Strengthening Intrusion Detection Systems utilizing Distributed Deep Learning: A Model Parallelism Approach for Efficient Training on High-Dimensional Data

Nour El-din Fathy[1], Yasmine N. M. Saleh[2], Khaled Mahar[3]

[1]Department of Computer Science College of Computing and Information Technology Arab Academy for Science, Technology & Maritime Transport Alex, Egypt, Email: Noureldeen.morsy@te.eg
[2]Department of Computer Science College of Computing and Information Technology Arab Academy for Science, Technology & Maritime Transport Alex, Egypt, Email: Yasmine_nagi@aast.edu
[3]Department of Computer Science College of Computing and Information Technology Arab Academy for Science, Technology & Maritime Transport Alex, Egypt, Email: kh.mahar@aast.edu

Received: 13.07.2024 Revised: 10.08.2024 Accepted: 24.09.2024

ABSTRACT
Intrusion detection systems (IDSs) are among the most outstandingwidespread and renowned detection models for providing the best safeguards to network traffic. IDS has the ability to discover malware and abnormal behaviors during or after data processing, whether in online or offline mode. In the literature, IDS has been approached by various deep learning techniques. The main problem is the high dimension features for large volumes of data of various intrusion detection datasets, and large deep learning (DL) models with huge numbers of hyperparameters, training such models is a major challenge that needs to be approached because such models is too big to be accommodated on one device. So, we need an efficient approach with low communication overhead for faster training and higher testing accuracy with fewer false alerts is strongly required. Consequently, the deployment of distributed deep learning (DDL) approach could address these challenges. DDL supports working on bigger datasets, especially with high-dimensional data and large numbers of hidden layers, with better performance in less time, whether using data parallelism to split datasets or model parallelism to split large models. In addition, deploying the optimization library Deep Speed, which was released by Microsoft Research (last version v.5.10), makes distributed training and inference easier, more efficient, and more effective. Solving and overcoming the limitations of training large deep learning model by standing fora novel deep convolutional neural network model using parallelization (Splitting) approach, in order to reduce the communication volume and computational load balance of the resulting partition, speeds up the training procedure by utilizing several (GPUs) in contrast to single processorfor training the whole deep convolutional neural network model. Improving accuracy, detect unknown attacks and reduce false alert rates by improving classification performanceby using optimal architecture for deep convolutional neural network model.In this paper, an enhanced IDS is proposed by applying a model parallelism that has the capability to split the model hidden layers and distribute them to more than one graphics processing unit (GPU). This approach has been applied to the high-dimensional UNSW-NB15 data set. Given that neural networks with convolutions (CNNs) have been applied with success in numerous image analysis use case where there is a spatial relationship between features, working with high-dimensional data where there is no relationship between its features is inappropriate for CNN modeling. To take upon this challenge, the high-dimensional data was converted into images because the conversion of data into images has the potential to enhance the representation of the correlation between features, whereby CNNs can learn to assist in prediction. Our proposed model achieved an improvement in the time of training by more than 5X speedup compared to conventional deep learning models and an archived high detection rate of 99%. In addition, the proposed model appreciably reduces the false alert rate by approximately 80%, which outperforms conventional deep learning models. The comparison utilizing cutting-edge methods demonstrates that the suggested strategy is more efficient and appropriate for large models and complex data.

Keywords: Intrusion Detection System, Distributed Deep Learning, Model Parallelism, Data Parallelism.

1. INTRODUCTION

Nowadays, with the increasing reliance on networks and the Internet, a huge amount of information is exchanged among different types of communicating devices, which elevates the probability of malicious attacks and fundamentally necessitates that the data should be securely transferred between these communicating devices [1]. Intrusion is defined as any type of prohibitive activities that can cause damage or wastage to a network system. Intrusion detection systems (IDS) can be software or hardware that recognize any abnormal or malicious actions by monitoring and tracing the network data flow in case of online mode or offline mode. IDS generates alerts if it detects an intrusions or vulnerabilities. Consequently, any IDS must provide security utilities like confidentiality, integrity, authentication and non-repudiation [2].

IDS can be classified according to source of data and detection methods shown in Fig 1.

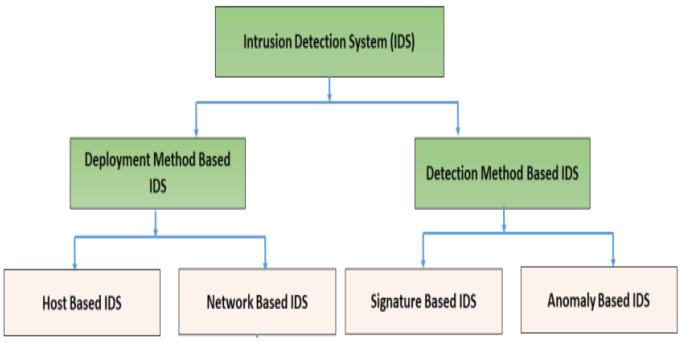

Fig 1. IDS Classification [3].

Source of data can be split into host-based IDS and network-based IDS, while detection methods can be divided into two groups anomaly-based IDS and misuse-based IDS [3]. In case of host-based intrusion detection system (HIDS), the IDS is installed on a single personal computer (PC) or host and it monitors all log files and any transactions performed through this PC, this implies that it observes the I/p and O/p packets from the terminals only. In network-based intrusion detection system (NIDS), IDS tracks all traffic information directed through the network and attempts to detect malicious data. One of the most famous examples of NIDS is firewalls [4].

Anomaly-based has sturdy generalizability and capability to recognize known or unknown attacks. This is because it can differentiate between normal and abnormal behaviors performed in the network. In addition, it can analyze network patterns statically or dynamically. However, in the case of misuse-based, aka signature-based, the requisite conception to illustrate attack behaviors is the as signatures. It can only detect the attacks by comparing it with the attacks saved in a database as it can identify the attack or the abnormal behaviors by using its signatures [5].

The field of intrusion detection has witnessed a recent surge when applying deep learning methods. Numerous intrusion detection systems (IDS) have been proposed in the literature that employ different deep learning techniques to address security threats and privacy issues. Because the use of deep learning networks and emphasizes deep intrusion detection systems (IDS) techniques to accomplish this goal using a variety of techniques at various stages of the intrusion detection process in order to obtain improved results.

A subset of machine learning algorithms called deep learning (DL) uses several layers to gradually take out more complex features from the unprocessed input. Image processing, which splits perception into multiple layers, is its primary focus. Higher layers may classify human-like concepts like faces, letters, or numbers, while lower layers may classify edges [6].DL uses computers to program human behavior in natural ways. For example, autonomous cars are designed to recognize stop signs, pedestrians approaching from lampposts, intruders, and obstacles on their own[7].

DL is abundant classifiers working together, which are instituted on linear regression followed by several activation functions. It reduces the cost function by changing the weight values through the backpropagation algorithm so the error rates will be decreased. DL takes more time and needs higher performance machines in the training and testing processes. Accordingly, nowadays distributed deep learning approaches are deployed to overcome these problems with better performance and less time specially when using incremental deep learning [9].

In order to address the challenge of training DNN modelswhich calls for substantial amounts of data and computational powera number of distributed and parallel approaches have been put forth. Thus, it comes

substantial to load the model on multiple machines which called parallelization strategy for preferable performance. Parallelization Methods for DL come with an abundance of options for splitting concepts. In this paper, the main predominant model and data parallelism are the two parallelization techniques in DL that are being explored[9].

Within data parallelism, data samples are split into mini-batches; each node or worker includes one of the mini-batches, a replica of the model of the neural network, and independently computes gradients (usually using the Mini-Batch SGD). Data parallelism is an effective approach for training a neural network that involves distributing a large-scale DNN within all computational workersas shown in Fig (2)[9].

As a result, there needs to be synchronization of the machines' respective model parameters. Consequently, each machine needs to compute its own gradient. The weight parameters of all GPUs are then updated by broadcasting the aggregated gradients. In case of asynchronous training, the gradient from one GPU is used to update the model parameters without waiting for the other GPUs to finish. Finally in both cases of training, the aggregated gradients use parameter server and All-Reduce function to share data among GPUs [10].

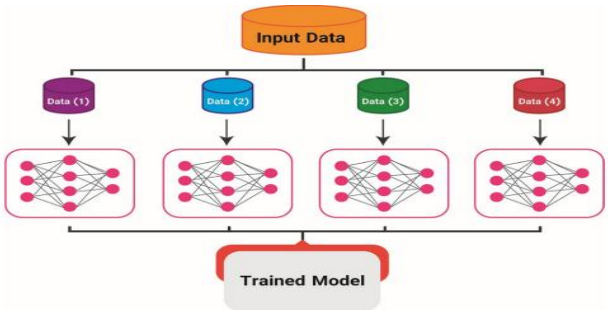

Fig 2. Data Parallelism [10]

In model parallelism, the deep learning model, for example CNN, is split, and each worker is loaded with a different part of the DL model for training as shown in Fig (3). The training data is fed into the workers that hold the DL model's input layer. They compute their output signal in the forward pass, and that signal is then distributed to the workers holding the subsequent layer of the deep learning model. Gradients are calculated in backpropagation, starting from the workers holding the output layer and propagating to the workers holding the DL model's input layers. Thus, the smaller memory footprint is model parallelism's primary benefit. Each worker needs less memory because the model is divided. When the entire model is too big to fit in a single device, this is helpful[10][11].

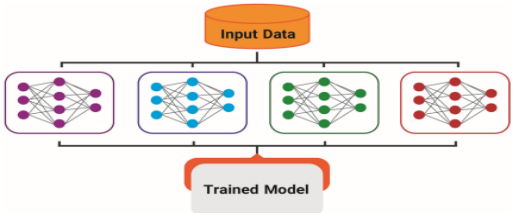

Fig 3. Model Parallelism [10]

The expected contribution of this paper are as follows:

1. Demonstrate a distributed approach which utilizes multilevel optimization using model parallelism that offers reduced latency, high efficiency, and low memory footprint. Solving and overcoming the limitations of training large deep learning model by representing a novel deep convolutional neural network in order to reduce communication volume and the computational load balance of the resulting partition.
2. Solve the problems of Increased computation and Performance degradationbecause of how features are represented and classified for large-scale datasets by using Vec2image conversion.
3. Speeds up the training procedure by employing multiple (GPUs) in contrast to a single processor for training the whole deep convolutional neural network model.
4. Improving accuracy, detect unknown attacks and reduce false alert rates by improving classification performanceby using optimal architecture for deep convolutional neural network model.

5. Solve the problem of overfit or underfit that happen in most applications using deep learning models. i.e., our model will be approximately optimal fit.

The rest of this paper is organized as follows: Section II introduces an overview of related work. Section III presents the proposed approach and Section IV depicts the experimental results. Finally, section V outlines the conclusion and future work.

2. Background and Related work

It is quite challenging to train big deep learning models with a lot of training data, especially when the models have high dimensions. Many difficulties arise because it is primarily carried out in a distributed infrastructure made up of several compute nodes, each of which may have several GPUs installed. First, because of communication bottlenecks, processing resources need to be utilized efficiently. In order to reduce expenses and offer flexibility, second, the network and computer storage resources are usually shared among various users or during training procedures.

An explainable CNN framework for characterizing feature selection is the Vec2image modeland classifier learning that employs an integrative image approach to assess feature values and feature correlation structure. Vec2image has performed superior to other comparable techniques in a variety of binary classification and multiclass classification tasks, especially when it comes to handling imbalance and high dimensional datasets.

Gradient descent is the basis for feed forward neural network training, which uses backpropagation to update and optimize network parameters. Deep neural networks require a lot of time to train; a large-scale training project may take days or weeks to complete. To expedite the DNN training process, parallelization is required.

For each worker to generate the subsequent gradient in parallel training, all gradients must synchronize to a single gradient. The most crucial aspect of the distributed and parallel training is the frequent gradient synchronizations for every mini batch. The scalability of gradient descent algorithms is restricted by these high-frequency synchronizations.

A. Distributed approach

Du et al. [12] High throughput and low latency could be achieved by using distributed CNN inference model parallelism. Using model parallelism is not simple, because the structure of the initial CNN model has close couplingby design. Suggest DeCNN, a more efficient inference technique that optimizes model parallelism for dispersed inference on end-user devicesby utilizing a decoupled CNN structure. Three distinct schemes make up the novel DeCNN. Scheme 1 optimizes at the structural level. It uses channel shuffle and group convolution to split the main CNN structure for model parallelism. Partition-level optimization is used in Scheme 2. The convolutional layers are divided using a channel group, and the fully connected layers are divided using an input-based technique, which further reveals a high degree of parallelism. Scheme-3 optimizes at the communication level. It hides communications using inter-sample parallelism to improve performance and robustness, particularly over bad network connections.

Alvarez et al. [13] Currently, one of the biggest problems to be solved is the training of large DNN models. data-parallelism has become the most popular distributed training approach. Unfortunately, there are a number of drawbacks to this solution, including the memory limitations that arise when training extremely large models and its high communication requirements. The most significant issues with the previous approach were resolved by the model-parallelism that has been proposed as a solution to these constraints.The solutions that have been suggested presume a uniform distribution, which makes them unfeasible when dealing with devices that have varying computational capacities, a situation that is frequently encountered in high-performance computing environments. In order to overcome previous shortcomings, this work proposes a novel model-parallelism technique that considers heterogeneous platforms and implements a load balancing procedure between uneven machines of an HPC platform.

Wang et al. [14] Recently, it has been proposed to use the Alternating Direction Method of Multipliers (ADMM)as a possible deep learning optimization technique that could replace Stochastic Gradient Descent (SGD). This is so that problems with gradient disappearing and poor conditioning can be resolved by ADMM. Furthermore, it has demonstrated strong scalability in numerous extensive deep learning applications.This work presents a new framework, called parallel deep learning ADMM (pdADMM), which enables the independent parallel updating of neural network parameters at each layer. Under mild circumstances, the suggested pdADMM's convergence to a crucial element is demonstrated. Benchmark dataset experiments illustrate that the suggested pdADMM can significantly accelerate the training of a deep feed-forward.

Yi et al. [15] This research proposes a fast module using tensorflow for executing deep neural network with multiple GPUs. suggest using analgorithm called white box to quickly compute methods that use little computational power. Similar research has recently been conducted to maximize device placement through reinforcement learning. With this strategy, the same computing node utilized in training can be used to find optimal device placement and execution order in a matter of minutes. Give an algorithm that divides operations along the critical path to handle mixed and fine-grained data, and take advantage of model parallelism to further accelerate training for each iteration. Make a list of scheduling algorithms as well in order to decide where to put the devices and what order to execute each operation in.

Table 1. Summary for section A

Reference	Deep Learning Model	Distributed Approach	# of GPUs	Speed Up
12	Y	Model Parallelism	4 GPUs	3x
13	Y	Model Parallelism	3 GPUs	2.5x
14	Y	Model Parallelism	4 GPUs	4x
15	Y	Data + Model Parallelism	4 GPUs	3x

B. Vector to image conversion

Noever et al. [16] In this study, the UNSW-NB15 network attack dataset is recast as an intrusion detection problem in image space.The resulting grayscale thumbnails, achieved by using one-hot encodings. By utilizing the convolutional neural network architecture of the MobileNetV2, the work illustrates a 97% accuracy in differentiating between attack and normal traffic. But in case of multiclass class classification this work achieves 54% accuracy.

Zhuet al. [17] In many applications, especially speech and imaging, convolutional neural networks have proven to be effective. CNN modeling is not appropriate for most tabular data since it does not assume a spatial relationship between features. As a solution to this problem, create an innovative algorithm called the image generator for tabular data, which uses feature assignment to create images by grouping related features together in the same pixel location.Utilize IGT to convert drug molecular descriptors and cancer cell line gene expression profiles into the appropriate visual representations. When IGTD is used instead of other transformation techniques, the compact image representations it creates better preserve the structure of the feature neighborhood.

Golubev et al. [18] Techniques that involve converting tabular data into images have received a lot of attention from scientists lately.Deep neural networks can be applied to a variety of analysis tasks through the transformation of features to images without requiring spatial relations. This study examines current methods for feature transformation that turn network traffic features into images and weighs the benefits and drawbacks of each. Additionally, the authors suggest a method for converting raw network packets into images and assess its effectiveness in identifying network attacks in cyber-physical objects, as well as its resilience to new and unknown attacks. The authors also propose an approach to transform raw network packets into images and evaluate its efficacy in detecting network attacks in cyber-physical objects and its ability to withstand novel and unidentified attacks.

Pak et al. [19] Systems for detecting network intrusions (NIDS) are some of the best ways to find irregularities and attacks on a network. However, because hackers constantly change their attack strategies, it can be difficult to stay on top of network security regulations. Using convolution neural networks (CNN) and image processing, this paper offers a widely used and efficient framework for network intrusion detection systems (NIDS). The suggested framework uses the Gabor filter to further improve the images after first converting data from network traffic into images. Next, a CNN classifier is used to categorize the images. Four benchmarking datasets—the CSE-CIC-IDS2018, CIC-IDS-2017, ISCX-IDS 2012, and NSL-KDDwere used to evaluate the effectiveness of the suggested approach.

Table 2. summary for section B

Reference	Deep Learning Model	Distributed Approach	IDS data set	Image data set
16	Y	N	Y	Grayscale
17	Y	N	N	Grayscale
18	Y	N	Y	RGB
19	Y	N	Y	RGB

In this paper, an enhanced distributed approach is comprehensively proposed to solve all the previous highlighted problems using model parallelism. The proposed model balances the overhead and achieves high performance for large-scale DL model that uses intrusion detection datasets with high dimensionally data for training and testing.Our proposed model achieved an improvement in the time of training by more than 5X speedup compared to conventional deep learning models and an archived high detection rate of 99%. In addition, the proposed model appreciably reduces the false alert rate by approximately 80%, which outperforms conventional deep learning models.

3. Proposed Approach

The focus in designing the proposed system is to enhance the detection accuracy, reduce false alerts, and use less training time especially in online mode. Also, this paper seeks to decrease the computational overhead for large deep learning model as will be explained later.

Fig 4. shows an overview of the proposed framework. Using the Synthetic Minority Oversampling Technique (SMOTE) [20][21], one can effectively enhance classification performance even with unbalanced training data. The provided training data are sampled for new data in the following manner:

(x1new, y1) = (x1 raw + random (0,1) (x1 random – x1 raw), y1) (1)

(x2new, y2) = (x2 raw + random(0,1) (x2 random – x2 raw), y2) (2)

…

(xnnew, ym) = (xmraw + random(0,1) (xm random – xm raw), ym) (3)

Oversampling is the most utilized approach to deal with class-imbalanced datasets. SMOTE has the advantage of providing more data from the minority class, overcoming data limited availability, and enhancing classification performance. It has been demonstrated that SMOTE is useful to resolve imbalanced classification problems and can help produce a more realistic representation of the underlying density.

KNN is most beneficialin situations where obtaining labeled data is either prohibitively expensive or impossible. It can achieve high accuracy in a variety of prediction-type problems. KNN is a simple algorithm that can be used to accurately and precisely learn an unknown function. The target function's local minimum serves as its foundation.

Calculate K nearest neighbors from the whole training set, and then xi randis chosen at random from them.

Subsequently, a new synthetic sample, xi new, can be computed by putting a boundary between the selected closest neighbors and the minority example.

Therefore, the updated training data that is balanced D = x1, y1, …, xn, yn are generated.

High feature dimension and a substantial data volume are two traits of the typical high dimensionally imbalanced datasets, UNSW-NB15. As this is one of the biggest challenges that must be solved, we use aVec2image technique to convert Input Data (file.CSV) from numeric format into image format (24-bit RGB color).

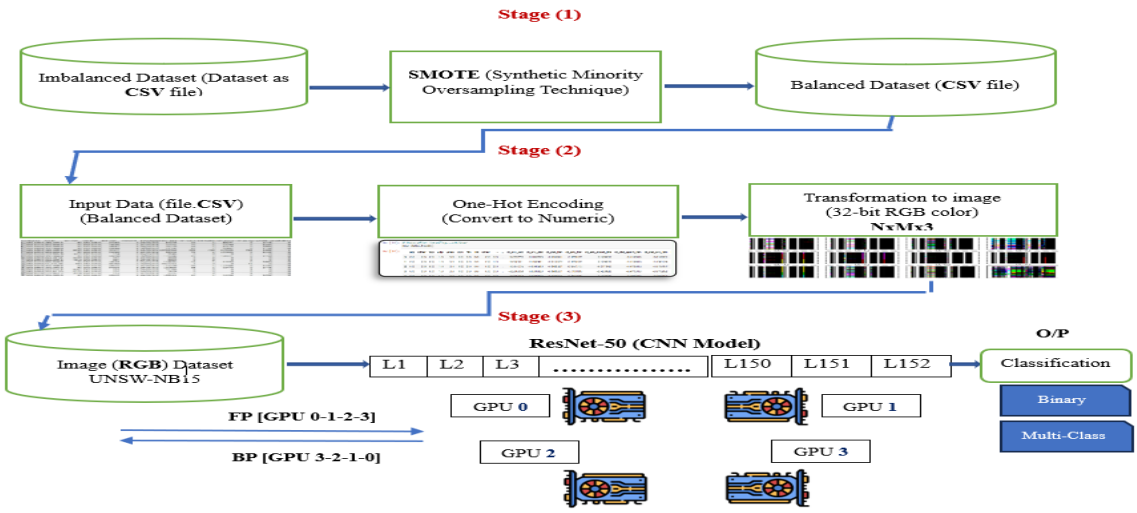

Fig 4. Proposed Frame work

First, we convert the dataset to numeric form, all the categorical (nominal) features are converted to numeric values using one-hot encoding. The full UNSW-NB15 dataset has **39** numeric and **3** nominal features, where the nominal features are **proto, service, and state** [22].

Table 3. Final Features after Encoding

	UNSW-NB 15 Dataset Features	
	Before Encoding	After Encoding
Full Dataset	42	194

Second,Create a blank image file with the same dimensions as the new CSV file's height and width for rows and columns. Next, every feature value from the updated CSV file will be transformed into a 24-bit RGB color value, ranging from [0x000000] to [0xFFFFFF].
Nine classes (fuzzers, analysis, backdoors, denial-of-service, exploits, generic, reconnaissance, shellcode, and worms) are converted with UNSW-NB15.

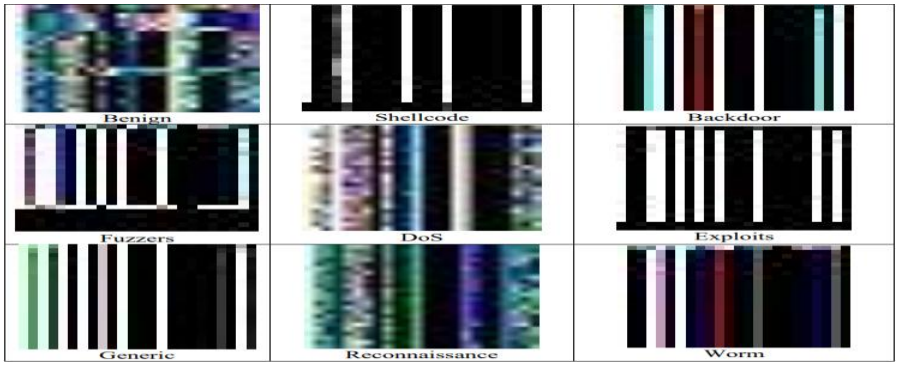

Fig 5. RGB sample of 9 classes that have been transformed for UNSW-NB15

In the third stage **a)** input image, **b)** distributed model and I output label represent the class of the input. So, through this stage we train and test our model to overcome the previous limitations and achieve our contributions as discussed before.

A) Input: RGB file of UNSW-NB15 dataset of high dimensional data, we choose this dataset due to the range of attack types they contain as well as the fact that they have every feature we require for server identification and arrival times.
B) Employ the ResNet-50 model. We assess residual nets up to 50 layers deep—eight times deeper than VGG nets—on the ImageNet dataset. ResNet-50 performs better than AlexNet but requires ten times as many computations, meaning that it takes more time and energy to train.

We conduct the model using 4 GPUs with model parallelism strategy. Astrain the model with 4 GPUs by splitting our model layers among these GPUs through forward pass and backward pass. Each GPU is responsible for the weight updates of the assigned model layers. We minimize communication time by placing neighboring layers on the same GPUs.
Every iteration of training requires that the forward propagation be completed in GPU #0 first. GPU#1, GPU#2, and GPU#3 are in order of waiting for the output from GPU#0, GPU#1, GPU#2, and GPU#3 respectively. after forward propagation has been completed. The model parameters for the final layers in GPU #3 are updated, and the gradients for those layers are computed. After that, the gradients go back to GPU #2's earlier layers, etc.

4. Experimental Results

Performance Metrics

1-ACC: is the likelihood that all of the samples that the system correctly classified account for all of the samples
[23].
Accuracy= TP+TN / TP+TN+FP+FN [23]
2-DR: shows the likelihood that, in the event that an attack occurs in the system environment[23].
DR= TP/ FN+TP [23]

3-FAR: is a possibility that the system will mistakenly interpret the regular data as an attack that poses a threat to the system and raise a false alarm. The possibility that this component contains all of the typical network connection data is known as the FAR[23].
FAR= FP/ TN+FP [23]
We present results on the following two sections:
a- Binary Classification
b- Multi-class Classification
The outcomes of the high-dimensional UNSW-NB15 data set using BCNN and MCNN. Normal and anomalous are classified using binary classification. Ten labels are identified by the multiclass classification.

Table 5. Hyper Parameter

Parameter	Value
Batch Size	512
I/P Layers Neurons	48
O/P Layer Neurons	Binary Class (0,1) Multi-Class (9 Classes)
Hidden Layers	50
Number of Epochs	150
Number of GPUs	(GPU0, GPU 1, GPU 2, GPU3)
Learning Rate Range	(0.01 till 0.5)
Activation Function	RELU, SOFTMAX

Data Balancing

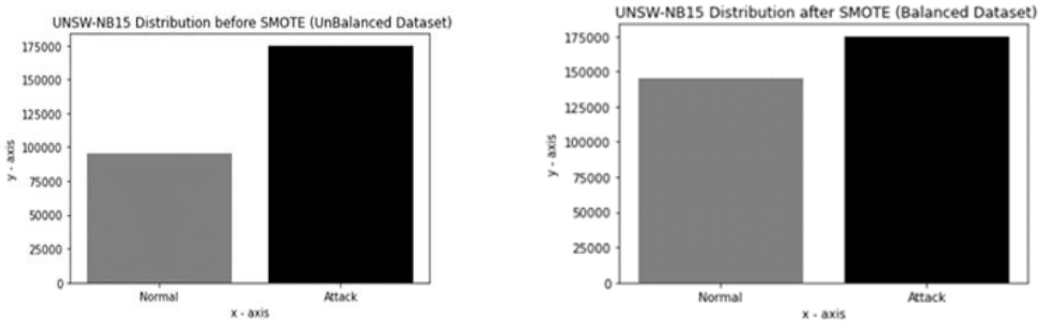

Fig 6. Data distribution (a) before SMOTE (b) after SMOTE in UNSW-NB15

We conduct our results through three experiments and compare between them, also compare our results with related works to show the enhancement in time and accuracy through our model.
Exp1: we train our dataset through stage #1 and stage #2 only. As with using deep learning model without parallelism.
Exp2: we train our dataset through stage #1 and stage #3 only. As using the high dimensional data without image transformation with model parallelism.
Exp3: we train our dataset using stage #1, stage #2 and stage #3

Experiment 1
we give our results through binary classification and multi-class classification, total training accuracy about 95%

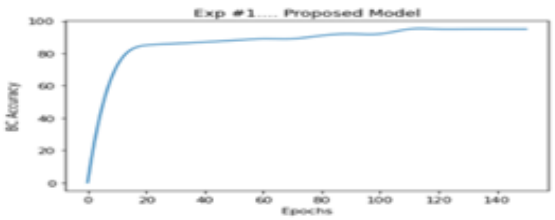

Fig 7. Training accuracy for binary class

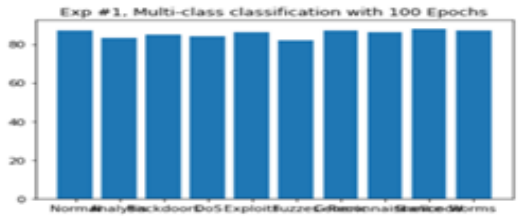

Fig 8. Training accuracy for multiclass

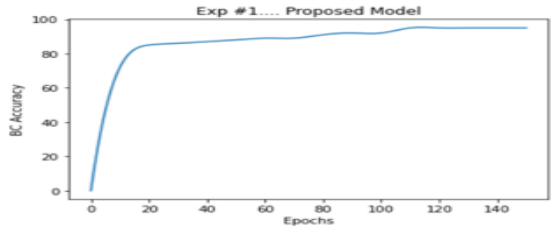

Fig 9. Testing accuracy for binary class

Fig 10. Testing accuracy for multiclass

Experiment 2

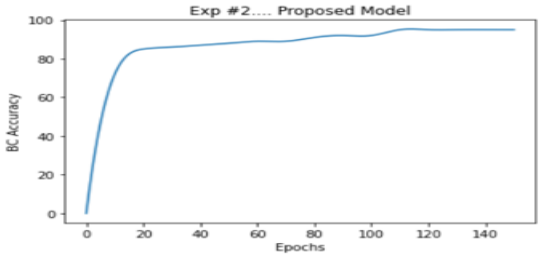

Fig 11. Training accuracy for binary class

Fig 12. Training accuracy for multiclass

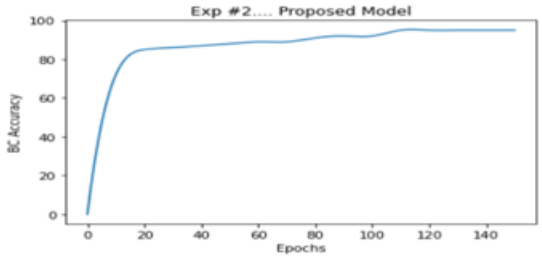

Fig 13. Testing accuracy for binary class

Fig 14. Testing accuracy for multiclass

Experiment 3 (Proposed Model)
a-Binary Classification

1- ROC curves of UNSW-NB 15 using Deep CNN

ROC (AUC = 78%)

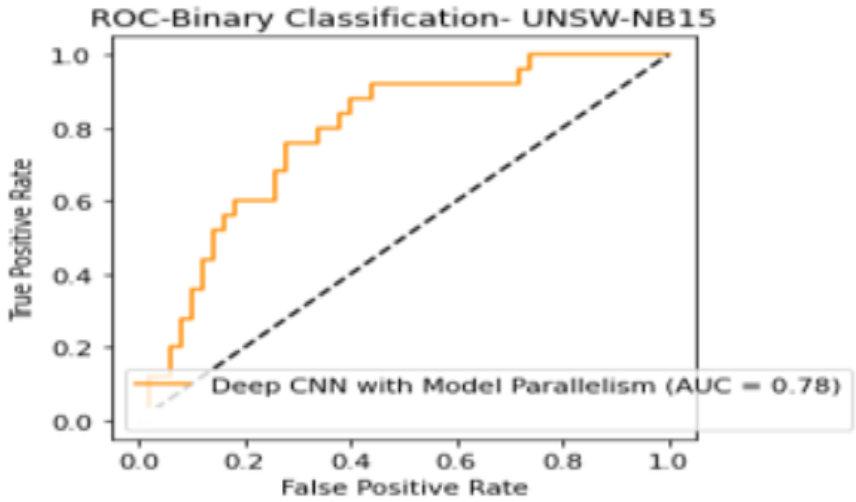

Fig 14. TPR Vs FPR

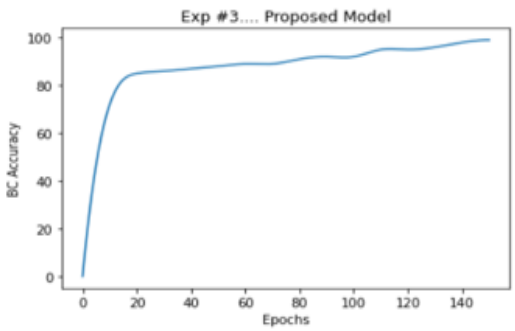

Fig 15. Binary Classification

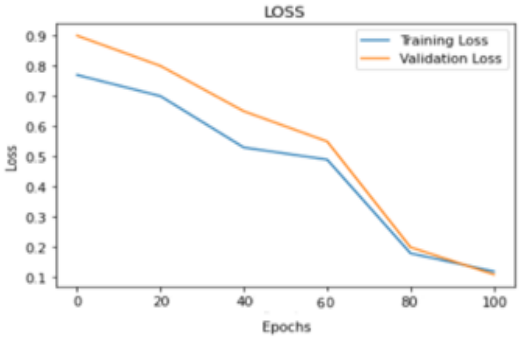

Fig 16. Loss & Validation

This results This indicates that our model approximately optimal fit, i.e., This new model does not overfit or underfit.

Training & testing Accuracy for proposed model

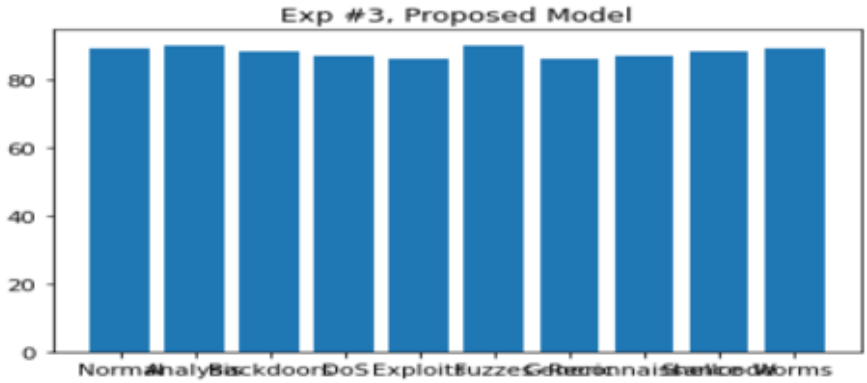

Fig 16. Training for Multiclass Classification

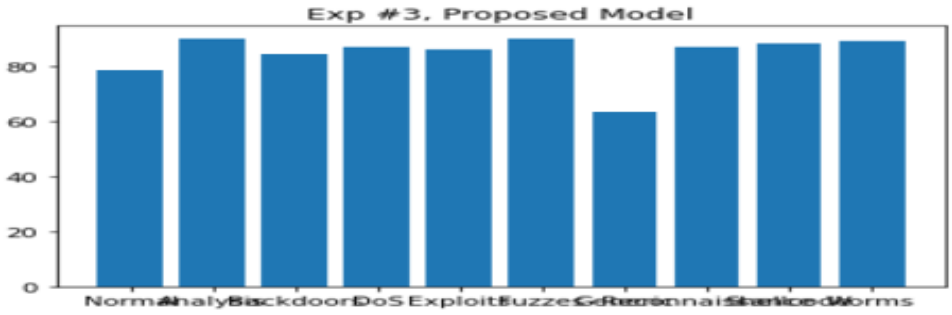

Fig 17. Testing for Multiclass Classification

Table 6. Details test results for binary classification

	Accuracy	Precision	Recall	F-score
EX 1	0.95	0.94	0.96	0.95
EX 2	0.93	0.95	0.93	0.94
EX 3	0.99	0.98	0.99	0.97

Table 7. Details test results for multiclass classification

	Normal	Fuzzers	Analysis	Backdoors	DoS	Exploits	Generic	Reconnais	Shell code
EX1 ACC	0.96	0.95	0.95	0.94	0.95	0.91	0.92	0.91	0.98

EX2 ACC	0.94	0.93	0.96	0.96	0.98	0.90	0.97	0.95	0.92
EX3 ACC	0.95	0.97	0.96	0.97	0.95	0.96	0.97	0.98	0.97

5. Conclusion and Future work

IDS have been approached by various deep learning techniques. But with the large volume of various datasets and the large deep learning models with a lot of hidden layers, so it needs an efficient approach for faster training and higher testing accuracy with less false alerts as possible. So Distributed Deep Learning approaches is the best solution to support all these needed improvements because they have the ability to work on bigger datasets and larger hidden layers with higher performance in less time whether using data parallelism to split datasets or model parallelism to split large models. Specially using deep speed learning packages which give the splitting approaches more supports. In our work we enhanced IDS by apply model parallelism which split the model layers into more than one GPU.

We enhanced IDS by applying a model parallelism which has the capability to split the model hidden layers and distribute them to more than one GPU.

This approach has been applied to the high dimensional UNSW-NB15 data set. It has achieved improvement in the time of training by more than 5X speedup than conventional deep learning models and archived high detection rate 99%. It also appreciably reduces false alerts rate by approximately 80% which outperforms conventional deep learning models.

In future work, we intent to apply pipeline, Deep Speedincorporates updated pipeline parallelism support. Pipeline parallelism divides a model's layers into stages that can be processed concurrently, which increases the memory and computational efficiency of deep learning training. Also, it solves the problem of delay between GPUs.

REFERENCES

[1] Si-Ahmed.A, Al-Garadi.M. Survey of Machine Learning Based Intrusion Detection Methods for Internet of Medical Things. arXiv:2022.09657v4, 7 Mar 2023.

[2] Oughannou.Z, EL Rhadiouini.Z, CHAOUI.H. Anomaly-Based Intrusion Detection System to Detect Advanced Persistent Threats: Environmental Sustainability. E3S Web of Conferences 412, 01106 (2023).

[3] Azam.Z, Islam.M, Huda.M. Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. IEEE Access, Received 29 June 2023, accepted 6 July 2023, date of publication 18 July 2023, date of current version 4 August 2023.

[4] Al-safaar.D, Al-yaseen.W.A survey of network intrusion detection systems based on deep learning approaches, Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2023, vol. 23, no2.

[5] Pinto.A, Hererra.L, Donoso.Y. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure, MDPI, Sensors 2023, 23, 2415.

[6] Chelladurai.K, Sujatha.N. A Survey On Different Algorithms Used In Deep Learning Process, E3S Web of Conferences 387, 05008 (2023).

[7] Masood.S.Neural Networks and Deep Learning: A Comprehensive Overview of Modern Techniques and Applications, Department of Computer Science, University of California, Santa Cruz, 2023.

[8] Alzubaidi.L, Zhang.J, Amjad J. Humaidi. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Datavolume 8, Article number: 53 (2021).

[9] Patil.M, Ckickerur.S. Study of Data and Model parallelism in Distributed Deep Learning. International Conference of Machine Learning and Data Engineering, 2023.

[10] Dehghani.M, Yazdanparast.Z. From distributed machine to distributed deep learning: a comprehensive survey. Journal of Big Data (2023) 10:158.

[11] Choi.H, Lee.B, Chun.S. Towards accelerating model parallelism in distributed deep learning systems. Plos One, November 2, 2023.

[12] Du.J, Zhu.X, Shen.M, Du.Y. Model Parallelism Optimization for Distributed Inference via Decoupled CNN Structure. IEEE Transactions On Parallel And Distributed Systems, VOL. X, NO. Y, 2020.

[13] Alvarez.S, Haut.J, Paoletti.M. Heterogeneous model parallelism for deep neural networks, 0925-2312/2021 Elsevier B.V. All rights reserved.

[14] Wang.J, Chai.Z, Zhao.L. Toward Model Parallelism for Deep Neural Network based on Gradient-free ADMM Framework. arXiv 22 Jun 2021.

[15] Yi.X, Luo.Z, Meng.C. Fast Training of Deep Learning Models Over Multiple GPUs, ACM ISBN 978-1-4503-8153-6/20/12, 2020.

[16] A. Noever.D, Noever.S. DEEP Learning Classification Methods Applied To Tabular Cybersecurity Benchmarks. International Journal of Network Security & Its Applications (IJNSA) Vol.13, No.3, May 2021.
[17] Zhu.Y, Brettin.T, Xia.F. Converting tabular data into images for deep learning with convolutional neural networks. Scientific Reports (nature.com), Springer, 2021.
[18] Golubev.S, Novikova.E, Fedorchenko.E. Image-Based Approach to Intrusion Detection in Cyber-Physical Objects. MDPI, 25 Nov 2022.
[19] Pak.W, Siddiqi.M. An Optimized and Hybrid Framework for Image Processing Based Network Intrusion Detection System. Computers, Materials & Continua 2022, 3921-3949. https://doi.org/10.32604/cmc.2022.029541Received 06 March 2022; Accepted 07 May 2022; Issue published 16 June 2022.
[20] https://www.blog.trainindata.com/overcoming-class-imbalance-with-smote/
[21] Halim.A, Dwifebri P.M, Nhita.F. Handling Imbalanced Data Sets Using SMOTE and ADASYN to
[22] Improve Classification Performance of Ecoli Data Sets. Building of Informatics, Technology and Science (BITS)
[23] Volume 5, No 1, June 2023 Page: 246−253.
[24] Zoghi.Z, Serpen.G. UNSW-NB15 computer security dataset: Analysis through visualization. 27 June 2023 https://doi.org/10.1002/spy2.331Citations: 2.
[25] Reis.B, Karatas.G, Sahingoz.O. Intrusion Detection Systems with GPU-Accelerated Deep Neural Networks and Effect of the Depth. 2018 6th International Conference on Control Engineering & Information Technology (CEIT), 25-27 October 2018, Istanbul, Turkey.