**HPE IN-SEMESTER PROJECT REPORT**

Team members

1.)Kancharla N V V S S Pavan Kumar

BTech 3rd year CSE

MANIPAL INSTITUTE OF TECHNOLOGY

Ph no: +91 9912897750

Mail Id: [pavanknvs@gmail.com](mailto:pavanknvs@gmail.com)

2.)Kosuri Venkata Phanendra Varma

BTech 3rd year CSE

MANIPAL INSTITUTE OF TECHNOLOGY

Ph no: +91 8501904248

Mail Id: [phanendravarma3@gmail.com](mailto:phanendravarma3@gmail.com)

**INTRODUCTION**

In this project we need to verify and validate a paper [2] named “A Preliminary Study of

Neural Network-based Approximation for HPC Applications”. This paper [2] mainly

concentrates on two concepts one is Newton Raphson and second is LJ Potential from

molecular dynamics , these two algorithms takes most of the computation time in their

program so they wanted to replace it with neural network models . They proposed Some NN

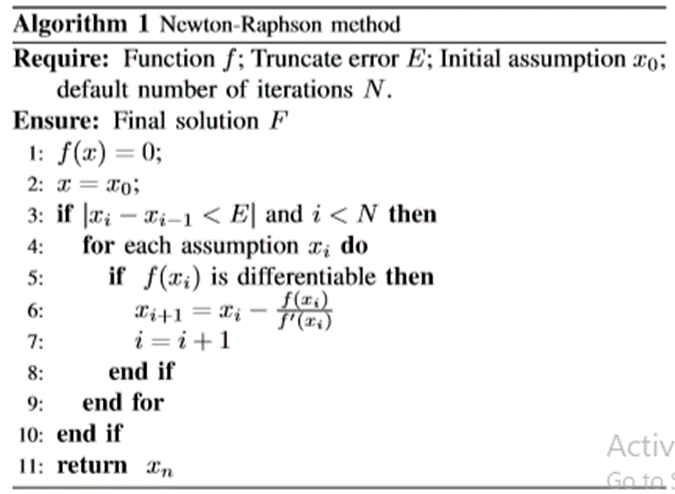
models to replace these two algorithms and our aim is to verify and validate those models

and if those models are not giving satisfactory results when we compared those results with

results in paper we can come up with a NN model which gives better accuracy.

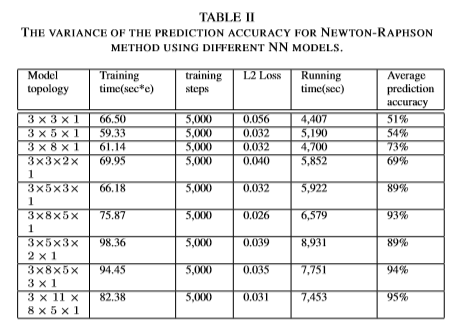
**Newton Raphson Model**

Algorithm for generating Newton Raphson Dataset



In the following table we have results of different neural network models for replacing

Newton Raphson code.



we implemented all the models with different activation functions ,regularization

techniques, optimizers except last model most of other models failed to give similar results

which they mentioned in paper, in those failed models some are giving similar results when

our data set is confined for example the input for this model is a, b , c when all three

variables are positive then some models are giving satisfactory results but it can’t be used to

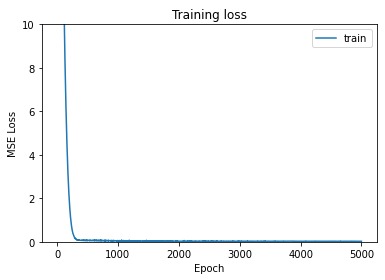
replace it with original algorithm because those variables can be negative also, but when it

comes to last model 3\*11\*8\*5\*1 we got training MSE loss 0.03 and average prediction

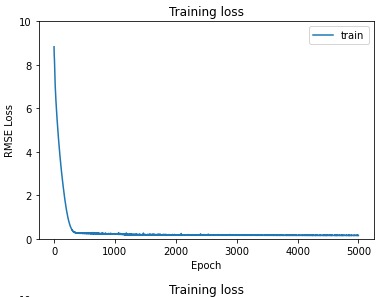
accuracy is 95% which is same as mentioned in paper

All the results of topology (3\*11\*8\*5\*1 ) are mentioned below:-

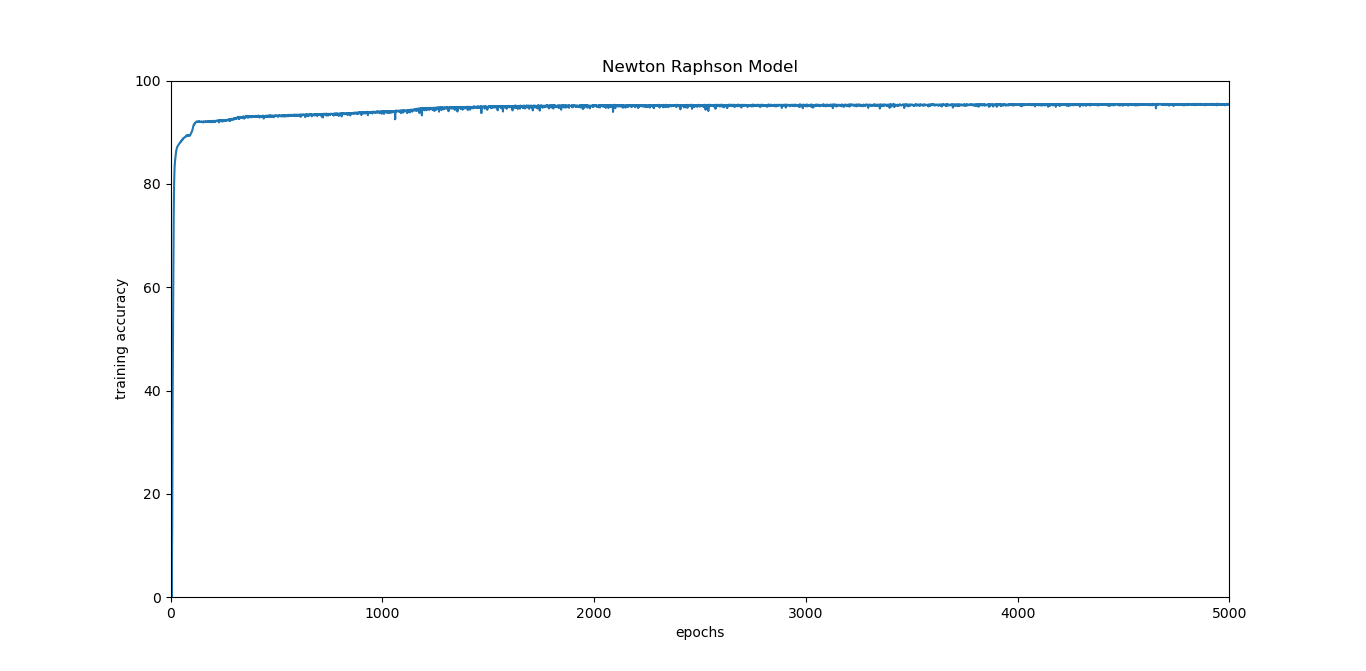
MSE Training loss of this model



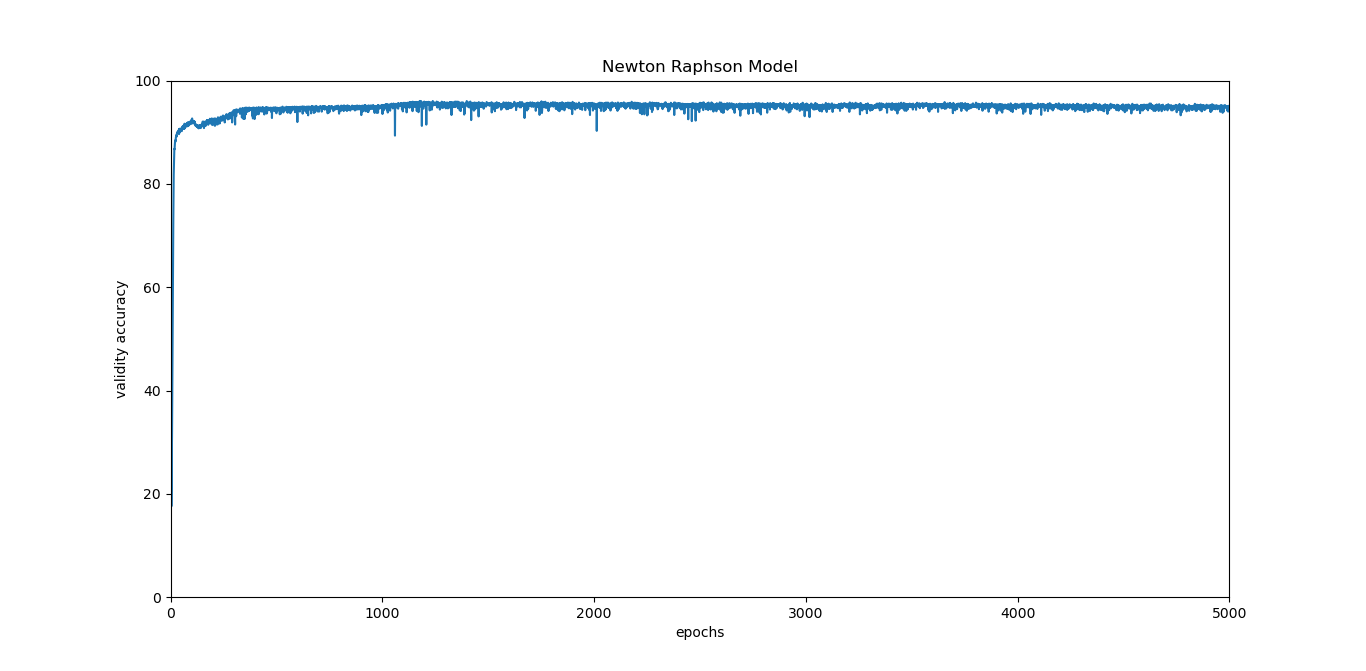
RMSE Training loss of this model



Training accuracy of this model



Validity accuracy of this model

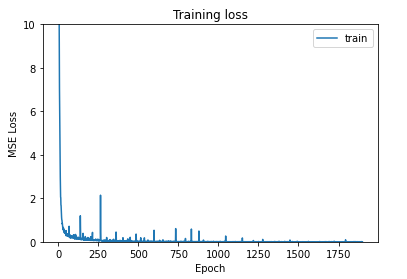


We tried to increase the accuracy of the above model. For the topology 25\*40\*30\*25\*1 we

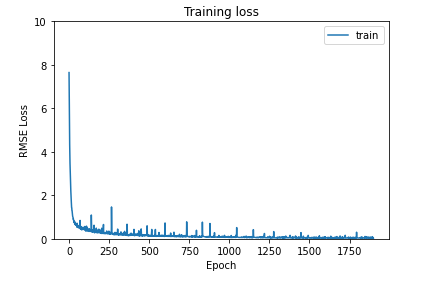
have achieved accuracy close to 99%.

The results of topology (25\*40\*30\*25\*1 ) are mentioned below:-

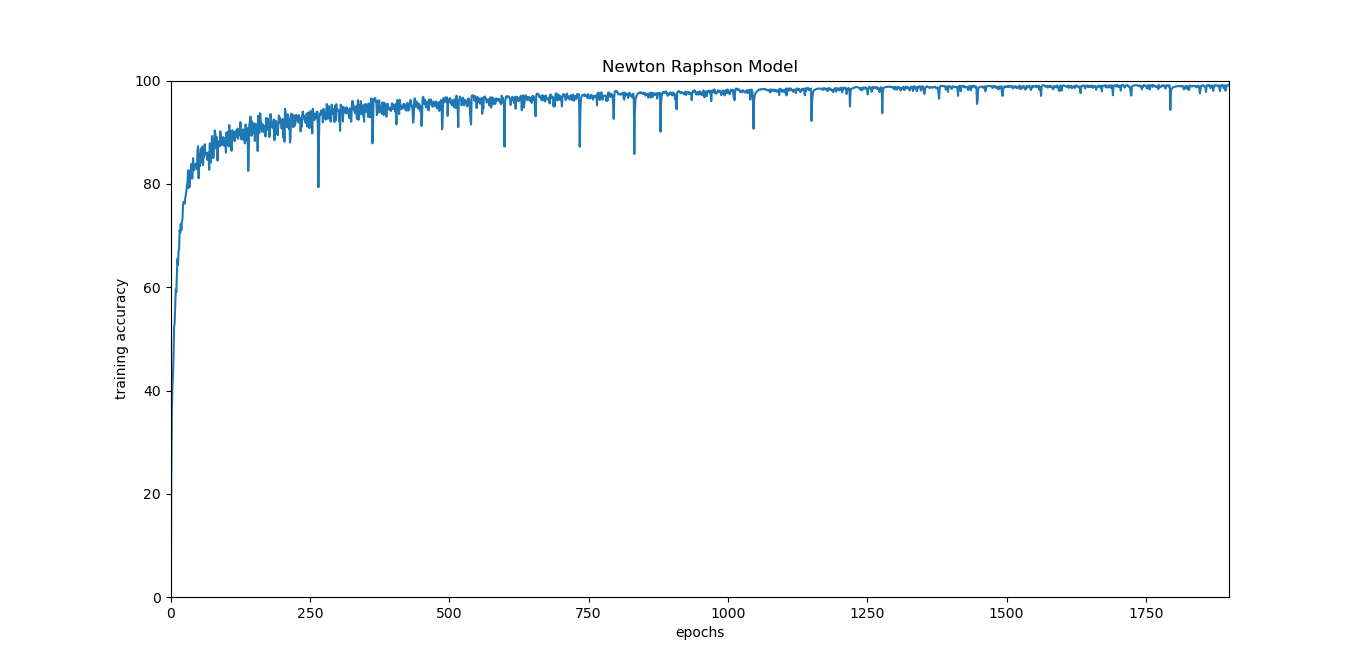
MSE Training loss of this model



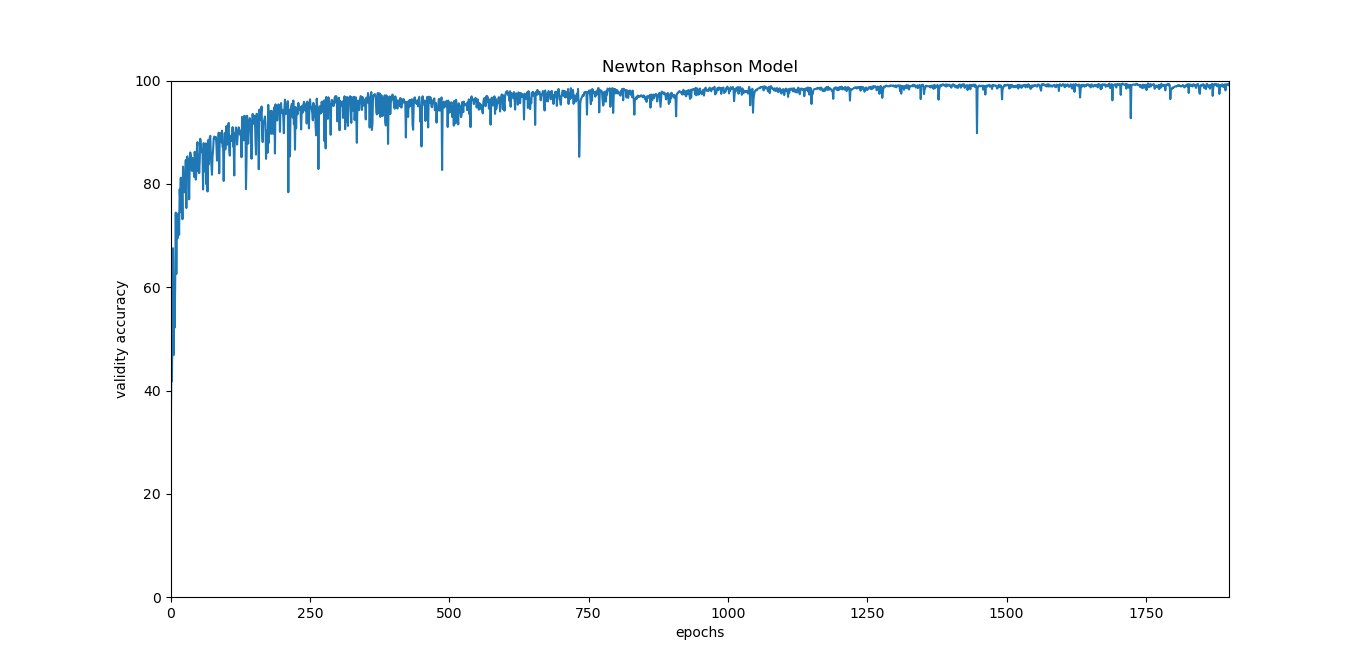
RMSE Training loss of this model



Training accuracy of this model



Validity accuracy of this model



For this model we tried with our test data set and we got MSE 0.0012 and RMSE 0.034 and

MAPE is 0.876.

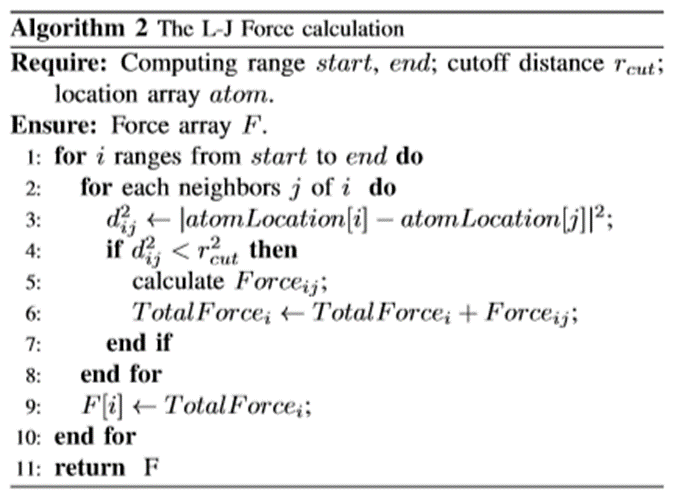
So finally for this algorithm only the topology (3\*11\*8\*5\*1 ) in the table works with positive

and negative and gives close results w.r.t paper , for better accuracy our topology

(25\*40\*30\*25\*1 )works good and gives satisfactory results.

**LJ POTENTIAL IN LAMMPS**

Algorithm for generating LJ Potential Dataset



Input:-x, y, z (Atom location)

From our knowledge that we gained from paper [2] and other online resources, we came to

conclusion that atom location should be the input for the NN model and it has to be three

components x, y, z.

Output:- tf (total force on that atom)

Assumption:-In paper [2] they have approximated force formula as potential formula.

The first step is to prepare the dataset to train our model.

This is the most difficult part, So ,we will be explaining how we have prepared our dataset,

In the paper [2], they have mentioned the input problem they used is (i.e in.lj.5) [1].

we have attached the input script in drive for your reference.

In order to prepare the dataset we need to have atom locations. After thorough research

we got to know how to extract the atom locations from the input script. we have run the

input script in the LAMMPS software. we used a command dump in input script to extract

the atom locations. After getting the atom locations we need to find the total force on all

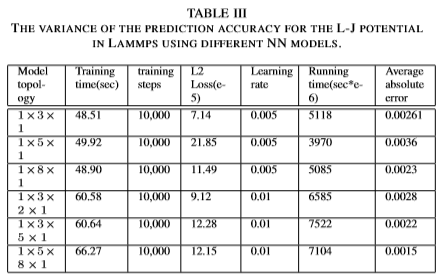
the atoms using the algorithm in paper. we have implemented the algorithm in python. The

code for it is uploaded in drive.

All the inputs for this algorithm are taken from input script, Finally, after preparing the

dataset, we moved on to create the model.

The results that we have to validate for the models is mentioned below:-



After creating the models in tensorflow, we tried to validate these results for the topologies

mentioned in paper [2], we have noticed that our dataset has three inputs and there is only

one neuron in input layer for all their topologies still we tried to train it with our dataset

and we used with many activation functions ,regularization techniques, optimizers but we

didn’t get similar results.

So we finally came to a point that we cant get the same results for the topologies

mentioned in paper [2]. So we decided to prepare a model that would give more accurate

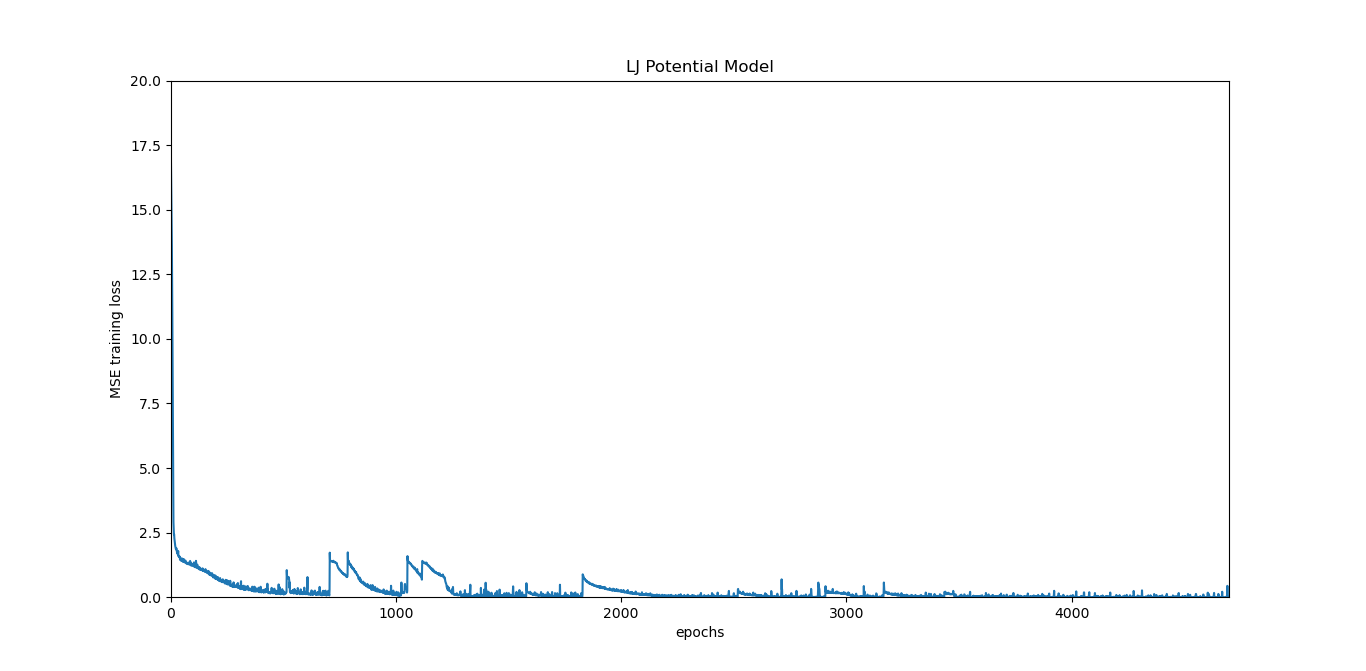
results than in paper [2].

we have prepared a model with topology (15\*45\*15\*1) and we have achieved an accuracy

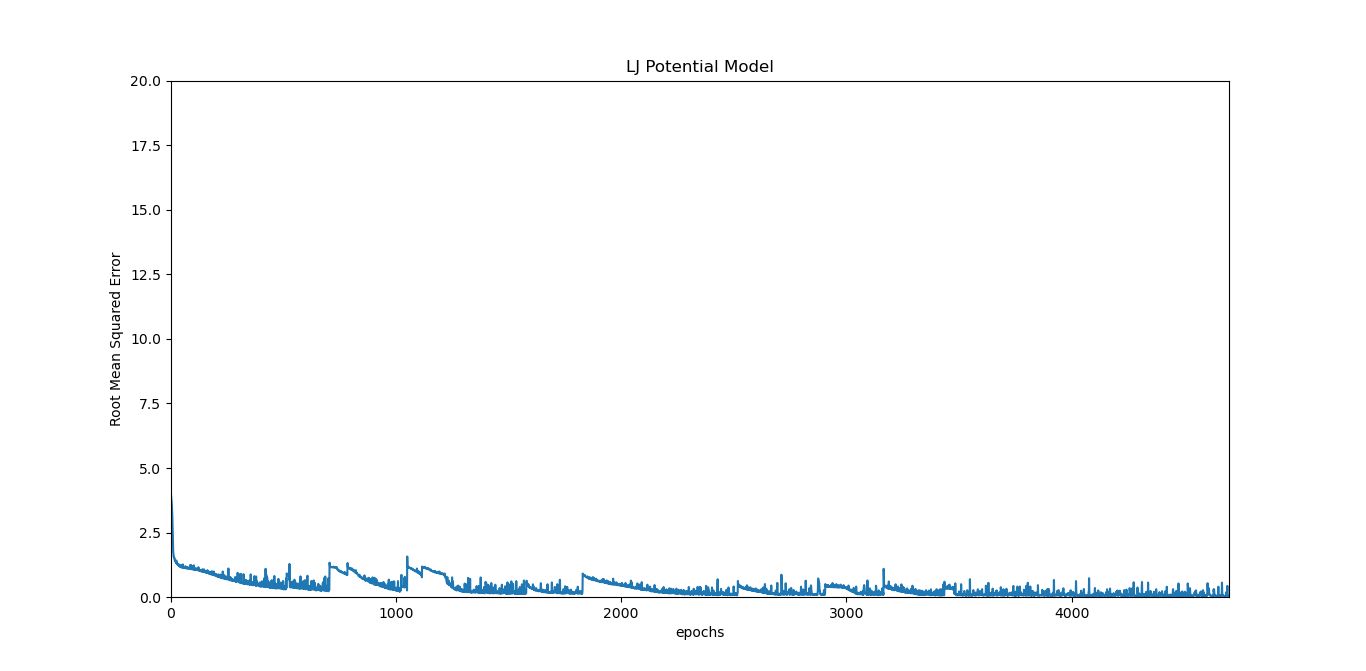
close to 99%.

All the results of this model are mentioned below:-

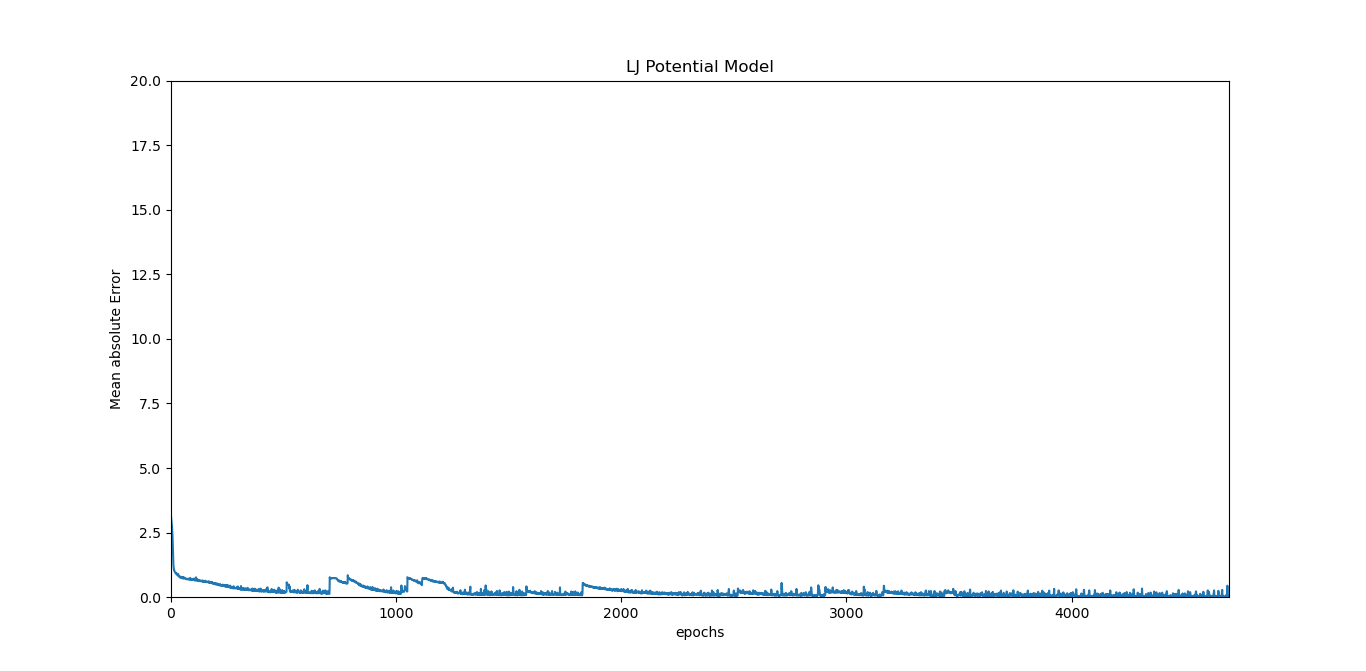
MSE Training loss of this model



RMSE Validity loss of this model



MAE Training loss of this model



For this model MSE on test set is 0.0026 and MAE is 0.0028

**CONCLUSION :**

Finally we understood that most of the models mentioned in paper [2] may work for

confined data range but main aim of this paper is to replace the algorithm codes with these

models to decrease the computation time and for parallel computing but input of the

algorithm can be any value may or may not in confined range ,if not in range it creates a

problem so instead of parallel computing we may end up in getting wrong results.

**REFERENCES**

1. <https://github.com/lammps/lammps/blob/master/examples/accelerate/in.lj.5.0>
2. <https://paperswithcode.com/paper/a-preliminary-study-of-neural-network-based>