Outsmarting the outliers in attributed network representation learning

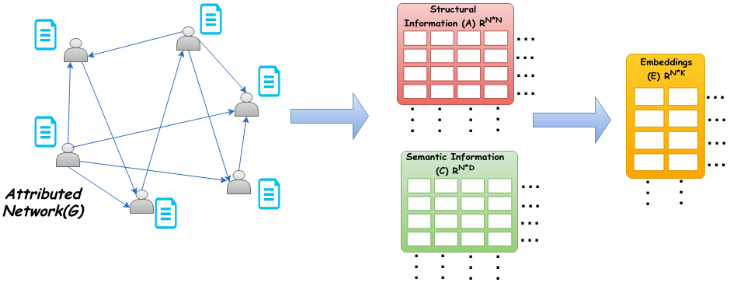
Lokesh N

Indian Institute of Science, Bangalore & IBM Research

Thanks to : Prof. Annie Bennet for hosting me.

February 11, 2020

Attributed Network Reperesentation learning



- Mathematically learn a function $f : R^{N+D} \rightarrow R^k$
- $k << N + D$

Related work

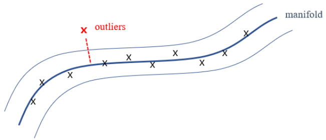
Comparison of the properties of the state-of-the-art baseline algorithms with that of ONE and DONE.

Method	Consider Attributes	Unsupervised	Outlier handling	Deep Network
node2vec	No	Yes	No	No
LINE	No	Yes	No	No
SDNE	No	Yes	No	Yes
TADW	Yes	Yes	No	No
GraphSAGE	Yes	Yes	No	Yes
DGI	Yes	Yes	No	Yes
SEANO	Yes	No	Yes	Yes
ONE	Yes	Yes	Yes	No
DONE	Yes	Yes	Yes	Yes

For the sake of fair comparison, we consider mostly **unsupervised** embedding algorithms

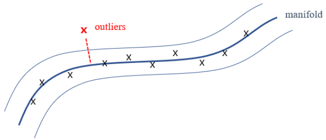
Motivation

- Network has outliers
- Outliers affect the representation of all the (good) nodes and in general the embedding space/manifold



Motivation

- Network has outliers
- Outliers affect the representation of all the (good) nodes and in general the embedding space/manifold



Not so optimal solution

- Preprocess the dataset by running anamoly detection algorithm
- remove $\times$
- learn the representations of $\times$ only

Hypothesis - I

- We hypothesize that every node has some tendency to exhibit outlierness/anomalous nature
  - We observe only partial information about each node in the network
  - Also the partial information we observe is lossy

Hypothesis - I

- We hypothesize that every node has some tendency to exhibit outlierness/anomalous nature
  - We observe only partial information about each node in the network
  - Also the partial information we observe is lossy
- *Conclusion :* We should not draw a **hard** line and say that some nodes are anomalous and others are not
- Hence earlier strategy (pipe-lined approach) is sub optimal

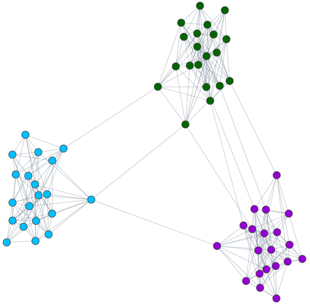
Hypothesis - I

- We hypothesize that every node has some tendency to exhibit outlierness/anomalous nature
  - We observe only partial information about each node in the network
  - Also the partial information we observe is lossy
- *Conclusion :* We should not draw a **hard** line and say that some nodes are anomalous and others are not
- Hence earlier strategy (pipe-lined approach) is sub optimal

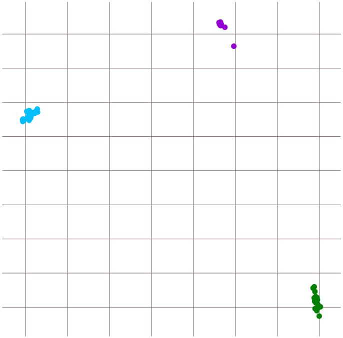
Our work : We explicitly model the outlierness of each node (in closed form) and **discourage** the contribution of nodes with greater outlierness towards representation learning

Motivating Example

Ideal World



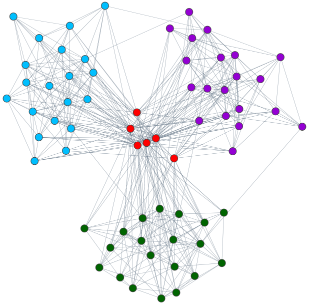
Synthetic Network with a very good modular structure



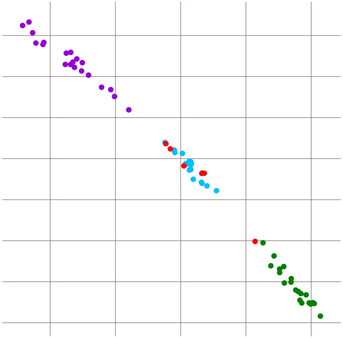
TSNE visualization of node2vec embeddings

Because the world is ideal n2v is at it's best

Corrupted world



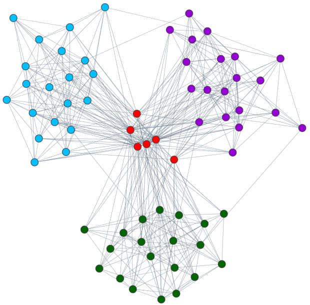
Synthetic Network with just 6 outliers



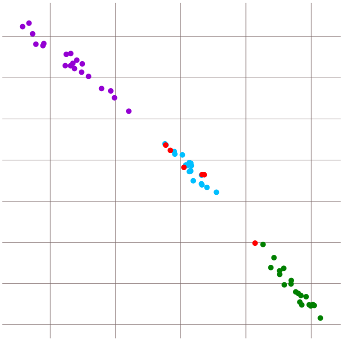
TSNE visualization of node2vec embeddings

The outliers draw the random walks across communities violating the homophily assumption that n2v makes. Hence n2v is at its worst

Corrupted world



Synthetic Network with just 6 outliers

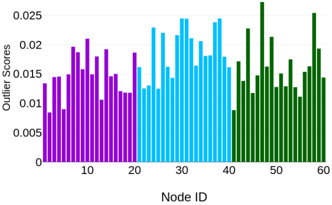


TSNE visualization of node2vec embeddings

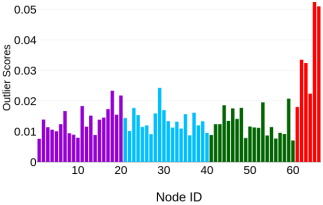
The outliers draw the random walks across communities violating the homophily assumption that n2v makes. Hence n2v is at its worst

ONE/DONE to the rescue

How does the outlier scores magic work



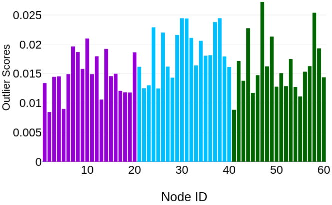
Outlier scores on synthetic network without outliers



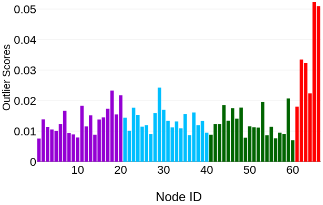
Outlier scores on synthetic network with outliers

$$\mathcal{L}_{str} = \sum_{i=1}^{N} \sum_{j=1}^{N} \log\left(\frac{1}{O_{1i}}\right)(A_{ij} - G_{i\cdot} \cdot H_{\cdot j})^2$$

How does the outlier scores magic work



Outlier scores on synthetic network without outliers



Outlier scores on synthetic network with outliers

$$\mathcal{L}_{str} = \sum_{i=1}^{N} \sum_{j=1}^{N} \log\left(\frac{1}{O_{1i}}\right)(A_{ij} - G_{i\cdot} \cdot H_{\cdot j})^2$$

Hence, the magic works

Informal Problem statement

Learn robust representation of the network in the presence of outliers. Basically, **outsmart the outliers** and learn good node representations.

Datasets

- To the best of our knowledge, there is no dataset with ground truth outlier scores
- What do we do now ?

Datasets

- To the best of our knowledge, there is no dataset with ground truth outlier scores
- What do we do now ? Create our own dataset

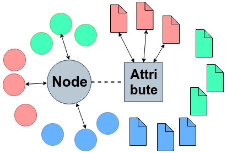
Datasets

- To the best of our knowledge, there is no dataset with ground truth outlier scores
- What do we do now ? Create our own dataset
- In literature a common practice is to perturb/**corrupt** existing nodes in dataset
- We **plant** new outlier nodes in dataset
- Our planting scheme is robust, meaning no prepossessing algorithm can trivially detect them

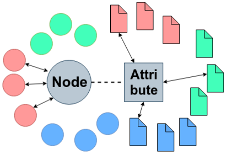
But what is an outlier (or) what exactly is outlierness ?

- To be frank, we don't know
- There is no standard definition of outlier in literature
- Loosely speaking, outlier is any node that exhibits behavior different from majority of the nodes in the dataset
- Hence, Mr xxxx is an outlier and so is Dr Jon Kleinberg

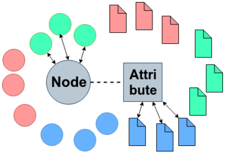
Our notion of an outlier



Structural outlier



Attribute outlier



Combined outlier

We capture structural outlierness using the term o_{1i}, attribute outlierness using o_{2i} and combined outlierness using o_{3i} in the algorithm formulations of ONE and DONE

Contributions of ONE [1]

- We propose an unsupervised algorithm called ONE (**O**utlier aware **N**etwork **E**mbedding) for attributed networks.
- This is **the first work** to propose a completely unsupervised algorithm for attributed network embedding integrated with outlier detection.
- Also we propose a **novel** method to combine structure and attributes efficiently.
- Thorough experimentation

[1] Accepted in AAAI'19

ONE - Solution Approach

- Our solution primarily has three components

ONE - Solution Approach

- Our solution primarily has three components
- Structure embedding
  - Factorize the Adjacency matrix
  - $\mathcal{L}_{str} = \sum_{i=1}^{N} \sum_{j=1}^{N} \log\left(\frac{1}{O_{1i}}\right)(A_{ij} - G_{i\cdot} \cdot H_{\cdot j})^2$

ONE - Solution Approach

- Our solution primarily has three components
- Structure embedding
  - Factorize the Adjacency matrix
  - $\mathcal{L}_{str} = \sum_{i=1}^{N} \sum_{j=1}^{N} \log\left(\frac{1}{O_{1i}}\right)(A_{ij} - G_{i\cdot} \cdot H_{\cdot j})^2$
- Attribute embedding
  - Factorize the attribute matrix
  - $\mathcal{L}_{attr} = \sum_{i=1}^{N} \sum_{d=1}^{C} \log\left(\frac{1}{O_{2i}}\right)(C_{id} - U_{i\cdot} \cdot V_{\cdot j})^2$

ONE - Solution Approach

- Our solution primarily has three components
- Structure embedding
  - Factorize the Adjacency matrix
  - $\mathcal{L}_{str} = \sum_{i=1}^{N} \sum_{j=1}^{N} \log\left(\frac{1}{O_{1i}}\right)(A_{ij} - G_{i\cdot} \cdot H_{\cdot j})^2$
- Attribute embedding
  - Factorize the attribute matrix
  - $\mathcal{L}_{attr} = \sum_{i=1}^{N} \sum_{d=1}^{C} \log\left(\frac{1}{O_{2i}}\right)(C_{id} - U_{i\cdot} \cdot V_{\cdot j})^2$
- Let G = structure embedding matrix
- Let U = attribute embedding matrix
- Final node embeddings is $\frac{G+U}{2}$

- Last component : couple G and U
  - L_2 norm on G-U
  - $\mathcal{L}_{combined} = \sum_{i=1}^{N} \sum_{k=1}^{K} \log\left(\frac{1}{O_{3i}}\right)\left(G_{ik} - U_{i\cdot}\right)^2$

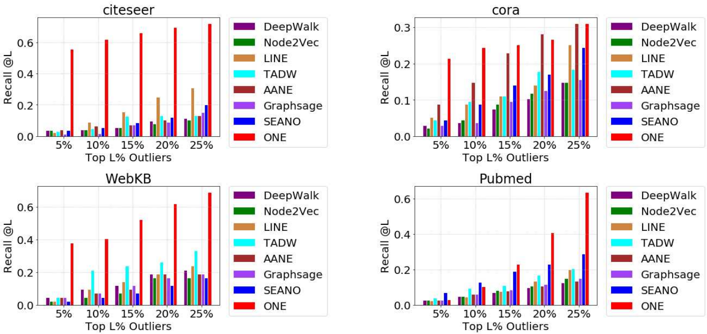
- Last component : couple G and U
  - L_2 norm on G-U
  - $\mathcal{L}_{combined} = \sum_{i=1}^{N} \sum_{k=1}^{K} \log\left(\frac{1}{O_{3i}}\right)\left(G_{ik} - U_{i\cdot}\right)^2$
- We have minimized the eucledian distance between the structure and embedding spaces, what if they are not **aligned** ?
- We introduce a Linear transformation matrix **W** that aligns the two spaces
- $\mathcal{L}_{combined} = \sum_{i=1}^{N} \sum_{k=1}^{K} \log\left(\frac{1}{O_{3i}}\right)\left(G_{ik} - U_{i\cdot} \cdot (W^T)_{\cdot k}\right)^2$

Can W be an arbitrary linear transformation ?

- **Nope**, we want linear transformation
- But only upto **rotation**, no scaling
- Mathematically, W should be **orthogonal**
- But how do we enforce the orthogonality constraint ??
  - To our fortune, we have **procrustes** problem to our rescue
  - Closed form solution to W can be formulated from SVD(GU^T)

Beauty of closed form solutions

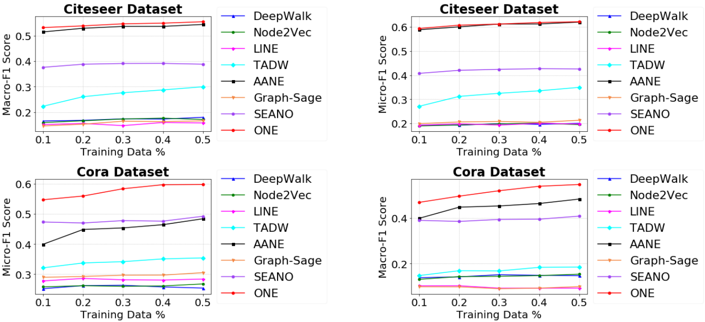
Outlier Detection Perfromance

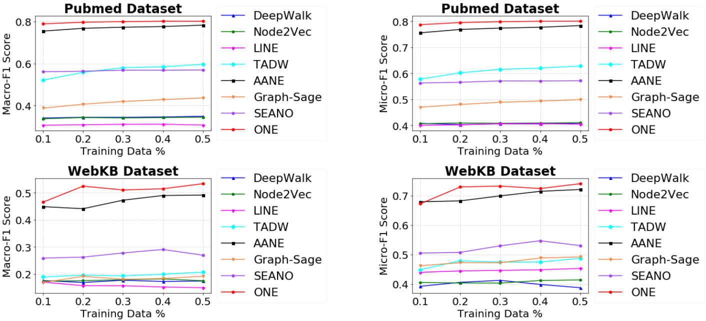


Note : ONE and SEANO have explicit outlier interpretations readily available. For rest of the algorithms we train the embeddings first and then run Isolation Forest on the embeddings to generate the outlier scores.

Classification Performance

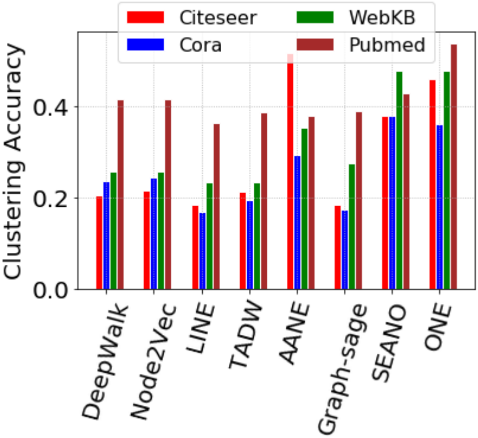
All results reported are obtained by running Random Forest algorithm for classification





Pubmed Dataset
Macro-F1 Score
Training Data %
Pubmed Dataset
Micro-F1 Score
Training Data %
WebKB Dataset
Macro-F1 Score
Training Data %
WebKB Dataset
Micro-F1 Score
Training Data %
DeepWalk
Node2Vec
LINE
TADW
AANE
Graph-Sage
SEANO
ONE

Clustering Perfromance



Limitations of ONE

- ONE is not scalable to large graphs because it is $O(N^2)$ algorithm
- ONE is a linear model and hence cannot capture non-linear intricacies in the dataset

Limitations of ONE

- ONE is not scalable to large graphs because it is $O(N^2)$ algorithm
- ONE is a linear model and hence cannot capture non-linear intricacies in the dataset

We precisely address the above two concers in our next solution DONE

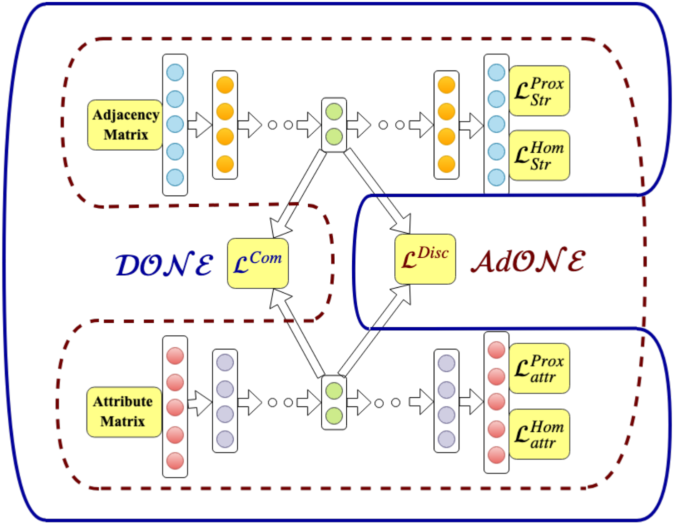
Moving on to **DONE**

Contributions of DONE [2]

- We have proposed an autoencoder based deep architecture (DONE) to minimize the effect of outliers for network embedding, in an unsupervised way.
- We use SGD, **along with the derived closed form update rules** for faster optimization of the parameters of the network.
- To the best of our knowledge, this is **the first deep architecture** for outlier aware attributed network embedding.
- A further extension of DONE is **AdONE** which is also a deep model

[2] Accepted in WSDM'20

DONE Architecture



Note that the hidden layers have non linear activations and hence this model is nonlinear unlike ONE

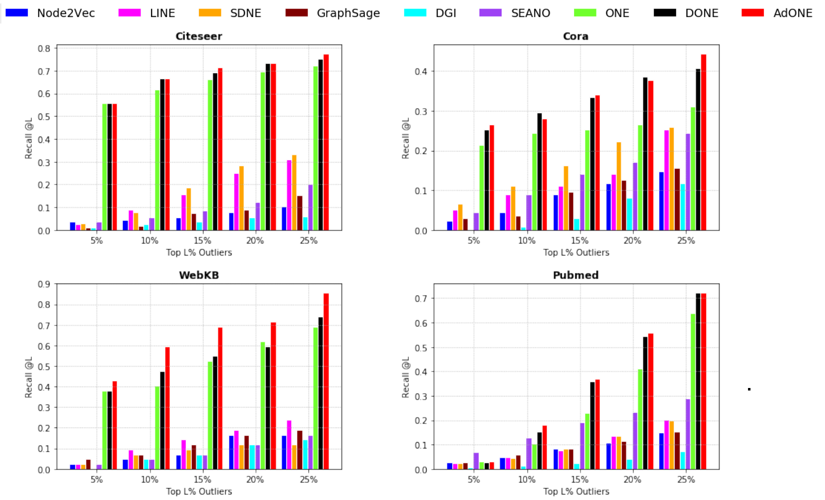
Further Details

We skip further details in the interest of time.

However some details noteworthy are

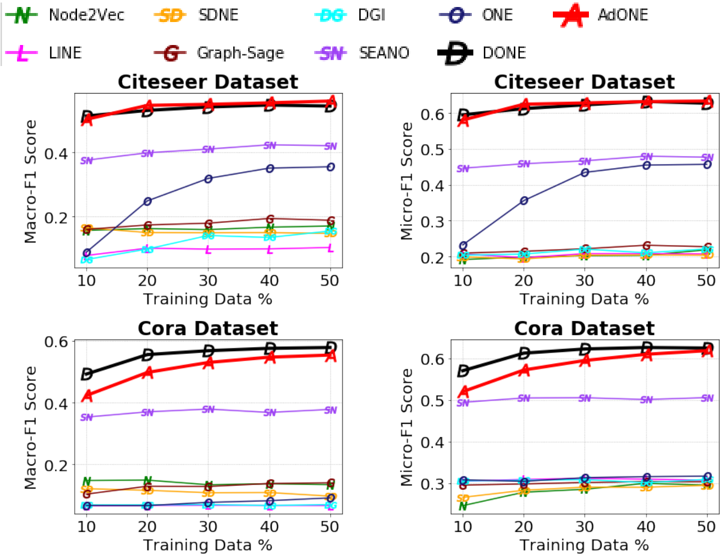
- DONE being non-linear model can capture non-linear intricacies in the dataset where ONE failed
- Because of stochastic gradient descent and tensorflow implemented massively parallel GPU based updates, DONE scales to larger datasets also

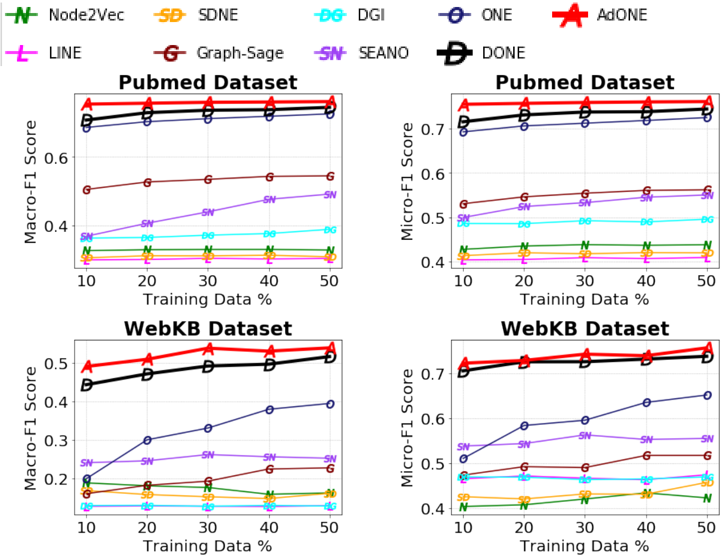
Outlier Detection Perfromance



Classification Performance

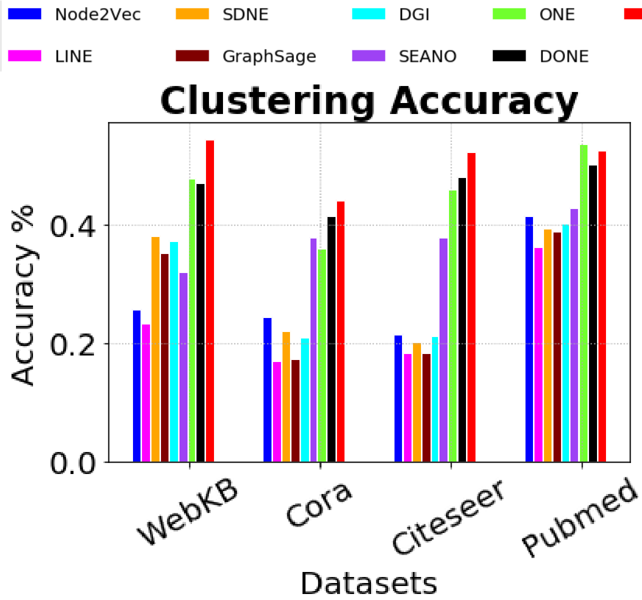
All results reported are obtained by running **Logistic regression** algorithm for classification





Node2Vec
SDNE
DGI
ONE
AdONE
LINE
Graph-Sage
SEANO
DONE
Pubmed Dataset
Macro-F1 Score
Training Data %
Pubmed Dataset
Micro-F1 Score
Training Data %
WebKB Dataset
Macro-F1 Score
Training Data %
WebKB Dataset
Micro-F1 Score
Training Data %

Clustering Performance



We evaluated DONE on the unseeded datasets also and found convincing results.
This validates, empirically though, that results are better not just because of the seeding process we did but because of the superior nature of out algorithm.

Publications out of our work

- ONE accepted in **AAAI'19**
- DONE and AdONE accepted in **WSDM'20**

Acknowledgements

- My sincere thanks to Sambaran B for collaborating with me for ONE
- My sincere thanks to Sambaran B and Saley Vishal Vivek for collaborating with me for DONE/AdONE

The End