Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi Core DNN based IDS for Botnet Attacks using KPCA Reduction Techniques

Sharmila B S (sharmilabs@nie.ac.in)
NIE: National Institute of Engineering https://orcid.org/0000-0002-2495-543X
Rohini Nagapadma
NIE: National Institute of Engineering

Research Article

Keywords: IDS, KPCA, KPCA, Multi-core, optimizers, IoT, Artificial Intelligence

Posted Date: September 21st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-795895/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Noname manuscript No.
(will be inserted by the editor)

Multi Core DNN based IDS for Botnet attacks using KPCA reduction techniques

Sharmila B S · Rohini Nagapadma

the date of receipt and acceptance should be inserted later

Abstract Research on network security has recently acquired attention in the field of the Internet of Things. In the context of security, most of the IoT devices with the internet are connected directly which results in the exploitation of private data. Nowadays, the fraudster will release novel attacks very frequently especially for IoT devices. As a result, the traditional sophisticated Intrusion Detection System (IDS) model is not suitable for the identification of vulnerabilities in IoT devices. In our research work, we propose MCDNN for IDS. MCDNN is Multi Core DNN with having parallel optimizer. Rather than a traditional dataset, this paper experiment is conducted on the BoT-IoT dataset. Since IoT devices generate a huge volume of data, this work focuses on reducing huge datasets using Kernel Principal Component Analysis(KPCA) reduction technique with optimizer parallelly. To decrease false alarm rate and maintaining less computational power multi-core is introduced in our research work. This helps identification of vulnerabilities in IoT devices using deep learning techniques faster. Experimental results indicate that designing MCDNN based IDS with different optimizers parallelly achieved higher performance than those of other techniques.

Keywords IDS · KPCA · KPCA · Multi-core · optimizers · IoT · Artificial Intelligence

Sharmila B S
Department of Electronics and communication Engineering
The National Institute of Engineering,
Mysuru, Karnataka,India E-mail: sharmilabs@nie.ac.in

Rohini Nagapadma
Department of Electronics and communication Engineering
The National Institute of Engineering,
Mysuru, Karnataka,India
E-mail: rohini_nagapadma@nie.ac.in

1 Introduction

In recent years, Cybercrime is a threat to IoT devices as every device needs to communicate over the internet. Furthermore, this exploitation results in compromising with confidentiality of private data. Several traditional techniques are proposed to protect intrusions. Unfortunately, the signature-based techniques are not suitable for resource-constrained devices because attackers are improving their techniques and tools to breach these devices especially in the field of home automation, healthcare, business, and automated industries. Security risk involved in healthcare and Industrial IoT is very high. Since it may lead to loss of human life. Conventional cryptographic algorithms are not designed for distributed IoT environments. Since these algorithms involve huge calculations, a large amount of data, and key exchange, which is not suitable for resource-constrained devices. As most of the devices have small RAM and low computational power[1][2]. Therefore existing cryptography[3] and firewalls[4] are not suitable for resource-constrained devices. As a result, frequent incidents of cyber attacks uplifted the need of progress in security [5]. Additionally, data which will be exchanged among these devices will be enormous. This made our research focus on identifying attacks more rapidly [6] [7]. Therefore efficient IDS for resource-constrained devices is required to identify and stop malicious activities.

Over the last few years, the introduction of machine learning and DNN proved as an effective method for forensic analysis[8]. Machine learning provides a security framework to implement security policies. However, in [9] discussed the issues related to conventional machine learning techniques. Traditional supervised machine learning is more or like a machine as it requires a lot of interventions of domain experts and humans. Most of the time applying these machine learning algorithms at some point either accuracy becomes constant or sometimes it may decrease also.

On the other hand, deep learning or DNN is also a subdivision of machine learning that achieves great flexibility by introducing backpropagation. This leads to solving complex problems with fewer false alarms. Deep learning performs better for a huge volume of data and machine learning for a small dataset. Therefore in our work, we proposed MCDNN based IDS. Usually, DNN has at least one input, an output layer, and in between one hidden layer. But in the deep neural network, more hidden layers result in a deep network. With this deep network pattern formation in data is effective. On the other hand, DNN will increase its performance as we increase the size of the dataset epochs and layers. So, DNN proved to be the best solution for predictive problems. Finally, in our research work, we propose three hidden layers of MCDNN networks with different feature selection methods and also analyse the effect of different optimizers.

2 Related work and motivation

Accuracy is a big challenge for Intrusion detection solutions for IoT networks, this leads to approach DNN in an efficient way for improving the accuracy and performance in terms of time complexity and decreasing false alarm [10][11]. Mercaldo et. al [12] proposed a deep neural network and supervised learning method for the identification of malware in mobile devices. They have considered platform independent static approach and obtained precision and recall of about 0.912 and 0.918 respectively for detection. To detect the low frequent attacks SAVAER-DNN method has been introduced by Yanqing Yang et[13]. This method implemented using WGAN-GP for UNSW-NB15 and NSL-KDD datasets. Zuchao Ma et. al[14] proposed Distributed Consensus based Trust Model(DCONST) approach in IoT networks for the detection of tamper, drop and replay attacks using K-means machine learning algorithm. In [15] DDoS Detection framework is proposed for resource constrained IoT devices focused on benign and malicious traffic using a machine learning algorithm. [16] Explained the vital role of network security essential for IoT and also provided a survey on exiting various attacks. Zhou et al. [17] proposed KPCA based IDS system using Extreme Learning Machine (ELM) algorithm. The KPCA is used to reduce dimensionality in same feature space.[18] implemented IDS system using KPCA-DEGSA-HKELM approach to increase F1-score and reduction in FAR.

Minh Tuan et al. [19] proposed Genetic Convolution neural networks having GA, FCM and CNN as three layers for NSL-KDD dataset. This approach selected KNN with GN as the first layer for feature selection followed by CNN for model selection, finally deep features extracted by CNN as model validation. Nour Moustafa et al. [20] collected network traffic from various websites of both normal and attack traffic. For feature selection, this work used the Association Rule Minings algorithm, where ARM [21] techniques is to identify strong association rules. Based on these rules, either using Apriori or FB-Growth method interdependency between two or more features in a sample will be identified. Since, ARM will produce a huge volume of association rules as the dataset increases, which is difficult to manage in real world problems. Then the extracted features are sent to a OGM technique for detection of unusual activities. This approach can achieve 4.43% and 2.72% of FAR for UNSW-NB15 dataset and web attack dataset respectively.

In [22] Iram bibi et al. constructed DL driven architecture like GRU, LSTM, DNN and CNN and compared it with standard matrices. Here GPU accelerated GRU model for android devices is proposed. REU is derived from Recurrent Neural Network. This model is designed for the detection of benign, torjan and backdoor attacks. It was observed that DNN performs better compared to the proposed architecture. Furthermore, the Chi-square technique is introduced for the important feature extraction citeMoustafa2018Aug. Since chi-square χ^2 method is the simple computation for realistic IDS. Once χ^2 is calculated then it is subjected to cross-entropy. If any variation in cross-entropy, then it will be considered as an attack. With this technique, they can

reach an accuracy of 95.98% for 300,000 sample size. Finally for identification of vulnerabilities in IoT devices using machine learning, requires a proper dataset. Over the years, for network traffic, many datasets are created for Intrusion detection, prevention systems and other forensic applications. The most popular dataset for research in information security is NSL-KDD and KDD-99. Most of the conventional IDS was designed based on this network traffic. This dataset mainly contains signatures of DoS, Probe, R2L and U2R attacks [23]. Since we are focused on security for IoT devices, this work will address mainly DoS, DDoS, Theft and Reconnaissance attacks. The working dataset is BoT-IoT dataset [24][25][26] [27] [28] created by UNSW Canberra Cyber. But the dataset is very huge and difficult to apply DNN. So in this research different dimensionality reduction techniques were compared. In the BoT-IoT Dataset, the original dataset is having 72,000 records of network traffic that is almost 16.7 GB. Since it is huge for computation in laptops, this research considered 5% of the dataset having 36,00,000 rows of traffic. This contains signatures of DoS, DDoS, Theft and Reconnaissance as the main category and like UDP, TCP, service scan, OS Fingerprint, HTTP, keylogging, and Data exfiltration a subcategory.

2.1 Paper contribution and organization

This section summarizes the steps involved in designing MCDNN for the detection of IoT-BoT attacks. Since the original dataset is very huge this work considered the 5% of the dataset only for UNSW-NB15. The simulation results of this paper lead to a comparison of the dimensionality reduction techniques. The main contribution of this paper is described as follows:

- The dataset is pre-processed by scaling to unit variance.
- Preprocessed data is applied to KPCA for feature reduction to four components.
- Constructed MCDNN model for IDS using different hyperparameters. During the learning stage, metrics may stop for further improvement. To overcome this situation reduce the learning rate during learning processes has been implemented.
- Performance analysis in terms of accuracy for different optimizers is analysed and best results are selected using the voting technique.

The remainder of this article is organized as follows: in section 3 we discussed different dimensionality reduction techniques. The proposed IDS and its performance analysis are explained in section 4. The detailed results are discussed in section 5. Finally conclusion is reported in the last section.

3 Dimensionality Reduction techniques

Since the nature of IoT devices is in large volume, the collected dataset will also be huge. It is difficult to implement DNN directly. In [29] discussed the

need for feature reduction to decrease the complexity of computation. As a result, this research focused on dimensionality reduction of features in the dataset to train more efficiently in less computational time.

3.0.1 Principal Component Analysis(PCA)

PCA is the most popular method for reducing datasets without any statistical information loss [30]. This technique extracts the covariance matrix of feature [31]. Maximum variance decides the direction of projection. Based on variance, the maximum content of information features is selected. The steps followed for dimensionality reduction as follows:

- Considering n samples $\overrightarrow{x}_1.....\overrightarrow{x_n}$ having m features, compute μ mean equation[1] of all features in a single vector $\mathbb{R}^m$.

$$\vec{\mu} = \frac{1}{n}(\vec{x_1}.....\vec{x_n}) \tag{1}$$

- Compute eigenvalues and corresponding eigenvectors.
- Select m eigenvectors having large information by considering eigenvalues.

3.0.2 Linear Discriminant Analysis(LDA)

Another dimensionality reduction technique is LDA. This method is applied to reduce the dataset to lower dimensional space based on the number of classes especially for huge datasets [32]. The procedure mainly involves in finding linear combination of feature set that achieves maximum separation from dataset between different class. and minimum separation of dataset within different class.

3.0.3 Singular Value Decomposition

Another feature selection technique is Singular Value Decomposition (SVD). This is a matrix decomposition technique, where a complete dataset can be decomposed into smaller dimensions having combinational features[33], [34].

3.0.4 Kernel Principal component analysis (KPCA)

A variant of PCA for dimensionality reduced is KPCA technique. PCA is applied for the linear dataset. Whereas, KPCA is applied for the nonlinear dataset in which it uses kernel methods to perform computation in original space only. There are many kernel methods like Gaussian, polynomial, Radial Basis Function(RBF).[35][36].

4 Proposed Intrusion Detection System using MCDNN

Fig 1 shows the complete multi-core DNN model which consists of following steps:

4.1 Preprocessing

To illustrate the performance of MCDNN model with various optimizers, we have experimented using NSL-KDD dataset and BoT-IoT dataset also known as UNSW-NB15 dataset [37]. UNSW-NB15 contains some categorical columns like protocol, source address and destination address. During preprocessing step, the first protocol features are remapped to integers ie (icmp:0, tcp:1, udp:2, arp:3, ipv6-icmp:4) and the source and destination port is converted from hexadecimal to integer form. The details of attacks present in the dataset are shown in Table 1. Firstly, all categories are converted from string to integer. Secondly, the distribution of data needs to be rescaled for centering data. Additionally when the features of the dataset is having different measurements it is important to centralize the complete data around mean 0 and standard deviation 1. This removes bias and all values will provide equal contribution during study [38]. To standardize the complete dataset is subjected to transform using the following equation:

Table 1: Details of attack category in dataset

Attack Names	Attack Numbers	Count
Normal	0	477
UDP	1	1981230
TCP	2	1593180
Service_Scan	3	73168
OS_FingerPRint	4	17914
HTTP	5	2474
Keylogging	6	73
Data_Exfiltration	7	6

$$y = (x - \mu)/\sigma \tag{2}$$

where, x is dataset, μ is mean of training set and σ is standard deviation.

4.2 Deep Neural Network

In this section, we describe the model of MCDNN based IDS. This proposed work consists of pre-processing stage followed by feature reduction technique. Once the dataset is prepared from above steps, it is applied to MCDNN model as shown in below fig 1. In this model, DNN consists of three hidden layers having 512, 256, 512 nodes and leaky relu as a activation function at each layer respectively. Also at the output, which is in the last layer will have an activation function called softmax, since we need to identify different types of attacks including normal traffic in our work. Softmax is preferred for multi class classification problems in DNN [39][39]. In this research work, we have used different optimizers like adam, Nadam, Adabelief[40] and Adamax[41]

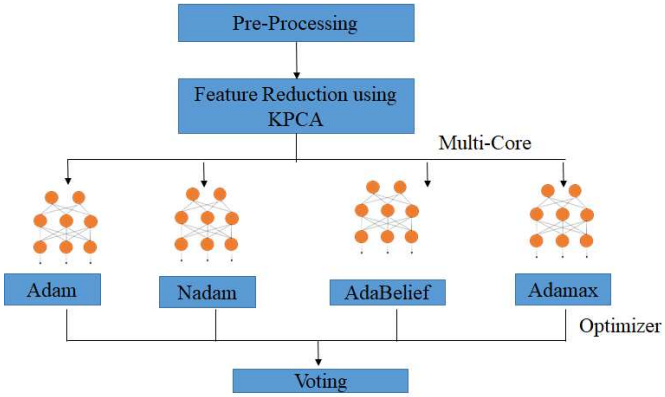


Fig. 1: MCDNN model for IDS

for parameter update for every epoch. The adam in eq. [3] is adaptive learning method has an advantage of momentum from Stochastic Gradient Descent(SGD) and squared learning rate from RMSProp. Nadam in eq. [4] is a combination of Adam and Nesterov accelerated gradients(NAG).

$$\triangle\theta_t = -\frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\hat{m}_t \tag{3}$$

$$\triangle\theta_t = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}(\beta_1\hat{m}_t + \frac{(1-\beta_1 g_t)}{1-\beta_1^t}) \tag{4}$$

Here $\triangle\theta_t$ is a update parameter refereed as either weights or bias. In above equations η represents learning rate. v_t and $\hat{m}_t$ are estimates of the first and second moment of the gradients respectively. The below eq. [5] represents AdaBelief optimiser. Compared to adam, Adabelief does not have any extra variables. Nevertheless, in adam the paremeter update is in the direction of $m_t/\sqrt{v_t}$, where v_t is the exponential moving average of g_t and in AdaBelief, update direction is $m_t/\sqrt{s_t}$, where s_t is the exponential moving average of $(g_t - m_t^2)$. The eq. [6] shows another optimizer called Adamax. The difference between Adam and Adamax is, it extends the L2 norm of past gradients to L-infinity norm. In eq. [6], u_t represents $max(\beta_2.v_{t-1}, |g_t|)$.

$$\triangle\theta_t = \theta_t - \frac{\eta}{\sqrt{\hat{s}_t + \epsilon}} \tag{5}$$

$$\triangle\theta_t = \theta_t - \frac{\eta}{\sqrt{u_t}}\hat{m}_t \tag{6}$$

During simulation, it was observed that each optimizer provides a better Detection Rate(DR) for few categories of attacks but not all. So, we have attempted to propose a hybrid optimizer that applies multiple optimizers to select the best weights and learning rate. This results in an improved detection

```
Epoch 11/30
78125/78125 [==============================] - 243s 3ms/step - loss: 0.0061 - accuracy: 0.9973 - val_loss: 0.0056 - v
al_accuracy: 0.9975
Epoch 12/30
78125/78125 [==============================] - 243s 3ms/step - loss: 0.0061 - accuracy: 0.9973 - val_loss: 0.0078 - v
al_accuracy: 0.9963
Epoch 13/30
78125/78125 [==============================] - 243s 3ms/step - loss: 0.0062 - accuracy: 0.9973 - val_loss: 0.0057 - v
al_accuracy: 0.9975
Epoch 14/30
78120/78125 [============================>.] - ETA: 0s - loss: 0.0060 - accuracy: 0.9974
Epoch 00014: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
78125/78125 [==============================] - 244s 3ms/step - loss: 0.0060 - accuracy: 0.9974 - val_loss: 0.0058 - v
al_accuracy: 0.9975
Epoch 15/30
78125/78125 [==============================] - 244s 3ms/step - loss: 0.0034 - accuracy: 0.9988 - val_loss: 0.0028 - v
al_accuracy: 0.9992
```

Fig. 2: Reduction of learning rate

rate. However, implementing different optimizers sequentially will increase the detection time. Therefore our research work introduced muti-core to run all optimizers parallelly and the final prediction will be based on a majority voting mechanism.

In each core, a different optimizer is applied to gain better accuracy. The prediction of each optimizer is given to the Voting block. This final block selects the best based on the majority from the prediction of each optimizer.

During training the model it often stops learning as shown in fig 2. So, in this work, we explored the ReduceLROPlateau library from Keras. This library monitors the quality of training, if the improvement is not found then, the learning rate will be reduced automatically by a factor of 0.1. Fig 2 shows the scenario. During epochs 11 to 13, it stopped increasing accuracy, so ReduceLROPlateau is applied at epoch 14, and accuracy began to increase.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 256)               2560
_________________________________________________________________
dense_1 (Dense)              (None, 512)               131584
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 2056
=================================================================
Total params: 267,528
Trainable params: 267,528
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3: Sequential Model of DNN

All the experiments are conducted in a Laptop having configuration Intel Core i5-7200U CPU at 2.50GHz and Jupyter notebook having Keras and tensor flow. The Sequential model of DNN is showed in fig 3 shows the summary

Table 2: Comparative analysis of MCDNN based IDS with KPCA for IoT-Botnet dataset with other techniques

Model	Optimizer	Accuracy	Precision	Recall(DR)	F1-score
PCA with DNN	Adam	99.838	88.582	93.069	90.527
	Nadam	99.863	94.603	91.917	92.625
	Adabelief	99.858	91.464	90.741	91.046
	Adamax	99.83	87.58	95.875	90.774
SVD with DNN	Adam	99.86	89.416	93.142	91.139
	Nadam	99.847	93.932	90.317	91.276
	Adabelief	99.828	94.169	90.872	92.243
	Adamax	99.802	93.371	90.754	91.737
LDA with DNN	Adam	99.601	90.54	88.954	89.569
	Nadam	99.643	86.478	92.003	87.889
	Adabelief	99.663	88.163	92.417	89.478
	Adamax	99.616	84.983	89.939	85.762
KPCA with DNN	Adam	99.70	98.179	96.93	97.54
	Nadam	99.994	97.57	93.9	95.62
	Adabelief	99.925	95.86	94.179	95.002
	Adamax	99.946	97.895	95.039	96.4153
MCDNN based IDS	**Hybrid**	**99.937**	**97.589**	**94.761**	**96.095**

of the Sequential model using the Keras library. This provides the complete description of each layer, input-output nodes, number of hidden layers and their units, and also activation function in advance.

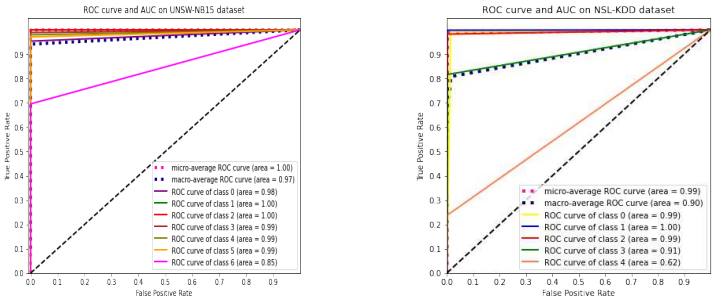
4.3 Results and discussions

In this research work, we experimented hybrid Intrusion Detection system using Deep learning techniques. Since the dataset of network traffic is huge and difficult for computation in the constrained device, the first step we focused on dimensionality reduction. To identified the optimized reduction technique, we experimented and analyzed using PCA, SVD, LDA and KPCA with DNN model as shown in table 2. It was observed that DNN with KPCA achieved higher Detection Rate(DR) rate compared to all other reduction methods. So, this method is selected for further optimization using the Multi-core technique. Finally, it is proved that our proposed method Multi-core IDS with Hybrid DNN using KPCA is more effective. The comparisons are shown in in table 2. The hybrid DNN method is also applied to the NSL-KDD dataset as shown in table 3. From experimental observations, it can be seen that the MCDNN also performs better for the NSL-KDD dataset in terms of accuracy and Detection Rate.

For better visualization of the sensitivity and specificity of our model, we considered the AUC-ROC curve. Higher AUC(Area Under Curve) indicates a higher degree of measure of separability. Whereas ROC (Receiver Operation Characteristics)curve is a probability curve plotted against a True Positive Rate (TPR) vs False Positive Rate (FPR). Fig 4 shows the ROC-AUC curve

Table 3: Comparitive analysis of Multi-core IDS using Hybrid DNN with KPCA for IoT-Botnet dataset with other techniques

Model	Optimizer	Accuracy	Precision	Recall(DR)	F1-score
KPCA with DNN	Adam	98.80	90.45	82.68	85.57
	Nadam	98.90	89.83	82.32	84.91
	Adabelief	98.82	89.61	83.26	85.81
	Adamax	98.85	88.25	82.58	84.60
MCDNN based IDS	**Hybrid**	**98.81**	**87.67**	**80.53**	**83.10**



(a) AUC-ROC for UNSW-NB15 dataset (b) AUC-ROC for NSL-KDD dataset

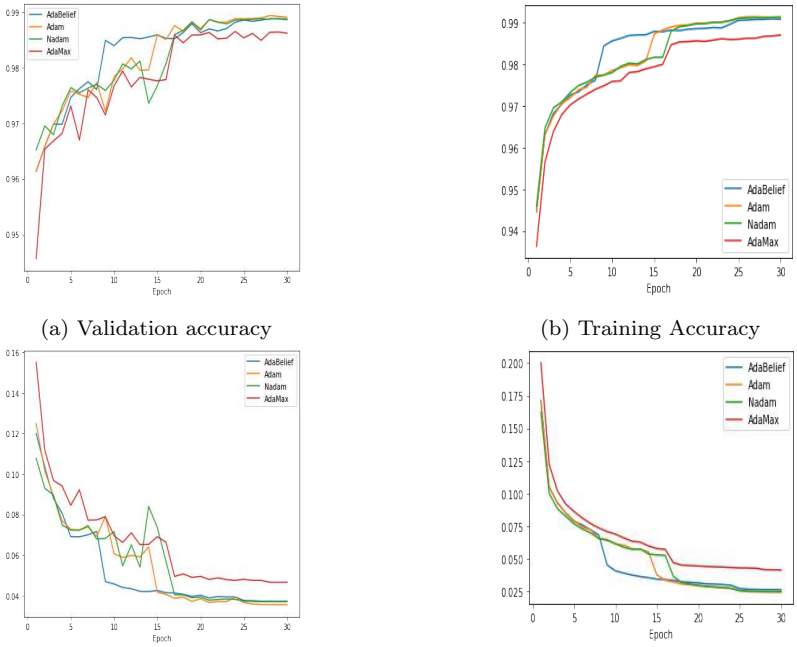
Fig. 4: AUC-ROC of different dataset

for NSL-KDD and UNSW-NB15 datasets. The figure is plotted FPR as the x-axis and TPR as the y-axis.

The learning behaviour for each epoch during training of MCDNN model should be observed to assess progress in the performance. The below fig 5 shows the complete analysis during training and validation phase for accuracy and loss. The proposed method considered 30 epochs having Leaky Relu as an activation function with 64 batch sizes.

In order to validate MCDNN with KPCA model, a recent well known methods for intrusion detection system are studied from literature and compared. The table 4 shows the complete analysis of all the methods for both UNSW NB15 and KDDCUP-99 dataset. The state of art methods includes CNN(convolution neural network) with BiLSTM (Bi-directional long short-term memory) [42], SAVAER-DNN (Supervised Adversarial Variational Auto Encoder With Regularization)[13], ELM (Extreme Learning Machine) with KPCA [17],BAT combination of BLSTM and attention mechanism [43],TL-BOSA(Teaching Learning based optimization and Simulated Annealing)[5]. TSE-IDS(Two Stage Classifier Ensemble for IDS)[44]. Feed-Forward Neural Network [45]. For a fair comparison, only the NSL-KDD dataset and UNSW-NB15 dataset are considered. The table shows the comparison results in terms of optimizers, accuracy, precision, DR and F1-score.

Apart from performance evaluation, since it is Multi core based method, we also analysed time consumption of both datasets. The time is calculated for single core for each optimizer sequentially and all optimizers parallely us-



(a) Validation accuracy

(b) Training Accuracy

(c) Validation loss

(d) Training Loss

Fig. 5: DNN performance with different optimizer

Table 4: Comparison of proposed method with benchmark methods on UNSW-NB15 and NSL- KDD

Method	Dataset	Accuracy	Precision	Recall (DR)	F1-score
CNN-BiLSTM[42]	NSL-KDD	83.58	85.82	75.49	78.3
SAVAER-DNN[13]		89.36	95.19	79.96	81.25
ELM with KPCA[17]		98.18	N/A	N/A	N/A
BAT[43]		84.25	N/A	N/A	N/A
SVM-R[5]		92.56	85.25	90.11	89.85
Hybrid feature selection with two level classifier ensemble[44]		85.797	88	86.8	N/A
Proposed method MCDNN		**98.81**	**87.67**	**80.53**	**83.10**
CNN-BiLSTM[42]	UNSW-NB15	77.16	82.63	79.91	81.25
SAVAER-DNN[13]		93.01	95.21	91.94	93.54
SVM-R [5]		89.06	85.97	88.98	87.63
Hybrid feature selection with two level classifier ensemble[44]		91.27	91.6	91.3	N/A
Feed Forward Neural Network[45]		99.5	N/A	N/A	N/A
Proposed method MCDNN		**99.94**	**97.59**	**94.76**	**96.09**

Table 5: Comparison of time complexity for sequential and multi core DNN method

Number of core	Optimizer	Time Consumption in seconds	
		UNSW-NB15	**NSL-KDD**
1 (Sequential)	adam	3597.80	309.30
	Nadam	5105.33	629.33
	Adabelief	3597.80	702.18
	Adamax	5105.33	435.09
4 (Parallel)	MCDNN based IDS (Proposed Method)	355.03	82.96

ing four core. The experiment showed that using MCDNN method consumed 355.03, 82.96 for UNSW-NB15 and NSL-KDD dataset respectively. The reduction is almost 25% down compared to sequential method. The detailed information is shown in table 5. This faster training and high accuracy are the strength of our proposed approach.

5 Conclusion

In this article, we proposed Multicore for Deep Neural Network (MCDNN) to improve the time complexity during the training phase. The method use operates different optimizers parallelly using the Linux platform. Furthermore, the efficiency of the proposed method is evaluated using the UNSW-NB15 dataset and NSL-KDD datasets. The empirical evaluation showed that the performance of MCDNN reduced time complexity to approximately 25%. Additionally, to increase the potential of the model, different performance metrics of reduction techniques are also analyzed. The emulations show that KPCA performs better compared to PCA, SVD and LDA reduction techniques.

6 Acknowledgement

I thank Cyber Range Lab of Center of UNSW Canberra Cyber for providing IoT-BoT dataset.

7 Declaration

– Funding: The work presented came from the PhD research undertaken at Department of Electronics and Communication Engineering, The National Institute of Engineering, Mysuru, Karnataka, Inida.
– Conflict: There is no conflict of interests that are directly or indirectly related to the work submitted for publication.

- Availability of data and material: Yes, data and material for transparency will be made available. The datasets analysed during the current study of UNSW-NB15 and NSL-KDD are available in the [46] and [47].
- Code availability: N/A.
- Author's contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Sharmila B S and Rohini Nagapadma.

References

[1] Jorge Martiez Carracedo et al. "Cryptography for Security in IoT". In: *2018 Fifth International Conference on Internet of Things: Systems, Management and Security* (). DOI: 10.1109/IoTSMS.2018.8554634.

[2] A Duraisamy and M Subramaniam. "Attack Detection on IoT Based Smart Cities using IDS Based MANFIS Classifier and Secure Data Transmission Using IRSA Encryption". In: *Wireless Personal Communications* (2021), pp. 1–22.

[3] Shruti Kalsi, Harleen Kaur, and Victor Chang. "DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for Key Generation". In: *J. Med. Syst.* 42.1 (Dec. 2017), p. 17. ISSN: 0148-5598. DOI: 10.1007/s10916-017-0851-z.

[4] Naman Gupta, Vinayak Naik, and Srishti Sengupta. "A firewall for Internet of Things". In: *2017 9th International Conference on Communication Systems and Networks (COMSNETS)* (Jan. 2017), pp. 411–412. ISSN: 2155-2509. DOI: 10.1109/COMSNETS.2017.7945418.

[5] Alok Kumar Shukla. "An efficient hybrid evolutionary approach for identification of zero-day attacks on wired/wireless network system". In: *Wireless Personal Communications* (2020), pp. 1–29.

[6] Waleed Bul'ajoul, Anne James, and Siraj Shaikh. "A New Architecture for Network Intrusion Detection and Prevention". In: *IEEE Access* 7 (Jan. 2019), pp. 18558–18573. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2895898.

[7] Upendra Kumar et al. "Isolation of ddos attack in iot: A new perspective". In: *Wireless Personal Communications* 114 (2020), pp. 2493–2510.

[8] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks". In: *Digital Signal Process.* 73 (Feb. 2018), pp. 1–15. ISSN: 1051-2004. DOI: 10.1016/j.dsp.2017.10.011.

[9] Abdulla Aburomman and Mamun Bin Ibne Reaz. "Review of IDS Develepment Methods in Machine Learning". In: *International Journal of Electrical and Computer Engineering (IJECE)* 6.5 (Oct. 2016), pp. 2432–2436. ISSN: 2722-2578. DOI: 10.11591/ijece.v6i5.pp2432-2436.

[10] Li-Hua Li et al. "A Feature Selection Based DNN for Intrusion Detection System". In: *2021 15th International Conference on Ubiquitous*

Information Management and Communication (IMCOM). IEEE, Jan. 2021, pp. 1–8. DOI: 10.1109/IMCOM51814.2021.9377405.

[11] Abdulrahman Al-Abassi et al. "An Ensemble Deep Learning-Based Cyber-Attack Detection in Industrial Control System". In: *IEEE Access* 8 (May 2020), pp. 83965–83973. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2992249.

[12] Francesco Mercaldo and Antonella Santone. "Deep learning for image-based mobile malware detection". In: *J. Comput. Virol. Hack. Tech.* 16.2 (June 2020), pp. 157–171. ISSN: 2263-8733. DOI: 10.1007/s11416-019-00346-7.

[13] Yanqing Yang et al. "Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder With Regularization". In: *IEEE Access* 8 (Feb. 2020), pp. 42169–42184. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2977007.

[14] Zuchao Ma, Liang Liu, and Weizhi Meng. "Towards multiple-mix-attack detection via consensus-based trust management in IoT networks". In: *Computers & Security* 96 (Sept. 2020), p. 101898. ISSN: 0167-4048. DOI: 10.1016/j.cose.2020.101898.

[15] Pooja Chaudhary and B. B. Gupta. "DDoS Detection Framework in Resource Constrained Internet of Things Domain". In: *IEEE* (2020), pp. 15–18. ISSN: 2378-8143. DOI: 10.1109/GCCE46687.2019.9015465.

[16] Yuchen Yang et al. "A Survey on Security and Privacy Issues in Internet-of-Things". In: *IEEE IoT J.* 4.5 (Apr. 2017), pp. 1250–1258. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2694844.

[17] Yuan Zhou et al. "Network Intrusion Detection Based on Kernel Principal Component Analysis and Extreme Learning Machine". In: *2018 IEEE 18th International Conference on Communication Technology (ICCT)*. IEEE, Oct. 2018, pp. 860–864. DOI: 10.1109/ICCT.2018.8600104.

[18] Lu Lv et al. "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine". In: *Knowledge-Based Systems* 195 (May 2020), p. 105648. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2020.105648.

[19] Minh Tuan Nguyen and Kiseon Kim. "Genetic convolutional neural network for intrusion detection systems". In: *Future Gener. Comput. Syst.* 113 (Dec. 2020), pp. 418–427. ISSN: 0167-739X. DOI: 10.1016/j.future.2020.07.042.

[20] Nour Moustafa, Gaurav Misra, and Jill Slay. "Generalized Outlier Gaussian Mixture technique based on Automated Association Features for Simulating and Detecting Web Application Attacks". In: *IEEE Trans. Sustainable Comput.* (Feb. 2018), p. 1. ISSN: 2377-3782. DOI: 10.1109/TSUSC.2018.2808430.

[21] Le Hoang Son et al. "ARM–AMO: An efficient association rule mining algorithm based on animal migration optimization". In: *Knowledge-Based Systems* 154 (Aug. 2018), pp. 68–80. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.04.038.

[22] Iram Bibi et al. "A Dynamic DL-Driven Architecture to Combat Sophisticated Android Malware". In: *IEEE Access* 8 (2020), pp. 129600–129612. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3009819.

[23] Saida Farhat et al. "Comparative Study of Classification Algorithms for Cloud IDS using NSL-KDD Dataset in WEKA". In: *2020 International Wireless Communications and Mobile Computing (IWCMC)* (). ISSN: 2376-6506. DOI: 10.1109/IWCMC48107.2020.9148311.

[24] *The BoT-IoT Dataset*. [Online; accessed 10. Nov. 2020]. Nov. 2020. URL: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php.

[25] Nour Moustafa and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set". In: *Information Security Journal: A Global Perspective* 25.1-3 (2016), pp. 18–31.

[26] Mohanad Sarhan et al. "Netflow datasets for machine learning-based network intrusion detection systems". In: *arXiv preprint arXiv:2011.09144* (2020).

[27] Nour Moustafa, Jill Slay, and Gideon Creech. "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks". In: *IEEE Transactions on Big Data* 5.4 (2017), pp. 481–494.

[28] Nour Moustafa, Gideon Creech, and Jill Slay. "Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models". In: *Data analytics and decision support for cybersecurity*. Springer, 2017, pp. 127–156.

[29] Shital Gulghane et al. "A Survey on Intrusion Detection System Using Machine Learning Algorithms". In: *Innovative Data Communication Technologies and Application*. Cham, Switzerland: Springer, Jan. 2020, pp. 670–675. ISBN: 978-3-030-38039-7. DOI: 10.1007/978-3-030-38040-3_76.

[30] *A Naive Bayesian Network Intrusion Detection Algorithm Based on Principal Component Analysis*. [Online; accessed 10. Nov. 2020]. Nov. 2020. URL: https://www.computer.org/csdl/proceedings-article/itme/2015/8302a325/12OmNAnuTox.

[31] T. Jolliffe Ian and Cadima Jorge. "Principal component analysis: a review and recent developments". In: *Philos. Trans. Royal Soc. A* 374.2065 (Apr. 2016). ISSN: 1471-2962. DOI: 10.1098/rsta.2015.0202.

[32] [Online; accessed 8. Nov. 2020]. June 2019. URL: https://arxiv.org/abs/1906.02590.pdf.

[33] Alan Kaylor Cline and Inderjit S. Dhillon. *Computation of the Singular Value Decomposition*. CRC Press, Jan. 2006.

[34] Sanjay Rawat, Arun K. Pujari, and V. P. Gulati. "On the Use of Singular Value Decomposition for a Fast Intrusion Detection System". In: *Electron. Notes Theor. Comput. Sci.* 142 (Jan. 2006), pp. 215–228. ISSN: 1571-0661. DOI: 10.1016/j.entcs.2004.12.043.

[35] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Kernel principal component analysis". In: *Artificial Neural Networks — ICANN'97*. Berlin, Germany: Springer, June 2005, pp. 583–588. ISBN: 978-3-540-63631-1. DOI: 10.1007/BFb0020217.

[36] Chuang Ma et al. "Network Attack Detection Based on Kernel Principal Component Analysis and Decision Tree". In: *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, Oct. 2020, pp. 84–91. DOI: 10.1109/CyberC49757.2020.00023.

[37] Aanshi Bhardwaj, Veenu Mangat, and Renu Vig. "Hyperband Tuned deep neural network with well posed stacked sparse AutoEncoder for detection of DDoS attacks in cloud". In: *IEEE Access* 8 (2020), pp. 181916–181929.

[38] Abdullah Elen and Emre Avuçlu. "Standardized Variable Distances: A distance-based machine learning method". In: *Appl. Soft Comput.* (Oct. 2020), p. 106855. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2020.106855.

[39] Chigozie Nwankpa et al. "Activation Functions: Comparison of trends in Practice and Research for Deep Learning". In: *ResearchGate* (Nov. 2018). URL: https://www.researchgate.net/publication/328826136_Activation_Functions_Comparison_of_trends_in_Practice_and_Research_for_Deep_Learning.

[40] Juntang Zhuang et al. "AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients". In: *arXiv* (Oct. 2020). eprint: 2010.07468. URL: https://arxiv.org/abs/2010.07468v5.

[41] Somenath Bera and Vimal K. Shrivastava. "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification". In: *Int. J. Remote Sens.* 41.7 (Apr. 2020), pp. 2664–2683. ISSN: 0143-1161. DOI: 10.1080/01431161.2019.1694725.

[42] Kaiyuan Jiang et al. "Network intrusion detection combined hybrid sampling with deep hierarchical network". In: *IEEE Access* 8 (2020), pp. 32464–32476.

[43] Tongtong Su et al. "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset". In: *IEEE Access* 8 (2020), pp. 29575–29585.

[44] Bayu Adhi Tama, Marco Comuzzi, and Kyung-Hyune Rhee. "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system". In: *IEEE Access* 7 (2019), pp. 94497–94507.

[45] Liu Zhiqiang et al. "Modeling Network Intrusion Detection System Using Feed-Forward Neural Network Using UNSW-NB15 Dataset". In: *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE. 2019, pp. 299–303.

[46] Nour Moustafa and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *2015 military communications and information systems conference (MilCIS)*. IEEE. 2015, pp. 1–6.

[47] *Search UNB*. URL: https://www.unb.ca/cic/datasets/nsl.html.