Exploring Sampling Techniques in Large Graphs and Networks

Anna Myakushina

University of Rochester
Spring 2023 Honors Thesis
Advisor: Alex Iosevich

Abstract

Graphs are an important branch of data science that are essential to visualizing and representing relationships, particularly in spaces with many observations. Working with large graphs (10,000 nodes or more), however, can be extremely difficult- runtime increases dramatically, and results are often difficult to visualize and interpret due to how busy the model becomes. For this reason, efficient and accurate sampling techniques are necessary to scale down data to a manageable magnitude. This study will explore sampling methods of three different classes: random selection techniques, sampling by exploration, and deletion methods. Various existing methods are introduced and explained, and the probability of node selection is calculated for each method in order to compare and discuss advantages and pitfalls of each method. In total 15 existing methods are explored and discussed, including Forest Fire Selection (FFS), a popular method utilizing graph traversal.

The last section of this study then proposes a new sampling technique, Spontaneous Forest Fire Sampling (SFFS), which has been modified from FFS. Both FFS and SFFS are modeled on a Facebook social network at three different forward burning probabilities ($p_f = 0.2, 0.5$, and 0.7) and four different sample sizes (5%, 10%, 15%, and 20% of the population). A modification to both FFS and SFFS to include more edges, named edge restoration, is also proposed and modeled. Finally, SFFS is tested separately at "jumping frequencies" of $N = 1, 2, 3, 5$, and 7. In total, 80 samples are generated, modeled, and compared using Kolmogorov-Smirnov D-statistics. It is found that depending on sampling goals, either edge-restored SFFS or traditional FFS performed the best.

Contents

1 Introduction

1.1 Background and Motivation

Graphs are powerful tools that are regularly used to visualize and learn about spaces, with social networks being a common application. They can be used to present relationships is ways that would otherwise be be infeasible to do given the inherent quantity and complexity of the data, and to detect patterns and groupings that may not be obvious otherwise.

Though graphs are certainly better equipped than text when it comes to representing large, complex relationship data, they still have limitations. A graph with tens or hundreds of thousands of observations would be extremely computationally expensive to visualize and work with- runtime of such operations could extend for hours, if not days. Furthermore, even if such a space is able to be visualized, the results would be useless. With so many observations and connections between them, the generated model would be far too busy to easily observe underlying patterns.

All of these issues are ones that were encountered first-hand in a previous project [10]. The importance and necessity of accurate and representative sample sizes was quickly recognized and explored. This study is an extension of the work started then. It explores and introduces a classification system for existing sampling methods, calculates the probability of node selection in each graph, and finally introduces and tests a new sampling method, as well as proposes a modified usage of an existing one.

1.2 Definitions and Notation

Definition 1.2.1 (Graph) *A graph is a set of points connected by lines. The points are called* ***nodes*** *and the lines are called* ***edges****.*

There are two types of graphs:

Definition 1.2.2 *A* ***directed graph*** *is a graph in which edges have a direction, and are denoted visually with an arrow rather than a line. A graph with undirected edges is called an* ***undirected graph****.*

That is, in a directed graph a node v_1 can be connected to v_2 without v_2 being connected to v_1. In an undirected graph, if v_1 is connected to v_2 then v_2 is connected to v_1 and vice versa. For the sake of simplicity, this study will focus on undirected graphs. In applied settings, graphs are called networks:

Definition 1.2.3 (Network) *A network is a type of graph in which nodes represent observations, and edges represent connections between them. Networks can be directed or undirected.*

Lastly, there is one more term of note:

Definition 1.2.4 (Incidence) *An edge is said to be incident to a node v if v is one of its endpoints.*

The following symbols will be used throughout the paper:

Symbol	**Meaning**
G	The full, unsampled graph
V	The set of all nodes in G
E	The set of all edges in the full graph G
n	$n = \|V\|$; the number of nodes G contains
m	$m = \|E\|$; the number of edges G contains
G'	The graph sample
V'	The set of sampled nodes
E'	The set of sampled edges
p	The percent of V that is desired for G' to retain; ideally $\|V'\| = np$
n'_p	$n'_p = np$, the number of nodes desired to be in the sample graph
$\|A\|$	The number of nodes the graph A has
$\|\|A\|\|$	The number of edges the graph A has
$\deg(v)$	The degree of node v
$G[S]$	The graph induced by the subset of nodes S
$p(v)$	$p(v \in V')$; the probability of the node v being in the sample graph
G'_t	The graph sample at time t
S_t	The set of nodes that gets sampled at time t
V'_t	$\{\cup_{i=1,\ldots,t-1} S_i\}$; The set of all nodes that have been sampled up until time t
V_t	$\{v \in V \| v \notin V'_t\}$; The set of all nodes that have not been sampled up until time t
n_t	$\|V_t\|$; The total number of unsampled nodes at time t
n'_t	$\|V'_t\|$; The total number of sampled nodes at time t
m_t	The number of edges between the unsampled nodes at time t
m'_t	The number of edges already sampled at time t
$p_t(v)$	The probability of the node v getting selected at time t
E_v	The set of edges incident to v
V_e	The set of two nodes that are the endpoints of edge e
V_v	The set of nodes that share an edge with the node v
$e(v_1, v_2)$	The edge connecting nodes v_1 and v_2 (if none exists, $e(v_1, v_2) = \varnothing$)

Table 1: Establishment of Notation.

1.3 Goals and Desired Characteristics

When sampling from large graphs, different methodologies are preferable based on the goals of the sample. Specifically, samples may address sampling from one of two perspectives. The first is with a ***"back-in-time"*** sampling goal. With this approach, the sampled graph attempts to "time travel" and replicate what the original graph looked like at the time that it had n'_p nodes. That is, if $G_{t_{n'_p}}$ denotes the graph G at the time when it had n_p nodes, the goal of the sampled graph $G'_{t_{n'_p}}$ would be to sample n'_p nodes from G in such a way that the result resembles $G_{t_{n'_p}}$ and replicates its topology [6]. Such a goal may be preferable when working with dynamic graphs meant to model changes over time- say, for instance, a graph modeling the growth in popularity of a social media platform over time. Taking a back-in-time sample then would allow one to estimate what connections on the platform looked like at any given point in time. There are some distinct challenges associated with a back-in-time sampling problem. Namely, samples are drawn from G, which is just a final, static view of the space. If information regarding node ages is available, this can be incorporated into the sampling algorithm to generate a more accurate sample [6].

This study, however, will take on a ***"scale-down"*** sampling goal instead. With this approach, the goal is to sample from G to create a sample graph G'_{n_p} with n_p nodes that preserves many of the properties of G, but is of a far smaller magnitude [11]. Essentially, this approach attempts to create a "mini model" of G that contains fewer nodes while preserving its topology. Such a goal may be preferable when working with static graphs containing an unfeasibly large quantity of data. By generated an accurate scale-down sample, a graph is created that can still provide valuable information about the full space while being far easier to work with.

To measure how "accurate" a scale-down sample is, ***degree distributions*** will be compared. Ie, for each graph, the degree of each node will be calculated and then the distributions of node degrees will be analyzed. An accurate sample will preserve the degree distribution of G. To quantify this comparison, Kolmogorov-Smirnov D-statistics will be used, which measures the maximal distance between empirical cumulative distribution function F' of the sampled graph's degree distribution and the cumulative distribution function F of the original graph. It is calculated as follows [12]:

$$D = \max_x\{|F'(x) - F(x)|\}$$

A smaller D-statistic indicates less deviance from the original distribution, suggesting that the sample better replicates it.

2 Existing Sampling Methods

Throughout the literature, a number of different sampling methods for graphs and networks have been proposed. Laregly, these can be broken up into three broad categories : (1) random selection techniques (2) sampling by exploration and (3) deletion methods.

2.1 Random Selection Techniques

Random selection techniques methodically take advantage of randomization to sample nodes, edges, or both in a way that attempts to scale down the size of the graph while preserving its topology. Methods utilizing random selection techniques fall into one of two categories: sampling by node selection, and sampling by edge selection.

2.1.1 Sampling by Node Selection

These methods prioritize node selection in constructing the sample graph. Specifically, they each sample n'_p nodes, so that $|V'| = n'_p$ and $G' = G[V']$. Examples of sampling by node selection include the following:

1. **Random Node (RN) Sampling**: In this method, n'_p nodes are uniformly selected at random:

$$p(v) = \frac{n'_p}{n} = \frac{np}{n} = p$$

   RN sampling was explored in a previous work and is discussed later in the paper (section 3). Though a very intuitive approach, this method can sometimes fail to capture the intricacies of a graph's topology. Other methods have been found to preserve graph structure far better.

2. **Random Degree Node (RDN) Sampling**: Unlike RN sampling, this is an example of a non-uniform sampling method. Rather than assigning each node a uniform probability, each node's selection probability is directly proportional to its degree:

$$p(v) \propto deg(v),\ \forall v \in V$$

   Evidently, this process is biased towards nodes of high degrees, and others have noted that as a result it does poorly with preserving the degree distribution of the original graph [6].

3. **Random PageRank (RPN) Sampling**: Inspired by Google's PageRank© algorithm, this is another non-uniform sampling technique. This method, however, assigns node selection probabilities that are proportional to each node's *PageRank* score $PR(v)$, which ranks nodes based on their relative popularity. This parameter is a function of not just the degree of the node, but of its degree relative to the rest of the nodes in the graph [8]:

$$p \propto PR(v),\ \forall v \in V$$

   PageRank score can be thought of as a measure of where a node lies on the graph's degree distribution, and by extension the probability of it getting randomly selected [1]. Given this adjustment, this algorithm had been found to be less biased towards higher degree nodes [6].

2.1.2 Sampling by Edge Selection

While some of these methods still give consideration to nodes, they are distinct in that all or many of the selections they make originate at the edges of the graph. Examples of sampling by edge selection include the following:

1. **Random Edge (RE) Sampling**: Analogously to RN sampling, this technique uniformly selects k edges at random [6], so that

$$G' = G[V_E], V_E = \cup_{e \in E'} V_e$$

   Samples drawn by RE sampling tend to have large diameters due to being sparsely connected and thus do not preserve graph structure [6]. Furthermore, since

$$p(v) = \frac{k \deg(v)}{m}, \ \forall v \in V$$

RE faces the same issue as RDN in that it is biased towards high degree nodes.

2. **Random Node-Edge (RNE) Sampling**: A variation of RE, this method performs a single step of both RN and RE at each iteration. Each iteration consists of two steps:

   (1) RN: one node v is uniformly picked at random and added to G'

   (2) RE: one edge incident to v is uniformly picked at random, and both the edge and its other endpoint are added to G'

   [6] So if $t = 1$ is the first step, we have that when t is odd we have a step of RN and when t is even we have a step of RE. Since at each step we are either sampling one node, or an edge with one vertex already in V_t', we have that for any t

$$|S_t| = 1$$

   Let $s_t \in S_t$ denote the single element in S_t. Then for any $v \in V$,

$$p_t(v) = \begin{cases} \frac{1}{n}, & t \text{ odd} \\ \frac{1}{|E_{s_{t-1}}|}, & t \text{ even and and } e(v, s_{t-1}) \neq \varnothing \\ 0 & t \text{ even, and } e(v, s_{t-1}) = \varnothing \end{cases}$$

   We can make also modify this algorithm to make it more strict by only choosing from nodes and edges that have not been sampled yet. Then:

$$p_t(v) = \begin{cases} 0, & v \in V_t' \\ \frac{1}{n_t}, & v \notin V_t' \text{ and } t \text{ odd} \\ \frac{1}{|E_{s_{t-1}}|}, & v \notin V_t' \text{ and } t \text{ even and } e(v, s_{t-1}) \neq \varnothing \\ 0 & v \notin V_t', \ t \text{ even, and } e(v, s_{t-1}) = \varnothing \end{cases}$$

   In either case, this process is continued up until $n_p' \leq |V_t'|$. Let t_0 be the stop time. Then

$$G' = (V_{t_0}', E_{V_{t_0}'}), \quad E_{V_{t_0}'} = \{e(s_i, s_{i-1}) | i = 2, ..., t_0 - 1\}$$

   That is, the sample graph is the one consisting of (a) the set of nodes randomly selected during the RN steps, (b) the edges selected during the RE steps, and (c) the nodes connecting to (a) via (b). With the incorporation of RN, higher-node bias is eliminated [6]. However, this method even more so suffers from the issue of sparse connections.

3. **Hybrid (HYB) Sampling**: This unique approach combines RNE and RE. At each step, RNE is used with a set probability P and RE is used with a probability of $1 - P$ [6]. Since RNE is a two-step process, we have that at each t, the type of step that will be done is influenced by the previous two steps:

   (1) If regular RE was done at step $t-1$, then step t will be the RN step of RNE with probability P and another regular RE step with probability $1 - P$

   (2) If the RN step of RNE was done at step $t-1$ (ie step $t-1$ was beginning RNE), then step t will be the RE step of RNE

   (3) If the RE step of RNE was done at step $t-1$ (ie both previous steps were RNE), then step t will be the RN step of RNE with probability P and another RE step with probability $1-P$

   Then

$$p_t(v) = \begin{cases} \frac{P}{n} + \frac{(1-P)\deg(v)}{m} & \text{Previous step was a RE step (either as part of RNE or on its own)} \\ \frac{1}{|E_{s_{t-1}}|}, & \text{Previous step was RN and } e(v, s_{t-1}) \neq \varnothing \\ 0, & \text{Previous step was RN and } e(v, s_{t-1}) = \varnothing \end{cases}$$

   We can again modify this algorithm to make it more strict by only choosing from nodes and edges that have not been sampled yet. For a node v, let

$$Z_v = E_v - \{e(v, s) | s \in V_t'\}$$

   Then with the algorithm modification,

$$p_t(v) = \begin{cases} 0, & v \in V_t' \\ \frac{P}{n_t} + \frac{(1-P)(\deg(v)-|Z_n|)}{m_t} & v \notin V_t' \text{ and previous step was RE (either as part of RNE or on its own)} \\ \frac{1}{|E_{s_{t-1}}|}, & v \notin V_t', \text{ previous step was RN, and } e(v, s_{t-1}) \neq \varnothing \\ 0, & v \notin V_t', \text{ previous step was RN, and } e(v, s_{t-1}) = \varnothing \end{cases}$$

   By alternating between RE and RNE, this method is able to reduce higher-node bias without sacrificing as much in terms of graph sparsity.

2.2 Sampling by Exploration

An alternative to strategic random selection is the utilization of crawling techniques. These can again be split into two broad categories: graph traversal and random walks.

2.2.1 Graph Traversal

Graph traversal methods are designed to select each node at most once in order to prevent the issue of cycling. The order in which they visit each node, however, varies. Some methods include the following:

1. **Random Node Neighbor (RNN)** In this method, at each iteration a node $v \in V_t$ is uniformly selected at random, along with V_v and E_v [6]. This is repeated until the desired sample size is reached. Hence for any $v \in V_t$, we have

$$p_t(v) = \frac{1 + deg(v)}{n_t}$$

   Because a new jump is made at each iteration and only connections with immediate neighbors are preserved, this sampling method does a poor job of preserving community structures [6].

2. **Breadth-Search-First (BSF)** With this sampling method, an initial starting node v_0 is uniformly selected at random. Then all of its immediate neighbors are sampled, and then the immediate neighbors of those nodes are sampled, and so on until the desired number of nodes is reached [4][5]. In this case, the probability of a node getting sampled at time t is 1 if it has an edge incident to a node in S_{t-1}, and 0 otherwise. Because node selection relies so heavily on proximity to previously sampled nodes, this method works well for graphs in which all nodes are closely connected, but not ones with distinct clusters.

3. **Depth-Search-First (DSF)** This sampling method, on the other hand, uniformly selects a starting node v_0 at random, then uniformly selects one of its incident edges at random, as well as the other endpoint of that edge. Then an edge from that node is uniformly selected at random, as well as the other endpoint of that new edge. This process is then repeated until a node is reached that has no edges connecting to upsampled nodes, in which case a new starting point is chosen (revisitation is allowed when selecting a new starting node, but not at other steps in the process) [4][5]. This process is continued until the desired number of nodes has been reached. If v_t is the node that is sampled at time t, then for $v \in V_t$

$$p_t(v) = \begin{cases} \frac{1}{n}, & deg(v_{t-1}) = 0 \\ \frac{1}{deg(v_{t-1})}, & deg(v_{t-1}) > 0 \text{ and } e(v, v_{t-1}) \neq \varnothing \\ 0, & deg(v_{t-1}) > 0 \text{ and } e(v, v_{t-1}) = \varnothing \end{cases}$$

   In this case, $deg(v)$ is updated at each time t to include only edges incident to other upsampled nodes in degree calculations. This method, unlike BSF, does better with more sparsely connected graphs and those with distinct clustering.

Forest Fire Sampling (FFS), another example of sampling by exploration through graph traversal, is later presented in great detail (section 4). Furthermore, a modified approach, named Spontaneous

Forest Fire Sampling (SFFS) is proposed and explored later (section 4) as well.

2.2.2 Random Walks

While graph traversal visits each node no more than once, revisitation is permitted in random walks [3]. Some methods include the following:

1. **Random Walk (RW) Sampling**: In this sampling method, a starting node v_0 is picked uniformly at random, and then a random walk on the graph is simulated. A "fly back" probability c (often, $c = 0.15$) is selected, and at every step the algorithm returns back to the starting node and restarts the random walk with probability c. Let s_t denote the node selected at time t, and s_0 the starting node [6]. For a simple random walk, then we have that for any $v \in V$,

$$p_t(v) = \begin{cases} c, & v = s_0 \text{ and } e(v, s_{t-1}) = \varnothing \\ c + \frac{1-c}{|E_{s_{t-1}}|}, & v = s_0 \text{ and } e(v, s_{t-1}) \neq \varnothing \\ 0, & v = s_{t-1} \neq s_0 \\ 0, & v \neq s_0, s_{t-1} \text{ and } e(v, s_{t-1}) = \varnothing \\ \frac{1-c}{|E_{s_{t-1}}|}, & v \neq s_0, s_{t-1} \text{ and } e(v, s_{t-1}) \neq \varnothing \end{cases}$$

This process is continued until $|V'_t| = n'_p$.

One issue that frequently arrises with RW sampling is that the algorithm may get "stuck"- if the starting node happens be part of an isolated portion of the graph, it is unable to travel to the rest of the graph and therefore cannot gather a representative sample. To get around this issue, the algorithm can be slightly modified. If after a certain large number (say, $100 \cdot n$) of steps $V'_t < n'_p$, a new starting node is uniformly selected at random and the process is repeated.

2. **Random Jump (RJ) Sampling**: RJ sampling is very similar to RW but instead of c being the probability of jumping back to the starting node, it is the probability of jumping to any new random node [6]. So with probability $1 - c$ the random walk continues for another step, and with probability c a new node is uniformly selected at random and jumped to. Let s_t denote the node selected at time t. Then for any $v \in V$,

$$p_t(v) = \begin{cases} c, & v = s_{t-1} \\ c, & v \neq s_{t-1} \text{ and } e(v, s_{t-1}) = \varnothing \\ c + \frac{1-c}{|E_{s_{t-1}}|}, & v \neq s_{t-1} \text{ and } e(v, s_{t-1}) \neq \varnothing \end{cases}$$

RJ sampling was explored in a previous work and is discussed later in the paper (section 3).

2.3 Deletion Methods

Both random selection and sampling by exploration begin from a null set and build up to the desired number of nodes. Deletion methods, on the other hand, begin with the full graph and work backwards to iteratively reduce the number of nodes in the sample graph until the desired number is reached. For this section, $p_t(v)$ shall denote the probability of "surviving" deletion and staying in the sample graph, ie of NOT getting selected.

1. **Deletion of Random Node (DRN)** Analogous to RN sampling. Instead of randomly choosing n'_p nodes to add to G', however, DRN uniformly selects nodes at random for removal one at a time [5]. Thus we have that for any $v \in V_t$ at time t, the probability of not getting removed (and thereby staying the sample) is

$$p_t(v) = 1 - \frac{1}{n_t} = \frac{n_t - 1}{n_t}$$

   Normally this is done in iterations, with some predetermined proportion d of the remaining nodes being deleted one at a time each iteration [5]. The probability of a node remaining after an iteration then is $1 - d$.

2. **Deletion of Random Edge (DRE)** Analogous to RE sampling. Edges are chosen uniformly at random and deleted one at a time. If this results in any isolated nodes, they get deleted as well. This process is then repeated until the desired sample size is reached. For any $v \in V_t$ then we have

$$p_t(v) = \begin{cases} 0, & \deg(v) > 1 \\ 1 - \frac{1}{m_t} = \frac{m_t - 1}{m_t}, & \deg(v) = 1 \end{cases}$$

   Again, this is normally done in iterations, with some predetermined portion d of the remaining edges being deleted at each iteration.

3. **Deletion of Random Node/Edge (DRNE)** Analogous to RNE sampling. A node s_t is selected uniformly at random, and then one of its incident edges is uniformly selected at random and removed. If this results in isolated points, isolated points get removed. Hence if $t = 1$ is the time at which the first DRN is performed, we have that DRNE will perform a step of DRN if t is odd and a step of DRE if t is even. If t is odd, let s_t be the node that is selected and deleted at time t. Then

$$p_t(v) = \begin{cases} \frac{n_t - 1}{n_t}, & t \text{ odd} \\ 1 - \frac{1}{|E_{s_t}|} = \frac{|E_{s_{t-1}}| - 1}{|E_{s_t}|}, & t \text{ even}, \ \deg(v) = 1, \ e(v, s_{t-1}) \neq \varnothing \\ 0, & \text{All else} \end{cases}$$

3 Previous Work

3.1 Motivation

The contents of this paper are an extension of a project that was completed in December, 2022 titled *Sampling Methods for Social Networks with Applications in Spotify Latent Space Models* [10]. The original goal of this project was to learn about modeling social networks in a Bayesian setting, but a significant issue emerged almost right away: there was simply too much data to be able to reasonably work with or extract the desired information from. This resulted in a pivot in research goals, and the paper instead became an exploration of sampling techniques in the context of large graphs. This original project was merely a first introduction into the field and of a far smaller magnitude than this work; this paper has greatly expanded upon what was originally done. Nonetheless, the original findings are not only relevant to the subject of this paper, but they were also a significant guide during the research process. This paper would not be complete without those previous findings, and so this section has been included

Because the original project was intended as a first introduction into the field, only two sampling techniques were explored: Random Node (RN) sampling (see section 2.11) and Random Jump (RN) sampling (see section 2.2.2). Each method, however, was performed at four different values of p: 0.10, 0.15, 0.25, and 0.50. These methods were applied to a social network modeling artists on Spotify and their collaboration, where each node was an artist, and edges between artists represented collaborations. The goal was to model this space and attempt to identify clusters based on genre and/or country.

3.2 Summary of Original Study

3.2.1 Data

Data was obtained through Kaggle [2]. This graph was undirected and consisted of 156,422 artists, as well as all of their collaboration between September 28th, 2013 and October 9th, 2022. For the sake of computation, the graph was restricted to include only artists with popularity in the United States. This brought the number of nodes down to 1,185, with 46,933 edges between them. The remaining artists covered a total of 559 genres. As this would not be feasible to visualize, the data was further restricted to only include artists that fell into one of the following three genres: rock, rap, and/or indie. The resulting subset consisted of 614 nodes, with 27,462 edges between them. For a more detailed explanation of the data cleaning process, see [10].

Once the data was cleaned, a 614x614 adjacency matrix was generated. Each row-column pair represented a pair of artists- if those artists had at least one collaboration, a 1 was entered. If not, 0. Once this was done, all rows and columns with entirely zero entries were removed- these signified artists with no collaborations with anyone else in the subset, and thus would have been isolated points in the graph. Once this was completed, there were 470 artists left: 87 in rock, 354 in rap, and 29 in indie.

3.2.2 Methods

Before sampling, the built-in ergmm() function in R was used to perform latent space modeling on the all 470 artists. This was done to provide a point of reference and determine what the network looked like and what patterns sampled spaces should be mirroring. Modeling was done twice: once from the frequentist perspective using MLE, and a second time from a Bayesian perspective using posterior means as point estimates.

Four samples were drawn using each method and the following sample sizes: 47, 71, 118, and 253 (10%, 15%, 25%, and 50% of $N = 470$). Each of the eight sample spaces was then modeled twice using ergmm() in R, from both a frequentist and Bayesian perspective. When plotting the 50% RN sample, it was found that a few isolated points were skewing the entire graph, so these were removed and the space was modeled again. All models were compared to the full space to assess the graph topologies. Lastly, Kolmogorov-Smirnov D-statistics were calculated for each sample space to compare their degree distribution to that of the full space.

3.2.3 Visualization, Evaluation, and Conclusions

Modeling the full space (Figure 1) did not reveal any significant clustering, but color coding artists by genre revealed that most of the rap artists tended to be positioned towards the center, while rock and indie artists were on the perimeter of the graph.

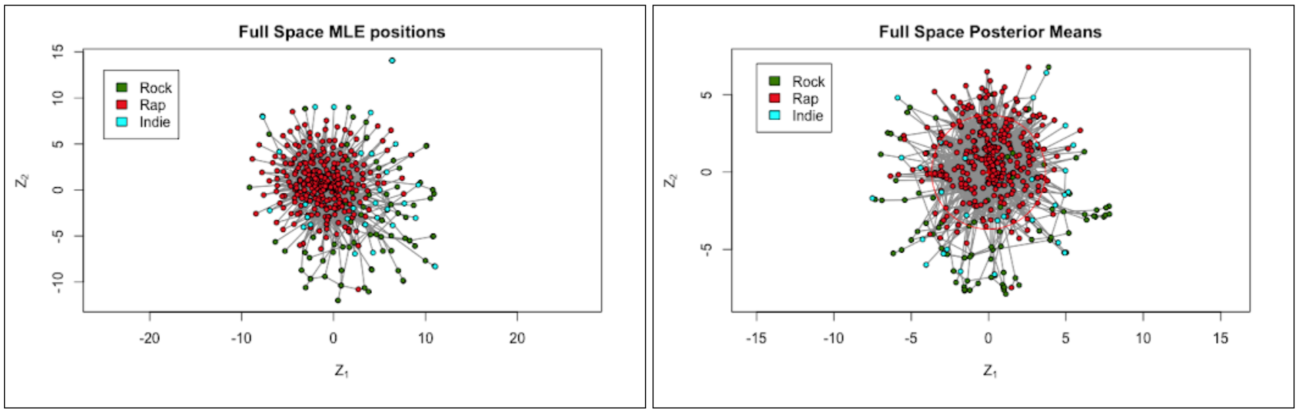
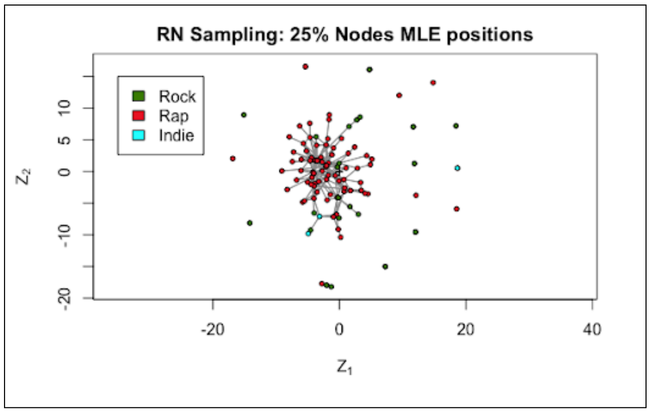


Figure 1: Full Space in Latent Space from both a frequentist and Bayesian perspective

In general, it was found that a 10-15% sample size was not sufficient for either method to properly preserve graph structures. Both methods did well at 25-50%, with RN slightly outperforming RJ The previously observed pattern of rap artists being located near the center while rock and indie artists disperse on the outside began to emerge in these samples (Figure 2). D-statistics supported these findings, with D improving as sample size increased and overall being better with RN than RJ (Table 2).



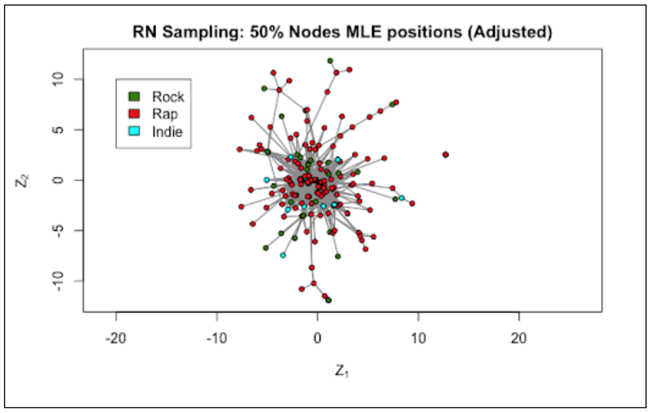


Figure 2: 25% and 50% adjusted sample using RN sampling

		Sample Size (Value of p)			
		0.10	**0.15**	**0.25**	**0.50**
Sampling Technique	Random Node (RN)	0.5957	0.5429	0.4065	0.1851
	RN (adjusted)				0.1494
	Random Jump (RJ)	0.7197	0.7337	0.6995	0.6122

Table 2: Table of Kolmogorov-Smirnov D-Statistics for each sample

Low D-statistics and poor graph structure preservation were likely due to an insufficient population size. The fact that only at 50% did the sample perform well suggests that sampling smaller graphs may be pointless, as very large proportions of the data need to be sampled in order to to generate accurate representations of the full space. Because of this, a much larger space was sampled from later in this study (section 4).

4 Spontaneous Forest Fire Sampling (SFFS)

4.1 Forest Fire Sampling (FFS)

The Forest Fire graph generation model [4] was first introduced by Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, and was then later adapted by Leskovec and Faloutsos into a sampling technique [7]. Forest Fire Sampling (FFS) is graph traversing exploration technique, and the algorithm is as follows:

1. From the graph G, uniformly select a node v at random
2. Randomly select a number x from the geometric distribution with mean $\frac{p_f}{1-p_f}$, where the parameter p_f denotes the forward burning probability.
3. Let W denote the set of edges connected to v that have not yet been traversed, and let

$$n_W = \min(x, |W|)$$

Uniformly selected n_W edges in W at random, and let $w_1, w_2, ..., w_{n_W}$ denote the nodes at the other ends of these selected edges.

4. Repeat steps (2) and (3) recursively at each $w_1, w_2, ..., w_{n_W}$ until the desired number of nodes have been sampled. To avoid cycling, nodes are visited at most once- that is, in these subsequent steps only edges connected to previously upsampled nodes are drawn from. If at some time t, $W = \varnothing$ for each "end node" w_i (also known as the fire dying) before the desired number of nodes have been sampled, restart the process by uniformly selecting a new node v at random from the nodes that have not yet been sampled [6].

Note that for the purposes of this study, the FFS algorithm has been modified from that proposed by Leskovec and Faloutsos in order to handle cases when $x > |W|$. Further modifications can be made to accommodate directed graphs by including a second parameter p_b, the backward burning probability, but this modification will not be explored as the focus of this study is on undirected graphs.

Define $Z_{vt} = \{e(v, s)|s \in S_{t-1}\}$ for a node v. Let

$$Y_t = [\cup_{s \in S_t} E_s] - [\cup_{s \in S_t} E_s] \cap [\cup_{w \in V'_t} E_w]$$

That is, let Y_t be the set of edges connecting to the nodes that is sampled at time t whose other endpoint has not been sampled already. Then at time t, for any $v \in V_t \backslash V'_t$:

$$p_t(v) = \begin{cases} \frac{1}{n_t}, & \text{If a new fire is started at } t \\ 0, & \text{If an existing fire is continued at } t \text{ and } Z_{vt} = \varnothing \\ \frac{|Z_{vt}|}{|Y_{t-1}|}, & \text{If an existing fire is continued at } t \text{ and } Z_{vt} \neq \varnothing \end{cases}$$

Since $|Z_{vt}|$ is related to $deg(v)$ (that is, high degree nodes are more likely to have edges connecting them to a node sampled in the previous step) this algorithm is biased towards higher degree nodes. Furthermore, if the graph has very isolated distinct groups, FFS may get "stuck" at one group and fail to visit the other(s).

4.2 Modifying FFS: Spontaneous Forest Fire Sampling (SFFS)

In an attempt to address these issues, in this paper I propose a modified version of FFS, which I shall refer to as Spontaneous Forest Fire Sampling (SFFS). The algorithm is as follows:

1. From the graph G, uniformly select a node v at random

2. Randomly select a number x from the geometric distribution with mean $\frac{p_f}{1-p_f}$, where the parameter p_f denotes the forward burning probability.

3. Let W denote the set of edges connected to v that have not yet been traversed, and let

$$n_W = \min(x, |W|)$$

Uniformly selected n_W edges in W at random, and let $w_1, w_2, ..., w_{n_W}$ denote the nodes at the other ends of these selected edges.

4. Repeat steps (2) and (3) recursively at each $w_1, w_2, ..., w_{n_W}$ N times, where N is a predetermined, fixed number of iterations (henceforth referred to as the "jumping frequency"). To avoid cycling, nodes are visited at most once- that is, in these subsequent steps only edges connected to previously upsampled nodes are drawn from. If at some step the fire dies before the Nth iteration, go to step (5).

5. If at this point $V_t' < n_p$, uniformly selecting a new node v at random from V_t and repeat steps (2)-(4) until $V_t' \geq n_p$

SFFS forces a new jump more often than FFS, which prevents the algorithm from getting "stuck" in any one part of the graph and encourages exploration into other areas. This allows the sample to capture more clusters that may otherwise be missed or underrepresented. As in FFS, for any $v \in V_t \backslash V_t'$

$$p_t(v) = \begin{cases} \frac{1}{n_t}, & \text{If a new fire is started at } t \\ 0, & \text{If an existing fire is continued at } t \text{ and } Z_{vt} = \varnothing \\ \frac{|Z_{vt}|}{|Y_{t-1}|}, & \text{If an existing fire is continued at } t \text{ and } Z_{vt} \neq \varnothing \end{cases}$$

However since a new fire is started more frequently, there is less bias towards high degree and closer-connected nodes.

4.3 Modeling

4.3.1 Data

A dataset containing node and edge information of a Facebook social network was obtained through Kaggle [11] and used for the purposes of demonstration and exploration. It consisted of two columns containing node identification numbers. Each row in the dataset represented an edge between the node listed in the first and second column. In total, the dataset contained 22,470 nodes with 171,002 edges connecting them.

Each node in the dataset represented a verified Facebook page, and each edge represented a mutual like between two pages. All information was collected in November 2017 by the author of the dataset using the Facebook Graph API. To ensure a manageable set of data, the author of the dataset chose to include only pages falling under one of the following four groups pre-defined and assigned by Facebook in data collection: politicians, governmental organizations, television shows, and companies.

The full space was plotted (Figure 3) in order to have a point of reference to compare sample models to.

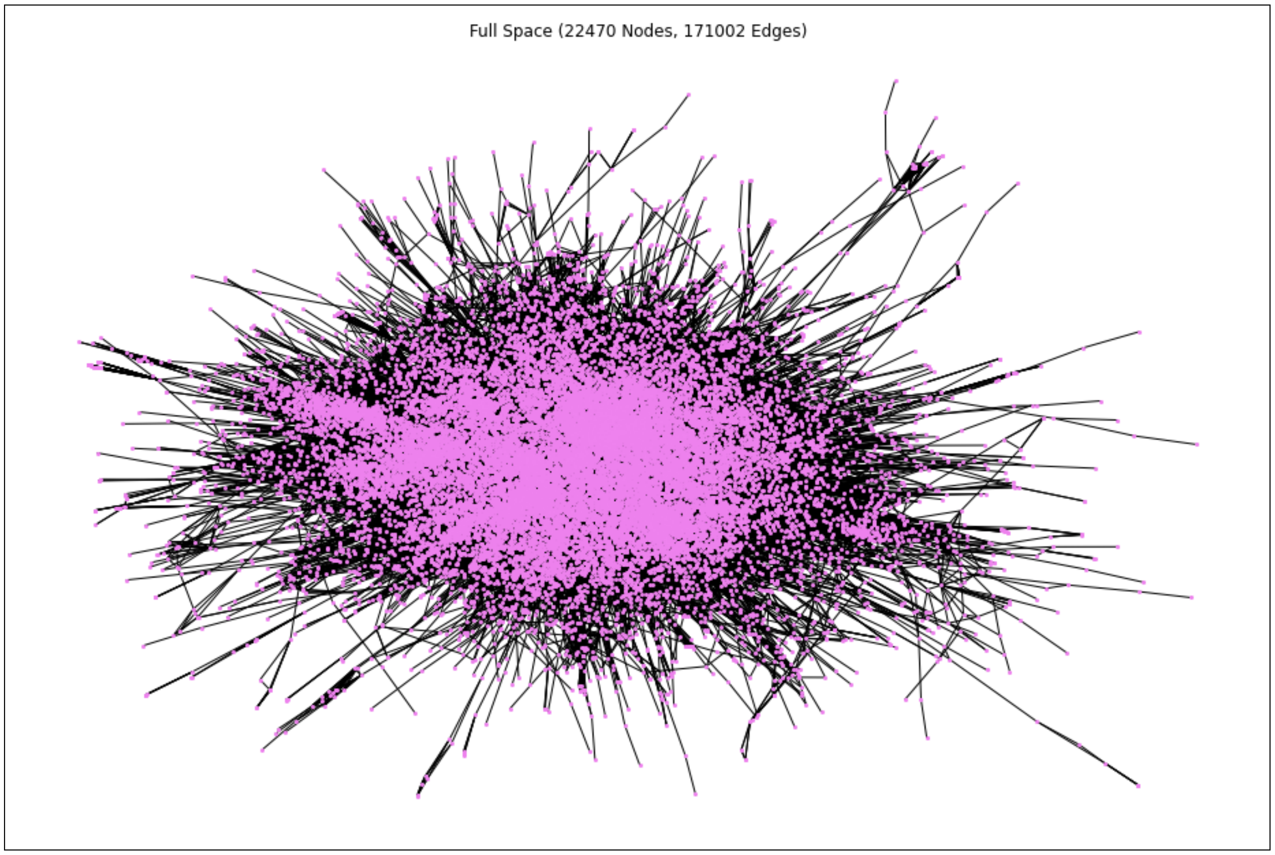


Figure 3: The entire graph before sampling.

4.3.2 Methods and Evaluation

Both FFS and and SFFS were modeled, analyzed, and compared in this study. Both FFS and SFFS were performed on the entire dataset at 4 different sample size (5%, 10%, 15%, and 20% of the population, resulting in sample sizes of n'_p = 1124, 2247, 3370, and 4494, respectively) and 3 different forward burning probabilities (p_f = 0.2, 0.5, and 0.7) for a total of 12 samples per method, 24 samples in total. Sample sizes were chosen in light of both the previous work and in consultation with the literature. Forward burning probabilities were selected based on [6], in which it was found that 0.2 worked well for back-in-time purposes and 0.7 worked well for scaling down goals. 0.5 was chosen both because it lies between these two values, and for its unbiasedness.

After using the sampling algorithms in the traditional way, they were also used exclusively as node selection technique. For this approach, the sample graph was that induced by the nodes selected in the original approach. This was again done 12 times for each algorithm (at the previous 4 different sample sizes and 3 different forward burning probabilities each) resulting in a total of 24 new samples. These methods were named Restored FFS and Restored SFFS for the way that they "restore" edges between nodes that were previously left out.

Finally, SFFS was analyzed separately at four additional "jumping frequencies" by setting $N = 1, 2, 5,$ and 7. It was found that beyond $N = 7$, SSFS was virtually identical to FFS. For this section of the analysis, the forward burning probability was fixed at $p_f = 0.7$ and again sample sizes of 5%, 10%, 15% and 20% of the population were used to obtain 16 new samples. The edge-restored version of SFFS was also performed for each of these samples, resulting in 16 more samples.

In total, 80 samples were drawn. For each sample, runtimes, the number of jumps made, node and edge proportion reduction, and Kolmogorov-Smirnov D-statistics were calculated. Each sample was also visualized in order to compare to the topology of the full space.

4.4 Results

4.4.1 Comparing FFS and SFFS

First, FFS and SFFS were both performed and visualized (Figures 4 and 5). Output for both can be found in Appendix A and B, respectively. Kolmogorov-Smirnov D-statistics, however, are summarized in Table 3 below.

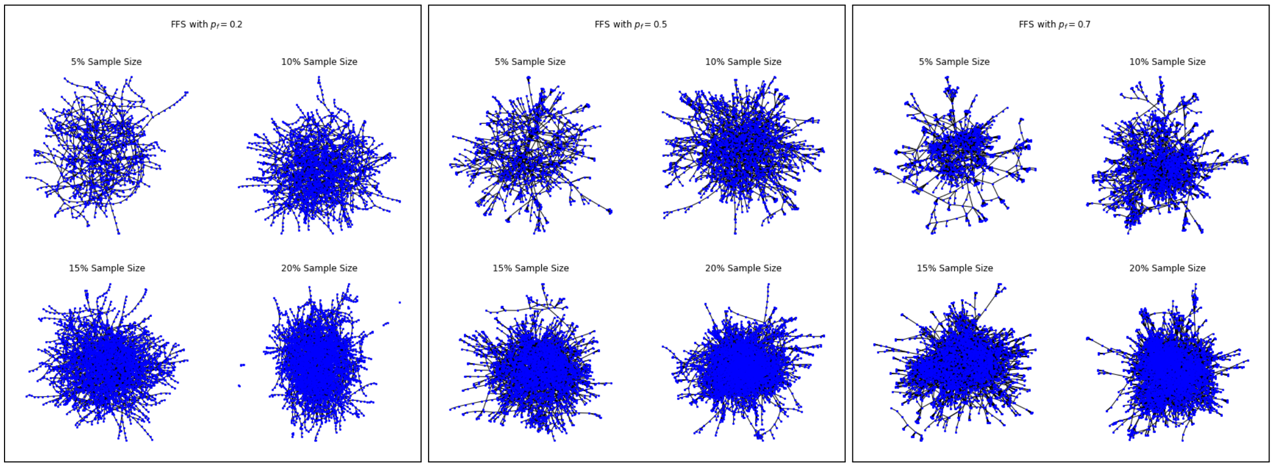


Figure 4: FFS at forward burning probabilities $p_f = 0.2, 0.5, 0.7$

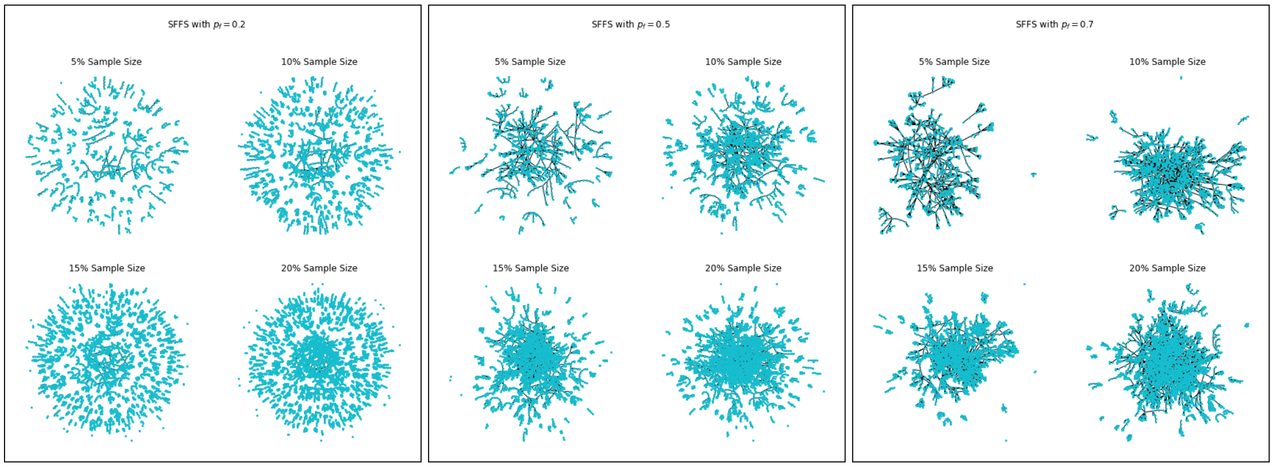
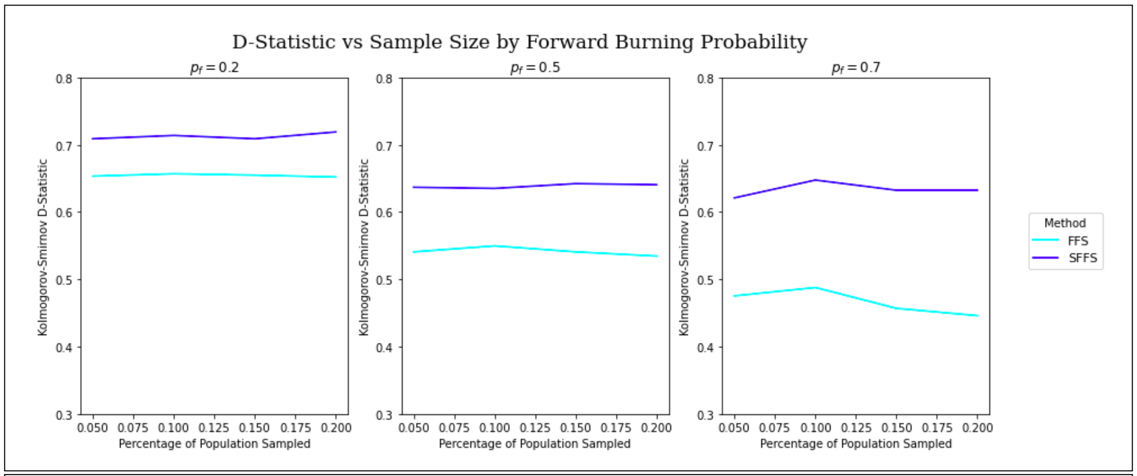


Figure 5: SFFS at forward burning probabilities $p_f = 0.2, 0.5, 0.7$

Percentage of Popuation Sampled	Sampling Method					
	FFS			SFFS		
	$p_f = 0.2$	$p_f = 0.5$	$p_f = 0.7$	$p_f = 0.2$	$p_f = 0.5$	$p_f = 0.7$
5%	0.6533	0.5408	0.4754	0.6702	0.5911	0.5190
10%	0.6569	0.5497	0.4878	0.6720	0.5772	0.5265
15%	0.6548	0.5408	0.4568	0.6696	0.5824	0.5215
20%	0.6520	0.5345	0.4460	0.6716	0.5770	0.5178

Table 3: Table of Kolmogorov-Smirnov D-Statistics for each sample.

In general, FFS slightly outperformed SFFS giving slightly lower D-statistics overall (Figure 6, top). SFFS appeared to be more sparsely connected than FFS at $p_f = 0.2$, though this difference was less noticeable after increasing p_f. FFS preserved the topology of the full space for all 12 samples, whereas SFFS did so for samples using $p_f = 0.5$ and 0.7. This could be explained by the fact that SFFS produced approximately 15.02% less edges, on average, than its corresponding FFS sample. One advantage of SFFS, however, is that for $p_f = 0.2$ and 0.5, it outperformed FFS in terms of runtime, suggesting that this algorithm is more efficient (Figure 6, bottom).



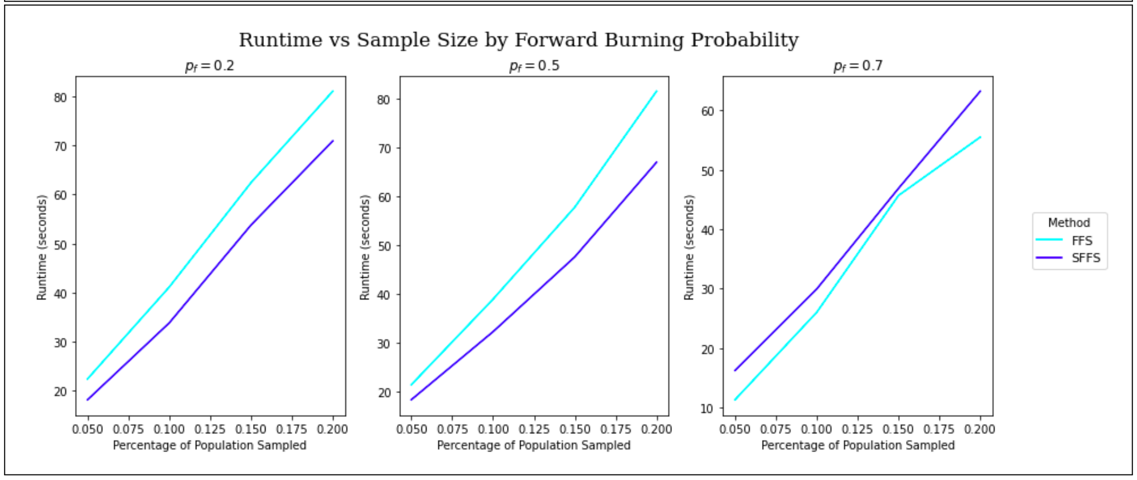


Figure 6: Comparison of FFS and SFFS by their D-statistics (top) and runtime (bottom).

4.4.2 FFS and SFFS as Node Selection Techniques

Next, edge-restoration was done for all 24 of the FFS and SFFS samples from section 4.4.1, and the resulting samples were visualized (Figures 7 and 8). Output for both can be found in Appendix C and D, respectively. Kolmogorov-Smirnov D-statistics, however, are summarized in Table 4 below.

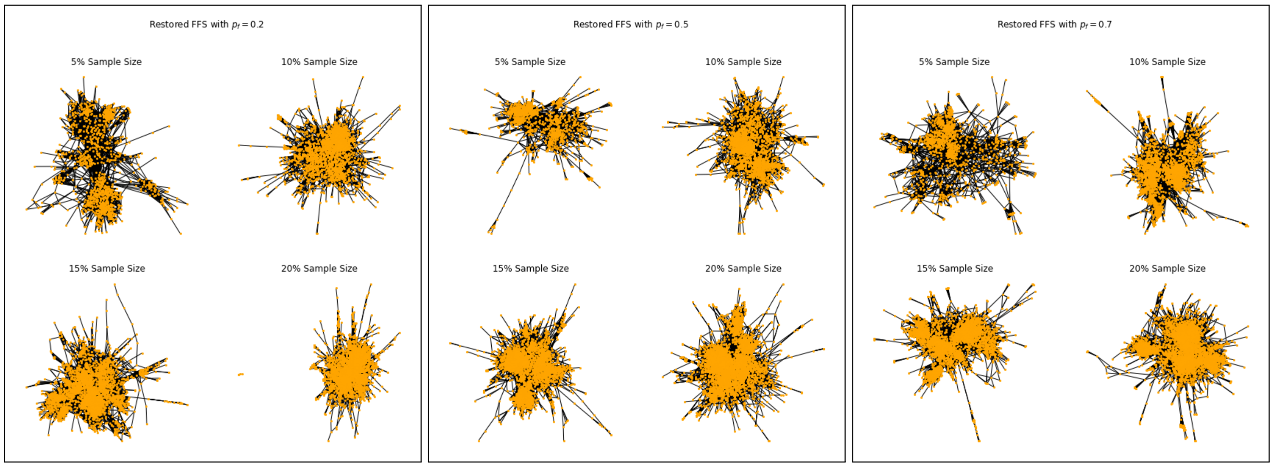


Figure 7: Edge-Restored FFS at forward burning probabilities $p_f = 0.2, 0.5, 0.7$

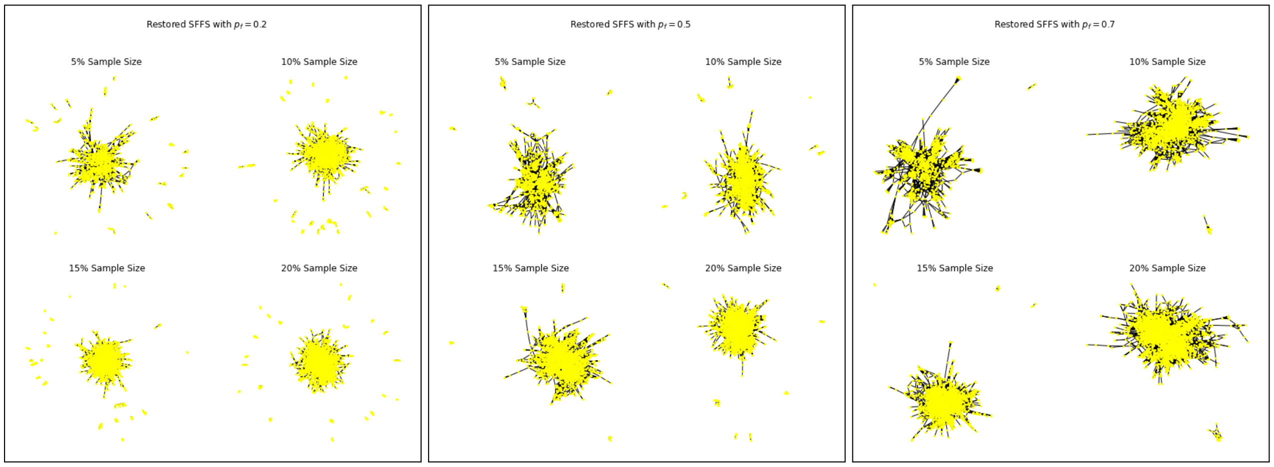


Figure 8: Edge-restored SFFS at forward burning probabilities $p_f = 0.2, 0.5, 0.7$

Percentage of Popuation Sampled	Sampling Method					
	FFS (Restored)			SFFS (Restored)		
	$p_f = 0.2$	$p_f = 0.5$	$p_f = 0.7$	$p_f = 0.2$	$p_f = 0.5$	$p_f = 0.7$
5%	0.2599	0.1698	0.1342	0.2038	0.1015	0.1582
10%	0.1775	0.1868	0.2426	0.1194	0.0708	0.1332
15%	0.2512	0.3035	0.2474	0.0483	0.1087	0.1699
20%	0.2191	0.2460	0.2626	0.0619	0.1216	0.1826

Table 4: Table of Kolmogorov-Smirnov D-Statistics for each edge-restored sample.

On average, restoring edges increased the number of edges in the sample by 89.79% in the FFS samples and by 85.93% in the SFFS samples. Interestingly, restoring the edges appeared to preserve degree distribution far better than the standard algorithm, as D-scores were significantly lower across the board. This suggests that FFS and SFFS have the issue of not sampling enough edges. Also of note, restored SFFS outperformed FFS in most cases after restoring edges. These results are summarized in Figure 9.

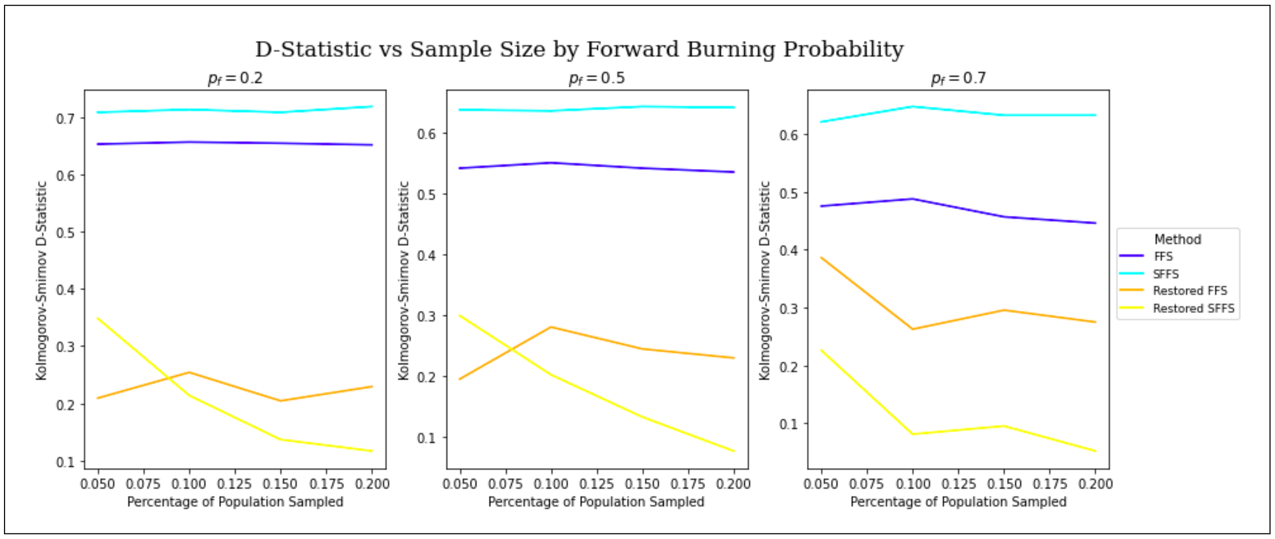


Figure 9: FFS and SFFS D-statistics before and after edge restoration.

For sample sizes of 10% of the population or greater, restored SFFS appears to be the best method, both in terms of runtime and preserving degree distribution. This, however, comes at the cost of more preserving more edges, and thus having a larger dataset to work with and process. If edge restoration is not an option (say, if constraints are placed on the size of the sampled dataset) FFS would be the preferred method, though the sampling distribution will be far less accurate.

4.4.3 SFFS at Different Jump Frequencies

Finally, SFFS was performed at various values of N to assess how jump frequency impacts SFFS performance. The forward-burning probability was fixed at $p_f = 0.7$, and then samples were drawn for $N = 1, 2, 5$, and 7. Results for $N = 3$ were taken from sections 4.4.1 and 4.4.2. All samples were visualized (Figures 10 and 11). Output for both the traditional sampling and edge-restored sampling can be found in Appendixes E and F, respectively. Kolmogorov-Smirnov D-statistics, however, are summarized in Table 5 below.

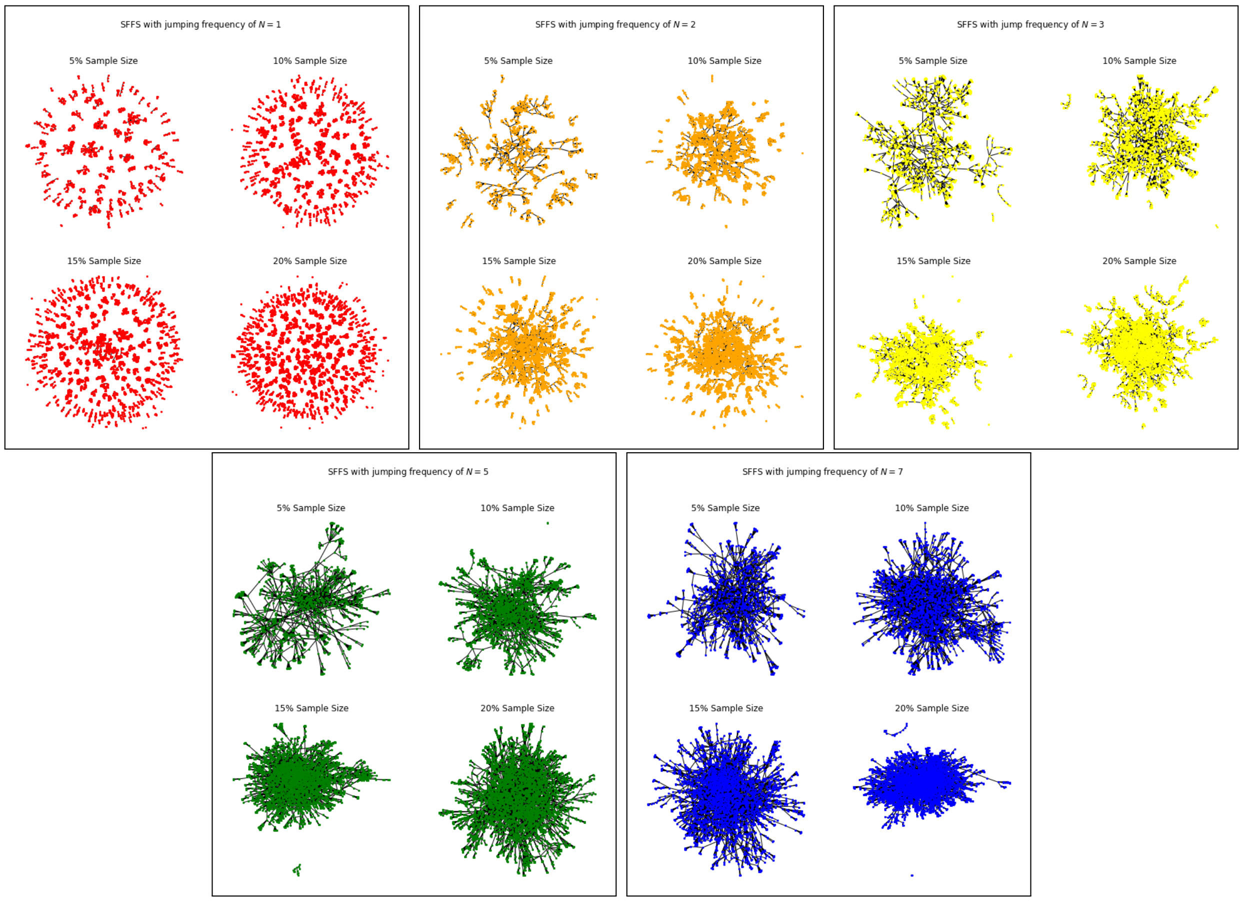


Figure 10: SFFS at jump frequencies $N = 1, 2, 3, 5, 7$.

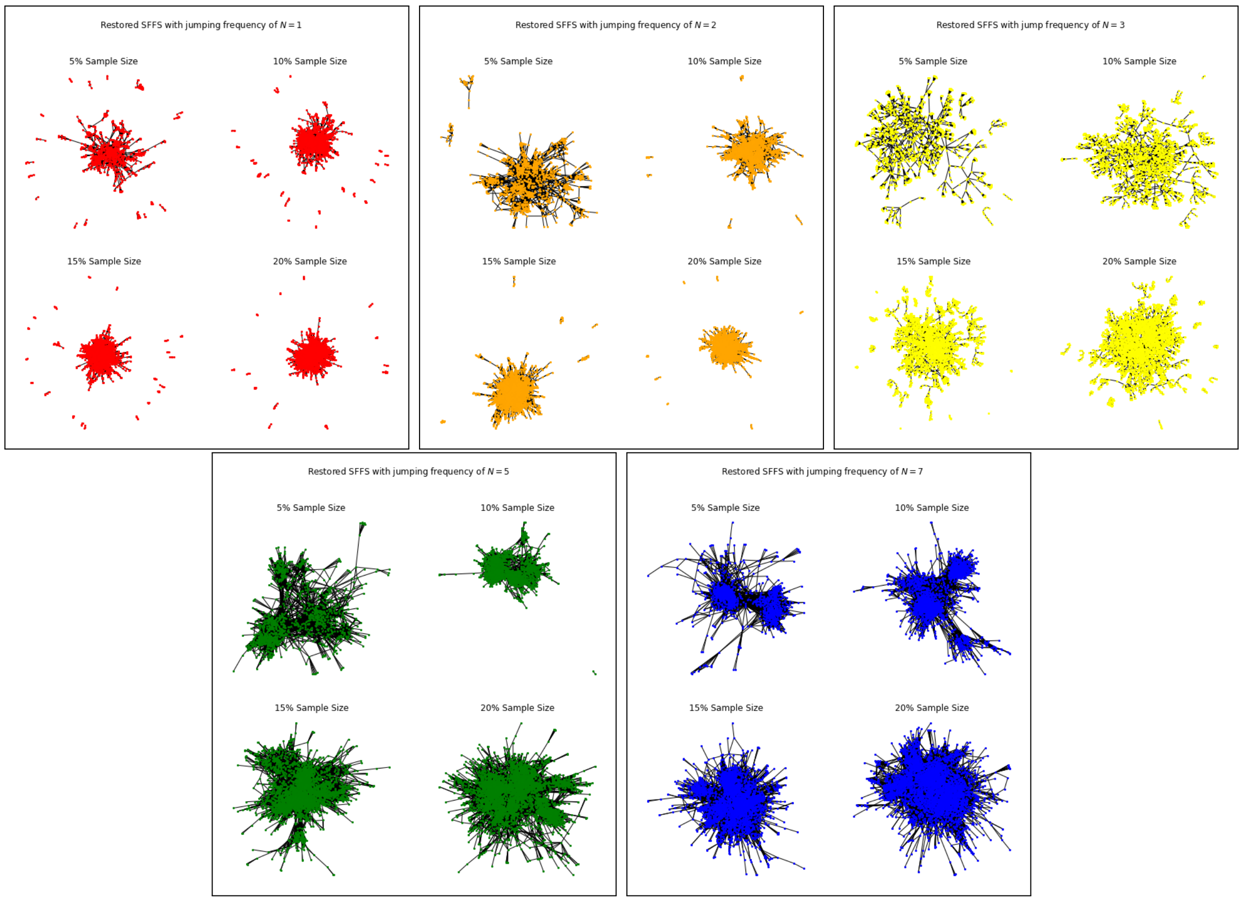


Figure 11: Edge-restored SFFS at jump frequencies $N = 1, 2, 3, 5, 7$.

Percentage of Popuation Sampled	**Sampling Method**									
	SFFS					**SFFS (Restored)**				
	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 1$	$N = 2$	$N = 3$	$N = 5$	$N = 7$
5%	0.5791	0.5186	0.5190	0.4785	0.4732	0.1518	0.0575	0.1582	0.1440	0.3364
10%	0.5794	0.5331	0.5265	0.5011	0.4749	0.0858	0.0805	0.1332	0.1895	0.2043
15%	0.5787	0.5358	0.5215	0.4799	0.4940	0.0633	0.1207	0.1699	0.1963	0.2260
20%	0.5797	0.5305	0.5178	0.5032	0.4880	0.0852	0.1368	0.1826	0.1626	0.1383

Table 5: Table of D-Statistics for SFFS and Restored SFFS samples at various jump frequencies.

Both SFFS and restored SFFS did a poor job of preserving graph structure for $N = 1$ and 2, as these graphs appeared to be sparesly connected. $N = 3, 5,$ and 7 appeared to be sufficient. Again, restored SFFS outperformed SFFS in terms of D-statistic. Increasing N appeared to have opposite effects on restored vs traditional SFFS. As N increased, traditional SFFS D-statistics decreased, while restored SFFS D-statistics increased (Figure 12). This is likely explained by the fact that traditional SFFS does not sample enough edges.

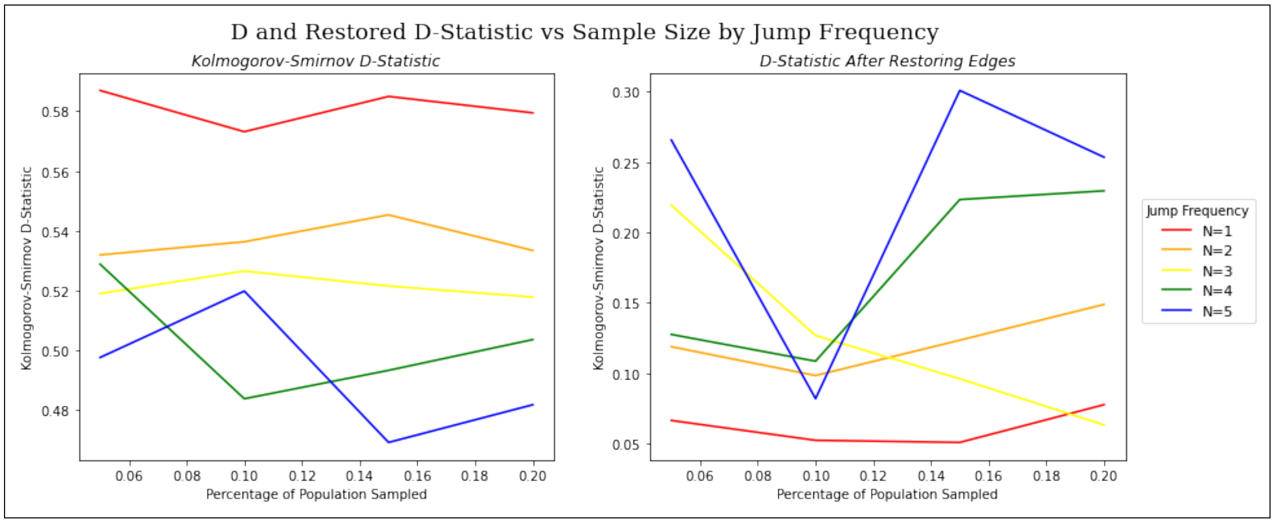


Figure 12: SFFS and edge-restore SFFS performance at various jump frequencies.

5 Concluding Remarks

Many different sampling methods have been introduced in the literature in order to address both the differing sampling goals one might have, as well as the specific topological characteristics of each individual graph. While a number of these have been introduced in this study, there are far more that exist within each of the three categories discussed. Analyzing node selection probabilities of each method provides an understanding of the merits and challenges of each method, and thus can help in selecting an appropriate sampling method. What works well for one graph may not work as well for another, and so there is no one universal ”ideal” method- in each case, it is best to attempt a number of methods, and then select the most representative sample.

This study also introduced Spontaneous Forest Fire sampling (SFFS), a new modification to traditional Forest Fire Sampling (FFS). While, for the particular graph analyzed in this study, traditional FFS outperformed SFFS, utilizing these graph traversal method for node selection, and then inducing a graph on the selected nodes, proved to be more efficient when utilizing SFFS as opposed to FFS. This increased accuracy, however, came at the cost of a larger dataset storing more edges, and so which of these methods (traditional FFS or edge-restored SFFS) one should choose depends on whether there are any predetermined constraints on the sample dataset.

This study could be continued in a number of ways. Both FFS and SFFS could be performed on other graphs with significantly different topologies to see how performance varies. Specifically, it is expected that SFFS would perform better on a graph with distinct clusters present, as the increased jump probability allows for a broader exploration of the space that does not rely on proximity to previously sampled nodes. Other modifications to FFS could also be introduced, such as incorporating deletion at every other step, to see if performance could be further improved. To better understand how these modifications might aid in increasing sample accuracy, random selection methods and their deletion method analogues could first be be modeled and compared.

References

[1] Brin, S., & Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. http://infolab.stanford.edu/ backrub/google.html

[2] Freyberg, J. Spotify Artist Feature Collaboration Network. https://www.kaggle.com/datasets/jfreyberg/spotify-artist-feature-collaboration-network?select=nodes.csv

[3] Gjoka, M., Kurant, M., Butts, C. T., & Markopoulou, A. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. https://ieeexplore.ieee.org/document/5462078

[4] Hu, P., & Lau, W. C. A Survey and Taxonomy of Graph Sampling. https://arxiv.org/pdf/1308.5865.pdf

[5] Krishnamurthy, V., Faloutsos, M., Chrobak, M., Lao, L., Cui, J.-H., & Percus, A. G. Reducing Large Internet Topologies for Faster Simulations. http://www.cs.ucr.edu/ michalis/PAPERS/sampling-networking-05.pdf

[6] Leskovec, J., & Faloutsos, C. Sampling from Large Graphs. https://cs.stanford.edu/people/jure/pubs/sampling-kdd06.pdf

[7] Leskovec, J., Kleinberg, J., & Faloutsos, C. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. https://www.cs.cornell.edu/home/kleinber/kdd05-time.pdf

[8] Mateos, G. PageRank: Ranking of Nodes in Graphs. https://www.hajim.rochester.edu/ece/sites/gmateos/ECE440/Slides/block_3_markov_chains_part_c.pdf

[9] Mayank, M. Visualizing Networks in python. Medium. Towards Data Science. https://towardsdatascience.com/visualizing-networks-in-python-d70f4cbeb259

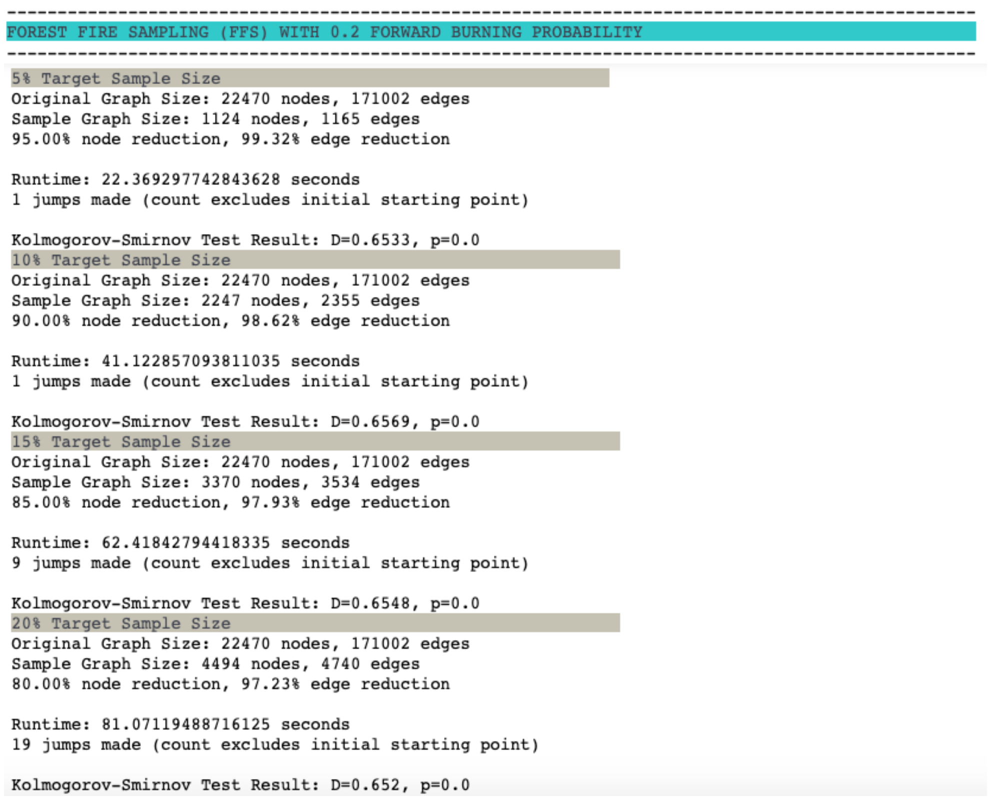
[10] Myakushina, A. Sampling Methods for Social Networks with Applications in Spotify Latent Space Models. https://docs.google.com/document/d/1jXhWbnWc10NSGVPNAOBtDKWN5ucBs5_N9ewJHY7RB5s

[11] Rozemberczki, B. MUSAE Facebook Page-Page Network. https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network?resource=download&select=musae_facebook_edges.csv

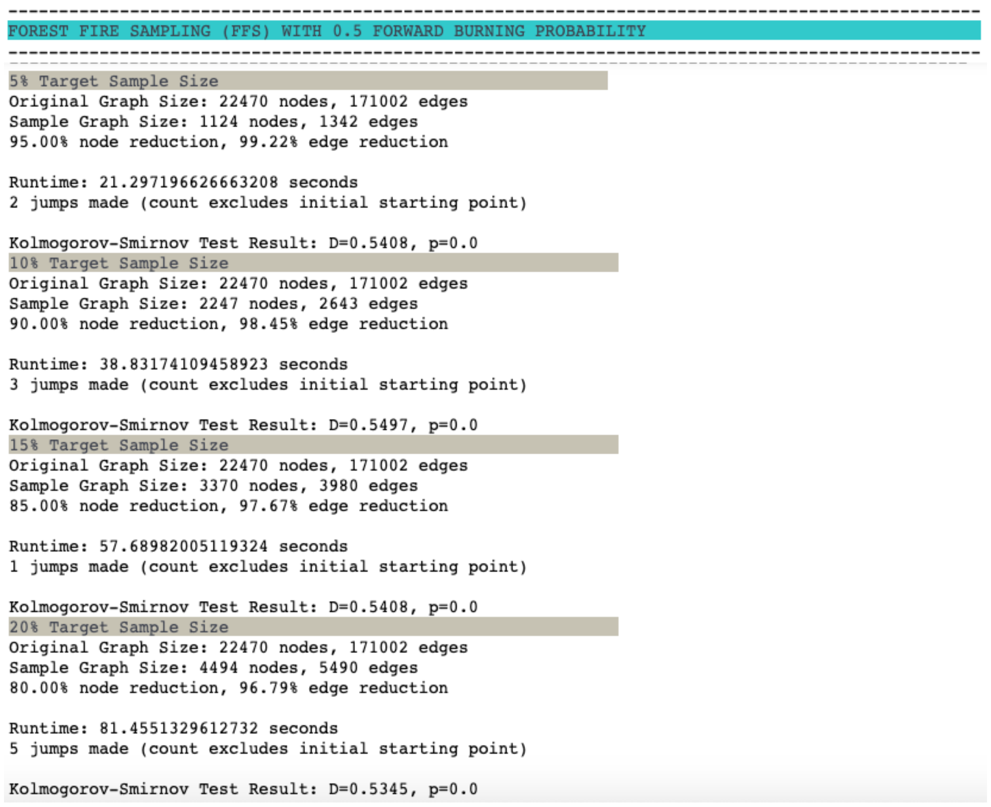
[12] Wicklin, R. (2019, May 15). What is Kolmogorov's D statistic? https://blogs.sas.com/content/iml/2019/05/15/kolmogorov-d-statistic.html

Appendix A

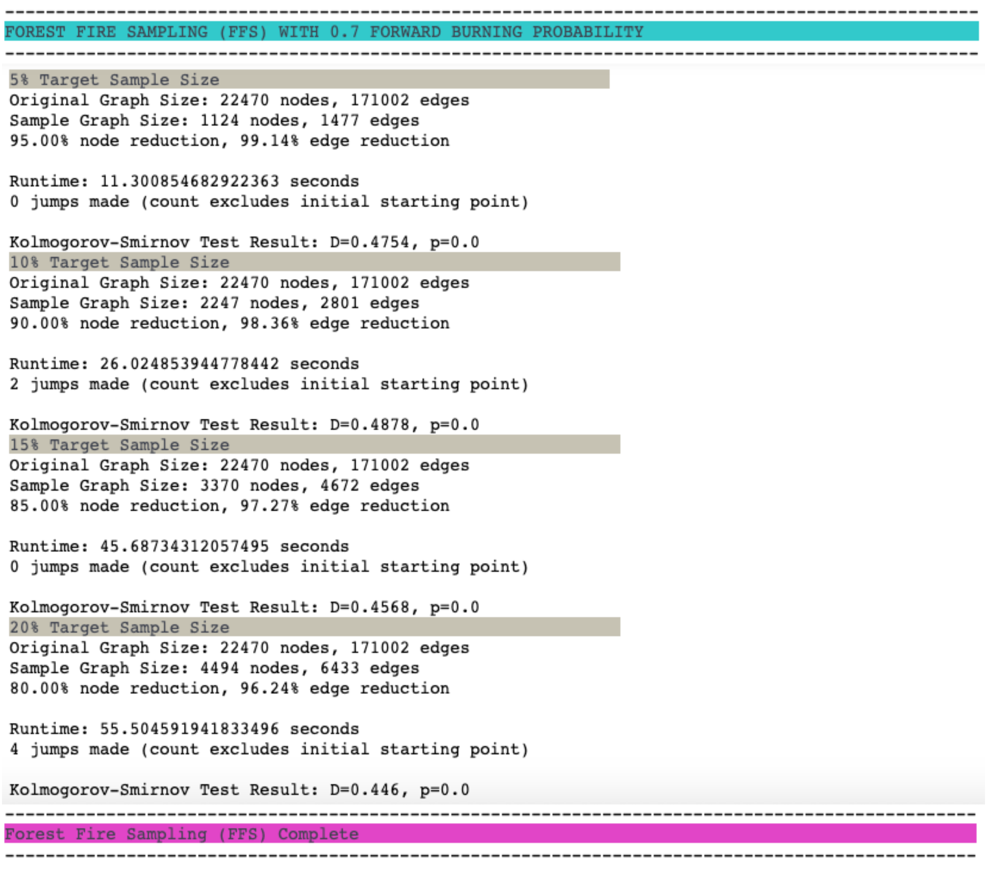
Forest Fire Sampling (FFS) Output



```
------------------------------------------------------------------------------------------------
FOREST FIRE SAMPLING (FFS) WITH 0.2 FORWARD BURNING PROBABILITY
------------------------------------------------------------------------------------------------
 5% Target Sample Size
 Original Graph Size: 22470 nodes, 171002 edges
 Sample Graph Size: 1124 nodes, 1165 edges
 95.00% node reduction, 99.32% edge reduction

 Runtime: 22.369297742843628 seconds
 1 jumps made (count excludes initial starting point)

 Kolmogorov-Smirnov Test Result: D=0.6533, p=0.0
 10% Target Sample Size
 Original Graph Size: 22470 nodes, 171002 edges
 Sample Graph Size: 2247 nodes, 2355 edges
 90.00% node reduction, 98.62% edge reduction

 Runtime: 41.122857093811035 seconds
 1 jumps made (count excludes initial starting point)

 Kolmogorov-Smirnov Test Result: D=0.6569, p=0.0
 15% Target Sample Size
 Original Graph Size: 22470 nodes, 171002 edges
 Sample Graph Size: 3370 nodes, 3534 edges
 85.00% node reduction, 97.93% edge reduction

 Runtime: 62.41842794418335 seconds
 9 jumps made (count excludes initial starting point)

 Kolmogorov-Smirnov Test Result: D=0.6548, p=0.0
 20% Target Sample Size
 Original Graph Size: 22470 nodes, 171002 edges
 Sample Graph Size: 4494 nodes, 4740 edges
 80.00% node reduction, 97.23% edge reduction

 Runtime: 81.07119488716125 seconds
 19 jumps made (count excludes initial starting point)

 Kolmogorov-Smirnov Test Result: D=0.652, p=0.0
```



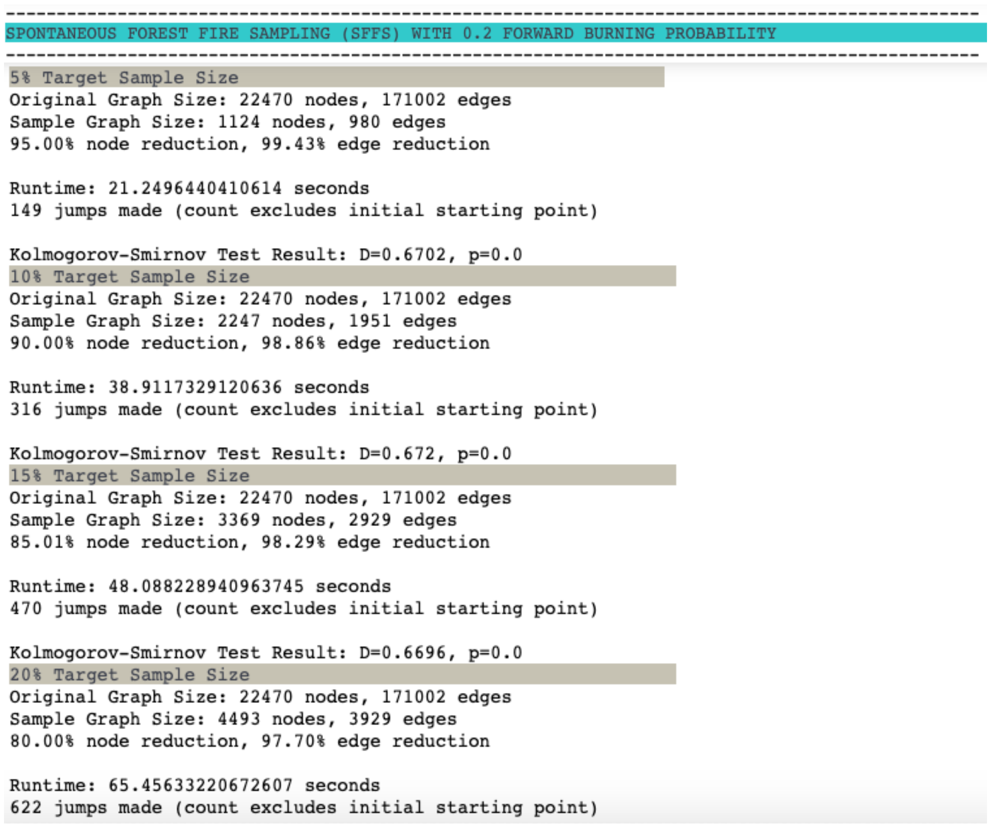
```
------------------------------------------------------------------------------------------------
FOREST FIRE SAMPLING (FFS) WITH 0.5 FORWARD BURNING PROBABILITY
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1342 edges
95.00% node reduction, 99.22% edge reduction

Runtime: 21.297196626663208 seconds
2 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5408, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2643 edges
90.00% node reduction, 98.45% edge reduction

Runtime: 38.83174109458923 seconds
3 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5497, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 3980 edges
85.00% node reduction, 97.67% edge reduction

Runtime: 57.68982005119324 seconds
1 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5408, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 5490 edges
80.00% node reduction, 96.79% edge reduction

Runtime: 81.4551329612732 seconds
5 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5345, p=0.0
```



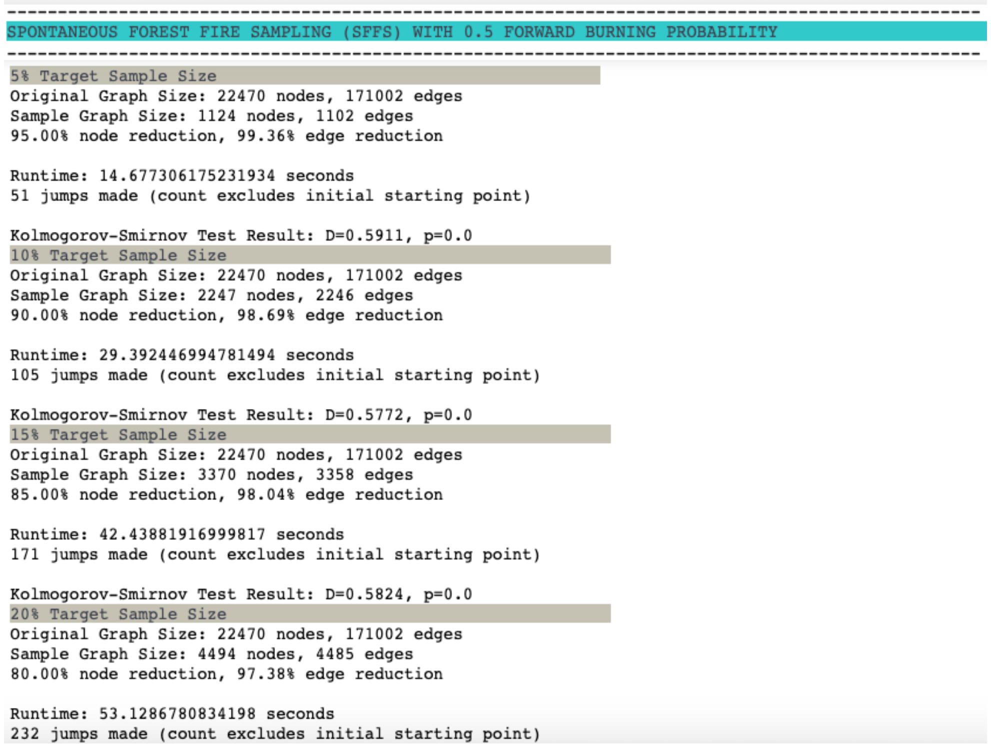
```
------------------------------------------------------------------------------------------------
FOREST FIRE SAMPLING (FFS) WITH 0.7 FORWARD BURNING PROBABILITY
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1477 edges
95.00% node reduction, 99.14% edge reduction

Runtime: 11.300854682922363 seconds
0 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4754, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2801 edges
90.00% node reduction, 98.36% edge reduction

Runtime: 26.024853944778442 seconds
2 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4878, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 4672 edges
85.00% node reduction, 97.27% edge reduction

Runtime: 45.68734312057495 seconds
0 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4568, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 6433 edges
80.00% node reduction, 96.24% edge reduction

Runtime: 55.504591941833496 seconds
4 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.446, p=0.0
------------------------------------------------------------------------------------------------
Forest Fire Sampling (FFS) Complete
------------------------------------------------------------------------------------------------
```

Appendix B

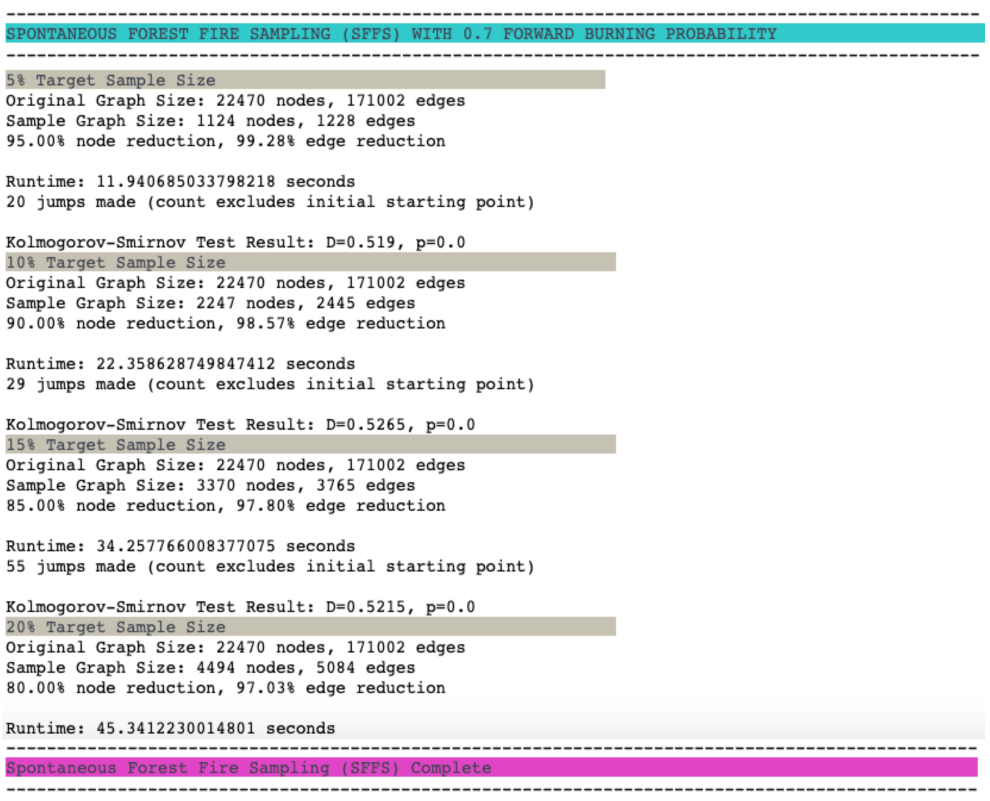
Spontaneous Forest Fire Sampling (SFFS) Output



```
------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.2 FORWARD BURNING PROBABILITY
------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 980 edges
95.00% node reduction, 99.43% edge reduction

Runtime: 21.2496440410614 seconds
149 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.6702, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 1951 edges
90.00% node reduction, 98.86% edge reduction

Runtime: 38.9117329120636 seconds
316 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.672, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3369 nodes, 2929 edges
85.01% node reduction, 98.29% edge reduction

Runtime: 48.088228940963745 seconds
470 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.6696, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4493 nodes, 3929 edges
80.00% node reduction, 97.70% edge reduction

Runtime: 65.45633220672607 seconds
622 jumps made (count excludes initial starting point)
```



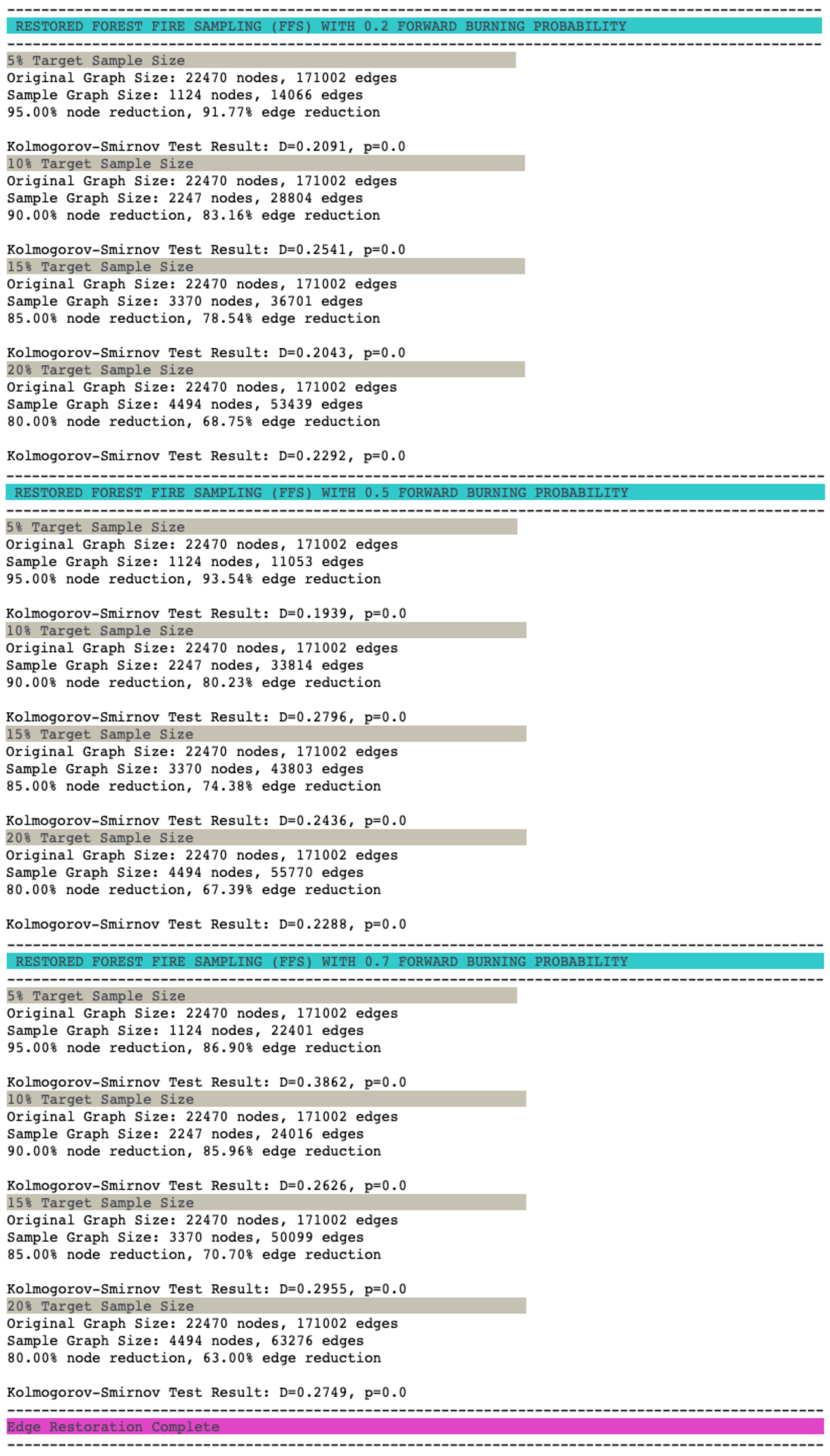
```
------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.5 FORWARD BURNING PROBABILITY
------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1102 edges
95.00% node reduction, 99.36% edge reduction

Runtime: 14.677306175231934 seconds
51 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5911, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2246 edges
90.00% node reduction, 98.69% edge reduction

Runtime: 29.392446994781494 seconds
105 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5772, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 3358 edges
85.00% node reduction, 98.04% edge reduction

Runtime: 42.43881916999817 seconds
171 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5824, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 4485 edges
80.00% node reduction, 97.38% edge reduction

Runtime: 53.1286780834198 seconds
232 jumps made (count excludes initial starting point)
```



```
-----------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.7 FORWARD BURNING PROBABILITY
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1228 edges
95.00% node reduction, 99.28% edge reduction

Runtime: 11.940685033798218 seconds
20 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.519, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2445 edges
90.00% node reduction, 98.57% edge reduction

Runtime: 22.358628749847412 seconds
29 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5265, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 3765 edges
85.00% node reduction, 97.80% edge reduction

Runtime: 34.257766008377075 seconds
55 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5215, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 5084 edges
80.00% node reduction, 97.03% edge reduction

Runtime: 45.3412230014801 seconds
-----------------------------------------------------------------------------------------------
Spontaneous Forest Fire Sampling (SFFS) Complete
-----------------------------------------------------------------------------------------------
```

Appendix C

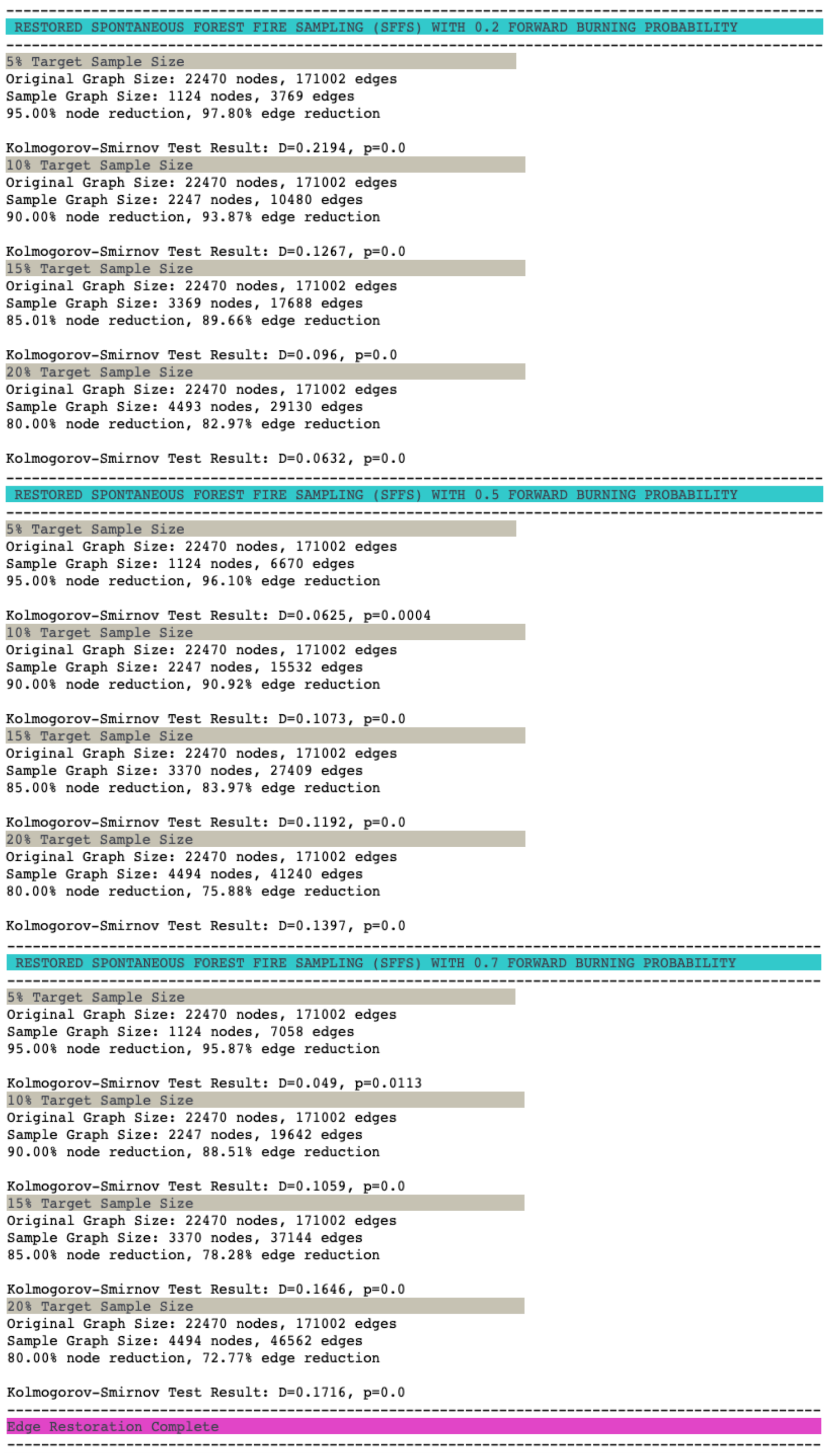
Edge-Restored FFS Output



```
----------------------------------------------------------------------------------------------
 RESTORED FOREST FIRE SAMPLING (FFS) WITH 0.2 FORWARD BURNING PROBABILITY
----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 14066 edges
95.00% node reduction, 91.77% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2091, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 28804 edges
90.00% node reduction, 83.16% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2541, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 36701 edges
85.00% node reduction, 78.54% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2043, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 53439 edges
80.00% node reduction, 68.75% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2292, p=0.0
----------------------------------------------------------------------------------------------
 RESTORED FOREST FIRE SAMPLING (FFS) WITH 0.5 FORWARD BURNING PROBABILITY
----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 11053 edges
95.00% node reduction, 93.54% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1939, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 33814 edges
90.00% node reduction, 80.23% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2796, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 43803 edges
85.00% node reduction, 74.38% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2436, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 55770 edges
80.00% node reduction, 67.39% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2288, p=0.0
----------------------------------------------------------------------------------------------
 RESTORED FOREST FIRE SAMPLING (FFS) WITH 0.7 FORWARD BURNING PROBABILITY
----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 22401 edges
95.00% node reduction, 86.90% edge reduction

Kolmogorov-Smirnov Test Result: D=0.3862, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 24016 edges
90.00% node reduction, 85.96% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2626, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 50099 edges
85.00% node reduction, 70.70% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2955, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 63276 edges
80.00% node reduction, 63.00% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2749, p=0.0
----------------------------------------------------------------------------------------------
Edge Restoration Complete
----------------------------------------------------------------------------------------------
```

Appendix D

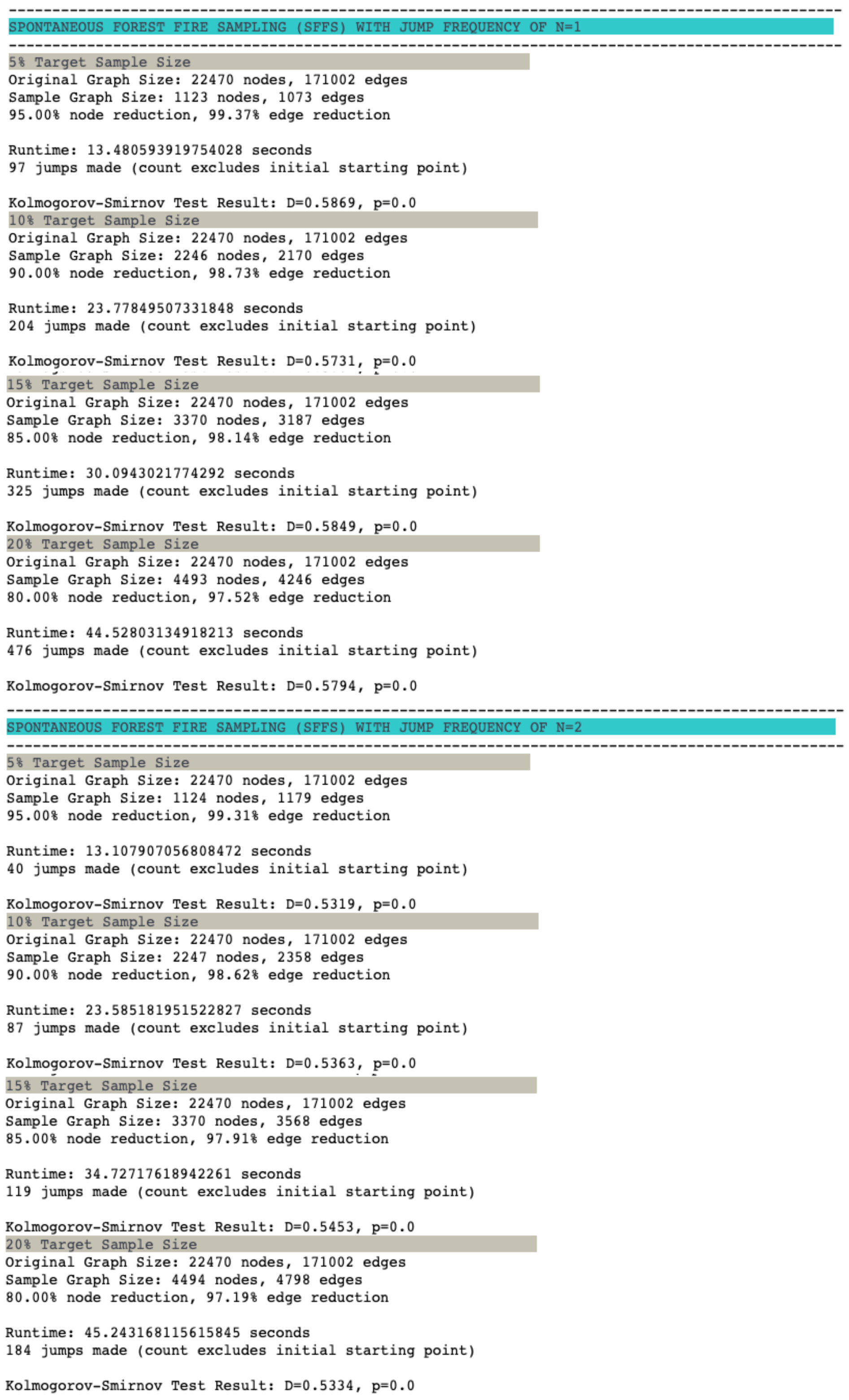
Edge-Restored SFFS Output



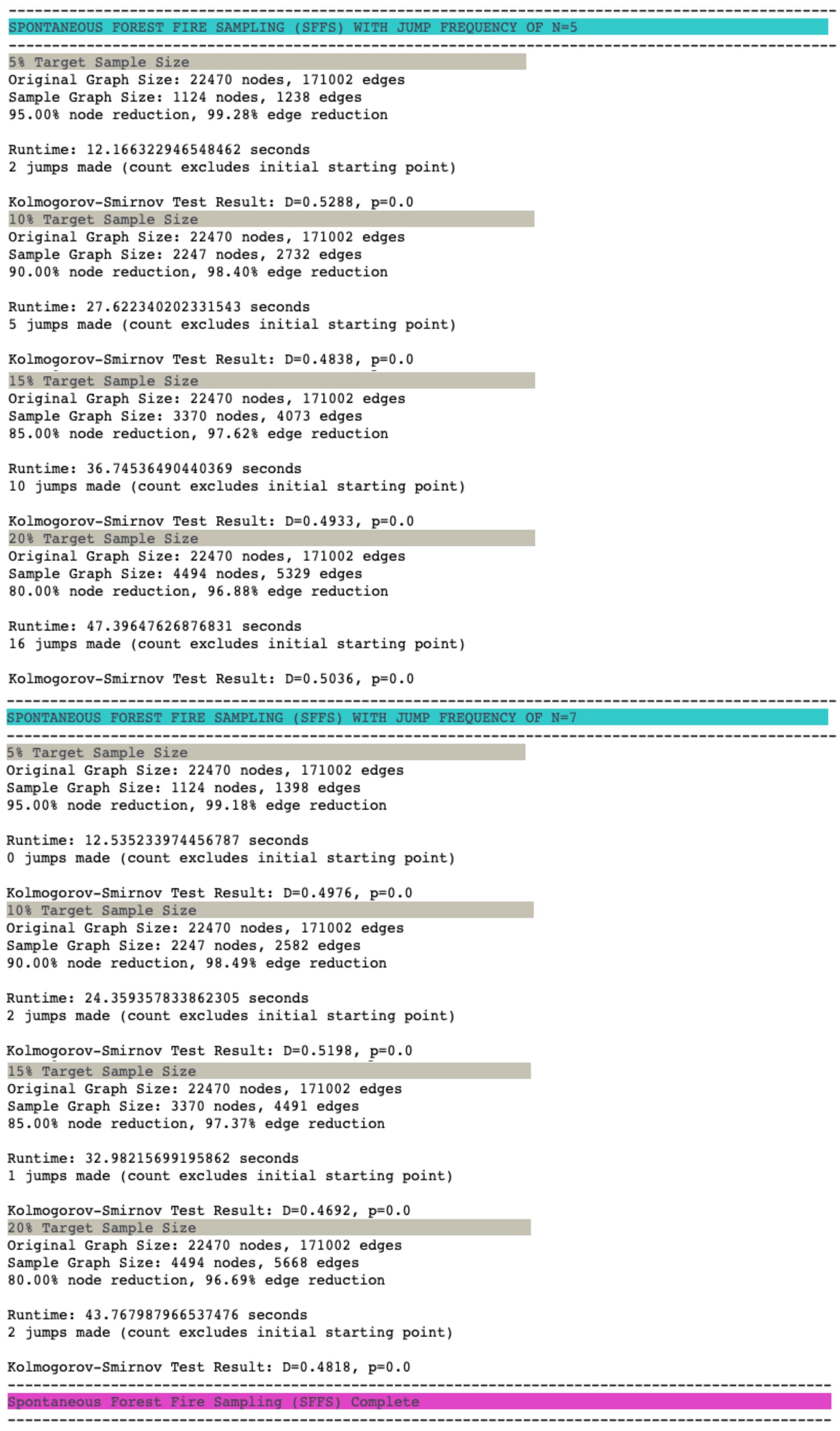
```
-----------------------------------------------------------------------------------------------
 RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.2 FORWARD BURNING PROBABILITY
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 3769 edges
95.00% node reduction, 97.80% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2194, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 10480 edges
90.00% node reduction, 93.87% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1267, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3369 nodes, 17688 edges
85.01% node reduction, 89.66% edge reduction

Kolmogorov-Smirnov Test Result: D=0.096, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4493 nodes, 29130 edges
80.00% node reduction, 82.97% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0632, p=0.0
-----------------------------------------------------------------------------------------------
 RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.5 FORWARD BURNING PROBABILITY
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 6670 edges
95.00% node reduction, 96.10% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0625, p=0.0004
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 15532 edges
90.00% node reduction, 90.92% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1073, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 27409 edges
85.00% node reduction, 83.97% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1192, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 41240 edges
80.00% node reduction, 75.88% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1397, p=0.0
-----------------------------------------------------------------------------------------------
 RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH 0.7 FORWARD BURNING PROBABILITY
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 7058 edges
95.00% node reduction, 95.87% edge reduction

Kolmogorov-Smirnov Test Result: D=0.049, p=0.0113
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 19642 edges
90.00% node reduction, 88.51% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1059, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 37144 edges
85.00% node reduction, 78.28% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1646, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 46562 edges
80.00% node reduction, 72.77% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1716, p=0.0
-----------------------------------------------------------------------------------------------
Edge Restoration Complete
-----------------------------------------------------------------------------------------------
```

Appendix E

Output for Testing SFFS at Various Jumping Frequencies



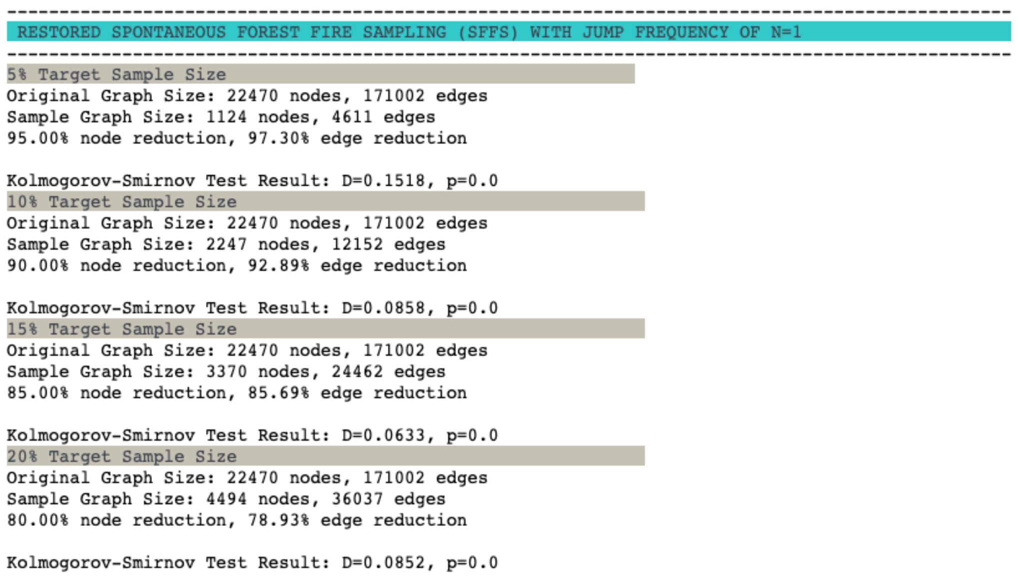
```
------------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=1
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1123 nodes, 1073 edges
95.00% node reduction, 99.37% edge reduction

Runtime: 13.480593919754028 seconds
97 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5869, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2246 nodes, 2170 edges
90.00% node reduction, 98.73% edge reduction

Runtime: 23.77849507331848 seconds
204 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5731, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 3187 edges
85.00% node reduction, 98.14% edge reduction

Runtime: 30.0943021774292 seconds
325 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5849, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4493 nodes, 4246 edges
80.00% node reduction, 97.52% edge reduction

Runtime: 44.52803134918213 seconds
476 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5794, p=0.0
------------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=2
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1179 edges
95.00% node reduction, 99.31% edge reduction

Runtime: 13.107907056808472 seconds
40 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5319, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2358 edges
90.00% node reduction, 98.62% edge reduction

Runtime: 23.585181951522827 seconds
87 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5363, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 3568 edges
85.00% node reduction, 97.91% edge reduction

Runtime: 34.72717618942261 seconds
119 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5453, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 4798 edges
80.00% node reduction, 97.19% edge reduction

Runtime: 45.243168115615845 seconds
184 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5334, p=0.0
```



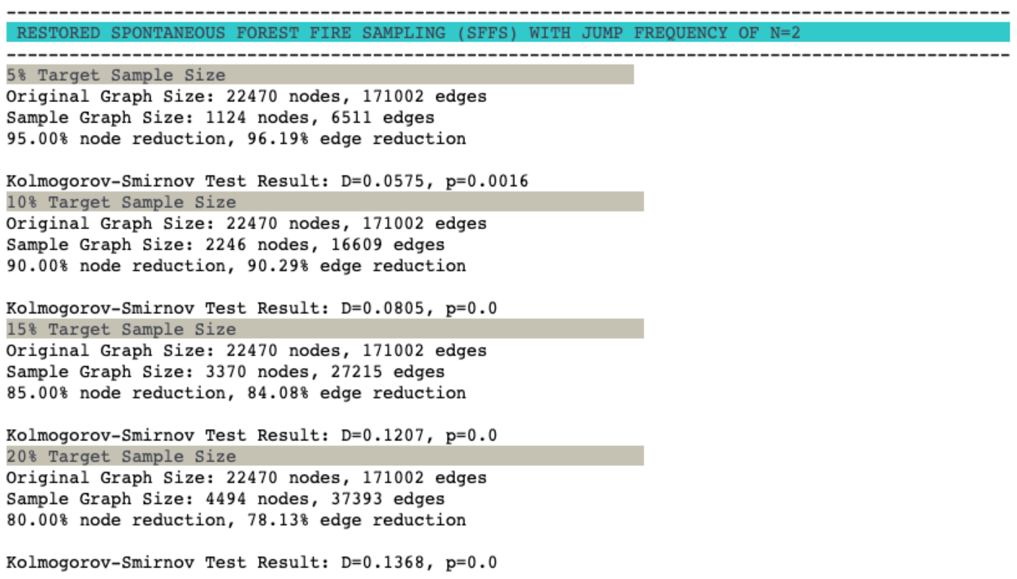
```
------------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=5
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1238 edges
95.00% node reduction, 99.28% edge reduction

Runtime: 12.166322946548462 seconds
2 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5288, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2732 edges
90.00% node reduction, 98.40% edge reduction

Runtime: 27.622340202331543 seconds
5 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4838, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 4073 edges
85.00% node reduction, 97.62% edge reduction

Runtime: 36.74536490440369 seconds
10 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4933, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 5329 edges
80.00% node reduction, 96.88% edge reduction

Runtime: 47.39647626876831 seconds
16 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5036, p=0.0
------------------------------------------------------------------------------------------------
SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=7
------------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 1398 edges
95.00% node reduction, 99.18% edge reduction

Runtime: 12.535233974456787 seconds
0 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4976, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 2582 edges
90.00% node reduction, 98.49% edge reduction

Runtime: 24.359357833862305 seconds
2 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.5198, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 4491 edges
85.00% node reduction, 97.37% edge reduction

Runtime: 32.98215699195862 seconds
1 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4692, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 5668 edges
80.00% node reduction, 96.69% edge reduction

Runtime: 43.767987966537476 seconds
2 jumps made (count excludes initial starting point)

Kolmogorov-Smirnov Test Result: D=0.4818, p=0.0
------------------------------------------------------------------------------------------------
Spontaneous Forest Fire Sampling (SFFS) Complete
------------------------------------------------------------------------------------------------
```

Appendix F

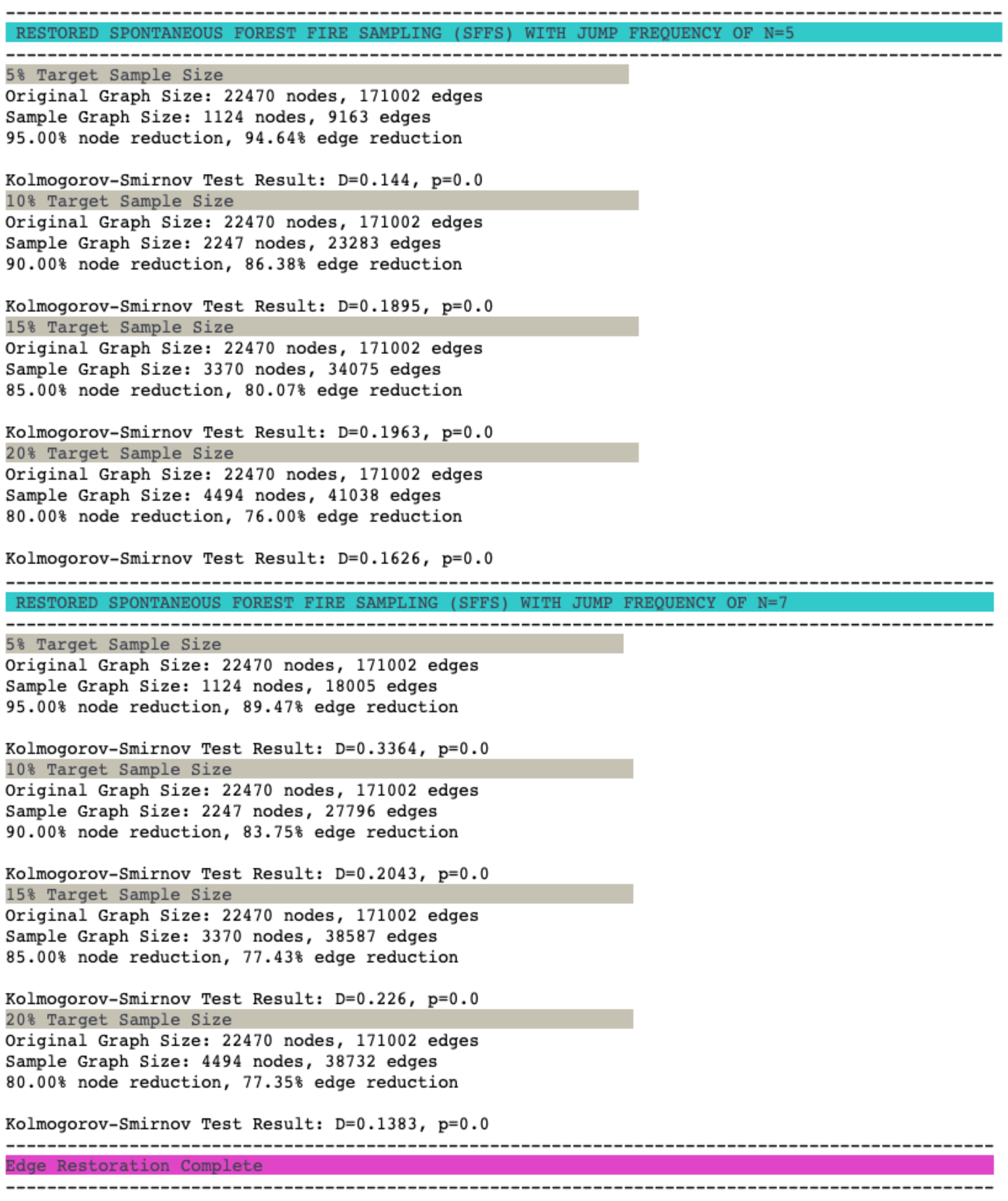
Output for Edge-Restored SFFS at Various Jumping Frequencies



```
-----------------------------------------------------------------------------------------------
 RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=1
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 4611 edges
95.00% node reduction, 97.30% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1518, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 12152 edges
90.00% node reduction, 92.89% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0858, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 24462 edges
85.00% node reduction, 85.69% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0633, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 36037 edges
80.00% node reduction, 78.93% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0852, p=0.0
```



```
-----------------------------------------------------------------------------------------------
 RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=2
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 6511 edges
95.00% node reduction, 96.19% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0575, p=0.0016
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2246 nodes, 16609 edges
90.00% node reduction, 90.29% edge reduction

Kolmogorov-Smirnov Test Result: D=0.0805, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 27215 edges
85.00% node reduction, 84.08% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1207, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 37393 edges
80.00% node reduction, 78.13% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1368, p=0.0
```



```
-----------------------------------------------------------------------------------------------
RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=5
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 9163 edges
95.00% node reduction, 94.64% edge reduction

Kolmogorov-Smirnov Test Result: D=0.144, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 23283 edges
90.00% node reduction, 86.38% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1895, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 34075 edges
85.00% node reduction, 80.07% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1963, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 41038 edges
80.00% node reduction, 76.00% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1626, p=0.0
-----------------------------------------------------------------------------------------------
RESTORED SPONTANEOUS FOREST FIRE SAMPLING (SFFS) WITH JUMP FREQUENCY OF N=7
-----------------------------------------------------------------------------------------------
5% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 1124 nodes, 18005 edges
95.00% node reduction, 89.47% edge reduction

Kolmogorov-Smirnov Test Result: D=0.3364, p=0.0
10% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 2247 nodes, 27796 edges
90.00% node reduction, 83.75% edge reduction

Kolmogorov-Smirnov Test Result: D=0.2043, p=0.0
15% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 3370 nodes, 38587 edges
85.00% node reduction, 77.43% edge reduction

Kolmogorov-Smirnov Test Result: D=0.226, p=0.0
20% Target Sample Size
Original Graph Size: 22470 nodes, 171002 edges
Sample Graph Size: 4494 nodes, 38732 edges
80.00% node reduction, 77.35% edge reduction

Kolmogorov-Smirnov Test Result: D=0.1383, p=0.0
-----------------------------------------------------------------------------------------------
Edge Restoration Complete
-----------------------------------------------------------------------------------------------
```