# Malware Analysis using Data Mining Techniques on GPGPU

Submitted to:

Dr. S K Sahay

Mr. Hemant Rathore

By:

Mayank Chaudhari

2016H1030014G

# Contents

- Motivation
- Objectives
- Data Source
- Methodology
- Result Analysis
- Conclusions

# Motivation

- The day by day increasing threat due to malwares
- Malware creation is easier than before
- My previous study over malware analysis

# Objective

- Perform comparative study of different classifiers and their parallel implementations
- Develop an approach to classify malwares and benign using GPU

# Data Source

- Malware dataset :
  - The malware dataset is taken from malicia-project
  - It contains 11088 samples
- Benign dataset:
  - The benign dataset is collected from different systems verified by virustotal
  - It contains 2900 samples

# Methodology

- In this project I focused on neural network implementation
- Data preprocessing
- Data normalization
- Designing deep neural networks
- Evaluation of different models

# Result Analysis

- The computational intensive nature of neural net makes it slow
- But it is able to out-perform Naïve Bayes in tems of accuracy
- There is a trade of in accuracy between computation time
- In case of our dataset a 3 layer architecture is sufficient
- Using a ReLU or PReLU function performs better than all sigmoid and tanh
- Increasing no of epochs increases accuracy due to overfitting so dropout needed

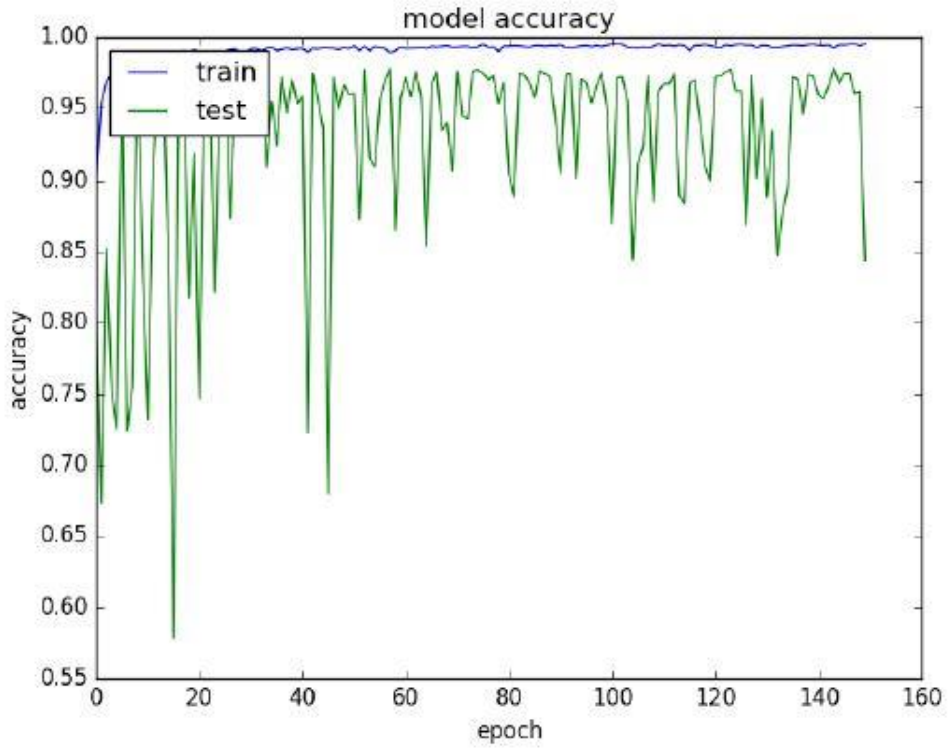



*figure 2: 4-layer network with dropout showing train test accuracy*

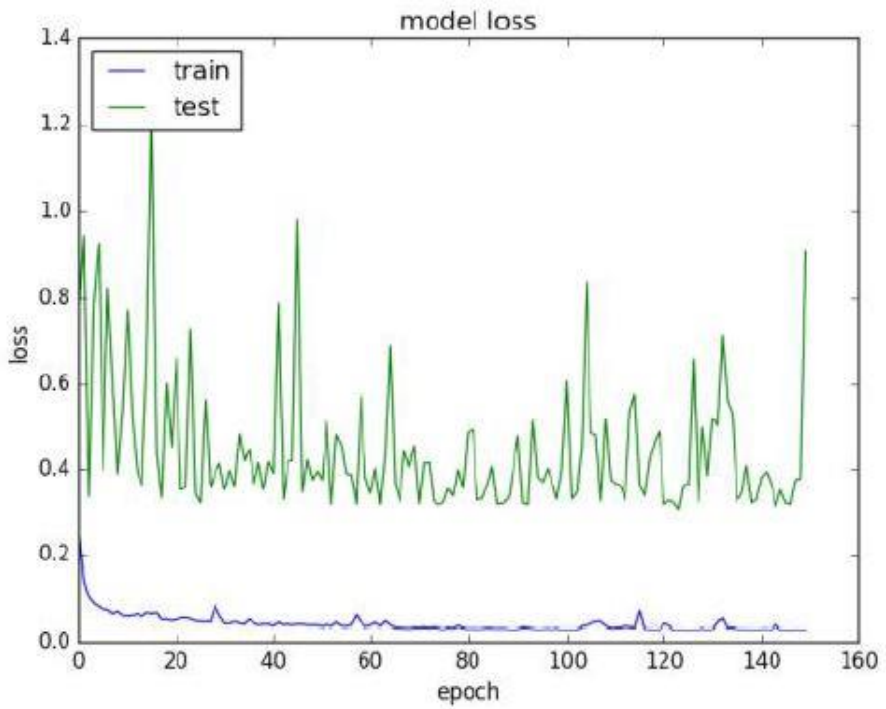



*figure 3: 4-layer network with dropout showing train test loss variation*

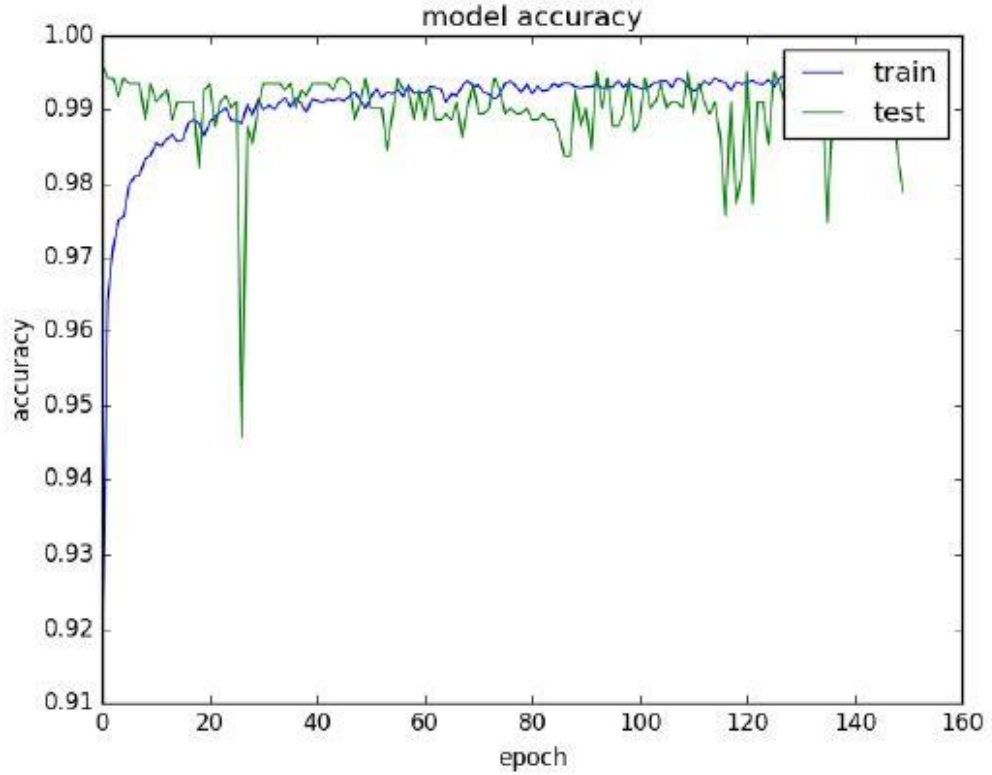



*figure 4: 3-layer network with dropout showing train and test accuracy*

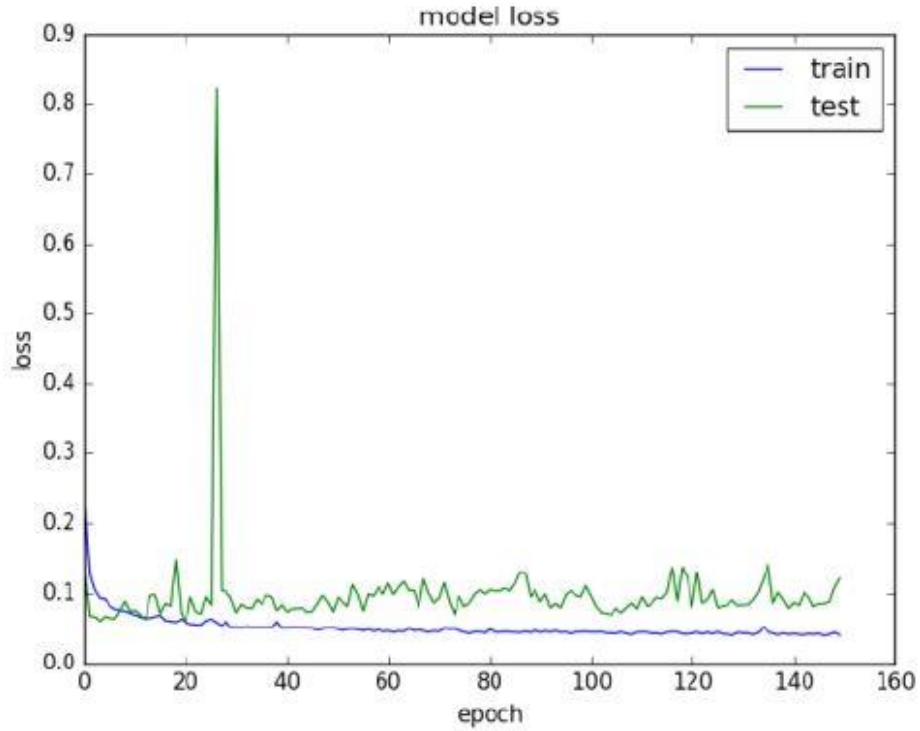



*figure 5: 3-layer network with dropout showing train-test loss variation*

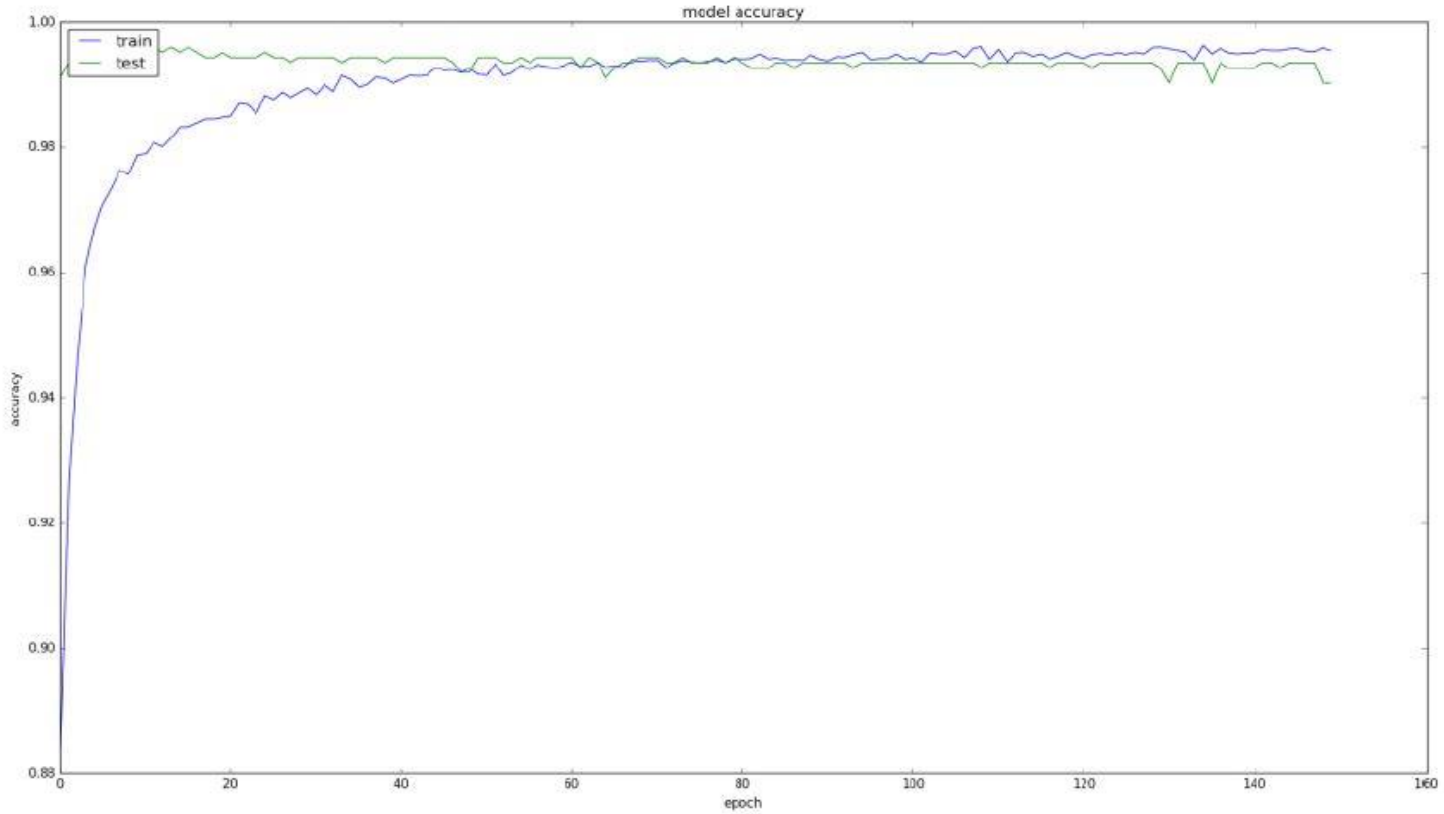



*figure 6: 3-layer design with dropout rate of .50*

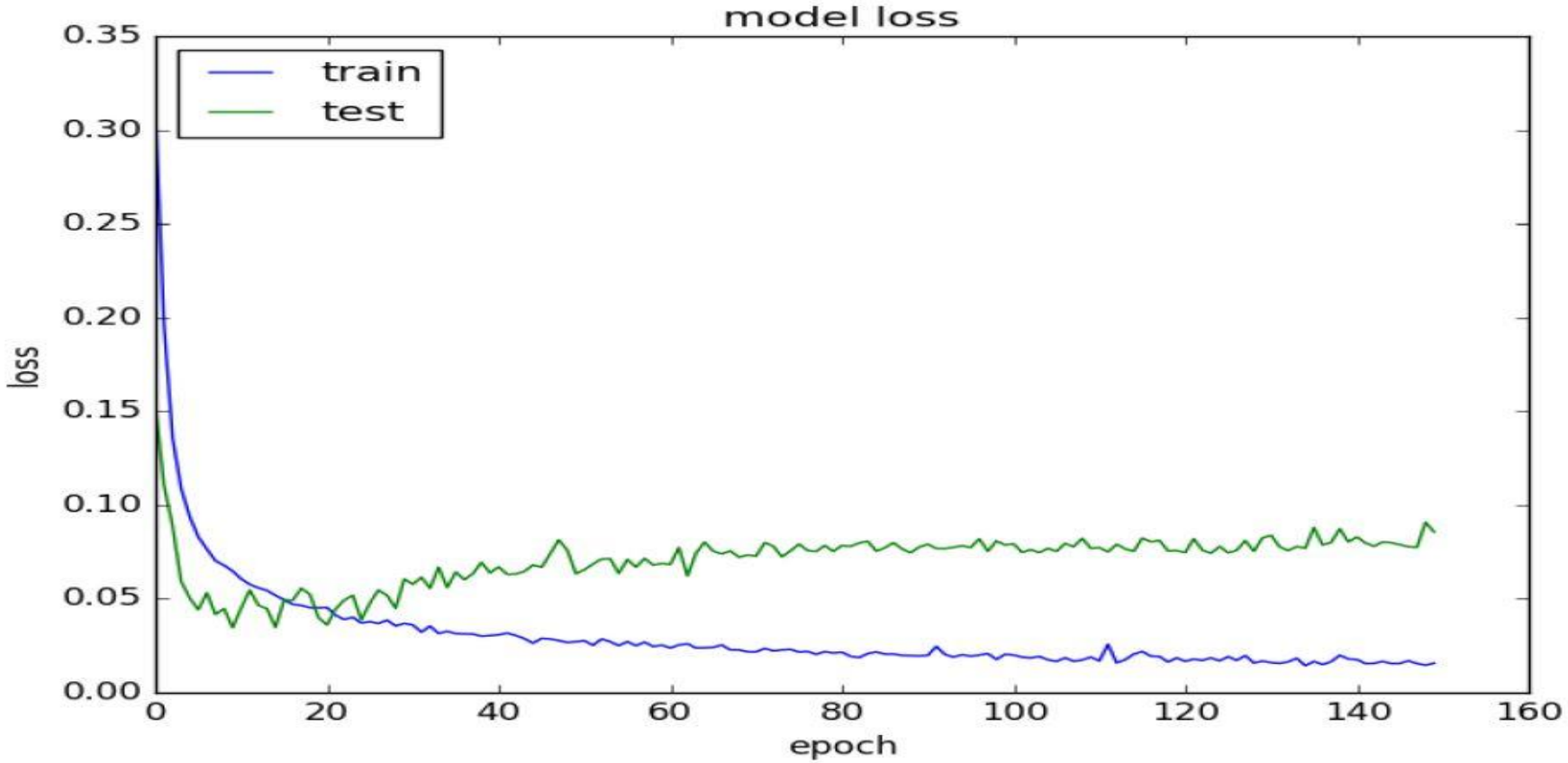
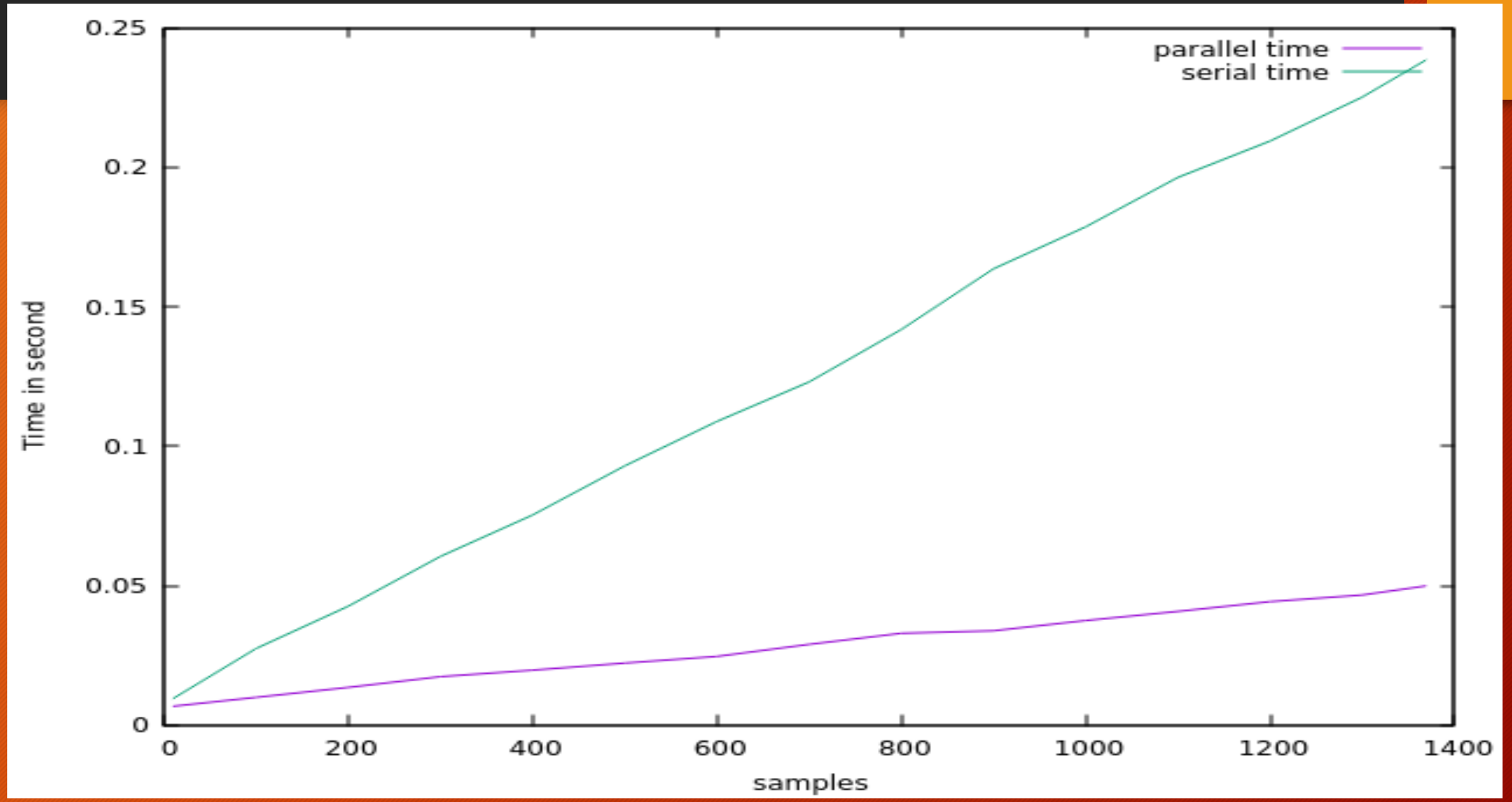
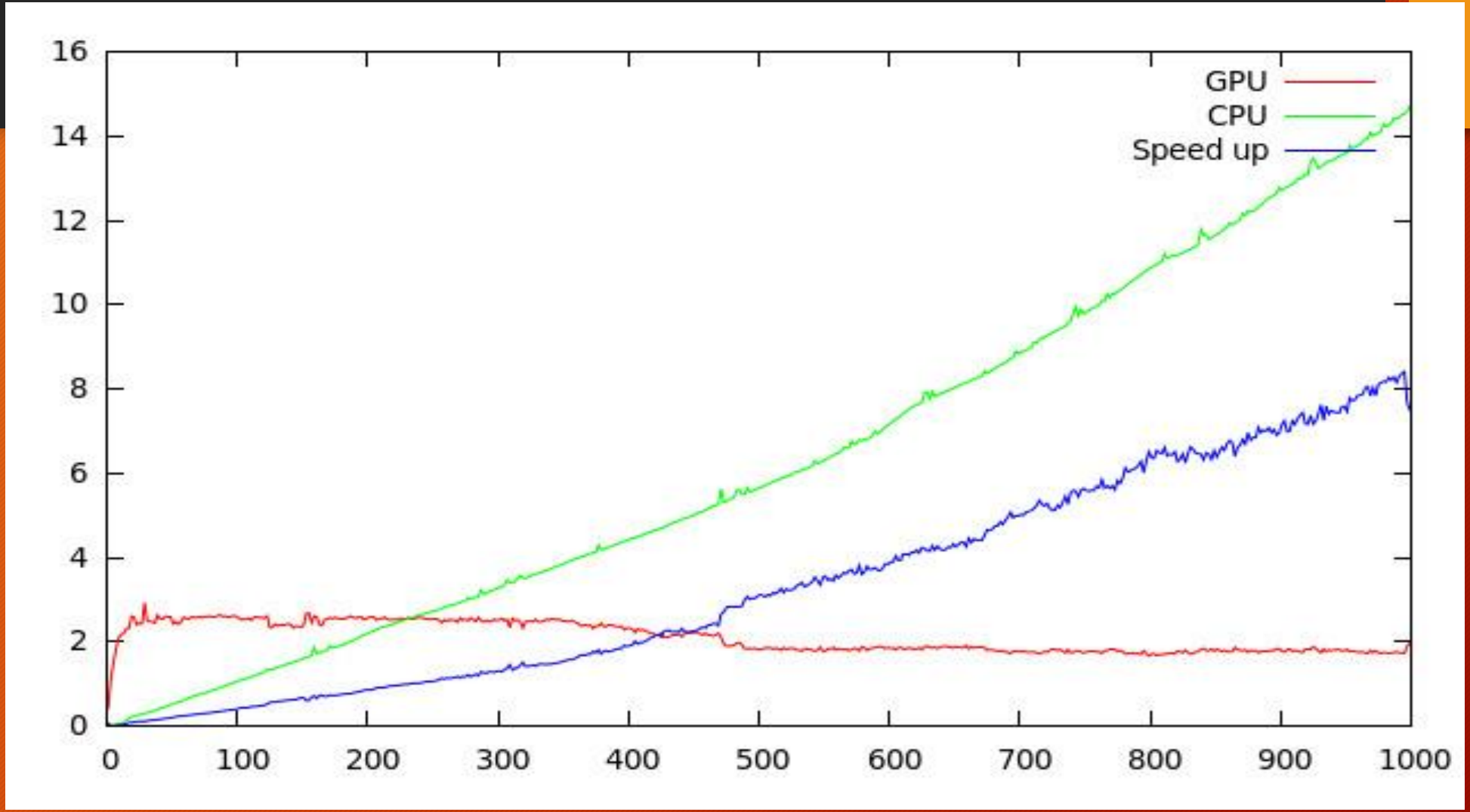



*figure 7: 3-layer design with dropout rate of .50*

| NO of layers hidden layers | Accuracy Testing(avg) | Best False Positive rate | Worst False Positive rate | 10-fold cross-validation |
|---|---|---|---|---|
| 1 sigmoid layer | 91% | 1% | 22% | ---------- |
| 1 sigmoid and 1 PReLU layer | 96.97% | 4.1% | 5.2% | 97.53(+/-58) % |
| 1 sigmoid and 2 PReLU layers | 97.77% | 3.0% | 4.7% | 98.10(+/-0.85) % |


parallel time
serial time
Time in second
samples
0.25
0.2
0.15
0.1
0.05
0
0
200
400
600
800
1000
1200
1400


GPU
CPU
Speed up
0
2
4
6
8
10
12
14
16
0
100
200
300
400
500
600
700
800
900
1000

# Conclusions

- Neural networks can provide better accuracy then Niave Bayes method
- The added testing computational time is not very huge
- Reinforcement learning support can be added easily
- Some other approaches needs to be explored which we encountered during our study

# THANK YOU