Network Anomaly Detection and User Behavior Analysis using Machine Learning

Priti H. Vadgaonkar
PG Student
SSVPS's B.S. Deore College of Engineering
Dhule, Maharashtra, India

ABSTRACT
Millions of people and hundreds of thousands of institutions communicate with each other over the Internet every day. In the past two decades, while the number of users using the Internet has increased very rapidly. Align to these developments, the number of attacks made on the Internet is increasing day by day. Although signature-based detection methods are used to avert these attacks, they are failed against zero-day attacks. In this study, the focus is to detect network anomaly using machine learning methods. For the implementation of proposed classifier, the graphics processing unit (GPU)-enabled TenserFlow will be used and for evaluation purpose the benchmark KDD Cup 99 and NSL-KDD datasets will be used for its wide attack diversity.On this dataset, several different machine learning algorithms will be trained and tested to make the model robust and accurate.

General Terms
Deep learning, anomaly detection, auto-encoders, network security, KDD.

Keywords
Anomaly detection, deep learning, auto encoder, PCA.

1. INTRODUCTION
Every day millions of people and institutions communicate with each other using the Internet. In the past few decades, while the use of Internet by the people has increased speedy, today this number has exceeded 4 billion and this increase is continuing speedy [2].

Comparable to these developments, the number of attacks happen on the Internet is increasing day by day. In opposition to these attacks, there are two basic methods used to detect the attacks in order to ensure information security; signature based identification, and anomaly based detection.

Signature-based methods use the database which are created to detect attacks. This method is pretty successful, but the databases need to be kept regularly updated and new attack information processed. As well, even if the databases are up-to-date, they are still vulnerable to the zero-day attacks. Since these attacks are not in the database, it is impossible to prevent these attacks. In anomaly-based approach the focus is on detecting unusual network behaviours by inspecting network flow. This method is successful in detecting attacks which are not encountered before, thus is effective against zero-day attacks [3].

Additionally, the usage of more than half of today's internet is encrypted using SSL / TLS (Secure Sockets Layer / Transport Layer Security) protocols, and this rate is increasing day by day [4]. Because of the incapability to observe the contents of the encrypted internet stream, signature-based methods are not effective on this type of data. Although, the anomaly-based approach examines the data by using its general properties such as size, connection time, and a number of packets. So, there is no need to see the message content and the analysis of encrypted protocols can also be done. Due to all these advantages, the detection and prevention of network attacks being done using anomaly-based detection method.

The provision of a powerful and effective Network Intrusion Detection System (NIDS) is one of the big challenges in network security. Despite the remarkable advances in NIDS technology, many of solutions still operate using less-capable signature-based techniques, in opposition to anomaly detection techniques. There are several reasons for this hesitation to switch, including the high false error rate, difficulty in obtaining reliable training data, the longevity of training data, and the behavioural dynamics of the system. Today's situation will reach a point by which reliance on such techniques leads to unprofitable and inaccurate detection. To create a widely-accepted anomaly detection technique that is capable of controlling limitations in modern networks is an objective of this challenge.

The classification of Intrusion Detection System is done based on where the detection takes place and based on what the detection method is used [5].

Detection takes place at:

1. Network based Intrusion Detection System (NIDS) :
To observe traffic from all devices on the network, Network intrusion detection systems (NIDS) are placed within the network. An analysis of passing traffic is performed on the whole subnet, and therefore the traffic that's passed on the subnets gets matches to the library of known attacks. Once an attack is spotted, or abnormal behaviour is noticed, the alert is often sent to the administrator. Installing NIDS on the subnet where firewalls are located to test if someone is trying to interrupt into the firewall is an example of NIDS. Ideally one would scan all inner and outer traffic, however, doing so might create a bottleneck that might reduce the general speed of the network. OPNET and NetSim are commonly used tools for imitating network intrusion detection systems. Another use of NIDS is, to link and drop harmful detected packets that have a signature matching the records within the NIDS, these systems are capable of comparing signatures for similar packets.

2. Host based Intrusion Detection System (HIDS) :
Independent hosts or devices are used to run Host intrusion detection systems (HIDS) on the network. The inner and outer packets from the device are monitored by HIDS and alert is send to the user or administrator if suspicious activity is detected on the packets. A snapshot of existing system files is taken and matches it to the past snapshot. If the critical system

files were adapted or found, an alert is sent to the administrator to inspect. Mission-critical machines, which are not expected to change their configurations can be an example of HIDS [5].

Detection methods are:

1. Signature based Intrusion Detection System (SIDS) :
The detection of attacks by considering specific patterns, such as network traffic byte sequences, or known vicious instruction sequences used by malware is suggested by Signature-based IDS. This terminology is derived from anti-virus software, which assigns to these detected patterns as signatures. It is crucial to detect new attacks, for which pattern is not available even if the signature-based IDS can easily detect known attacks.

2. Anomaly based Intrusion Detection System (AIDS) :
To detect unknown attacks the Anomaly-based intrusion detection systems were essentially introduced, to a certain extent due to the fast development of malware. The basic idea is to use machine learning to create a model of accurate activity, and then compare new behaviour against this model. The machine learning-based method has a better-popularized property compared to traditional signature-based IDS since these models can be trained as stated by the applications and hardware configurations. It may suffer from false positives: previously unknown legitimate activity may also be classified as anomalous even if the detection of previously unknown attacks is enabled by this approach. Most of the existing IDSs suffer from the time-consuming during detection process that reduces the performance of IDSs. Because of the efficient feature selection algorithm, the classification process used in detection becomes more reliable.

Anomaly detection identifies anomalous events or an unexpected behavior termed as anomalies or outliers [5].

2. RELATED WORK

Different Machine learning techniques including supervised, unsupervised, and semi-supervised, have been proposed to increase the performance of the anomaly detection systems. Supervised approaches such as k-nearest neighbor (k-NN), neural networks, and support vector machine (SVM) have been studied broadly for anomaly detection.

Dong and Wang accept a published and experimental comparison between the use of traditional NIDS techniques and deep learning methods [6]. The authors conclude that there is improvement in detection accuracy across all of sample sizes and anomaly types over the traffic using deep learning-based methods. The authors also show that problems associated with inconsistent datasets can be overcome by using the oversampling technique which is Synthetic Minority Oversampling Technique (SMOTE).

Zhao et al. [7] proposed a state-of-the-art survey of deep learning applications within machine health monitoring. The conventional machine learning methods are experimentally compared against four deep learning methods (auto-encoders, Restricted Boltzmann Machine (RBM), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Their conclusion is that deep learning methods give better accuracy than conventional methods.

This literature review point out several proposed deep learning methods specifically for NIDSs.

Alrawashdeh and Purdy [8] proposed a solution using RBM with one hidden layer to perform unsupervised feature reduction. To produce a DBN the weights are passed to another RBM. The pre-trained weights are passed through a fine-tuning layer consisting of a Logistic Regression classifier with multi-class soft-max. The KDD Cup ’99 dataset is used to evaluate propose a solution. The authors declared 97.90% detection rate and 2.47% false-negative rate. This is an improvement over the results of similar papers.

Potluri and Diedrich [11] present a method using 41 features and their DNN has 3 hidden layers (2 auto-encoders and 1 soft-max). The obtained results were mixed, the result focusing on less classes were more accurate than those with more classes. The authors allocated this to insufficient training data for some classes.

Kang and Kang [12] proposed the unsupervised DBN to train parameters to boot the DNN, which allows improved classification results. Their assessment shows improved performance in terms of classification errors.

In addition, there is other compatible work, including the DDoS detection system proposed by Niyaz et al. [13]. A deep learning-based DDoS detection system for a software-defined network (SDN) is proposed by them. Perform an evaluation using custom generated traffic traces. The authors claimed to achieve 99.82% of binary classification accuracy and 95.65% 8-class classification accuracy.

An automatic security auditing tool for short messages (SMS) is proposed by You et al. [10]. This method is based upon the RNN model. The authors claimed that their accuracy of evaluation is 92.7% than existing classification methods (e.g. SVM and Naive Bayes).

The work by Hou et al. [9] describes their commercial Android malware detection framework, Deep4MalDroid. Their method contains the use of stacked auto-encoders with the best accuracy obtained from 3 layers. The 10-fold cross-validation used, showing that as compared to shallow learning, their approach gives improved detection performance.

Lee et al. [14] give a deep-learning approach to fault monitoring in semiconductor manufacturing. A Stacked de-noising Auto-encoder (SdA) approach is used to provide an unsupervised learning solution. A comparison with conventional methods has shown that throughout different use cases, the accuracy is increased by up to 14% in different use cases.

The discovery from this literature shows that while the high accuracies of detection being achieved, there's still an opportunity for improvement. Such a fault includes the dependency on human operators, lengthy training times, uncertain or average accuracy levels, and also the heavy conversion of datasets. The realm remains in an infantile stage, for combining various algorithms and layering approaches to provide the foremost accurate and efficient solution for a selected dataset most researchers still experimenting with.

3. BACKGROUND

In this section, the background information required to understand the concepts behind the model proposed in this paper.

3.1 Deep Learning

Deep learning is an advanced sub-part of machine learning, which promotes Machine Learning closer to Artificial Intelligence. It eases the modeling of complex relationships

and concepts using multi-levels of representation. To construct successively higher levels of abstraction defined using the output features from lower levels, Supervised and unsupervised learning algorithms are used [15].

1) **Auto encoder**-The desired technique currently used within deep learning research is auto-encoders, which is used by the proposed solution in this paper. An auto encoder is an unsupervised neural network-based feature extraction algorithm, which learns the most effective parameters required to rebuild its output as near its input as possible. One of it advisable characteristics is that the capability to supply more a powerful and non-linear generalization than Principle Component Analysis (PCA).

This is performed by applying backpropagation and setting the target values to be adequate for the inputs. An auto-encoder typically has an input layer, output layer and a hidden layer. This hidden layer normally features a smaller dimension than that of the input.

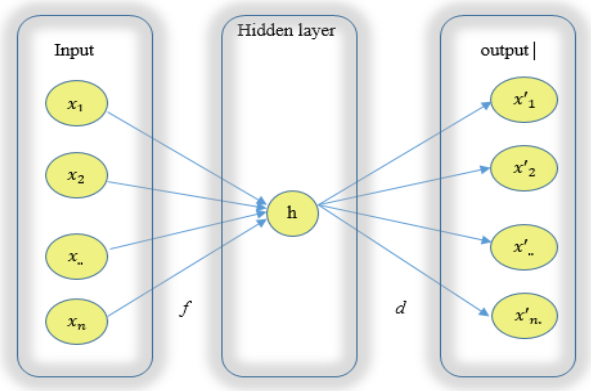

Figure 1: Example of a single auto-encoder.

First the input is passed through a typically lower-dimensional space which is an encoder, and then expanded to breed the initial data that is the decoder. Once a layer is trained, its code is passed to the following, to raised model highly non-linear dependencies within the input. This model focuses on minimizing the dimensionality of input data. There is a special layer - the code layer [16], at the center of the deep auto-encoder structure to achieve this. This code layer is employed as a compact feature vector for classification or for combine within a stacked auto-encoder.

Encoding is done by the hidden layer that it creates a low dimensionality version of high dimensionality data. By decreasing dimensionality, the most important features of the data distribution are forced to capture by the auto-encoder. In a supreme scenario, the auto-encoder generated data features will provide a more robust representation of the data points than the raw data itself.

The function shown in (1) is try and learn by auto-encoder.

$$h_{W,b}(x) \approx x \quad (1)$$

Here, h represents a non-linear hypothesis with the parameters W for weighting and b for bias, which might fit the given data (x).

Simply, it tries to find out an approximation to the identity of a function, where x' is most just like x. The learning process is expressed as a reconstruction error minimization function, as shown in (2).

$$L(x, d(f(x))) \quad (2)$$

Here, L is a loss function disciplined d (f (x)) for being dissimilar to x, d represents a decoding function and f represents an encoding function.

2) **Stacked Auto-Encoder**- far from a simple auto-encoder, a deep auto-encoder is formed using two symmetrical deep-belief networks.it contains four or five shallow layers for encoding, and the second set containing four or five layers for decoding. Deep learning can be applied to auto-encoders, thus the hidden layers are the straightforward concepts and multiple hidden layers are accustomed to provide depth, in a stacked auto-encoder. This increased depth is ready to scale down computational costs and the amount of training data required, as well as acquiring greater degrees of accuracy [23].

The output from each hidden layer is the input for a progressively higher level. Hence, the first layer of a stacked auto-encoder commonly learns first-order features in raw input. The second layer learns second-order features according to patterns that display in the first-order features. Following higher layers learn higher-order features. An example of a stacked auto-encoder is shown in Fig. 2.

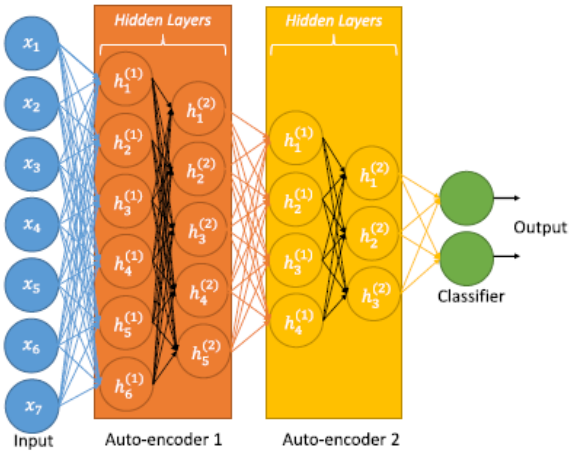

Figure 2: Example of stacked auto-encoder.

4. SYSTEM MODEL

In the anomaly detection task, the output is generated from the input. The input is network traffic on the internet and the output is the alert generated if the data is malicious.

In the detection model, the data is preprocessed. Then feature extraction is done on that data. The model is created using predefined parameters. Random forest machine learning algorithm is applied on the training data. The model is trained with lots of data. With the use of the test dataset performance of the model is evaluated. The detailed about the model is given in the following sub section.

4.1 Data Preprocessing

Data Preprocessing is one of the censorious steps in data mining process which prepare and transform of the original dataset. The varied steps are included in Data preprocessing, such as Data cleaning, Feature reduction, Feature construction [17]. Feature extraction and Feature selection are included in feature reduction. In data preprocessing, Feature extraction, selection, and construction all are independent methods. These methods can be combined based on the problem

analyzed like feature extraction followed by feature selection, feature construction followed by feature selection [18].

1. Feature Extraction and Selection:

Feature Extraction performs a transformation on data from high dimensionality to low dimensionality. Feature extraction is a process that discovers what evidence can be taken from audit data is most useful for analysis [19]. Here, the Principal Component Analysis method is employed for feature extraction. PCA is a linear method in dimensionality reduction for data analysis and compression. It is supported by transforming a comparatively large number of uncorrelated features by finding an orthogonal linear combination of the original features with the greatest variance [20].

Steps in PCA algorithm

1. Get the input data
2. Find the mean
3. Subtract the mean
4. Calculate the covariance matrix
5. Calculate the Eigen vector and Eigen value of the covariance matrix
6. Sort the Eigen value in decreasing order and forming feature vector
7. Derived the new dataset with reduced feature

The input to the PCA program is our dataset. We will find the Eigen value and Eigen vector from covariance by using the equation

$$|A - \lambda I| = 0 \quad (3)$$

Based on the Eigen value, sort the Eigen vector. The Eigen vector with the highest Eigen value represents the primary principle component of the data. For feature reduction, the K Eigen vectors with the highest Eigen values are selected.

2. Feature set for anomaly NIDS:

In anomaly detection, separate feature sets are built for every anomaly detector. A feature set of multiple connection derivative features is included for traffic based anomaly NIDS, which specifies the number of connections to a particular destination IP address and port [17].

4.2 System Workflow

The objective of this work is to identify the anomalous network communication and to find out the attacks and malicious intentions

To solve above problems, network anomaly detection system is generated as shown in fig 3

The above system works as follows,

1. The network traffic data is captured and passed to the next phase
2. It is then passed to the Intrusion/Anomaly detection phase where the following things happen:
   - Data pre-processing of the data to clean the missing/garbage values
   - Feature ranking & selection to slice data for the most important features
   - Then the machine learning based Classifier is trained and tested
   - Now the model is tested for anomaly detection and decision making
3. Then based on the detection, results alerts are generated.

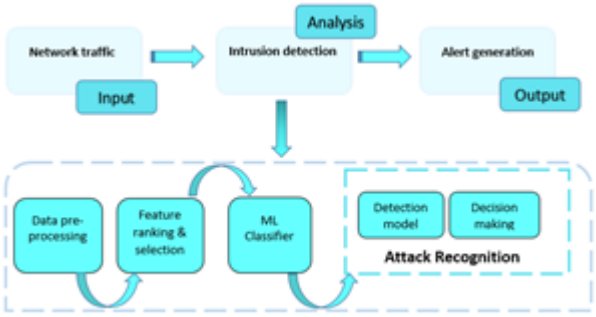

Figure 3: System architecture of network anomaly detection system

4.3 Methodology

1) Non Symmetric deep auto-encoder-

Non-symmetric deep auto-encoder is an auto-encoder featuring non-symmetrical multiple hidden layers. Basically, the present shift from the symmetric encoder-decoder pattern and towards utilizing non-symmetric i.e. just the encoder phase is involved in Non-symmetric deep auto-encoder. The reason behind this idea is to give the proper learning structure, it is possible to reduce both computational and time overheads, with minimum impact on accuracy and efficiency. As a hierarchical unsupervised feature extractor that balances well to accommodate high-dimensional inputs, Non-symmetric Deep Auto Encoder will use. A similar training strategy that is used for typical auto-encoder is applied to learn non-trivial features. An illustrated example of this is presented in the following Fig. 4 [22].

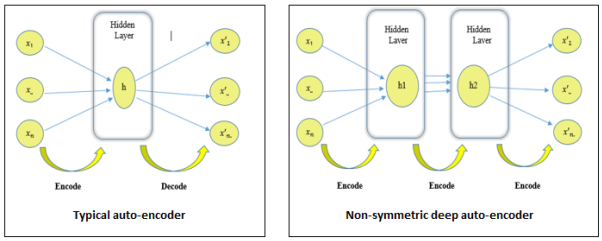

Figure 4: Comparison of a typical auto-encoder and a Non-symmetric deep auto-encoder.

The proposed Non-symmetric deep auto-encoder takes an input vector $x \in R^d$ and gradually maps it to the latent representations $h_i \in R^{d_i}$ (where d is the dimension of the vector) using a deterministic function shown in (4) below:

$$h_i = \sigma(W_i . h_{i-1} + b_i); i = 1, n \quad (4)$$

Here, $h_0 = x$, σ is an activation function (sigmoid function $\sigma(t) = 1/(1 + e^{-t})$ is used in this work) and n is the number of hidden layers.

Far from a conventional auto-encoder and deep auto-encoder, the non-symmetric deep auto-encoder does not contain a decoder. Its output vector is calculated by alike formula to (5) as the latent representation.

$$y = \sigma(W_{n+1} . h_n + b_{n+1}) \quad (5)$$

The model estimator $\theta = (W_i, b_i)$ is gained by minimising the square reconstruction error over m training samples $(x^{(i)}, y^{(i)})_{i=1}^{m}$, as shown in (6).

$$E(\theta) = \sum_{i=1}^{m} (x^{(i)} - y^{(i)})^2 \quad (6)$$

2) Stacked Non Symmetric deep auto-encoder-
This section describes the creative deep learning classification model created to deal with the problem spot with present non-symmetric deep auto-encoder as presented by Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi [22].

The model is basically relied upon using the non-symmetric deep auto-encoder technique for deep learning. This is achieved by stacking non-symmetric deep auto-encoders to form a deep learning hierarchy. A layer-wise unsupervised representation learning algorithm that is offered by stacking non-symmetric deep auto-encoder, which allow the model to learn the complex relationships between different features

Because of the data using which this model is proposed, the aim is to design the model that can handle large and sophisticated datasets. Even with the 42 features present in the KDD Cup '99 and NSL-KDD datasets being comparatively small. Here, the deep learning power of stacked non symmetric deep auto-encoder is combined with a shallow learning classifier. Random Forest is used as shallow learning classifier.

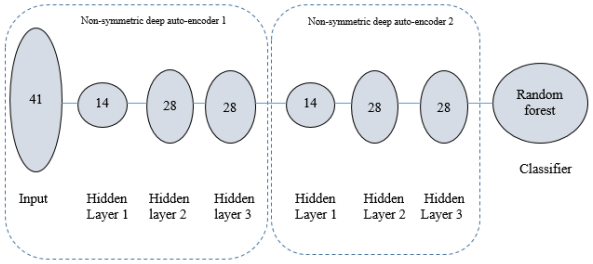

Figure 5: Stacked non-symmetric deep auto-encoder Classification Model.

Random Forest is an ensemble learning method, the principle of which is to group 'weak learners' to create a 'strong learner' [21]. In this instance, many individual decision trees (the weak learners) are combined to form a forest. RF can be considered as the bagging of these un-pruned decision trees, with a random selection of features at each split. It boasts advantages such as robustness to outlier's robustness, bias's low levels, and overfitting correction, all these are useful in a network intrusion detection system scenario.

In this model, to classify network traffic into normal and attacking the RF classifier is trained using the encoded description learned by the stacked NDAEs is used. As per Fig. 5, this model uses two NDAEs organized in a stack and is combined with the RF algorithm. 3 hidden layers are present in each NDAE, with each hidden layer contains the same number of neurons as features. By using numerous combinations (i.e. numbers of neurons and hidden layers) of cross-validating, these exact parameters are determined until the most effective is identified. For this experiment, we used the 5-fold cross-validation approach on the dataset using Scikit Learn.

5. EXPERIMENTAL RESULTS

Proposed model uses lots of training data so GPU is required to handle data efficiently. The proposed model is implemented using Github's atom IDE, Tensorflow and various packages provided by python. The model needs a machine with a 2.3 GHz Intel Xenon processor, 16 GB memory and NVIDIA GPU card coupled with 16 GB memory.

5.1 Dataset

Experiment is performed on two challenging datasets KDDCUP 99 and NSLKDD dataset.

5.1.1 KDDCUP 99

It consists of approx. 4,900,000 single connection vectors with 41 features each. These include Basic features, Domain knowledge features, and timed observation features. Each vector is labelled as either normal or as malicious. The use of 10% of the full-size dataset is common practice, as this provides reduced computational requirements with suitable representation. This 10% subset is produced and spread alongside the original dataset. Here the 10% subset is used, which contains 494,021 records for training and 311,029 testing records for testing.

5.1.2 NSLKDD

The structure of NSL-KDD dataset has basically the same as the KDD Cup '99 dataset (i.e. it contain 22 attack patterns or normal traffic, and area for 41 features). The whole NSL-KDD dataset is used for evaluations.

5.2 Methods to Compare

In this work, to compare the performance of proposed model, some advanced methods of anomaly detection are used.

KDDCUP 99

The 5-class classification performance of the proposed classification model is evaluated against the DBN model published in [8] and S-DAE model publish in [22], using the KDD Cup '99 dataset is given here.

By comparing the results of these three models, we can see that overall the proposed model, the effectiveness and accuracy of proposed model's results are better than those achieved by the model in [8] and S-NDAE [22].

NDLKDD

The paper [8] does not come up with evaluations using the NSL-KDD dataset. Hence the previously-discussed TensorFlow DBN model will use for comparisons. To boost comparability, two independent evaluations based on (A) 5-class classification as KDD Cup '99, and (B) 13-class classification from NSL-KDD are used.

1) 5-Class Classification: By using the same 5 generic class labels as used in the KDD Cup '99 dataset, we can compare the performance of the three models between the two datasets. It the performance results are presented in Table 2. From the table, it is proved that proposed model offers increased accuracy, precision when compared to the DBN and S-NDAE approach.

2) 13-Class Classification: according to previous discussion, proposed model is designed to work with larger and complex datasets. Thus, the model's classification capabilities are evaluated on a 13-class dataset. These 13 labels are those which are with more than the minimum 20 entries. The purpose of this analysis is to check and compare the stability of the model when the number of attack classes increases. So, we don't compare these results against another model.

5.3 Evaluation Metrics

The main idea behind the evaluation is to see how well the system reaches the goals and fulfills the requirements. These evaluation metrics are computed using confusion matrix

which presents four measures as follows:

- **True Positive (TP):** if an anomaly is correctly classified by model as an anomaly, it results as TP
- **False Positive (FP):** if a normal instance is incorrectly classified by model as an anomaly, it results as FP
- **True Negative (TN)** : if an anomaly is incorrectly classified by model as normal instance, it results as TN
- **False Negative (FN):** if a normal instance is correctly classified by model as normal instance, it results as FN

The following measures are used to evaluate the performance of our proposed solution:

Accuracy:
The proportion of the total number of correct classifications is measured by accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Precision:
The number of correct classifications penalised by the number of incorrect classifications is measured by precision.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

Recall:
The number of correct classifications penalised by the number of missed entries is measured by recall.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

5.4 Result Analysis

The result shows that the anomaly detection Model is showing remarkable output for the number of varying attack types as compared with other existing methods. The training algorithm of the anomaly detection model improves the performance of the system.

The proposed anomaly detection model uses the KDDCUP 99 dataset to conduct the experiments. Table 1 reports the experimental results of various models on the KDDCUP 99 testing set. From the results shown in Table 1, our anomaly detection model achieves better performance.

Table 1 KDDCUP 99 Performance

Method	Accuracy (%)	Precision (%)	Recall (%)
DBN	97.90	97.81	97.91
S-NDAE	97.85	99.09	97.85
Proposed System	99.94	99.85	99.93

As shown in above table 1, by comparing the results of all above models it is clear that the effectiveness and accuracy of the proposed system is better than the DBN and S-NDAE model.

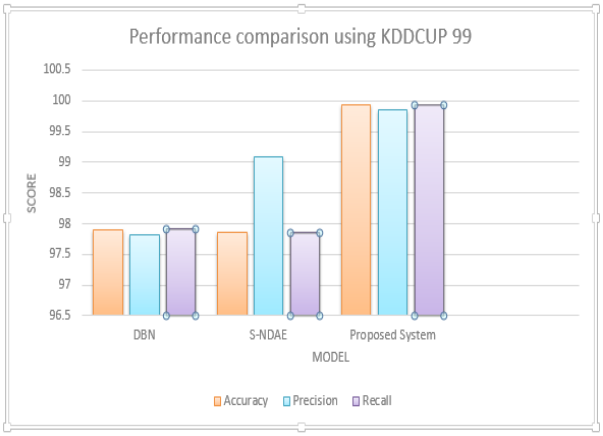

Figure 6: Performance comparison of various methods on KDDCUP 99 dataset

Figure 6 shows the performance comparison of various methods on the KDDCUP 99 dataset using accuracy, precision, and recall measures. The performance of the proposed anomaly detection model is compared against various method's performance stated in the literature.

The proposed anomaly detection model uses the NSLKDD dataset to conduct the experiments. Table 2 reports the experimental results of various models on the NSLKDD testing set.

Table 2 NSL-KDD 5-class Performance

Method	Accuracy (%)	Precision (%)	Recall (%)
DBN	80.58	88.10	80.58
S-NDAE	85.42	92.97	85.42
Proposed System	86.48	94.97	74.10

As shown in above table 2, by comparing the results of all above models using 5-class NSL-KDD classification it is clear that the effectiveness and accuracy of the proposed system is better than the DBN and S-NDAE model.

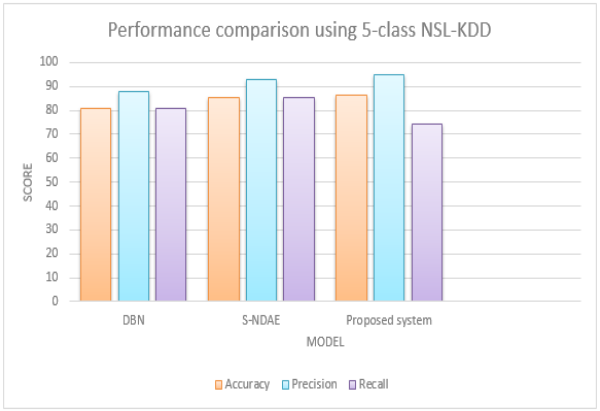

Figure 7: Performance comparison of various methods on 5-class NSLKDD dataset

Figure 7 shows performance comparison of various methods on 5-class NSL-KDD dataset using accuracy, precision, and recall measures.

Table 3 NSL-KDD 13-class performance

	Accuracy (%)	Precision (%)	Recall (%)
Proposed system	94.61	94.97	71.10

The above table shows the performance of NSL-KDD dataset with 13-class classification. The purpose of this analysis is to check the stability of proposed model when the number of attack classes increases. That's why, there is no comparison of these results against another model

6. CONCLUSION

The network anomaly detection system tries to detect anomalies happens on the network more effectively. In response to the problems faced by existing network intrusion detection system (NIDS) techniques, the novel non-symmetric deep auto encoder method for unsupervised feature learning is proposed. The model is built upon a novel classification model constructed from the stacked non-symmetric deep auto encoder and the Random Forest classification algorithm. The proposed model is implemented in TensorFlow and performed extensive evaluations on its capabilities. For evaluations purpose, KDD Cup '99 and NSL-KDD datasets are used and achieved very promising results. The results show that the proposed system offers high levels of accuracy, precision, and recall together with less training time.

7. REFERENCES

[1] S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati, "A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology," in Proc. IEEE EMBS Int. Conf. Biomed. Health Informat, FL, USA, 2017, pp. 141–144.

[2] K. Kostas, "Anomaly Detection in Networks Using Machine Learning", Research Proposal, march 2018, pp. 1-64.

[3] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters", Proceedings of the Twenty-eighth Australasian conference on Computer Science,2005, pp. 333-342.

[4] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset", Software Networking, 2017, pp. 177-200.

[5] Sonali Naikade, Akshaya Ramaswamy, Burhan Sadliwala, Prof. Dr. Pravin Futane Atmaja Sahasrabuddhel," Survey on Intrusion Detection System using Data Mining Techniques", International Research Journal of Engineering and Technology, may 2017, pp. 1780-1784.

[6] B. Dong and X. Wang," Comparison deep learning method to traditional methods using for network intrusion detection", Proc. 8th IEEE Int.Conf. Commun. Softw. Netw., Beijing, China, june 2016, pp. 581–585.

[7] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao,, "deep learning and its applications to machine health monitoring: A survey", Submitted to IEEE Trans. Neural Netw. Learn. Syst, 2016, pp. 1-14.

[8] Purdy, K. A., "Toward an online anomaly intrusion detection system based on deep learning", in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec 2016, pp. 195–200.

[9] S. Hou, S. Hou, A. Saas, L. Chen, and Y. Ye, " Deep4MalDroid: A Deep learning framework for android malware detection based on linux kernel system call graphs", in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Workshops,Omaha, NE, USA,Oct 2016, pp. 104–111.

[10] L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang , " A deep learning based RNNs model for automatic security audit of short messages", in Proc. 16th Int. Symp. Commun. Inf. Technol., Qingdao, China, sept 2016, pp. 225–229.

[11] S. Potluri and C. Diedrich," Accelerated deep neural networks for enhanced intrusion detection system", in Proc. IEEE 21st Int.Conf. Emerg. Technol. Factory Autom., Berlin, Germany, sept 2016, pp. 1–8.

[12] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security", PLoS One, june 2016.

[13] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDOS detection system in software-defined networking (SDN)", Submitted to EAI Endorsed Transactions on Security and Safety, 2017.

[14] H.-W. Lee, N.-R. Kim, and J.-H. Lee, "Deep neural network self-training based on unsupervised learning and dropout", Int. J. Fuzzy Logic Intell Syst, Mar 2017, pp. 1-9.

[15] L.Deng, "Deep learning: Methods and applications", Found. Trends Signal Process, Aug. 2014, pp. 197–387.

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", Science, 2006, pp. 504–507.

[17] Davis J.J., Clark A.J., " Data preprocessing for anomaly based network intrusion detection", Computer & Security, 2011, pp. 353-375.

[18] SomanK.P. DiwakarS., AjayV, "Insight into Data Mining Theory and Practice", PHI Learning Pvt Ltd, Third edition (2008).

[19] Sumathi S., Sivanandam S.N., "Data mining in security", Studies in Computational Intelligence (SCI), Springer 2006, pp. 629 -648.

[20] Neethu B., "Classification of Intrusion Detection Dataset using machine learning Approaches", International Journal of Electronics and Computer Science Engineering, 2012, pp. 1044-1051.

[21] L. Breiman, "Random forests," Mach. Learn., 2001, pp. 5–32.

[22] Nathan Shone , Tran Nguyen Ngoc, Vu Dinh Phai , and Qi Shi, N., "A Deep Learning Approach to Network intrusion detection", ieee transactions on emerging topics in computational intelligence, Feb. 2018, pp. 41-50.

[23] I. Goodfellow, Y. Bengio, and A. Courville,"Deep Learning", Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org