

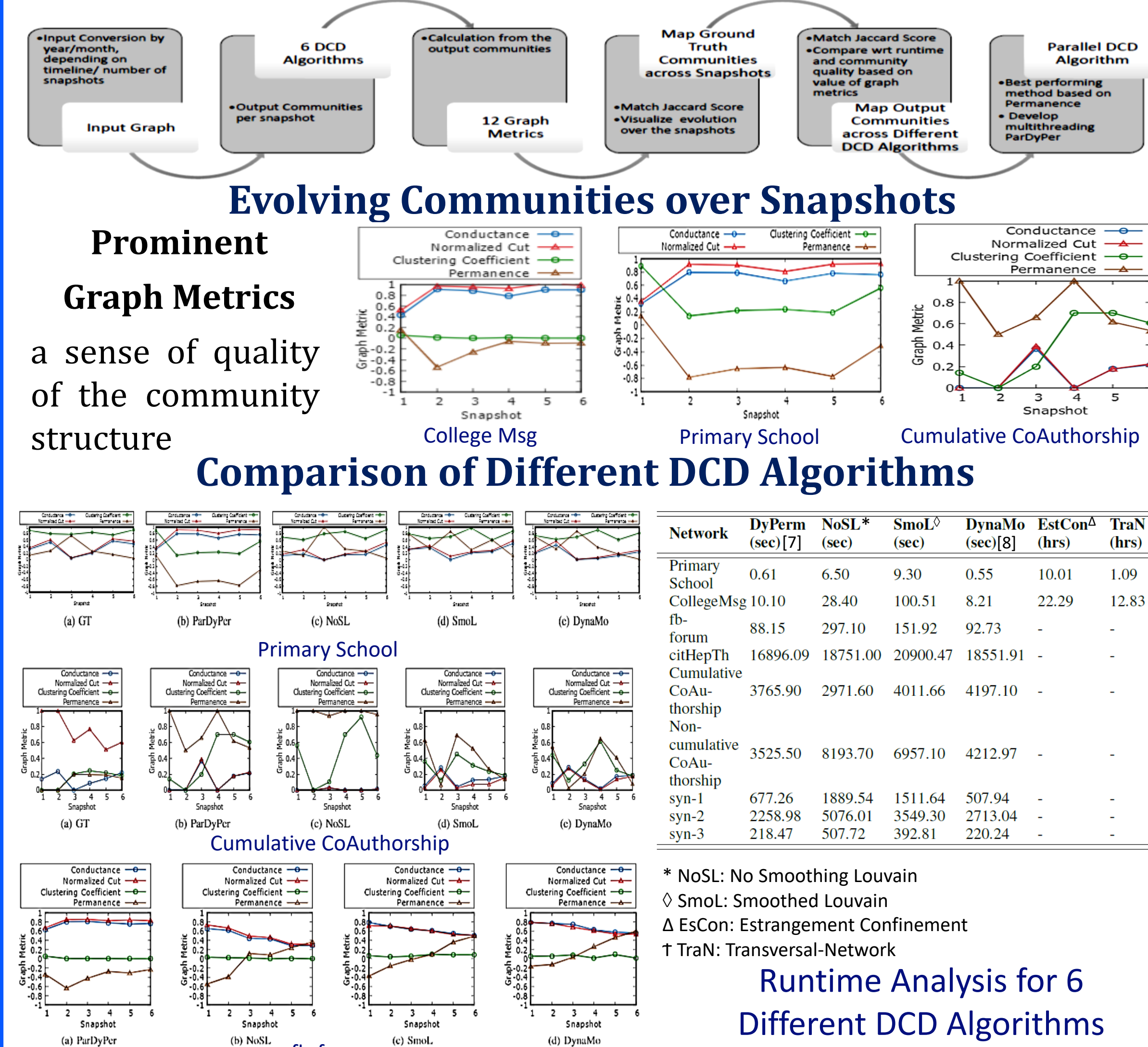
## Importance

- Computer Science: towards exascale High Performance Computing
- HPC and AI intersection
  - HPC using dense computer clusters to perform necessary calculations at blistering speeds to run the most advanced AI
- Processing large-scale graph data requires parallel computing
- Complex network analysis: exciting research area for different scientific domains
  - Online social media-Facebook, Twitter etc.
  - Biology-protein interaction, brain network
  - Online media, recommendation systems and many more

## Our Contribution

- Parallel Algorithm Design:
  - An optimized distributed-memory algorithm DPLAL with load balancing for scalable community detection in static graphs [1]
  - A multi-threaded algorithm for temporal community detection in dynamic graphs [2]
- Designing a scalable method for CD based on Graph Convolutional Network (GCN) via semi-supervised node classification using PyTorch with CUDA on GPU environment [3]
- Designing data parallel Sparse Deep Neural Network (DNN) using TensorFlow on GPU [4]
- Implementing Real-world Graph Mining Applications for large-scale dataset [5,6]

## Temporal Community Evolution



## Environment & Dataset

Louisiana Optical Network Infrastructure (LONI) QB2 cluster

- 1.5 Petaflop peak performance
- 504 compute nodes, 20 cores/node
- over 10,000 Intel Xeon processing cores @2.8 GHz.

## GPU: NVIDIA Tesla K20x GPU

- 2.6GHz, B/W: 250 GB/sec
- 6 GB GDDR5, SDRAM [24 64Mx16]

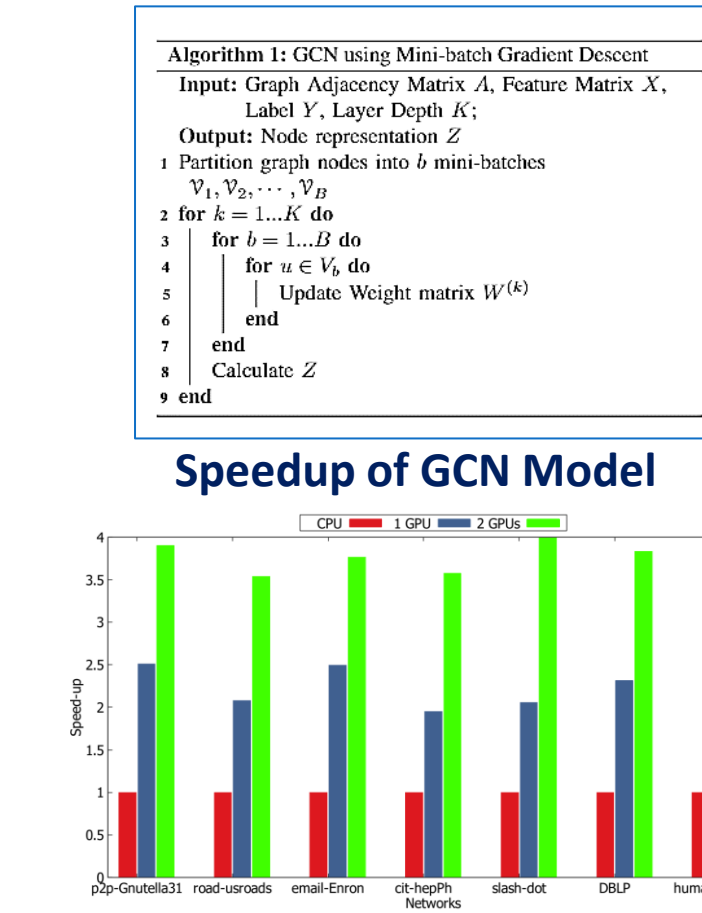
## Graph Dataset

- SNAP<sup>1</sup>
  - Network Repository<sup>2</sup>
- <http://snap.stanford.edu/data/index.html>
  - <https://networkrepository.com/>

## Performance Scalability in Neural Networks on GPU

### CD with Graph Convolutional Network (GCN)

- Traditional community detection algorithms
  - need to analyze the full network
  - computationally expensive
- Achieve the same or comparable result with less computational cost
  - Having community information of a part of the network, the rest communities predicted depending on those community

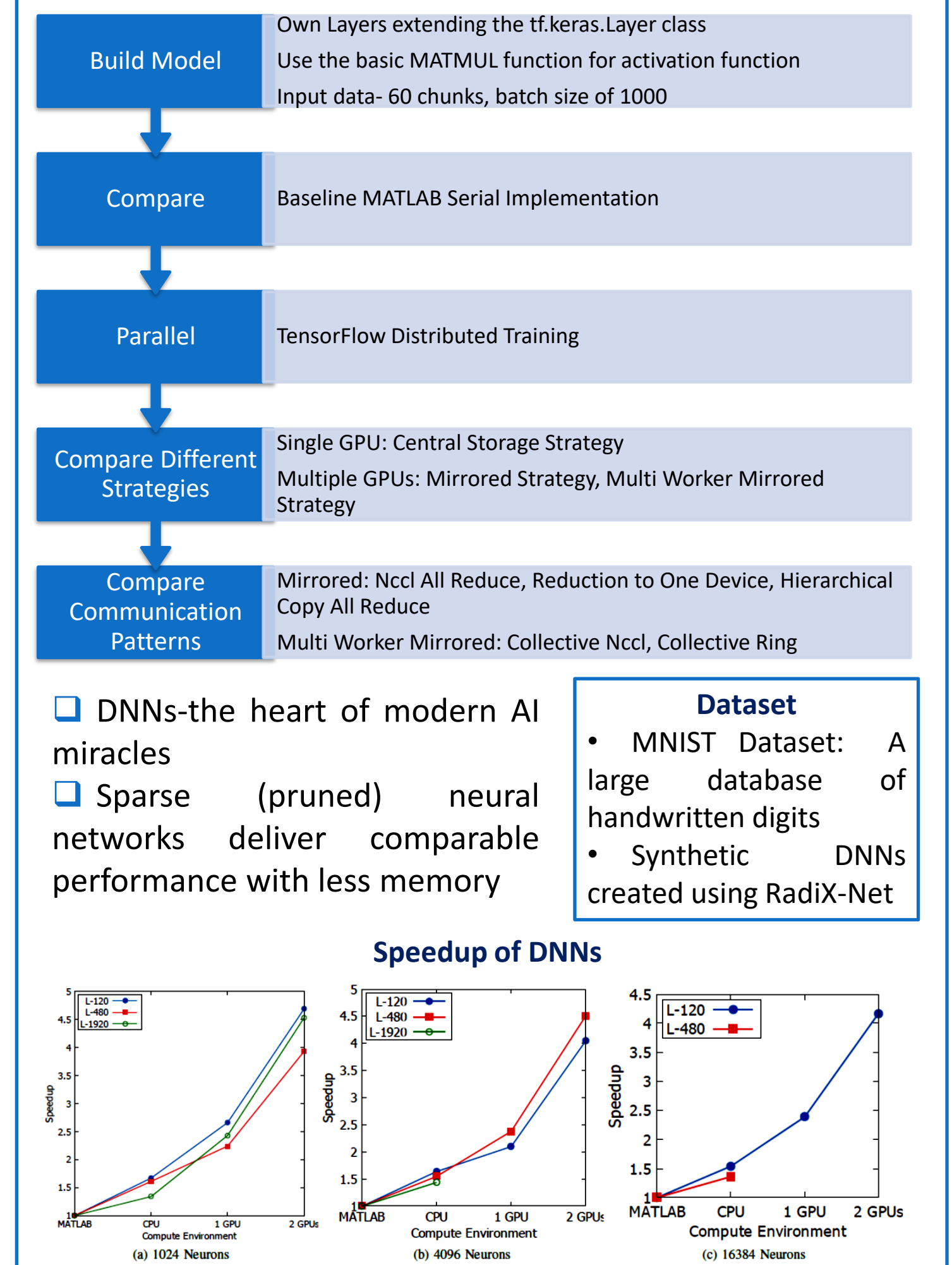


### GCN Model

- activation function ReLU
- Minibatch Gradient Descent: batch size 1200
- sampled by selecting nodes with dense neighborhoods
- Nodes tracked in consecutive two layers: avoid sparse connection problem

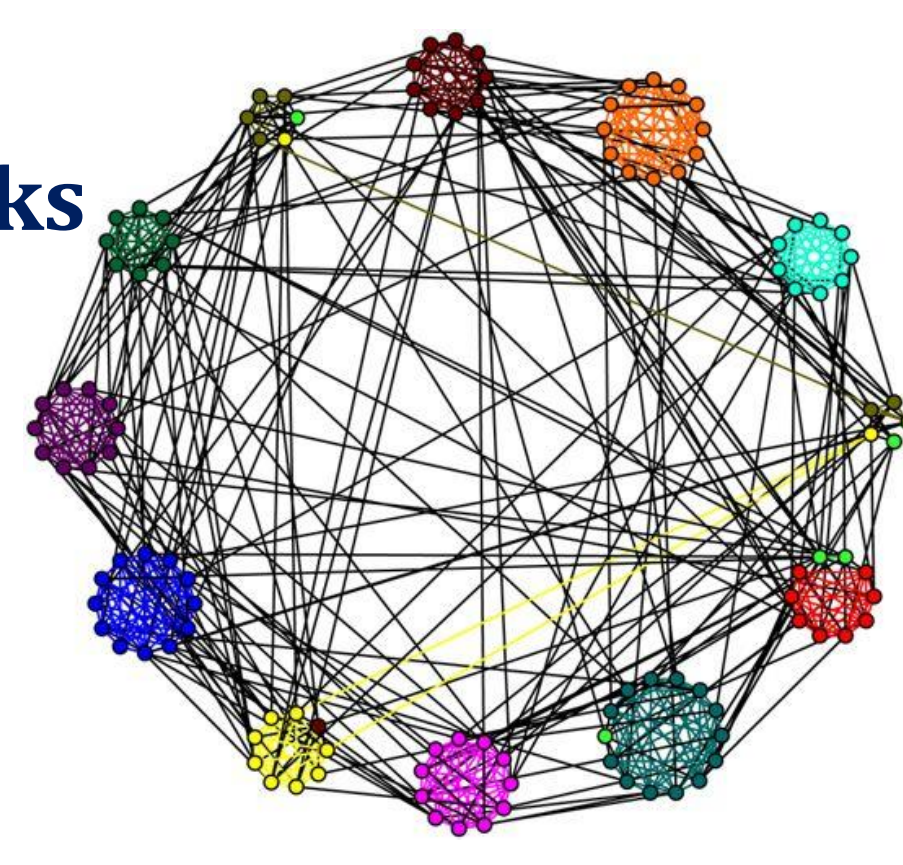
Method	arXiv Dataset Test Accuracy (%)	PPI Dataset F1 Score
MLP	55.50±0.23	N/A
Node2Vec	70.07±0.13	N/A
GCN	71.74±0.29	N/A
GraphSAGE	71.19±0.27	0.612
E		
Our Model	71.43±0.17	0.698

### Parallel Sparse DNN



## Parallel Community Discovery in Static and Dynamic Graphs

- Community detection (CD) is an important graph analysis kernel
- Communities reveal organizational structure of a system.
- Louvain [9] for Static Networks
- Detects community based on modularity optimization
- One of the best methods:
  - Computation time and
  - Quality of the detected communities



**Algorithm 1: DPLAL-Distributed Parallel Louvain Algorithm using Load-balancing**

**Data:** Input Graph G(V,E) [Edge List Format]

**Result:** (Vertex, Community) Pair

```

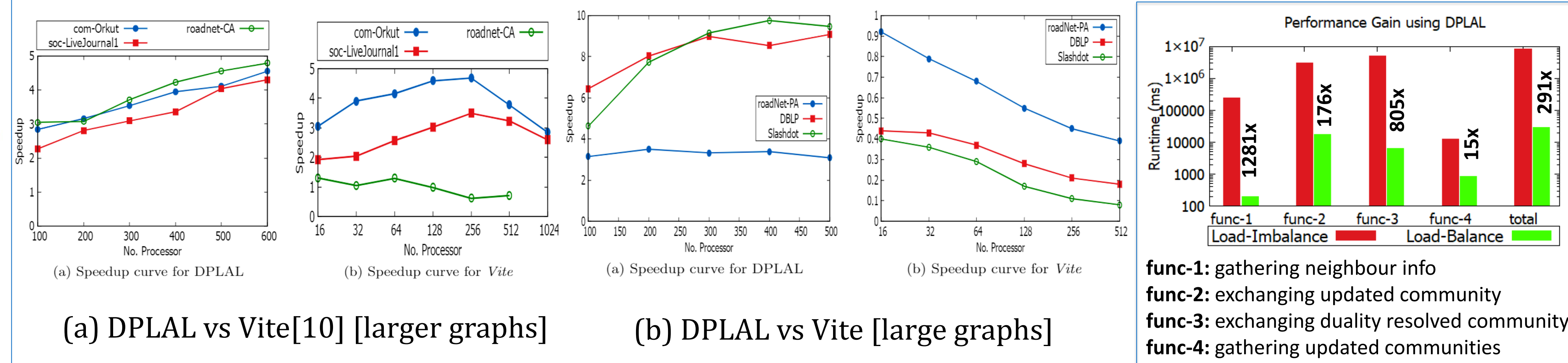
1 while increase in modularity do
2   G(V,E) is partitioned using METIS into p'
   partitions;
3   G'(V,E) pre-processed according to
   METIS-Output;
4   G'(V,E) converted to Adjacency List format to be
   given to each processor;
5   for Each processor Pi (executing in parallel) do
6     Gather_Neighbour_Info();
7     Compute_Community();
8     Exchange_Updated_Community();
9     Resolve_Community_Duality();
10    Exchange_Duality_Resolved_Community();
11    Find_Unique_Communities();
12    Compute_Modularity();
13    Generate_NextLevel_Graph();
14    if number_of_communities < i then
15      i ← number_of_communities;
16  end
17 end

```

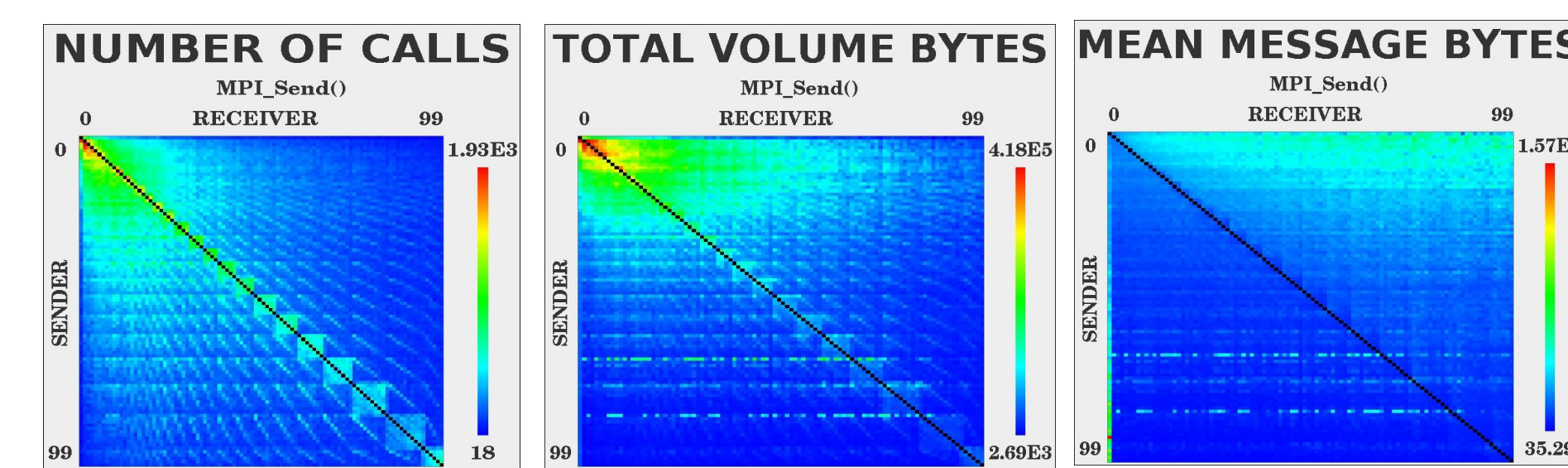
### Algorithms for Parallel Louvain

- Shared-Memory: OpenMP
- Distributed-Memory: MPI
- Hybrid: MPI + OpenMP
- DPLAL: MPI & METIS

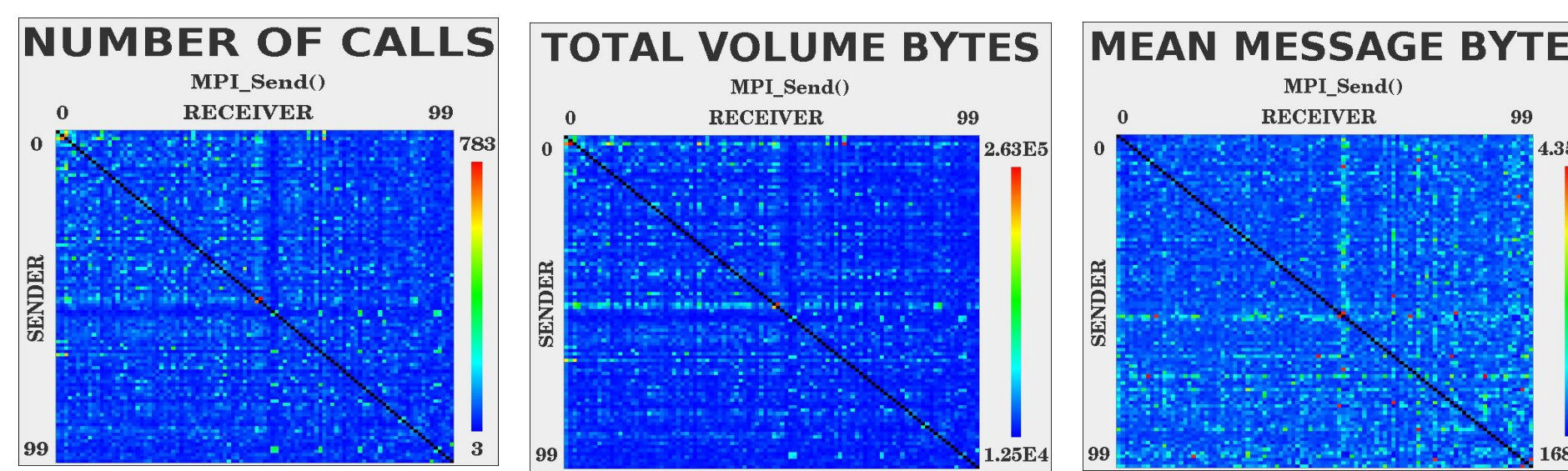
## Speedup factors of our Distributed Parallel Louvain Algorithm with Load-balancing (DPLAL)



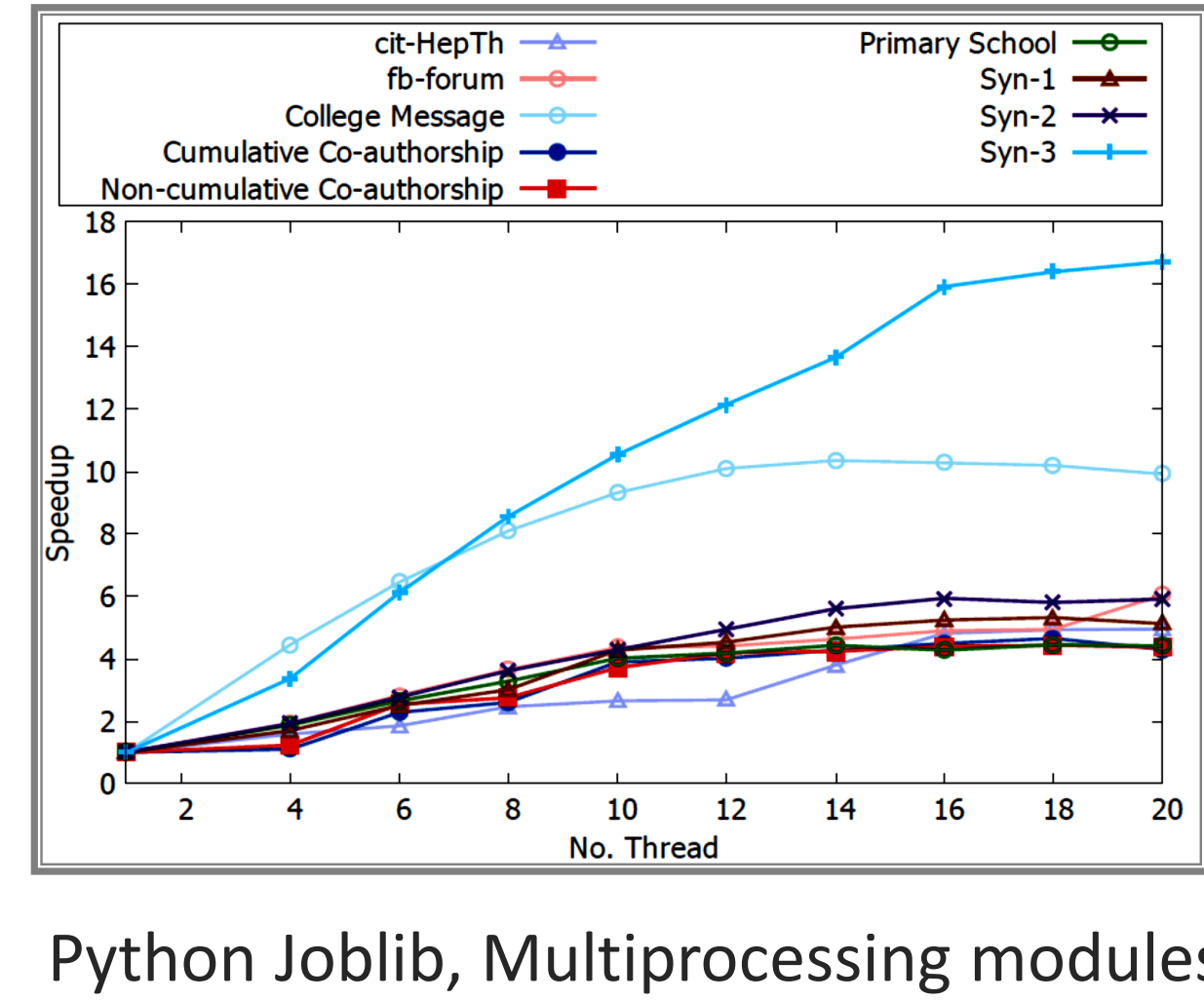
### MPI Communication in the Naive MPI-based Parallel Algorithm (Load-Imbalanced)



### MPI Communications in DPLAL (Load-balanced)



### Speedup factors of our Multi-threading parallel Temporal CD



## Temporal Community Detection

- Permanence, a vertex-based metric
- Advantages over other optimization metrics (i.e. modularity, conductance)
  - Local optimization rather than global optimization of the full network
  - No arbitrary tie-breaking (inaccurate or insignificant communities with high score) scenario like other functions
  - Optimization of modularity score NP-Hard Problem

Here,

$I(v)$  Internal connections of vertex  $v$

$E_{max}(v)$  Maximum connections to a single external community

$D(v)$  Total degree of vertex  $v$

Internal clustering coefficient [ratio of the existing connections and the total number of possible connections among the internal neighbors of  $v$ ]

$c_{in}(v)$

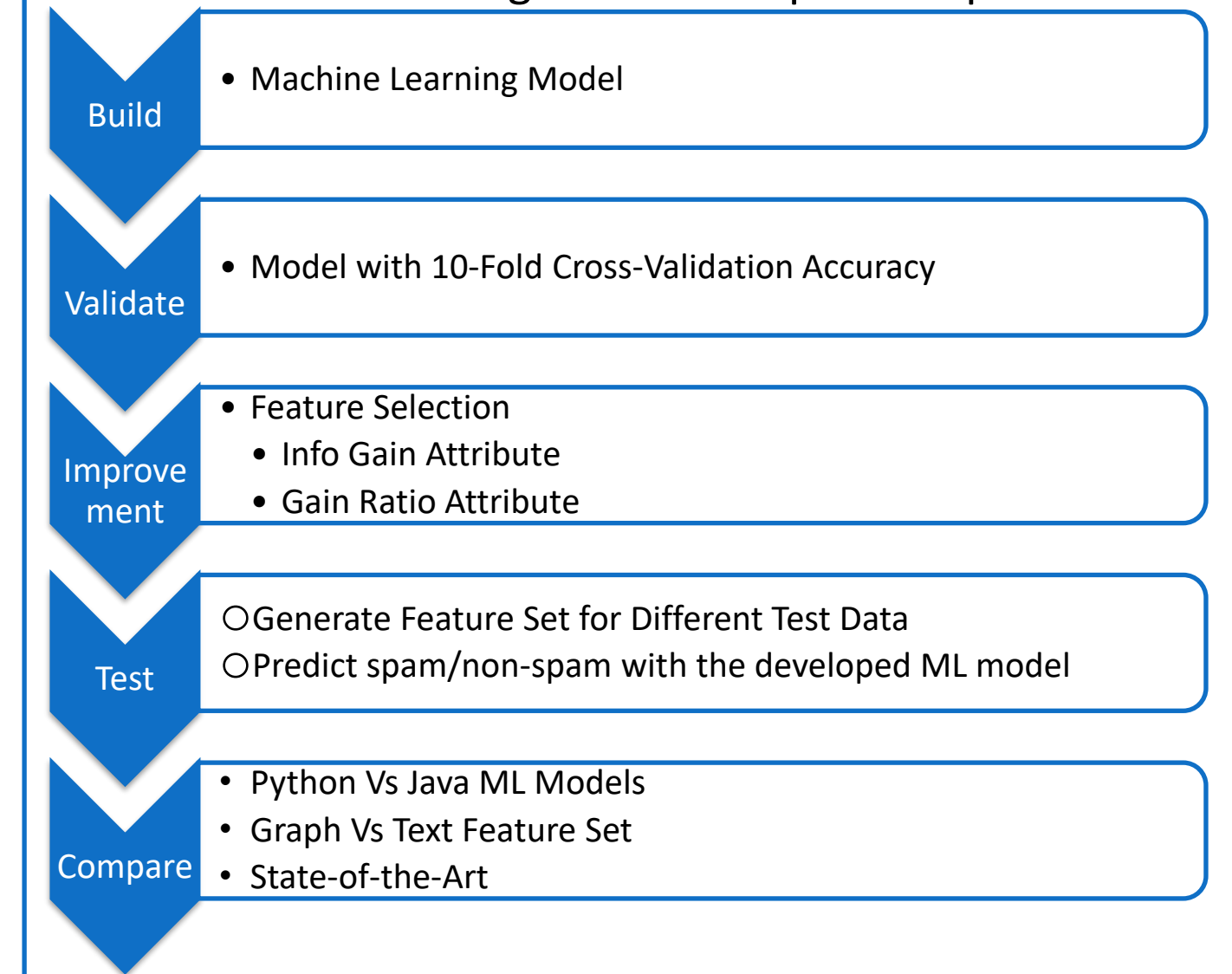
## References

- N.S. Sattar and S. Arifuzzaman, "Overcoming MPI Communication Overhead for Distributed Community Detection", Software Challenges to Exascale Computing, in Springer Communications in Computer and Information Science (CCIS) series, December 2018
- N.S. Sattar, S. Arifuzzaman, K.Z. Ibrahim and A. Buluc "Parallel Algorithm and Analysis for Understanding Evolving Community Structures in Temporal Graphs", Poster Presentation. SIAM Conference on Computational Science and Engineering (CSE21), Fort Worth, Texas, USA, February, 2021.
- N.S. Sattar and S. Arifuzzaman, "Community Detection using Semi-supervised Learning with Graph Convolutional Network on GPUs", proc. of 2020 IEEE International Conference on Big Data (IEEE Big Data 2020), Atlanta, GA, USA, December 2020
- N.S. Sattar and S. Arifuzzaman "Parallel Large Sparse Deep Neural Network on GPU", proc. of 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, USA, May 2020
- N.S. Sattar, S. Arifuzzaman, M. Zibran, and M.M. Sakib, "Detecting Web Spams in Webgraphs with Predictive Model Analysis", proc. of 2019 IEEE International Conference on Big Data (IEEE Big Data 2019), Los Angeles, CA, USA, December 2019.
- N.S. Sattar and S. Arifuzzaman, "Covid-19 Vaccination Awareness & Aftermath: Public Sentiment Analysis on Twitter Data and Vaccinated Population Prediction in USA", Appl. Sci. 2021, Volume 11, Issue 13, 6128, June 2021
- P. Agarwal, R. Verma, A. Agarwal, and T. Chakraborty, "Dyperm: Maximizing permanence for dynamic community detection," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 437-449.
- D. Zhuang, M. J. Chang, and M. Li, "Dynamo: Dynamic community detection by incrementally maximizing modularity," IEEE Transactions on Knowledge and Data Engineering, 2019.
- M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E, vol. 69, no. 2, p. 026113, 2004.
- S. Ghosh, M. Halappanavar, A. Tumeo, A. Kalyanaram, A.H. Gebremedhin, "Scalable distributed memory community detection using vite". In: 2018 IEEE High Performance extreme Computing Conference (HPEC), pp. 1-7. IEEE (2018).<https://doi.org/10.1109/HPEC.2018.8547534>

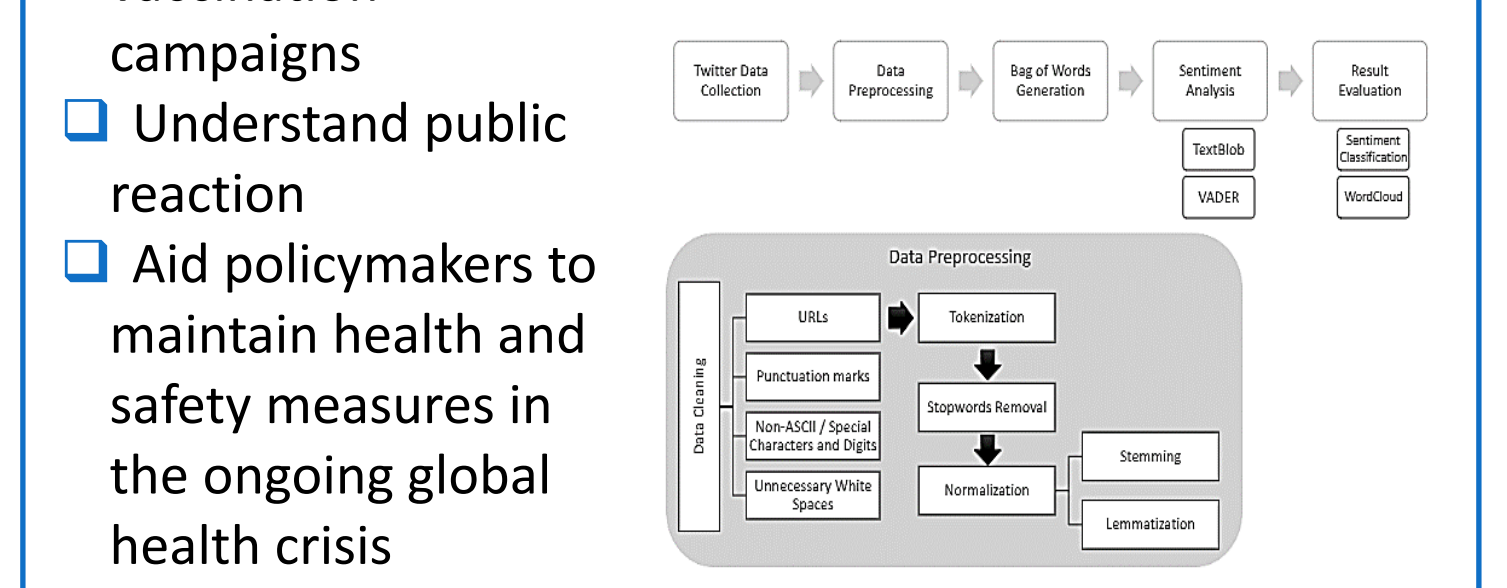
## Graph Applications with Machine Learning

### Detection of Webspams in WebGraphs

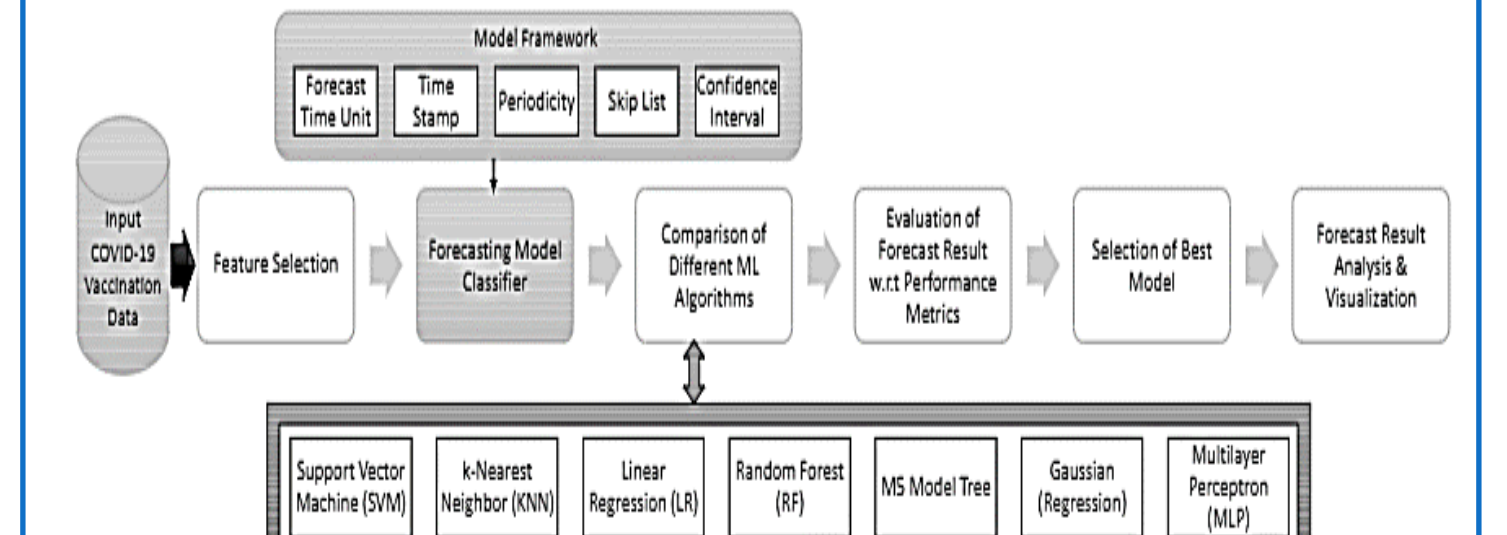
- Webgraph, great research potentials concerning web security: Detecting web spam-security vulnerabilities
- Detecting web spam based on graph features: Using Emerging graph mining techniques in a scalable manner
- Very few labelled dataset for spam detection: need for a machine learning classifier to predict spam



### Covid-19 Sentiment and Prediction on Twitter Data

- Help health and government officials to better comprehend and plan Covid-19 vaccination campaigns
  - Understand public reaction
  - Aid policymakers to maintain health and safety measures in the ongoing global health crisis
- Sentiment analysis on Twitter Data**
- A database of around 1.2 million tweets collected across five weeks of April-May 2021
- 

### COVID-19 vaccination forecasting model using WEKA



## Conclusion

- Presented different parallel implementations of community detection algorithms for static and temporal networks
- Identified bottlenecks for different methods of designing parallel implementations and provide an optimized solution
- Achieved 12x speedup for static CD, ~4-18x speed-up for temporal CD, 4x performance gain on GPU using PyTorch with CUDA for GCN) via semi-supervised node classification
- Identified a subset of graph metrics to understand the evolving community structure quality for different temporal networks
- Designed a scalable solution to the Sparse DNN challenge with 4.7x speedup on GPU using data parallelism of TensorFlow
- Detected web spam from webgraph and devised feature selection method from the graph data, an important application of graph mining
- Performed sentiment analysis on social network, Twitter and concluded important insights on public attitude towards COVID-19 vaccinations

## Acknowledgements

This work has been supported by Louisiana Board of Regents RCS Grant LEQSF(2017-20)-RDA- 25. (17-SC-2) and University of New Orleans ORSP Award CON000000002410. Some of the results are generated in collaboration with Dr. Aydin Buluc and Dr. Khaled Ibrahim at Lawrence Berkeley National Laboratory when the authors have started working under 2019 SRP Fellowship program.