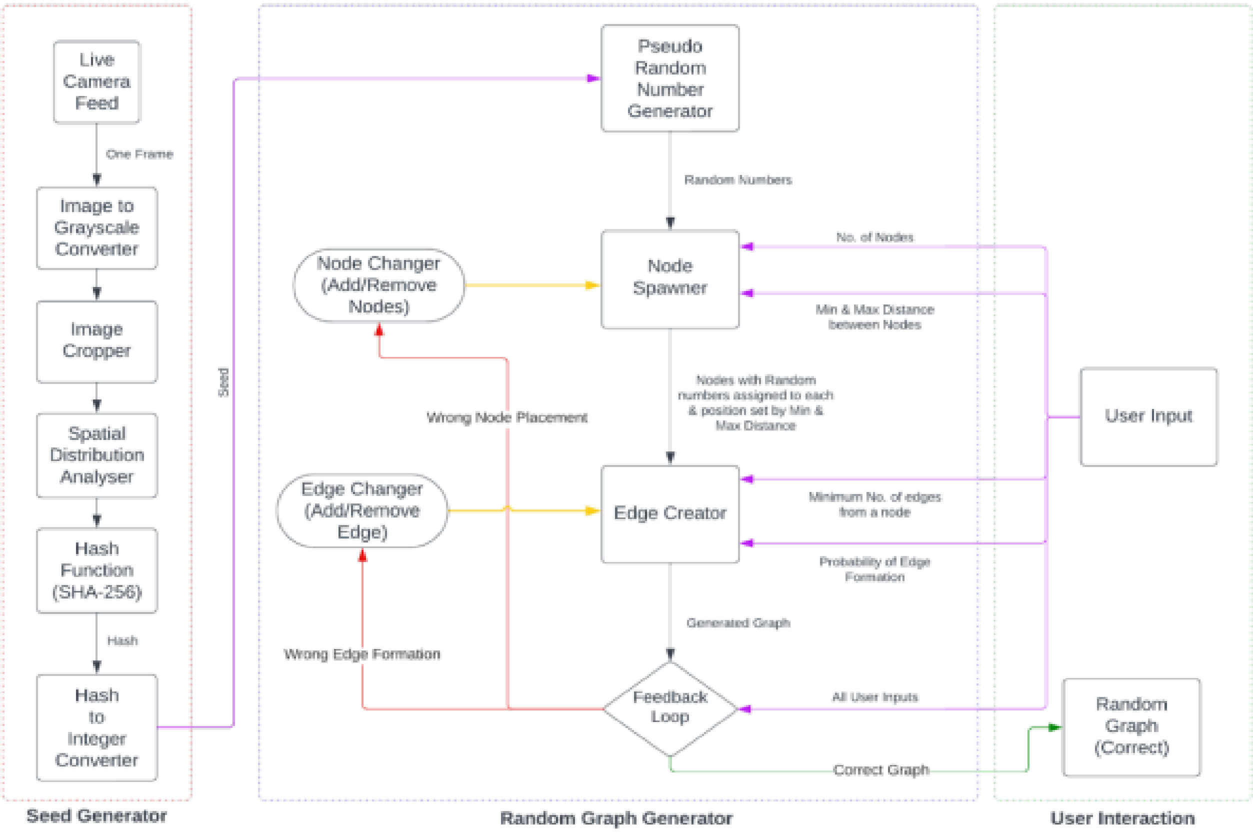


Figure 3.2.2.1: BLOCK DIAGRAM OF PROPOSED MODEL

# Chapter 4

# Implementation

The proposed system has been designed, implemented and tested stage by stage involving significant effort for several months. To implement this model, we have divided the model into parts which are further subdivided into parts. This has been done to ease the complexity of the model. In Fig we can see a flowchart of the proposed model.

## 4.1 Proposal

### 4.1.1 True Randomness

In this part we will demonstrate all the requirements and workings to implement true randomness over pseudo randomness and how a complex system has been implemented to generate different seeds to generate truly random numbers.

In traditional graph theory, randomness plays a pivotal role in the formation of networks. Randomness dictates how nodes (vertices) are connected by edges, leading to various structural properties of the graph. Traditional models, for instance, use a simple random process where each pair of nodes has a fixed probability of being connected by an edge. However, this approach often relies on pseudo-random number generators (PRNGs), which are algorithmically generated and thus inherently deterministic to some degree. PRNGs are commonly used in computing due to their speed and reproducibility. They generate sequences of numbers that appear random but are actually the result of deterministic computational processes. The limitation here is predictability; given the same initial 'seed', a PRNG will always produce the same sequence of numbers. This can be a significant drawback in scenarios where unpredictability is crucial, such as cryptographic applications or complex system simulations.

True randomness, by contrast, derives from physical phenomena that are inherently unpredictable. These phenomena do not follow a set pattern and are not reproducible, making them genuinely random. In graph theory, true randomness can introduce a level of unpredictability and complexity in the structure of the graph, which is more reflective of real-world networks and phenomena.
The proposed model uses a traffic footage feed for random number generation. This method captures the random, chaotic movement of cars and people in a busy

traffic intersection and translates it into digital data. The inherent unpredictability in the movement of the cars and people in addition to several factors, which is influenced by factors like weather, time of day, and even minor environmental changes, ensures that the randomness is not algorithmically generated.

A camera continuously captures the movements of traffic at the intersection. These images are then processed to extract random data, which can be used as seeds for random number generation. First an image frame from the live feed is selected and converted to grayscale and then cropped to only focus on areas of more chaos. This image is then sent to a spatial distribution analyser which collects pixel data from the image. This pixel data is sent to a hash function (SHA-256) which converts it into a hash. This hash is then converted to an integer seed. This seed is used by the PRNG to generate random numbers. The process leverages randomness in the physical world, which is fundamentally different from the computational processes used in PRNGs. It's a method that harnesses chaos theory, where small changes in initial conditions lead to vastly different outcomes.

This approach offers a higher degree of unpredictability compared to PRNGs. Since the movement of traffic is chaotic and not bound by a deterministic algorithm, the resulting random numbers are less predictable and more secure. For graph generation, this means the structure of the generated graph will be more varied and complex, better simulating the randomness found in natural and human-made networks.

In our model, this true randomness is used to determine the existence of edges between nodes. Instead of using a fixed probability or a pseudo-random process, the model uses random numbers generated from the seeds to decide whether or not to place an edge between each pair of nodes. This approach can lead to more complex and varied graph structures. The inherent unpredictability in edge formation makes the resulting graphs more akin to real-world networks, where connections are often not purely random but also not entirely predictable.

The use of true randomness can affect various properties of the graph, such as its degree distribution, clustering coefficients, and path lengths. These properties are crucial in understanding the behaviour and characteristics of networks. For example, in social networks, the clustering coefficient (which measures the degree to which nodes in a network tend to cluster together) can be significantly impacted by the randomness of edge formation. A true random process might lead to a more realistic simulation of social networks, where connections are not always logically predictable.

The unpredictability introduced by true randomness makes the model particularly suitable for simulating complex systems, such as biological networks, where the interactions between elements are not fully understood or are subject to random fluctuations. It also adds a layer of complexity to the modelling of technological and social networks, where randomness plays a significant role in the formation and evolution of connections.

While the true randomness approach offers significant advantages, it also poses challenges in terms of computational complexity. The process of capturing and processing physical phenomena to generate random numbers can be more resource-intensive than using a PRNG. This might impact the feasibility of the model for large-scale graphs or scenarios where quick generation of multiple graphs is required.

In our model that also considers other parameters like minimum edges or distances between nodes, integrating true randomness requires careful balancing. The model must ensure that the randomness does not conflict with these parameters, leading to graphs that fulfil all set criteria. This necessitates sophisticated algorithms and feedback mechanisms to continuously adjust the graph according to the parameters while maintaining the unpredictability of the edge formation.

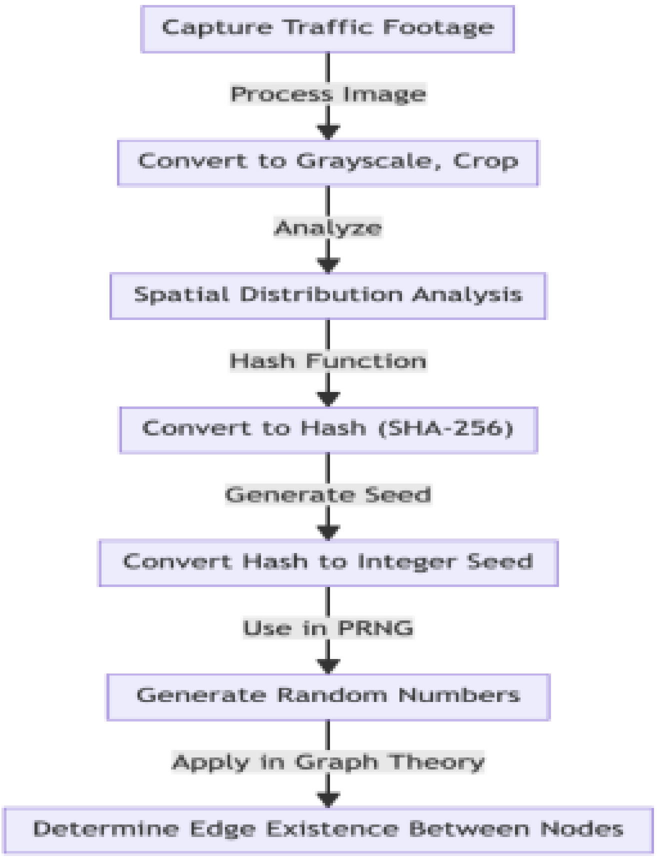


Figure 4.1.1.1: FLOWCHART OF SEED GENERATOR

### 4.1.2 Number of Nodes

In graph theory, nodes (or vertices) are fundamental. They represent the discrete elements of the network. The number of nodes in a graph is a primary factor that defines its scale and complexity. For instance, in social network analysis, nodes could represent individuals, in biological networks, they might represent cells or species, and in computer networks, nodes could be individual computers or servers.

The number of nodes in a graph directly correlates with the complexity of the system it represents. More nodes typically mean a more complex system with a greater number of relationships or interactions to consider. In practical terms, the number of nodes determines the potential number of connections (edges) in the graph, as each node can potentially connect with every other node.

In this model, the 'number of nodes' parameter is a crucial starting point for graph generation. It sets the stage for how expansive or detailed the graph is going to be. 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. The number of nodes affects various structural properties of the graph, such as its density, the potential for clustering, and the likelihood of isolated subgraphs. A higher number of nodes can lead to a more complex network with a greater possibility for intricate substructures and patterns to emerge.

A higher number of nodes offers more detail but can make the graph more complex and harder to analyse or simulate. There’s a trade-off between the granularity of the network representation and the computational feasibility of working with the graph. The number of nodes can affect the dynamics of the network, such as how quickly information or influence spreads through the network, or how resilient the network is to disruptions. In modelling scenarios, like simulating disease spread or information dissemination, selecting an appropriate node count is crucial for accuracy.

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 probability of edge formation, when combined with the number of nodes, determines the potential number of connections in the graph. This interplay is critical in defining the overall structure and connectivity of the graph. The minimum number of edges and spatial constraints (minimum and maximum distances between nodes) must be considered in relation to the node count. These parameters collectively influence the layout and connectivity of the graph, ensuring it aligns with the intended model criteria.

Feedback loops in this model serve to evaluate whether the generated graph aligns with the set parameters, including the number of nodes. This mechanism ensures the graph remains consistent with the predefined structure and intended complexity level. If the initial graph does not meet the criteria, adjustments are made. This might involve adding or removing nodes (and consequently edges) to achieve the desired graph characteristics.

### 4.1.3 Probability of edge formation

In graph theory, an edge represents a connection or a relationship between two nodes (vertices). 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. The higher the probability, the more connections (friendships) are likely to be formed.

The probability of an edge forming between any two nodes is crucial in determining the network's overall structure. It influences whether the graph will be sparse (few connections) or dense (many connections). This aspect is pivotal in modelling real-world networks, where connections aren't uniformly distributed but follow specific probabilities based on the nature of the network (like social networks, transportation networks, etc.).

In a traditional model, the probability of edge formation between any two nodes is constant. However, in the enhanced version, this probability can be more nuanced, potentially varying across different pairs of nodes. This variability allows for a more realistic representation of actual networks, where the likelihood of connections varies based on factors like distance, node characteristics, or external influences. By allowing for a variable probability of edge formation, the model can simulate a wider range of network types. For instance, in a social network, close friends (or nodes in close proximity) might have a higher probability of being connected than acquaintances (or distant nodes). This flexibility is key to modelling complex networks, like biological systems, where some connections are more probable due to biological functions or physical proximity.

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.

One of the main challenges in utilizing this feature is accurately determining the probabilities that reflect the real-world scenarios being modelled. This requires in-depth knowledge of the specific network and the factors influencing connection formation. Incorrect probabilities can lead to unrealistic network models that don't adequately represent the actual systems. While more nuanced probabilities offer more realistic models, they also increase the model's complexity. This can lead to greater computational demands, especially for large networks with a high number of nodes. Finding a balance between realistic representation and manageable computational complexity is essential for the practical application of the model.

The probability of edge formation must be considered alongside other parameters like the number of nodes, minimum edges, and spatial constraints. This interplay determines the final structure of the graph. For instance, if the minimum number of edges between nodes is high, but the probability of edge formation is low, the model might struggle to satisfy both criteria, requiring adjustments through feedback loops.

Feedback loops in this model are crucial for ensuring that the graph adheres to the set parameters, including the probabilities of edge formation. After initial graph generation, the model assesses whether the edges' distribution aligns with the predetermined probabilities. If discrepancies are found, the model iteratively adjusts the graph to better reflect these probabilities.

### 4.1.4 Minimum number of edges from node

In graph theory, edges are the lines that connect nodes (vertices). The number of edges emanating from a node is known as its degree. The concept of setting a minimum number of edges for each node in a graph introduces a baseline level of connectivity for each element in the network. This parameter ensures that every node in the graph is integrated into the network to a certain minimum extent, preventing the occurrence of isolated or under-connected nodes. In real-world networks, it's rare to find elements (nodes) that are completely isolated. Whether in social, biological, or technological networks, every element typically has at least some level of interaction or connection with others. Setting a minimum number of edges per node in the model helps simulate this aspect of real-world networks.

Unlike the traditional models, where the focus is primarily on the probability of edge formation between any two nodes, this improved model introduces the parameter of a minimum number of edges for each node. This ensures that the generated graph adheres more closely to the connectivity patterns observed in real-world networks.

The parameter stipulates that each node in the graph must be connected to at least a certain number of other nodes, which can be crucial in modelling networks where isolation of nodes is unrealistic or undesirable. This parameter allows the model to simulate networks where certain levels of interconnectedness are essential. For instance, in social networks, it's uncommon for individuals to have no connections at all, and in biological networks like neural networks, each neuron typically forms synapses with multiple other neurons.

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.

One challenge in utilizing this feature is accurately determining the appropriate minimum number of edges per node. This requires an understanding of the specific network being modelled and the nature of interactions within that network. Setting the minimum too high might result in an unrealistically dense network, while setting it too low may not adequately capture the network's interconnectedness. The minimum number of edges needs to be balanced with other parameters like the total number of nodes and the overall probability of edge formation. There is a risk of creating overly complex or overly simplistic networks if not properly balanced.

The minimum number of edges per node interacts with other parameters like the total number of nodes and the probability of edge formation. This interaction influences the overall topology of the graph, such as its density and clustering tendencies. For example, in a graph with a high number of nodes, a high minimum edge requirement could lead to a very dense network, potentially skewing certain network metrics like average path length or clustering coefficient.

Feedback loops in the model evaluate whether the generated graph meets the minimum edge criteria for each node. If not, the model iteratively adjusts the graph – this might involve adding edges to under-connected nodes while ensuring that other parameters like the overall edge probability are still respected.
These loops help maintain the balance between creating a graph that is both random and structurally consistent with the set parameters.

### 4.1.5 Minimum and Maximum Distance between Nodes

In traditional random graph models, spatial constraints are typically not considered. Graphs are generated based on connectivity probabilities without regard for the 'distance' between nodes. However, in many real-world networks, the physical or logical distance between nodes significantly influences their likelihood of being connected. 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. Incorporating minimum and maximum distances between nodes allows for a more realistic representation of networks. It enables the modelling of scenarios where proximity or remoteness plays a crucial role in determining connections. For instance, in a social network, people are more likely to form relationships with those geographically or socially closer to them, while in a transportation network, connectivity is influenced by geographical proximity.

The 'minimum distance' parameter sets a threshold below which nodes are not allowed to form connections. It simulates a barrier or a minimum level of separation required for a connection to be feasible or meaningful. The 'maximum distance' parameter, conversely, sets an upper limit on the separation distance allowable for a connection. It ensures that nodes are not too far apart to be connected, reflecting limitations like range, capacity, or relevance. These parameters can be applied differently depending on the type of network being modelled. In a geographical network, they could represent literal distances. In a social network, they might represent degrees of separation in relationships or similarity in interests.

In our model, all other parameters supersede these two parameters. However these parameters are easy to implement in graphs as graphs are formed in a spatial environment.

The introduction of minimum and maximum distances can significantly alter the topology of the generated graph. It can lead to the formation of clusters or the segregation of the network into distinct groups or components. This feature is particularly relevant in studying networks where clustering or compartmentalization is observed, like in community structures in social networks or habitat distributions in ecological networks. By setting these spatial parameters, the model can simulate real-world scenarios more effectively. For example, in urban planning, understanding how different areas of a city are connected considering physical distances can aid in optimizing infrastructure development.

One of the primary challenges is defining what constitutes 'distance' in a given network. This requires a deep understanding of the system being modelled and the factors that influence connectivity. The choice of distance metrics (Euclidean, Manhattan, etc.) and their appropriateness for the model must be carefully considered. Ensuring that the network remains connected and functional while adhering to the minimum and maximum distance constraints can be challenging. This is particularly true for networks that naturally have long-distance connections, like certain communication or transportation networks.

The spatial constraints must be balanced with other parameters like the number of nodes, the probability of edge formation, and the minimum number of edges. This interplay determines the final structure and characteristics of the graph. For example, a high probability of edge formation might conflict with strict spatial constraints, requiring careful calibration to ensure a realistic and functional network. Feedback loops in this model are used to assess whether the generated graph adheres to the spatial constraints set by the minimum and maximum distances. If the initial graph does not meet these criteria, the model iteratively adjusts the placement or connections of nodes to align with the spatial parameters. This might involve repositioning nodes, adding or removing edges, or even introducing new nodes to fill spatial gaps.

### 4.1.6 Feedback Loops

In complex systems, feedback loops are mechanisms through which a system processes output data to make adjustments to its future outputs. Essentially, a feedback loop involves taking a portion of the system's output and using it to influence future operations. These loops can be positive (amplifying effects) or negative (dampening effects), and they play a crucial role in maintaining the stability and adaptability of systems. In computational models, feedback loops are used to refine algorithms, enhance accuracy, and ensure that outputs align with specified criteria or objectives. They are particularly significant in models that require a high degree of precision and adaptability, such as in simulations of natural processes, optimization problems, or, in this case, random graph generation.

In our random graph model, feedback loops are integrated to ensure that the generated graphs accurately reflect the set parameters (number of nodes, probability of edge formation, minimum edges, spatial constraints). These loops analyse the generated graph and compare its properties with the desired criteria. If discrepancies are found, the model makes adjustments, such as adding or removing edges or nodes, and regenerates the graph. The feedback process is iterative. It involves repeatedly generating the graph, evaluating it, and making necessary adjustments until the graph meets all the specified parameters. This iterative refinement is crucial for dealing with the complexity and interdependencies of the parameters, ensuring that the final graph is not just a random construct but a well-defined representation of the specified model.

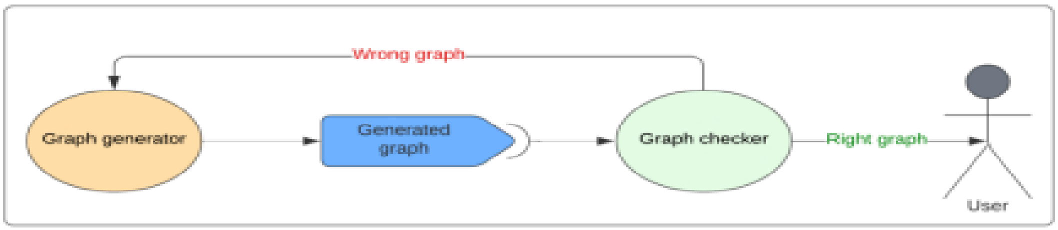


Figure 4.1.6.1: FLOW OF FEEDBACK LOOP

The implementation of feedback loops in the model requires sophisticated algorithmic design. The algorithms must be capable of evaluating multiple graph characteristics simultaneously and making decisions on how to adjust the graph to better meet the parameters. This involves complex decision-making processes, often leveraging techniques from fields like optimization theory and computational intelligence. The primary goal of the feedback loops is to ensure that the generated graph faithfully represents the model as defined by the parameters. This includes maintaining the balance between randomness and the structural constraints imposed by the parameters.

One of the major challenges in implementing feedback loops in this context is the computational complexity involved. Evaluating a graph against multiple parameters and iteratively adjusting it can be computationally intensive, especially for large graphs. Solutions to this challenge include optimizing the evaluation algorithms for efficiency, using parallel computing techniques, and employing heuristic methods to guide the graph adjustment process. Another challenge is balancing the different parameters against each other, especially when they may have conflicting requirements. For example, a high probability of edge formation might conflict with spatial constraints. To address this, the model might employ weighting systems or prioritization mechanisms to decide which parameters take precedence during the adjustment process.

The true randomness aspect of the model adds an additional layer of complexity to the feedback loops. The randomness affects how edges are formed between nodes, which the feedback loops must then evaluate and potentially adjust. The challenge here is to maintain the unpredictability and variability introduced by the true randomness while still ensuring that the graph adheres to the specified parameters. The feedback loops must strike a balance between the chaotic nature of true randomness and the structured requirements of the graph parameters. This involves allowing for enough randomness to ensure variability and complexity in the graph while preventing the graph from deviating too far from the set criteria.

## 4.2 Testing

Since the model is not fully complete yet, testing phase is not yet completed.

## 4.3 Result Analysis

Since the model is not fully complete yet, result analysis cannot be properly computed.

## 4.4 Quality Assurance

As for the quality assurance we have complied to every standard that has been set by the IEEE and done the documentation as well as testing of each and every component that has been used in this project.

Every logical component and function was individually programmed and tested before being integrated into the environment and as for IEEE 730 standards of software quality assurance we have tested the software required to run the projects.

# Chapter 5

# Standards Adopted

## 5.1 Design Standards

The standards for design that have been applied here in compliance with the IEEE are as follows:-

| | |
|---|---|
| 1. IEEE 829 | Software Test Documentation |
| 2. IEEE 830 | Software Requirements Specifications |
| 3. IEEE 1016 | Software Design Description |
| 4. IEEE 1028 | Standard for Software Reviews and Audits |
| 5. IEEE 1074 | Software Development Life Cycle |
| 6. IEEE 1914.1 | Standard for Packet-based Fronthaul Transport Networks |

## 5.2 Coding Standards

The variables ,functions and libraries used in the programming of the game are as follows :-

1. pygame: Imported library for creating video games and multimedia applications.

2. generate_random_graph: Function imported from game_mechanics module to create a random graph for the game.

3. spawn_cat_mouse: Function imported from game_mechanics module to place the cat and mouse in the graph.

4. cat_strategy: Function imported from game_mechanics module to determine the cat's movement strategy.

5. N_NODES: Constant defining the number of nodes in the graph.

6. EDGE_PROB: Constant defining the probability of edges being created in the graph.

7. graph: The generated random graph based on N_NODES and EDGE_PROB.

8. cat, mouse: Variables representing the positions of the cat and mouse in the graph.

9. screen: Pygame display surface for rendering the game graphics.

10. EXIT_BUTTON_X, EXIT_BUTTON_Y, EXIT_BUTTON_WIDTH, EXIT_BUTTON_HEIGHT: Constants defining the dimensions and position of the exit button.

11. exit_button: Pygame Rect object representing the exit button.

## 5.3 Testing Standards

1. IEEE 829 (Test Documentation): Adhere to this standard for test plan documentation.
2. IEEE 1012 (System and Software Verification and Validation): Follow for V&V processes.

# Chapter 6

# Conclusion and Future Scope

## 6.1 Conclusion

In conclusion, we can say that we have achieved a method for generating the random graphs by giving a few inputs and are using the chaos present in the nature to produce our graphs. This work has not yet been considered to be complete because we would like to include this work in any of the networking fields that are present as well as we would try to incorporate this in the quantum computing field as well. The generation of graphs by harnessing the randomness would prove to be an excellent way of paving the path for an age of digital security. We are also working towards the advanced AI and applied game theories.

## 6.2 Future Scope

We would like to encompass this algorithm in the following manner :-

1) Create a better way of encryption standard using the generated graphs.

2) Completing the game that we are working on.

# References

[1] 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

[2] 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

[3] Teh, J.S., Samsudin, A., Al-Mazrooie, M. *et al.* GPUs and chaos: a new true random number generator. *Nonlinear Dyn* **82**, 1913–1922 (2015)

[4] 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.

[5] Zhang, H., Kiranyaz, S. & Gabbouj, M. Outlier edge detection using random graph generation models and applications. *J Big Data* **4**, 11 (2017).

[6] 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.

[7] 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.

[8] Smith, J. A., & Jones, M. B. (2021). "Cybersecurity Threats in IoT Networks: A Machine Learning Approach," Journal of Network Security, vol. 2021.

[9] Chen, F., & Luo, X. (2021). "Evaluating the Effectiveness of Attack Graph Analysis in Network Security," *International Journal of Network Management*, vol. 2021.

[10] https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

# Probabilistic Random Graph Generations with Feedback Loops

SAPTANGSHU KAVIRAJ
2005330

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** I have worked towards the coding of the true random seed generator as well as the generation of random graphs. I have designed and coded the basic logic of the game which is implemented using the random graphs.

**Individual contribution to project report preparation:** I have worked towards preparing a basic overview of the entire proposed model. I have also worked towards designing the block diagram which allows ease of understanding to the reader.

**Individual contribution for project presentation and demonstration:** I have taken part in explaining the true random seed generator and its application and challenges for implementation in random graph generation model.

Full Signature of Supervisor: Full Signature of Student
……………………………... (Saptangshu Kaviraj)

## Probabilistic Random Graph Generations with Feedback Loops

SANDEEP SUBHANKAR SAHOO
2005328

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** I have contributed towards the research of the randomness that occurs in nature and finding the experiment that we have used. I have helped in implementing it with the design and development of the game. I have been working on using the generated graphs to create an encryption standard.

**Individual contribution to project report preparation:** I have created the gantt chart and have written the feedback loop section along with the literature review section.

**Individual contribution for project presentation and demonstration:** I have discussed the proposed model and the implementation of the algorithm in the game.

Full Signature of Supervisor: ……………………………...

Full Signature of Student (Sandeep S. Sahoo)

## Probabilistic Random Graph Generations with Feedback Loops

SAKSHYA VARDHAN SINGH
2005752

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** Extensive debugging and testing procedures, ensuring the robustness and reliability of the codebase to enhance the overall quality of the software.

**Individual contribution to project report preparation:** The Implementation and Report Structure with Proof reading.

**Individual contribution for project presentation and demonstration:** Discussed future work and development to be done in the project.

Full Signature of Supervisor: ……………………………...

Full Signature of Student
(Sakshya V. Singh)

## Probabilistic Random Graph Generations with Feedback Loops

DEVANSH SRIVASTAVA
2005309

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** I have contributed towards the research of the randomness that occurs in nature and network theory and in the random graph generation and finding the experiment that we have used. Studying the properties of these graphs, such as connectivity, distribution of node degrees, clustering coefficients, and path lengths. This can reveal insights into how feedback loops affect the structure and dynamics of the network.

**Individual contribution to project report preparation:** I have completed the literature survey/review denoting the works which helped in the generation of the graphs and knowing the concepts of feedback loops.

**Individual contribution for project presentation and demonstration:** I have discussed about the literature survey of the subdomains stating the tools used in the implementation.

Full Signature of Supervisor:
……………………………...

Full Signature of Student
(Devansh Srivastava)

## Probabilistic Random Graph Generations with Feedback Loops

MONJIMA MAJUMDAR
2005314

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** In this project I have worked for the research on the properties of true randomness and how they can be generated using a pseudo manner. We have also learned about how various tools were used to finally conclude about the existence of a state of pseudo randomness that can be achieved using them.

**Individual contribution to project report preparation:** I have worked towards the documentation of the introduction of the project. I have also contributed in the research regarding true randomness for the literature survey.

**Individual contribution for project presentation and demonstration:** I have taken part in the basic demonstration and explanation of the introduction and the literature review on true randomness.

Full Signature of Supervisor: ……………………………...

Full Signature of Student
(Monjima Majumdar)

## Probabilistic Random Graph Generations with Feedback Loops

SHIBASISH KAR
2005335

**Abstract:** This project aims to utilize the inherent randomness present in real-world systems to build a secure and highly adaptable networking environment. By incorporating the unpredictability found in natural systems, the goal is to simulate dynamic and resilient graphs with a wide range of capabilities. In essence, the project taps into the chaotic nature of real-world systems to enhance networking capabilities, fostering a secure and expansive networking landscape. The envisioned outcome is a dynamic infrastructure that adapts to the ever-changing challenges in networking, contributing to increased resilience in complex and dynamic environments.

**Individual contribution and findings:** In this project I have worked towards the research on the properties nodes and edges of a graph and how they can be randomly cojoined in an orderly manner.

**Individual contribution to project report preparation:** I have worked towards the proper documentation of the feedback loop used for debugging the graphs.

**Individual contribution for project presentation and demonstration:** I have taken part in the demonstration of the feedback loop part of the presentation.

Full Signature of Supervisor: ……………………………...

Full Signature of Student (Shibasish Kar)

# TURNITIN PLAGIARISM REPORT

A PROJECT REPORT.docx

ORIGINALITY REPORT

| 10% | 7% | 4% | 6% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to National Institute of Technology Warangal<br>Student Paper | 3% |
| 2 | www.coursehero.com<br>Internet Source | 2% |
| 3 | Submitted to University of the West Indies<br>Student Paper | 1% |
| 4 | pdfs.semanticscholar.org<br>Internet Source | <1% |
| 5 | Submitted to University College London<br>Student Paper | <1% |
| 6 | www.allcreaturespethospital.com<br>Internet Source | <1% |
| 7 | www.jetir.org<br>Internet Source | <1% |
| 8 | www.ncbi.nlm.nih.gov<br>Internet Source | <1% |