PROCESSOR PARTITIONING FOR ANOMALY DETECTION USING MACHINE LEARNING

Group: Anomaly Detection for Cybersecurity

Ruiyang(Wendy) Wang

John Korah, Professor of Computer Science

Contents

Our Group’s Motivation

In today's digital world, cyberattacks are growing more complex and harder to detect. To make deep learning more efficient , we need methods that can speed up computation without sacrificing accuracy.

Autoencoder neural networks offer a more adaptive solution by learning normal traffic patterns and identifying anomalies based on reconstruction error.

Dataset for testing:
CIC-IDS 2017.

Challenges

-Autoencoder not implemented yet

-Building the neural network (DNN) from scratch

-Accuracy validation

Research Objective

-Implement vertical partitioning using MPI.

-Split input features across processors.

-Run forward pass in parallel on each process.

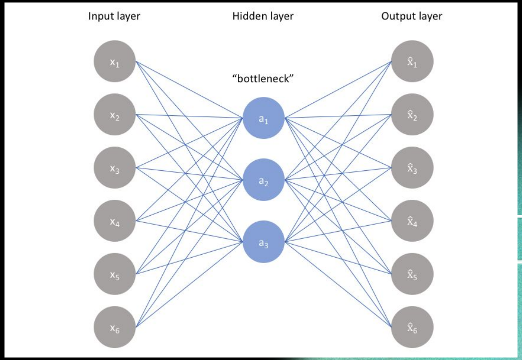
-Measure and compare runtime vs. serial execution.

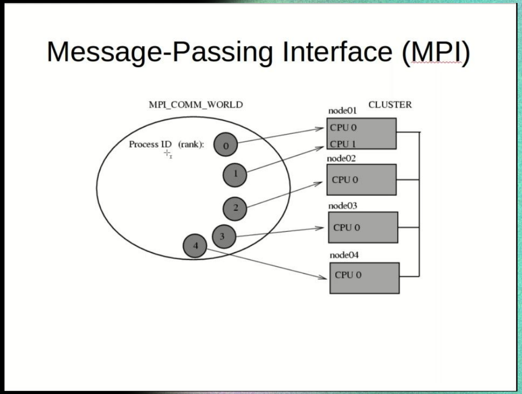
Literature review

-Autoencoder: a neural network designed to reconstruct its own input.

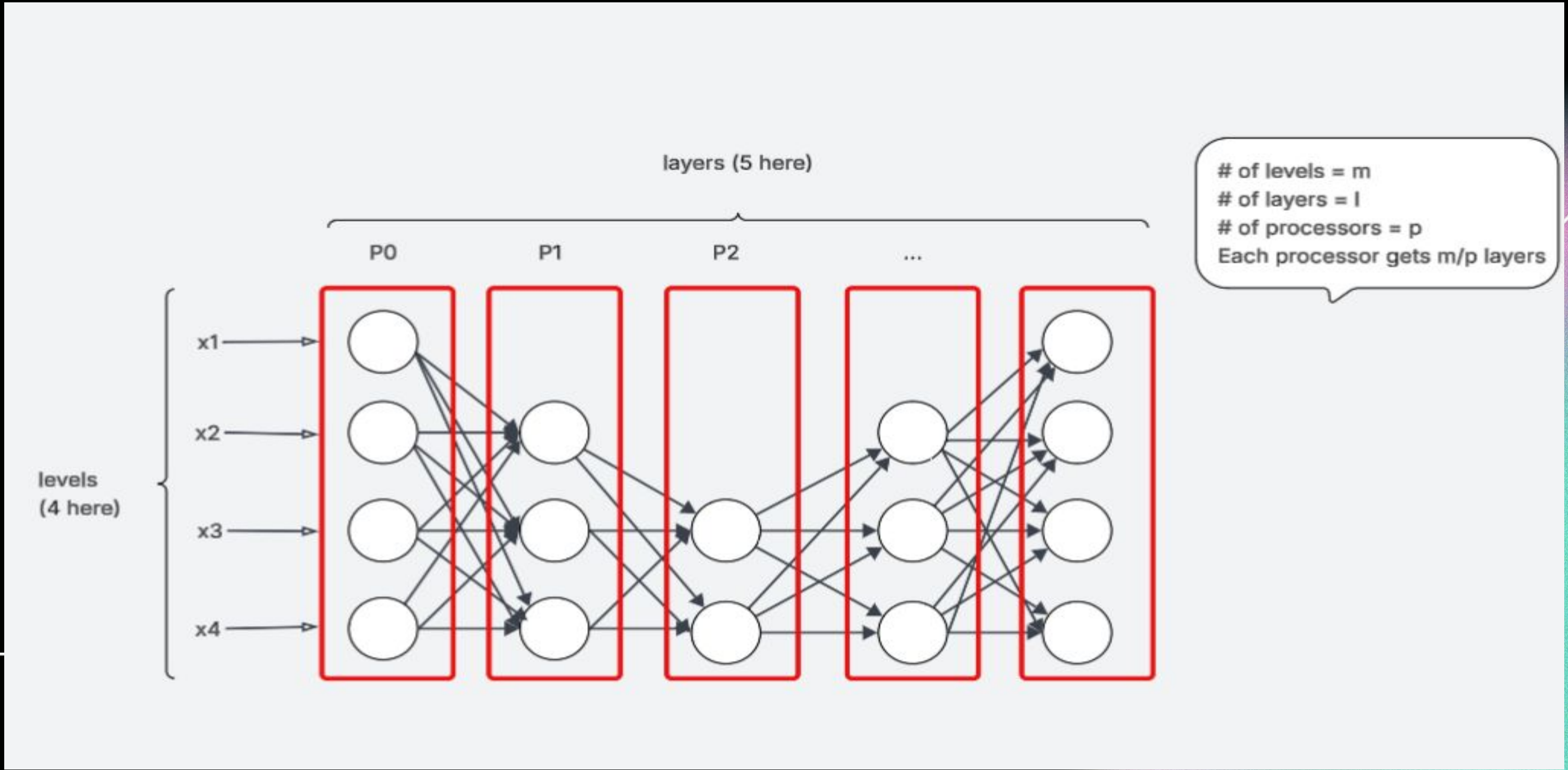
-CIC-IDS 2017, created by the Canadian Institute for Cybersecurity, is a widely used benchmark dataset for evaluating Intrusion Detection Systems (IDS).

-Message Passing Interface: a standard for writing programs that can run in parallel across multiple processors.





Research Methodology Vertical Partition



Research Methodology Pseudocode

```
Start MPI
Get this processor's rank and total number of processors

Load and normalize input data (X)

Split input columns among processors
Get local_data = X[:, my assigned columns]

Build local model using shared weights and biases

Run forward pass: local_output = model.forward(local_data)

Use MPI Allgather to collect outputs from all processors

If rank 0:
    Combine all local outputs into final output
    Print final output shape
```

View Code here:

Runtime Performance

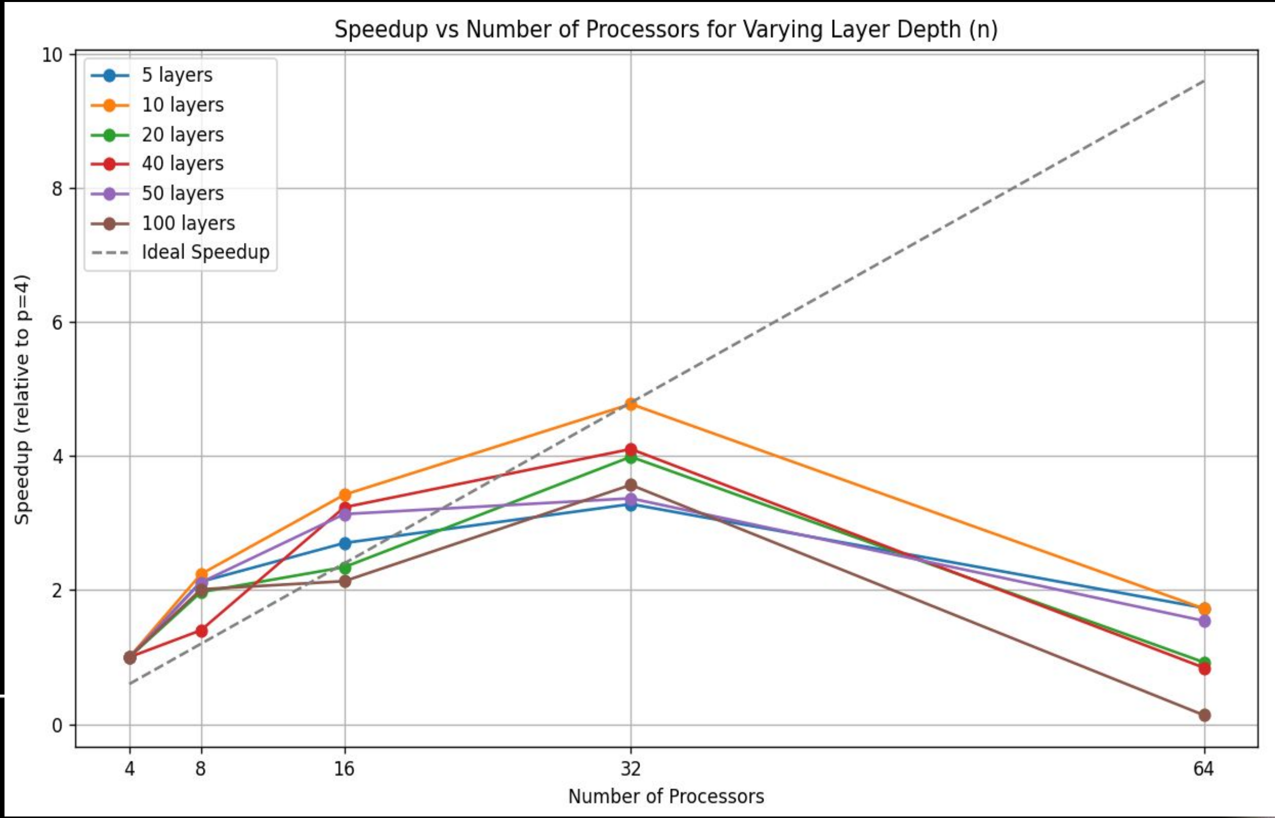
[Serial] Forward pass time: 0.102218 seconds

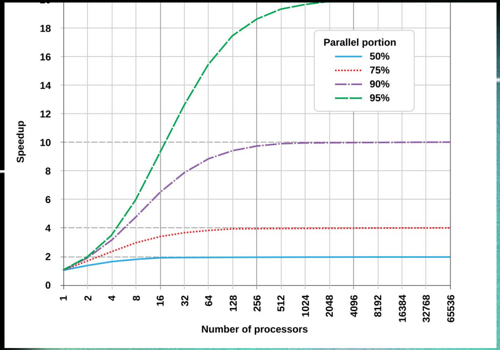
[Vertical Partition with MPI] (my record sheet)

n	p = 4	p = 8	p = 16	p = 32	p = 64
5	0.001544	0.000726	0.000571	0.00047	0.00089
10	0.002719	0.001214	0.000794	0.000569	0.001577
20	0.003863	0.00196	0.001648	0.000968	0.004186
40	0.007387	0.005268	0.002282	0.001799	0.008772
50	0.009178	0.004339	0.002927	0.002723	0.005952
100	0.01533	0.007617	0.007182	0.004293	0.114629

Test dataset: CIC-IDS 2017

Runtime Performance - Speedup





$$\text{Speedup} = \frac{\text{Serial Time}}{\text{Parallel Time}}$$

Results and Findings

-Implemented forward pass using MPI vertical partitioning.

-Reduced runtime significantly (up to 5.5× speedup)

-Best performance achieved around 16–32 processors

-Vertical partitioning is more effective than serial

Future Work

-Improve model accuracy

-Compare partitioning strategies

-Add training and anomaly detection

-Run on HPC cluster

Thank You For listening

Q&A

Sources

Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. International Journal of Engineering & Technology.

Chandola, Varun, et al. "Anomaly Detection: A Survey: ACM Computing Surveys: Vol 41, No 3." ACM Computing Surveys, 30 July 2009, dl.acm.org/doi/abs/10.1145/1541880.1541882?casa_token=fbNGVqgPwUUAAAAA%3AdGUVW8oFQxYn--fPjKf28PVwR092Fq-92cDjHrjsUaOePzC_cOCohNdBwBOj4cww4jI-tgtVqyr6.

Yaseen, Asad. "The Role of Machine Learning in Network Anomaly Detection for Cybersecurity." Sage Science Review of Applied Machine Learning, 6 July 2023, journals.sagescience.org/index.php/ssraml/article/view/126.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press