Multimodal Video Understanding using Graph Neural Network

Ayush Singh[1]*, **Vikram Gupta**[2]
[1]Indian Institute of Technology, Dhanbad, India, [2]ShareChat, India,
ayush.s.18je0204@cse.iitism.ac.in, vikramgupta@sharechat.co

Abstract

Majority of existing semantic video understanding methods process every video independently without considering the underlying `inter-video` relationships. However, videos uploaded by users on social media platforms like YouTube, Instagram etc. exhibit `inter-video` relationship which are a reflection of the interest, geography, culture etc. of the users. In this work, we explicitly attempt to model this `inter-video` relationship, originating from the creators of these videos using Graph Neural Networks (GNN) in a multimodal setup. We perform video classification by leveraging the creators of the videos and *semantic similarity* between videos for creating edges and observe improvements of **4%** in accuracy.

1 Introduction

Semantic understanding of videos has received lot of attention from both the research community and industry. Advancements in this field have been accelerated by the presence of large scale video understanding datasets like [14, 3, 8, 20, 17, 25, 16, 5, 4, 13, 29, 18] and deep learning models like [2, 6, 7, 12, 23]. Majority of these tasks process every video sample in isolation during the training and inference stage. The core assumption of this approach is that the videos are independent from each other and underlying `inter-video` relationships, if any, would be implicitly modelled by the network. However, in practical scenarios, videos originating from similar sources are often correlated, especially on social media platforms. For example, videos uploaded by an individual on a social media platform would be related by the interest or skill-set of the individual. An individual with a deep interest in *comedy* videos has higher probability of uploading and creating *comedy* videos. Similarly, an expert in *cooking* might create more *food* related videos. This `inter-video` relationship alludes to the presence of latent relationship existing between different videos which could be modelled for improving video understanding. While some of the previous works have modelled the `intra-video` spatio-temporal relationships [21, 28, 24, 19, 22], the efficacy of modelling `inter-video` correlations for improved semantic video understanding has not been explored. [1] explored the `inter-video` relationship but mainly from the *semantic similarity* perspective in an unsupervised settings. In this work, we explore the `inter-video` relationship between videos using Graph Neural Networks (GNN) in a multimodal setup using the `3MASSIV` dataset. `3MASSIV` dataset has been sourced from short-video application platform *Moj* and contains annotated social media videos along with masked identity of the uploaders of these videos. We leverage the creators of the videos and semantic similarity between them for connecting the videos to create the graph.

To the best of our knowledge, our work is the first attempt towards performing multimodal video understanding by modelling `inter-video` relations. We perform extensive experiments using Graph Convolutional Network (GCN) [15] and GraphSage [11] in unimodal and multimodal setup and demonstrate the advantages of modelling `inter-video` relationships.

*Work done during internship at ShareChat, India.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Our Method

2.1 Task

We model the semantic understanding of video as a node classification problem where every video $v_i \in V$ is a node in the graph G and is annotated with a category $y_i \in Y$. Two videos, v_i and v_j, created by the same `creator` are connected with each other using an edge $e_{ij} \in E$. By leveraging the semantic content encapsulated in the videos and `inter-video` relationships, the task is to classify each video v_i into one of the categories $y_i \in Y$. We model this graph $G =< V, Y, E >$ using Graph Convolutional Network (`GCN`) [15] and `GraphSage` [11] and use cross-entropy loss (L_{ce}) for classification. We present the details of `GraphSage` in next section and `GCN` in Section A.1.

2.2 GraphSAGE

We represent the videos (node) of the graph $G =< V, Y, E >$ by d-dimensional semantic features. Let, H be the feature matrix formed by stacking d-dimensional features $(h_1, h_2, ..., h_n)$ of n nodes or videos. Let, $W^{(l)}$ represents the weights of the l_{th} layer and $\mathcal{N}$ represent the neighbourhood function where, $\mathcal{N}(i)$ gives the list of nodes belonging to the neighbourhood of v_i node. For every node, `GraphSage` uniformly samples nodes from the neighbourhood and aggregates them using an aggregator function ϕ as shown in Equation 1.

$$h_{\mathcal{N}(i)}^{(l+1)} = \phi(h_j^l, \forall j \in \mathcal{N}(i)) \tag{1}$$

The aggregated features are then concatenated with the features of the node and transformed using weight matrix $W^{(l)}$ and passed to the next layer after applying an activation function φ and normalization.

$$h_i^{(l+1)} = \varphi\left(W^{(l)} \cdot \text{concat}(h_i^{(l)}, h_{\mathcal{N}(i)}^{(l+1)})\right) \tag{2}$$

$$h_i^{(l+1)} = h_i^{(l+1)} / \|h_i^{(l+1)}\|_2 \tag{3}$$

where, $h_i^{(l+1)}$ represents the transformed features of node i at layer $l+1$ and $\|h_i^{(l+1)}\|_2$ represents the norm of the features.

2.3 Video Representations

We represent the videos as nodes with features $h \in \mathbb{R}^d$ extracted from pretrained spatio-temporal and audio models. We used ResNext50-3D [27] (pretrained on Kinetics400) for extracting the spatio-temporal features and CLSRIL-23 [9] models for representing audio. These features are averaged across the temporal dimension for normalizing the temporal dimension. More details about feature extraction is present in Appendix Section A.2.

2.4 Edge Creation

We generate the edges between videos for the graph using two constraints:

(a) *Semantic Similarity:* We create an edge between two videos if the cosine similarity between the features of the nodes exceeds a predefined threshold. The optimum value of this threshold is selected using cross validation.

(b) *Creator Information:* An edge is created between two nodes if they belong to the same creator and *semantic similarity* is greater than the threshold.

3 Dataset

We experiment with `3MASSIV` [10] dataset which has been sourced from popular short-video platform *Moj*. `3MASSIV` contains the masked identities of the creators of the videos along with human annotated

Table 1: Accuracy over 3MASSIV test set using different models. We report mean and standard deviation of Top-1, Top-3 and Top-5 accuracy using 3 seeds. (.) shows the best Top-1 accuracy.

Model	**Top-1**	**Top-3**	**Top-5**
GCN$_{\texttt{base}}$	54.1 (54.3) ±0.15	74.5 (74.7) ±0.15	82.6 (82.7) ±0.50
GCN$_{\texttt{cos}}$	52.5 (52.6) ±0.11	73.9 (73.9) ±0.20	83.1 (83.3) ±0.23
GCN$_{\texttt{cre}}$	54.1 (54.1) ±0.11	75.2 (75.3) ±0.17	84.0 (84.0) ±0.00
GraphSage$_{\texttt{base}}$	53.6 (53.9) ±0.23	73.9 (74.4) ±0.38	82.4 (82.2) ±0.20
GraphSage$_{\texttt{cos}}$	53.3 (53.6) ±0.26	74.1 (73.5) ±0.51	82.4 (81.9) ±0.46
GraphSage$_{\texttt{cre}}$	57.7 (58.0) ±0.23	77.5 (77.5) ±0.05	85.1 (85.2) ±0.05

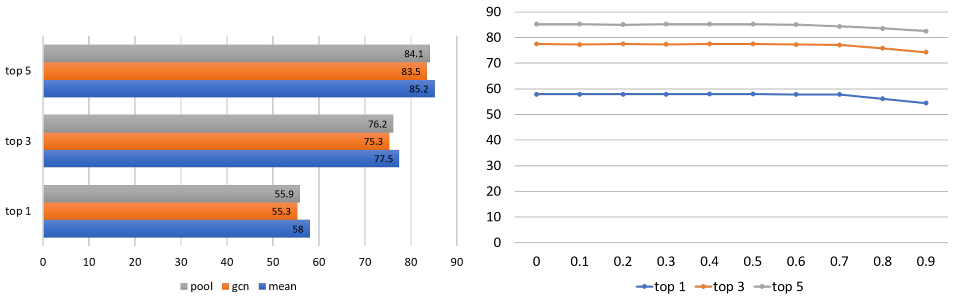
Table 2: Accuracy over 3MASSIV test set for GraphSage using visual and audio features for node representation and edge generation. We report mean and standard deviation of Top-1, Top-3 and Top-5 accuracy using 3 seeds. The value in (.) shows the best Top-1 accuracy.

Node features		**Edge Creation**			**Top-1**	**Top-3**	**Top-5**
Video	Audio	Creator	Video	Audio			
✓		✓	✓		57.7 (58.0) ±0.23	77.5 (77.5) ±0.05	85.1 (85.2) ±0.05
✓		✓		✓	57.9 (58.1) ±0.26	77.5 (77.5) ±0.06	84.9 (84.8) ±0.21
✓		✓	✓	✓	58.0 (58.1) ±0.10	77.4 (77.5) ±0.15	84.9 (85.2) ±0.30
✓	✓	✓	✓	✓	59.3 (59.5) ±0.17	79.4 (79.4) ±0.06	86.7 (86.5) ±0.29

labels. 3MASSIV contains 50K annotated videos belonging to 34 popular social media concepts and comprises of 11 Indic languages with an average duration of around 20 seconds. The videos in this dataset have been contributed by more than 20K social media users, thus providing rich and diverse graphical information for our task.

4 Architecture and Training Details

We use 2-layered graphical networks for our experiments. We used ReLU as the non-linearity and added a batch normalization layer between the two layers. We trained our models on A100 GPUs for 150 epochs with a learning rate of 0.001 and Adam Optimizer using 2048 as batch size. We used DGL [26] library. We use the training, validation and test split provided with the dataset for our experiments and report the mean and standard deviation of Top-1, Top-3 and Top-5 accuracy over 3 random runs .



(a) Different aggregators

(b) Distance threshold between nodes

Figure 1: Ablation experiments comparing different hyperparameters: (a) Effect of different aggregator function on accuracy in GraphSage model (b) Effect of distance threshold between nodes for creating edge on GraphSage model

Table 3: Transductive and Inductive settings

Model	**Setting**	**Top-1**	**Top-3**	**Top-5**
$\texttt{GraphSage}_{\texttt{cre}}$	Transductive	58.0	77.5	85.2
$\texttt{GraphSage}_{\texttt{cre}}$	Inductive	57.4	77.5	85.2

5 Results and Discussion

5.1 Unimodal Experiments

In Table 1, we report the performance using visual features. We train the models ($\texttt{GCN}_{\texttt{base}}$ and $\texttt{GraphSage}_{\texttt{base}}$) without edges to baseline the results in absence of `inter-video` relations. On using the *semantic similarity* ($\texttt{GCN}_{\texttt{cos}}$ and $\texttt{GraphSage}_{\texttt{cos}}$) for creating edges between videos, we do not observe gains. However, on leveraging the `creator` relationships ($\texttt{GCN}_{\texttt{cre}}$ and $\texttt{GraphSage}_{\texttt{cre}}$) between the videos, we observe substantial gains. This demonstrates the efficacy of the `creator` edges towards improving semantic understanding of videos. Overall, we observe that across all the configurations, `GraphSage` shows improved performance over `GCN`.

5.2 Multimodal Experiments

Videos are multimodal in nature due to presence of visual and audio channel. We explore the multimodality in different ways. (a) Firstly, we use visual features for node representation and generate edges using similarity between the visual features with creator identifier (b) In second setting, we used the visual features for node representation but generate edges using similarity between audio features along with creator identifier (c) Then, we generate the edges using concatenation of both visual and audio features and creator identifier while representing the node using visual features. All these approaches showed similar performance but using the audio features for edge creation improved the Top-1 performance by 0.2% over using only visual features. (d) Next, we concatenate the visual and audio information together for node representations and edge creation with creator information. We can see that this shows the best results and improves the accuracy by more than 1%. We show these quantitative results in Table 2.

5.3 Inductive and Transductive Settings

In Table 3, we compare *transductive* and *inductive* settings of training GNNs. In *transductive* setting, we use all the dataset splits for generating the training graph. However, we mask the labels of non-train set videos so that only the labels of train set are used for training. In the *inductive* setting, we use only the training videos and labels for creating the training graph and training the model. We observe that *transductive* setting shows higher performance than *inductive* settings as it processes all the splits of the dataset during graph creation.

5.4 Ablation

We study the effect of different aggregators like *pool*, *gcn* and *mean* on `GraphSage` in Figure 1 (a) and observe the best result using "mean" aggregator. Thus, we use "mean" aggregator for all our experiments. We study the impact of varying the cosine distance threshold. Higher threshold allows less number of edges while lower threshold results in more number of edges. From Figure 1 (b), we observe that 0.4 shows the best performance and we use it for other experiments.

6 Conclusions

In this work, we explored the inherent `inter-video` relationship that exists between the videos uploaded on social media platforms. We studied `GCN` and `GraphSage` under *inductive* and *transductive* settings and observed substantial performance gains by exploiting the creator information. To the best of our knowledge, this `inter-video` information has not been explored previously in the context of video understanding. Further, we explored the impact of using both visual and audio modalities as input and for creating graphical edges and observe improvement in video classification.

References

[1] Ali Aminian. Vidsage: Unsupervised video representational learning with graph convolutional networks. 2019.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021.

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021.

[5] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[6] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] Anirudh Gupta, Harveen Singh Chadha, Priyanshi Shah, Neeraj Chimmwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. Clsril-23: Cross lingual speech representations for indic languages. *arXiv preprint arXiv:2107.07402*, 2021.

[10] Vikram Gupta, Trisha Mittal, Puneet Mathur, Vaibhav Mishra, Mayank Maheshwari, Aniket Bera, Debdoot Mukherjee, and Dinesh Manocha. 3massiv: Multilingual, multimodal and multi-aspect dataset of social media short videos. *arXiv preprint arXiv:2203.14456*, 2022.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Quan Kong, Ziming Wu, Ziwei Deng, Martin Klinkigt, Bin Tong, and Tomokazu Murakami. Mmact: A large-scale dataset for cross modal human action understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[17] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.

[18] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020.

[19] Yang Liu, Keze Wang, Lingbo Liu, Haoyuan Lan, and Liang Lin. Tcgl: Temporal contrastive graph for self-supervised video representation learning. *IEEE Transactions on Image Processing*, 31:1978–1993, 2022.

[20] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):502–508, 2019.

[21] Masoud Pourreza, Mohammadreza Salehi, and Mohammad Sabokrou. Ano-graph: Learning normal scene contextual graphs to detect video anomalies. *arXiv preprint arXiv:2103.10502*, 2021.

[22] P Pradhyumna, GP Shreya, et al. Graph neural network (gnn) in image and video understanding using deep learning for computer vision applications. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1183–1189. IEEE, 2021.

[23] Michael S Ryoo, AJ Piergiovanni, Mingxing Tan, and Anelia Angelova. Assemblenet: Searching for multi-stream neural connectivity in video architectures. *arXiv preprint arXiv:1905.13209*, 2019.

[24] Jingkuan Song, Lianli Gao, Mihai Marian Puscas, Feiping Nie, Fumin Shen, and Nicu Sebe. Joint graph learning and video segmentation via multiple cues and topology calibration. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 831–840, 2016.

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[26] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

[28] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Hang Zhao, Zhicheng Yan, Lorenzo Torresani, and Antonio Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv preprint arXiv:1712.09374*, 2019.

A Appendix

A.1 Graph Convolutional Network (GCN)

GCN works on the spectral domain of the graph. Lets say we have n nodes representing n videos through n features $h_1, h_2, ..., h_n$ of d dimensions. These features can be stacked together to form a feature matrix X. We have an Adjacency matrix A which tells us about the edge connection between the nodes. We also enforce the self-edge connection between the nodes via

$$\tilde{A} = A + I$$

where, I represents identity matrix. We normalise $\tilde{A}$ with $\tilde{D}$ where $\tilde{D}$ represent degree matrix correponding to graph generated by $\tilde{A}$. $\tilde{D}_{ij}$ means row wise summation of $\tilde{A}$. The way normalization of $\tilde{A}$ is down is shown below:

$$\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$$

Then the update rule for GCN is:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}) \tag{4}$$

where, $H^{(l)}$ and $W^{(l)}$ means feature matrix and weight matrix at l^{th} stage. Here, $H^{(l)} = X$. σ represents the non-linearity added after each layer.

A.2 Preprocessing

We used ResNext50-3D [27] (pretrained on Kinetics700) for extracting spatio-temporal visual features from the `3MASSIV` videos. We sample the videos at 25 frame per second and scale the frames by resizing the shortest side to 240 pixels while maintaining the aspect ratio. We use 16 centrally cropped frames of dimension 112 x 112 x 3 as clips for extracting 2048-dimensional features from the last convolution layer and average the clip representations to generate a 2048-dimensional representation for the video. For audio features, we extracted the audio channel as mono-channel from our videos using ffmpeg[2]. We sample the audios at 16kHz and use CLSRIL-23 [9] models for audio feature extraction. The audio features are averaged across the clip to get a 512-dimensional vector.

B Results

Edge Type: We experiment with using the *semantic similarity* between the videos as edge weight and binary edges. In binary edge, if the edge exist, edge weight will be 1.0 else 0. From the experiments, we found that edge type does not impact model performance although semantic similarity weights shows slight improvement. We report the results of our ablation study in detail in Table 4, Table 5 and Table 6.

[2] https://www.ffmpeg.org/

Table 4: Ablation study on threshold value used for edge generation for GraphSage. We compare accuracy over 3MASSIV test set.

Model	Threshold	Top-1	Top-3	Top-5
GraphSage	0.9	54.4	74.3	82.5
GraphSage	0.8	56.1	75.8	83.6
GraphSage	0.7	57.8	77.1	84.4
GraphSage	0.6	57.8	77.3	85.1
GraphSage	0.5	58.0	77.5	85.2
GraphSage	0.4	58.0	77.5	85.2
GraphSage	0.3	57.9	77.4	85.2
GraphSage	0.2	57.9	77.5	85.1
GraphSage	0.1	57.9	77.3	85.2
GraphSage	0.0	57.9	77.5	85.2

Table 5: Ablation study on different aggregators.

Model	Aggregator	Top-1	Top-3	Top-5
GraphSage	mean	58.0	77.5	85.2
GraphSage	gcn	55.3	75.3	83.5
GraphSage	pool	55.9	76.2	84.1

Table 6: Ablation on Binary vs Non-Binary edge on GraphSage on two different semantic similarity threshold.

Model	Type of edge	Threshold	Top-1	Top-3	Top-5
GraphSage	Binary	0.7	57.7	77.5	84.9
GraphSage	Non-Binary	0.7	57.8	77.1	84.4
GraphSage	Binary	0.4	57.9	77.5	85.1
GraphSage	Non-Binary	0.4	5.0	77.5	85.2

Threshold	GCN	GraphSage	Avg Edges per node
0.9	53.8	54.4	3.37
0.8	54.1	56.1	12.32
0.7	50.6	57.8	21.88
0.6	46.4	57.8	26.29
0.5	45.7	58.0	26.84
0.4	45.7	58.0	26.85
0.3	45.7	57.9	26.85
0.2	45.7	57.9	26.85
0.1	45.7	57.9	26.85
0.0	45.7	57.9	26.85

Table 7: Ablation on GraphSage vs GCN performance on 3MASSIV dataset with respect with threshold and Avg edges per node