International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

A Density-based Clustering and Deep Learning Algorithm for Intrusion Detection in Sensor Networks

Chintala Tejaswini
Dept of IT, PG Scholar, VNR Vignana Institute of Engineering and Technology, Bachupally, Hyderabad, Telangana, India.

Abstract: *At present, network security is leading a dynamic role in wireless sensor networks and security becomes a precarious challenge in wireless sensor networks. Therefore, an Intrusion detection system (IDS) is a secure mechanism, which is aimed to identify and prevent from unapproved access, as it maintains harmless and protects network systems. Keeping these limitations, we present an approach called DBDNN, which combines Density-based spatial clustering of applications with noise (DBSCANSC) and the deep neural network (DNN). First, "split the given dataset into subsets depend on similarity features by core point", as in DBSCANSC. Secondly, "the distance between data points in training dataset and testing dataset calculated by using closely reachable points and which is fed input to deep neural network system". This study used KDD-Cupp99 datasets to check the implementation of the model. The experimental results indicate that the proposed DBSCANSC-DNN performs higher than Bayesian classifier (Bayes), Backpropagation Neural Networks (BPNN), Spectral Clustering and Deep Neural Networks (SCDNN) and the Support Vector Machine (SVM). Finally, the proposed method provides an effective technique for analysis of IDS in huge networks of anomalous attacks detection.*

KEYWORDS: Density-based clustering, Deep Neural Networks, an Intrusion Detection System, Wireless Sensor Networks.

I INTRODCTON

Wireless networks are now organized everywhere and became universal in nature, "Wireless technologies enable users to attach their movable devices to the network and connect the web to none physical network port and Wireless networks are gaining additional attention on its easiness of deployment". Protecting the data roaming through the wireless networks has turn into the foremost and necessary part of any online network. [1]

The principal purpose of an IDS is the recognition of real-time things. Let us consider," an IDS can produce the keys of an analysis of a network, recognizes while an attack is being dumped, and reports the incident to security administrators". For illustration, a few types of malware may perform port output with a specific finale goal to recognize the conceivable focuses on an attack [2].

Yet, intrusion detection has leaned on numerous ways in which first the training capability of early detection methodologies, which adds options and map data into vectors that is fed to classifier. Finally, the range of network varieties has caused large-scale knowledge by means of high-dimensional structures, that ancient intrusion detection methodologies are inappropriate [3].

II RELATED WORK

Recently, the deep learning process became a well-liked subject of analysis, and ways that supported have successfully been applied in numerous analysis and communication acknowledgment. In addition to, these traditional and mobile webs and taking consideration of varied intrusion ways accessible to malicious proxies, the technical desires of intrusion detection, square measure is further advanced than earlier.
Spasms happening on WSNs square measure is quite simple to hold ready than those on supported networks as a result of the delivery of WSN is fundamentally restricted, and since the multi-hop networking, information measure and usage of power battery in this. Hence, "coming up with an efficient IDS for WSN is incredibly

vital because of this new atmosphere, new attacks are devised (e.g., irregular transmission and packet reducing attacks)". The anomaly finding technique, for example, is widely used for security in WSNs [4].

Problem Statement:

However, intrusion detection has been insufficient in several ways. First, the learning capacity of traditional detection approaches that sum features of the raw data, map them into vectors, and then feed them to a classifier is limited. When network structure is complicated, learning efficiency further decreases. Second, this primary method only partially represents one or two levels of information; it is insufficient for identifying additional attack types. Third, in real network datasets, the types of network intrusion are similar to those in normal datasets, which confine classifiers from having enough information with which to categorize them. Next, the intrusion actions behave unpredictably, which causes IDS to make costly errors in detecting intrusions. Therefore, it is necessary to find an effective intrusion detection method. Finally, the variety of network types has generated large-scale data with high-dimensional structures, for which traditional intrusion detection approaches are unsuitable [5].

III EXISTING SYSTEM MODEL

Spectral Clustering Algorithm:

Spectral clustering to utilize the best eigenvectors that are got from the input information on a matrix and changes the clustering issue into a diagram cut issue. "The graph cut method gathers data points by attributes such that closely packed points are in a similar cluster, while the sparse are in other clusters". The formula of minimal cut is as follows:[6]

$$\text{Cut}(m, n) = \sum D_{ij}$$

Where $D_{ij\ is}$ the degree of balanced approaches that is "N cut & Ratio cut"; SC process with a Laplacian matrix, which is explained below:[7]

Input: Dataset, k clusters, mean σ & number of iteration
Output: The set of k clusters

Step 1: Calculate the affinity matrix $A \in R^n$ and defined as

$$A_{ij} = \exp(-|\,|S_i - S_j|\,/\,2\sigma^2)|$$

If $i \neq j$ then $A_{ij} = 0$

Step 2: The D is that the diagonal matrix and composed of features: $d_i = \sum_{j=1}^{n} A_{ij}$ and a Laplacian matrix which is the difference of affinity from degree matrix.

Step 3: Finding the largest of k eigenvectors of matrix L

Step 4: Generating matrix y, by renormalizing to each row and decreasing the distortion to each row in clustering by means of algorithm.

Finally, the unusual point is allotted near cluster j while the row of y_i goes to the cluster j and returns the group of clusters and their centers.

Deep neural network Algorithm:

The significance of the DNN is to create multi-layered networks and make learn of useful features of a huge amount of trained datasets." Forecast exactness is enhanced using DNNs, letting more data on the original dataset to be acquired DNN has profound models containing numerous hidden layers then each concealed layer alone directs non-straight changes from the past layer of the following" [8].

3.1. **Auto-Encoders***:* "The encoder system distributes input information used mainly for

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

dimensionality reduction purpose, and the decoder system adapts this response from the earlier step" and it is denoted as E_f. This function details the encoding process:

$$E_v = E_v (x^m)$$

Where x^m is a input point and E_v is an encoded vector from x^m.

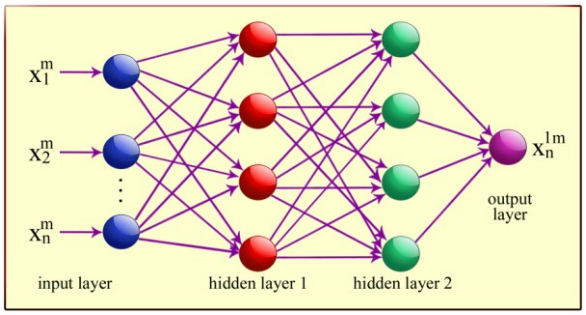


Figure 1: Network of "Auto-encoder & decoder in a DNN".

3.2. **Decoder:** The decoder reconstructs the function which is termed as E_d and is mentioned as follows:

$$x'^m = E_d (E_v)$$

Where "x'^m is the decoding vector found from E_v". There are definite algorithms for some encoding as well as reconstruction functions, including:

$$z = f(y) = \log sig(y) = 1 / 1 + e^{-xm}$$

$$E_f(x^m) = \begin{cases} 0 \text{ if } x^m \geq 1 \\ z \text{ if } 0 < x^m < 1 \\ 1 \text{ if } x^m \geq 1 \end{cases}$$

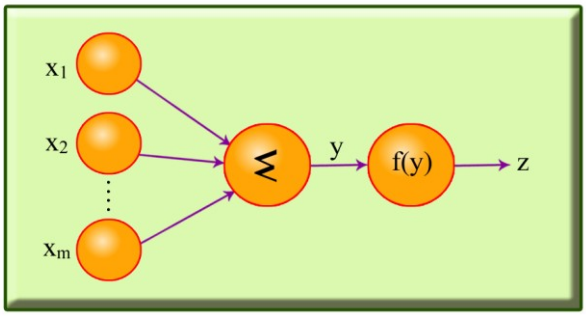


Fig 2: Re-construction function of DNN

3.3. **Sparse Auto-Encoder (SAE):** Sparse auto encoder is a kind of encoder by a sparsity enforces that directs a one layer network to absorb a code form that reduces reconstruction error while restricting the number of code-words necessary for reprocessing.

The basic sparse auto-encoder involves only one layer h, which is linked to response vector x, by the weight matrix W developing the encoding step [9].

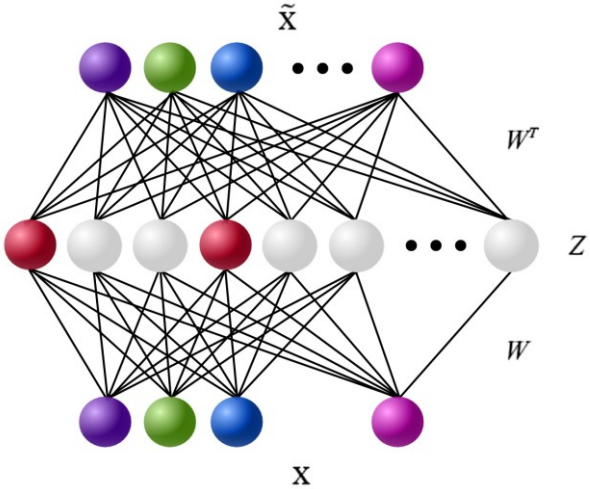


Figure 3: The Sparse auto-encoder network.

Now hidden layer output is given to reconstruction vector x', using a weight matrix W^t to get the decoder. The active function of f and b is in typical bias term.

$$z = f (Wx + b)$$

$$x' = f(W^t z + b')$$

Learning occurs to back propagation on the reconstruction error.

$$\text{Min}\|x - x'\|_2^2$$

The input vector x is converted to a sparse representation of hidden layer z and then reconstructed as x'.

Since the network setup is available, the following step is to include a sparsifying segment that drives the vector z to a sparse format. K-sparse auto-encoders finds the k highest activation in z and zeros from the rest. This error is back propagated only through the k active nodes of h. For low levels of k (very low sparse), k is scaled down gradually over the course of training [10].

IV PROPOSED APPROACH

4.1. DBSCAN Clustering:

DBSCAN is a popular clustering algorithm that try to discover clusters of data points which are dense in regions i.e.., closely packed points of different sizes and shapes. It is completely based on center-oriented approach, density is predicted for a specific point by calculating the number of points present within a known radius, Eps which is referred as a threshold point of a certain data point. It also classifies a point as a boarder point (point which is near to Eps) or core point (point which is within Eps) or noise point (which is outside of radius). The relation between data points also matters to discover the clusters that categorized as density-connected or density reachable. A cluster is recognized by keeping density as a condition and checking whether its high or density of data points indicates as cluster or outlier. This algorithm deals with the huge quantity of datasets which is of any shape and size.

To turn dataset into clusters, "it will start by identifying k closest neighbors of every data point and also identifying the point which is farthest, then the average distance of all this is calculated." Later on, the algorithm finds the data points which are density- reachable and declares it as a core point or border point. It repeats this procedure unless perfect clusters are made. Finally, it is thoroughly verified if there is any possibility to append any two data points of different clusters where their distance is less than the threshold point [11].

4.2. DBSCANDNN

The trained data sub-sets split the process of training and compute center points in DBSCAN from every training point. Next, every DNN learns the features of trained datasets unless it is same as clusters found. Third, "the tested data sub-sets are parted from test data sets by DBSCAN, where earlier cluster centers is applied on its first phase, and these sub datasets are efficient to discover attack kinds by using pre-prepared DNNs". At last, the result is intrusion detection.

Algorithm: DBDNN

Input: Datasets, Amount of clusters, Hidden layers HL and their nodes HLN.
Output: Accurate attacks classification.

Step 1: Riven given data into dualistic sets as training datasets and the testing datasets.

Step 2: Center points do the DBSCAN clustering and it is considered as training data subsets.

Step 3: Learning scale and parameters of sparsity computing is determined and weights & bias values are generally initialized.

Step 4: The nodes in the HLs are set and retrieves them based on clustered data subsets.

Step 5: Sparsity of that cost function computing generalized.

Step 6: Calculates the values of weights & bias and keeps updating it.

$$Z= f(Wx + b)$$
$$X'= f(W^t z +b')$$

Step 7: Repeat the process for multiple training subsets until the genuine results found.

Step 8: Adjust the sub DNN sets using back propagation to learn them.

Step 9: Now, the testing sub-datasets are taken to test their respective sub DNN set with respective to center points in testing & training sets.

Step 10: Finally, looking up to DNN values and the classification of attacks are realized.

The DBDNN model follows below-mentioned as shown in figure 4:

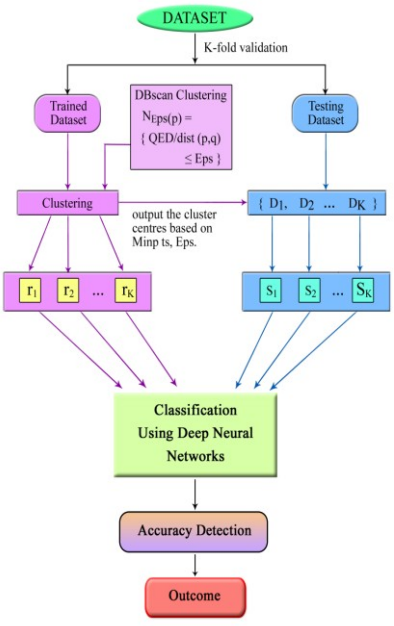


Figure 4: Architecture of DBDNN model

V EXPERIMENTAL STUDY

DATASET:

In this experiment, we consider KDD-CUP99 dataset to examine and compare DBSCAN, with SCDNNs and SVM. It incorporates 4,900,000 datasets, and each dataset has the function that is a part of the characteristic set. The queries about the dataset are categorized as regular queries and some categories of attacks [12].

The attacks generally fall into four natures: Denial of service (DoS) this attack overloads the server with a maximum number of requests. The crucial thing is to block the path of the application as an example of apache, smurf etc.

Second Probe attacks, on this kind of attack, the hacker searches weakness of device which is used to take advantage later if needed. Port sweeps is a simple example of this form of attacks.
Third, User to root (U2R) assault is any interest carried out via the person who gets right of entry to the device as an ordinary user to take advantage on the capability to get root access to device. This attack pursuit to get the credential statistics approximately.

Fourth, R2L root to level attack the gate crashed tries to abuse the framework vulnerabilities with a specific give up the intention to govern the remote system via the machine as a neighborhood client. In this work, accuracy and recall are utilized to determine the performance of the detection models [13].

Dataset-1	Model	Normal	DoS	Probe	U2R	R2L	Accuracy
KDD	DBSCANSCDNN	99	99	84	19	9	92.34
	SCDNN	99	97	81	17	8	91.89
	SVM	98.3	94	64	11	6	81.23
Dataset-2	**Model**	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**	**Accuracy**
	DBSCANSCDNN	98.32	97.33	77.34	7	5	91.64
	SCDNN	98.12	97.2	71	8.2	5.3	92.19
	SVM	97.23	97	66	3.4	3	88.33
Dataset-3	**Model**	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**	**Accuracy**
	DBSCANSCDNN	97.32	97.23	67.34	7.2	7.21	92.64
	SCDNN	98.21	97.22	66.45	4.2	6.3	91.91
	SVM	97.24	96.34	64.34	1	1	91.21
Dataset-4	**Model**	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**	**Accuracy**
	DBSCANSCDNN	99.32	76.45	54.34	4	3.45	79.67
	SCDNN	96.21	77.34	51.45	4	3.3	79.55
	SVM	95.49	71.34	53.64	0	1.23	77.21
Dataset-5	**Model**	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**	**Accuracy**
	DBSCANSCDNN	98.25	75.65	65.43	5	2.46	73.64
	SCDNN	98.19	74.35	61.45	5	2.23	72.54
	SVM	97.49	71.34	49.34	2	0	65.19
Dataset-6	**Model**	**Normal**	**DoS**	**Probe**	**U2R**	**R2L**	**Accuracy**
	DBSCANSCDNN	88.32	56.76	54.34	1	1.35	51.73
	SCDNN	87.21	57.38	56.35	1	1.2	49.28
	SVM	86.49	47.34	54.57	0	0	35.29

Table 1: Comparing intrusion network results among three models.

5.1. Estimation methods:

For this model, accuracy and recall are utilized to estimate and analyze the act of the detection models. To test that, by using following formulas are calculated[14]:

$$\text{Accuary} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

A True positive (TP) is said when it found what kind of attack in network, a true negative (TN) demonstrates ordinary network information characterized accurately as normal, a false negative (FN) means a given network is ordinary data flow, and a false positive (FP) implies that a typical case was declared as an attack. The exactness level of attacks demonstrates the general right location precision of the data sets, ER indicates the strength of the result, and review demonstrates the level of effectively recognized attacks categorization of all cases named attacks.

5.2. Comparison of results:

The proposed approach DBDNN accuracy levels over SCDNN and SVM are mentioned in below the table.

The next section, is comparing the results of intrusion detection network for six data subsets among different classifiers.

From table-2, considering general exactness, the DBDNN preforms superior to the other four techniques and has minimal fault rates. Besides, the proposed technique indicates particularly great execution for the inadequate User to Root and Root to Level attacks sorts in all data subsets, and gives a greater exactness rate. The top exactness rate for first dataset is 99%, gotten in DBDNN. The SVM has 98.3% exactness for typical information, demonstrating that DBDNN has better taken in the highlights of the information than different models. All techniques are viable for interruption to this dataset, aside from the others strategy, which has little exactness and 0% recognition precision for User to Root & Root to Level attacks

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue-17
December 2017

Dataset	Model	DoS	Normal	Probe	R2L	U2R	Accuracy	Recall
Dataset-1	Bayes	95.69	90.51	62.35	3.56	3.39	89.48	92.56
	BP	89.02	97.21	45.19	1.99	9.49	85.44	13.36
	SVM	94	98.3	64	6	11	81.23	78
	SCDNN	97	99	81	8	17	91.89	91.11
	DBDNN	99	99	84	9	19	92.34	90.81
Dataset-2	Bayes	96	96.71	62.3	4.3	4.8	90.6	91.01
	BP	97.4	91.1	63.6	6.41	6.01	91.9	92.4
	SVM	97	97.23	66	3	3.4	88.33	89.3
	SCDNN	97.2	98.12	71	5.3	8.2	92.19	90.1
	DBDNN	97.33	98.32	77.34	5	7	91.64	91.09
Dataset-3	Bayes	94.2	95.3	58.1	6.9	6.5	92.3	91.1
	BP	95.9	82.6	8.1	7.5	6.1	89.01	90.1
	SVM	96.34	97.24	64.56	1	1	91.21	89.03
	SCDNN	97.22	98.21	66.45	6.3	4.2	91.91	90.01
	DBDNN	97.23	97.32	67.34	7.21	7.2	92.64	92.01
Dataset-4	Bayes	75.1	94.5	40.1	0	1	65.1	55.7
	BP	72.4	97.3	67.5	0	0	75.5	57.9
	SVM	71.34	95.49	53.64	1.23	0	77.21	56.32
	SCDNN	77.34	96.21	51.45	3.3	4	79.55	65.58
	DBDNN	76.45	99.32	54.34	3.45	4	79.67	68.8
Dataset-5	Bayes	68.2	98.6	44.4	0	0	65.8	45.1
	BP	7.9	92.5	66.51	2	1	52.4	44.5
	SVM	71.34	97.49	49.34	0	2	65.19	47.1
	SCDNN	74.35	98.19	61.45	2.23	5	72.54	48.8
	DBDNN	75.65	98.25	65.43	2.46	5	73.64	49.1
Dataset-6	Bayes	49.9	83.3	25.5	0	0	39.9	35.7
	BP	41.6	75.6	87.1	2	0	42.3	32.3
	SVM	47.34	86.49	54.57	0	0	35.29	33.4
	SCDNN	57.38	87.21	56.35	1.2	1	49.28	39.8
	DBDNN	56.76	88.32	54.34	1.35	1	51.73	41.2

Table 2: Comparing intrusion network detection results among few classifiers for six datasets.

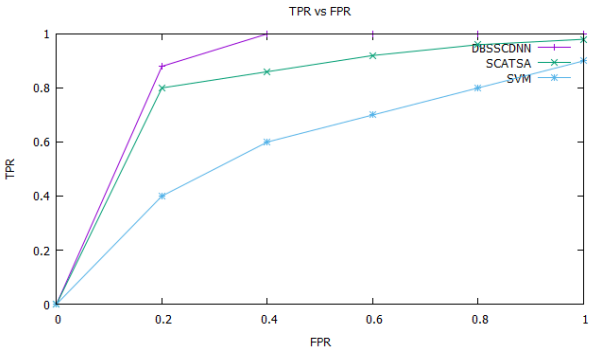
From dataset-2 as usual, DBDNN accomplishes comparing to other classifiers and has the minimal fault rate. It shows the best enactment for the U2R & R2L types of attacks of other classifiers. DBDNN holds 91.64% typical Normal, Probe, DoS activity with rates 98.32, 97.33, 77.34 separately and brings down accuracy for U2R & R2L types.

In dataset-3, the DBDNN performs top on DoS, with exactness levels of 97.32 and 97.23 individually. Furthermore, the Bayes demonstrate has the most amazing rates 7.2% & 7.21% separately for U2R information on all strategies. The General accuracy of all techniques is very low on Dataset-6, indeed the fact that the DBDNN performs better than different strategies in general precision and review.

It acquires high general precision and review. These outcomes demonstrate that DNNs are strong and prepared to do distinguishing that DBDNN is more reasonable for intrusion identification.

As appeared in above table, the five attacks are extreme. The last three attacks are the oddest. As indicated by the table results, the DBDNN achieves well in distinguishing small kind of attacks:/ R2L, probe & U2R.

To assess the execution of the DBDNN and the experimental outcomes of the other models in this area, a recipient operated curve (ROC) is computed to every model out there. The ROC work is generally used to demonstrate an algorithm's discriminative ability in this category. The general classification execution for each model, ROC is produced by using the true positivity rate (TPR) and the false positivity rate (FNR) [15].



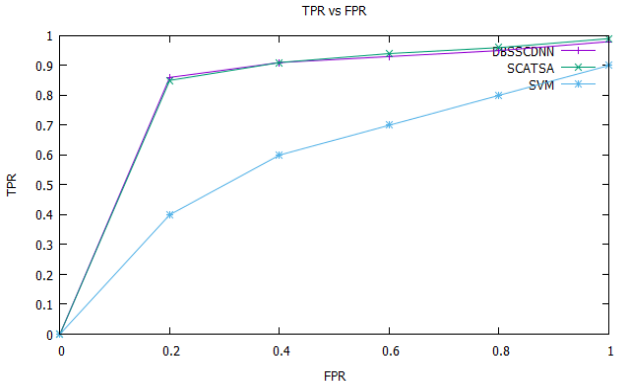


Figure 5: Recipient operated curves (ROC) of three models.

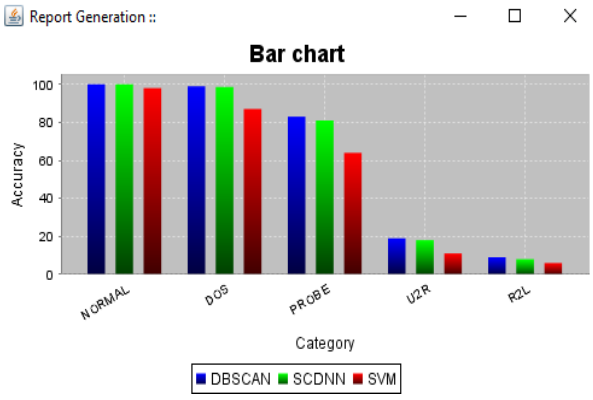


Figure 6: Comparison of attacks accuracy among three models.

In this circumstance, multi-class ROC & zone underneath curve (ZUC) strategies are utilized to point out ROC and compute every ZUC. This indicates that DBDNN has the biggest ZUC as 0.9 of the three models for the datasets. This demonstrates the DBDNN performed superior to different models and can get higher identification rates in systems.

The outcomes appeared in figure 5 show that the DBDNN produces greater precision than different techniques in six data subsets along these lines, "the proposed calculation accomplishes well on data sets with a scope of dispersions. The SCDNN has the best review of 92.19% and the DBDNN technique gets a finest precision of 91.64% in Dataset-2.

From the above examination, we see that the DBDNN calculation is not just great at recognizing the typical data flow, and in addition, Probe & DoS attacks, yet additionally acquired greater exactness for meager attack sorts U2R & R2L in the six data subsets. The DBDNN model is a sensible methodology to intrusion location in multifaceted systems.

CONCLUSIONS

This paper is presenting an approach that explains how DBSCAN benefited when combined with deep neural networks to identify attacks. In the initial step, "features of a network are caught by clusters and separated from k-sub datasets in a proposal to find further learning and patterns as of same clusters". In the next step, "deep learning models from the subsets created in the clustering procedure acquire extremely abstract features".

To conclude, testing subsets are utilized to recognize attacks. This is a proficient approach to enhance recognition rate exactness. Experimental results demonstrating that the DBDNN achieves superior to SVM, SCDNN strategies with the top precision rates over one dataset got from the KDDCUP99. Furthermore, the calculation is more equipped for grouping inadequate attack cases and successfully enhances detection accuracy.

REFERENCES

1. Denning, D.E. An intrusion-detection model. IEEE Trans. Softw. Eng. 1987, SE-13, 222–232. Sommer, R.; Paxson, V. Outside the closed world: On using machine learning for network intrusion detection.
2. Guide to Intrusion Detection and Prevention Systems (IDPS); Jasinski, R.; Pedroni, V. A.; Oliveira, L. S. (2017-01-01).
3.Zhang, Y.; Lee,W.; Huang, Y.A. Intrusion detection techniques for mobile wireless networks. Wirel. Network.
4. Dayu Yang, Alexander Usynin, and J. Wesley Hines, ―Anomaly-Based Intrusion Detection for SCADA Systems‖ IAEA Technical Meeting on Cyber Security of NPP I&C and Information systems, Idaho Fall, ID, Oct.2006.
5.Roman, R.; Zhou, J.; Lopez, J. Applying intrusion detection systems to wireless sensor networks.In Proceedings of the IEEE Consumer Communications & Networking Conference (CCNC 2006), Las Vegas
6. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. Adv. Neural Inf.
7. Shi, J.; Malik, J. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2000,22, 888–905.
8. Schmidhuber, J. Deep learning in neural networks: An overview. Neural Netw. 2015, 61, 85–117.
9. Olshausen, B.A.; Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 1996, 381, 607–609.
10. Ng, A. Sparse Autoencoder; CS294A Lecture notes; Stanford University: Stanford, CA, USA, 2011.
11. Jain Anil K. 1988. Algorithms for Clustering Data. Prentice Hall. Kaufman L., and Rousseeuw RJ. 1990. Finding Groups #~ Data: an Introduction to Cluster Analysis. John Wiley & Sons.
12. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. Science 2006,
313, 504–507.
13.Yi, Y.; Wu, J.; Xu, W. Incremental SVM based on reserved set for network intrusion detection.
Expert Syst. Appl. 2011, 38, 7698–7707.
14. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. A hierarchical SOM-based intrusion detection system.Eng. Appl. Artif. Intell. 2007, 20, 439–451.
15. Maxion, R.A.; Roberts, R.R. Proper Use of ROC Curves in Intrusion/Anomaly Detection; University of Newcastleupon Tyne, Computing Science: Tyne, UK, 2004.