ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NETWORK INTRUSION DETECTION BASED ON ONE-DIMENSIONAL CNN WITH CHIMP OPTIMIZATION ALGORITHM

Dr. V. GOKULA KRISHNAN[1], Dr. M. V. VIJAYA SARADHI[2], Dr. S. VENKATA LAKSHMI[3], S. KAVIARASAN[4], ABOTHU GEETHA[5]

[1]Professor, Department of CSE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Thandalam, Chennai, Tamil Nadu, India
[2]Professor & Head, Department of CSE, ACE Engineering College, Ghatkesar, Hyderabad, Telangana, India
[3]Professor, Department of AIDS, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India
[4]Assistant Professor, Department of CSE, Panimalar Engineering College, Poonamallee, Chennai, Tamil Nadu, India
[5]Assistant Professor, Department of CSIT, CVR College of Engineering, Mangalpally, Hyderabad, India

Email: [1]gokul_kris143@yahoo.com, [2]meduri.vsd@gmail.com, [3]venkatalakshmis227@gmail.com, [4]arasan.kavi@gmail.com, [5]songagitaabothu@gmail.com

ABSTRACT

The widespread development of the Internet of Things (IoT) is known throughout the world. The 2016 Dyn cyberattack revealed significant flaws among smart grids. IoT security has become one of the top concerns. The security of the entire IoT environment is affected by the affected networks that are connected to the risks posed by contacts. Nowadays, the diversity and complexity are evolved by defense attack vectors in recent times. It is one of the important things to prevent, identify or detect the new attacks in the IoT environment by analyzing the techniques. Therefore, network intrusion detection systems (NIDS) play an important role in protecting computer networks. Detection of security-related events using machine learning approaches has been extensively explored in the past. In particular, machine learning-based web browsing detection has attracted a lot of attention due to its ability to detect unknown attacks. Many classification techniques such as Decision tree (DT), Support Vector Machine (SVM) have been used for that purpose, but they were mostly classical schemes, like final trees. In this study, the use of deep learning technique is explored for NIDS. Initially, the noise samples are minimized in the majority segment by using One-Sided Selection (OSS) and then, Synthetic Minority Over-sampling Technique (SMOTE) is used to develop the minority samples for creating the balanced datasets. In this way, the research work is used to fully understand the characteristics of minority models and greatly reduce the sample training time. Second, we use a one-dimensional convolutional neural network (1D-CNN) with the Chimp optimization algorithm (COA) to extract the features, creating a hierarchical network model (HNM). The research work tested the classification accuracy of CNN-COA with existing techniques and its performance is verified by experiments in the NSL-KDD database. The proposed model achieved 87.19% of accuracy, 88% to 89% of precision and recall, where the existing model CNN achieved 81.75% of accuracy and 82% of precision and recall.

Keywords: *Attacks, Chimp Optimization Algorithm, Network Intrusion Detection Systems, One-Side Selection, Synthetic Minority Over-sampling Technique, UNSW-NB15.*

1. INTRODUCTION

Network security has become a major problem in computer systems, especially in decentralized and dynamic temporary networks. Temporary networks are organized without a standard security plan, which carries the issues of security. Although there are great mechanisms to increase network security [1], existing methods are not adequate to ensure security against changing attacks. For instance, the WannaCry ransaomware infected more than 150,000 devices in May 2017 [2]. In general, ID includes abuse caused by known attacks and unusual actions based on unknown threats [3]. The historical data is studied to detect the unknown attacks, where behavioral abuse is identified by using malicious traffic data [4]. Therefore, IDS is essential to effectively identify known attacks and gain more power to detect unknown inconsistencies simultaneously. IDS will automatically detect mischievous activity or policy violations by detecting random network patterns.

Most of the previous work has practiced leak detection, a traditional statistical study method. Inspired by the significant effect of in-depth study, several recent studies have used neutral networks to detect infiltration, including the multilayer perceptron, the convulsive neural network [9], and the continuous neural network [10]. There are three common problems faced by current ML/DL models that includes it had high level of intrusion with high false positive rates [11], it does not contain common databases and the need for cutting-edge solutions to maintain today's rapidly growing network traffic in a multidisciplinary environment. Additionally, unbalanced class data remains a problem that hampers the performance of most IDS [12]. When the total numbers of intrusion samples are less than normal samples, this problem is called data imbalance and it is considered as one of the major issues in NIDS. The degree of imbalance is a measure of inequality ratio between minority and majority samples in terms of sizes. Interest class events, that is, the minority class (aggression class) are often overlooked due to their representation compared to the majority class (normal class). However, while this process places great value on accuracy, IDS fail without effective protection against attacking traffic. Therefore, special technologies are needed to give due importance to the minority community.

In this study, we proposed a new NIDS algorithm that combines the hybrid model with a HNM for increasing the detection rate. There are two parts in this research work, first is to reduce the noisy samples in majority class and eliminate the data imbalance from the network traffic by adjusting the minority classes. Therefore, unbalanced data can be converted to balanced data for the next classification. For the data characteristics problem, the spatial and temporal features are extracted by using 1D-CNN with COA and create the HNM in NIDS. By using the proposed model, accurately we extract the network traffic data characteristics. Finally, the samples with better classification performance are attained after training in this work.

2. LITERATURE REVIEW

For dimensionality reduction, the author [13] developed the algorithm of discriminative low-rank preserving projection (DLRPP), where the author from [14] proposed the method called structured optimal graph based sparse feature extraction (SOGSFE) for supervised learning.

In order to develop the accurate end-to-end NIDS system, the author from [15] developed the multi-layer learning model by combining deep CNN with Multigrained Cascade Forest (gcForest). This technique uses the small-scale data with few hyper parameters and attained specific detection on imbalanced data, when compared with existing DL methods.

The intrusions are detected in cyber physical system by developing the CNN model in [16]. The dataset called NSL-KDD is used to test the performance of CNN with existing models. The simulated results proved that it achieved 80.07% of accuracy on 2 class and 77.15% of accuracy on 5 class data classification.

The author from [17] developed a hybrid model called CNN with LSTM for identifying the attacks in NIDS and used the CICIDS2017 dataset for analysis. It has the rate of 98.67% of accuracy than existing CNN and LSTM model. To improve the IoT security, DNN with kNN model along with Random Forest (RF) is developed by the author from [18] and used two types of datasets called NSL-KDD and CICIDS2017. In NSL-KDD, the method achieved the accuracy of 99.77% and the same method achieved 99.85% in other dataset.

In big data, the anomalies are detected and reduced the FPR by developing a DL approach in [19]. ML and DL methods were compared in the study. The DL method reduces the number of false positives by 10%. The author from [20] uses the ANN model with regularization and uses three different types of datasets called UNSW-NB15, CIDDS001 and NSL-KDD for training and testing process. The author from [21] made a comparison of RF and KNN performance by using CIDDS-001 dataset. In the study, the training results of the machine learning algorithms were compared based on the class label and the attack type label.

The minority classes are oversampled by SMOTE and used the six different types of ML algorithms in [22]. CIC-IDS2018 dataset is used for analysis process. The experimental results show that the proposed model is larger than the current models. However, the pre-techniques used in the NIDS domain have poor taxonomic performance, high FAR, and low detection rates in NIDs.

The existing models focused on detecting the attacks by the presence of oversamples or undersamples and this will lead to poor classification accuracy. This problem is solved by balancing the dataset and then, predicting the attack, which is carried out by the proposed model. The brief explanation of proposed model is provided in the following section.

3. PROPOSED METHODOLOGY

The general flow of the NIDS model in this document is shown in Figure 1. Normalize the data and other pre-processes of the hybrid model to obtain symmetry data. Finally, a HNM is used to classify the attacks.

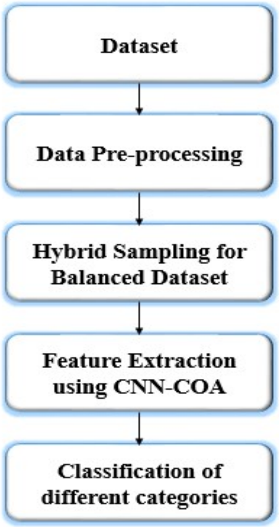
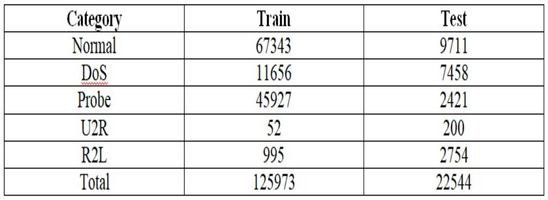

Dataset
Data Pre-processing
Hybrid Sampling for Balanced Dataset
Feature Extraction using CNN-COA
Classification of different categories

Figure 1: Working Flow of Proposed Methodology

3.1. Dataset Description

We considered the widely available and widely used leak detection data sets in earlier work: the NSL-KDD data set. The dataset has normal data and four different types of attacks include Probe, U2R, R2L and DoS. There are 42 dimensional features are presented in each intrusion record and it is categorized into 3-dimensional symbol feature, a traffic type label and 38-dimensional digital feature. Table 1 shows the description of the data set.

Table 1: Dataset Description



Category	Train	Test
Normal	67343	9711
DoS	11656	7458
Probe	45927	2421
U2R	52	200
R2L	995	2754
Total	125973	22544

3.2. Data Pre-processing

Data pre-processing includes three steps as follows:

3.2.1. Numerical processing

In the dataset, the symbol feature data are mapped to the digital feature vector by using one hot encoding method, because it has digital matrix as input. Three features includes service, flag and protocol_type are mainly focused by this processing technique. They have 11, 3 and 70 symbol attributes and all are separately hot coded. For example, the binary vectors (1,0,0), (0,1,0) and (0,0,1) are encoded from the NSL-KDD's attributes of protocol_type, TCP, UDP and ICMP.

3.2.2. Normalization processing

The data value of continuous features is completely different in this dataset. For instance, the feature of num_root value range is [0,7468] and the feature of num_shells value range is only [0,5]. To eliminate mathematical processing and measurements, a standard processing method is adopted to graph the range of values [0,1] for each character in a unified and uniform way. Default calculation formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

Among them, feature's maximum value is described as and minimum value is depicted as .

3.2.3. Conversion of matrix from normalized data

In order to get the format of grayscale images, each read data's network record is dimensionally transformed. RGB D1 is the default format in this process. The network data is reshaped into matrix for the convenient of CNN, for instance 11*11 matrix is reshaped from the 121 feature vectors.

3.3. Hybrid Sampling to Construct the Balanced Dataset

A normal data traffic and limited abnormal traffic are combined to form a network traffic data, which is the most common issues of unbalanced data classification. In this case, the accuracy of the prediction of some major classes is improved, when the overall error decreases and the accuracy of the prediction of the minority categories is generally very low. Random Over Sampling (ROS) and Random Under Sampling (RUS) are the two most common sampling techniques, where the imbalanced ratio (IR) of various traffic data is high in the NIDS. Samples with important information can only be lost using the RUS method. However, the use of the ROS method alone does not allow detailed information and classification, which can lead to inconsistent taxonomic performance. OSS [23] is a model-based system used by KNN as a result of the use of domain links. Tomek Links is used as a subsampling method and eliminates noise and marginal majority class examples. Boundary examples can be considered "unsafe" because small amounts of noise can cause the decision to fall on the wrong side of the border.

3.4. Spatial Features Extraction by 1-D CNN

There are six layers in the standard CNN models that includes convolution, pooling, activation, a fully connected, softmax and an output layers. The input data is passed through these layers so that eventually the original data is assigned to any class. In particular, the entry of a 1-D CNN is a 1 × N or N × 1 array. As shown in Figure 2 [24], an N × 1 array goes through a series of convolution and pooling layers, and finally, finds the class of the array in the output layer.

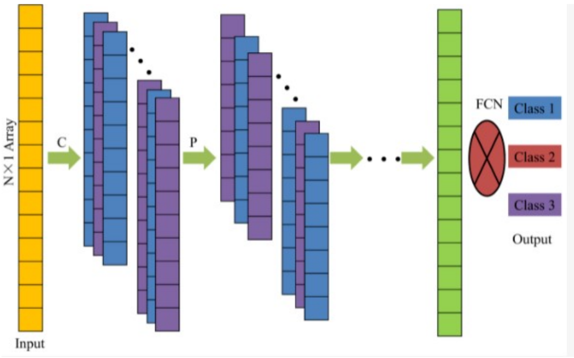


Figure 2: Architecture of a 1-D CNN.

In figure 2, C means convolution layer, P means pooling layer and FC means fully connected layer. Each Confusion kernel component multiplies the Confusion layer input data by the subsection item and summarizes the products to get an item on the feature map. Each time, subsection 1 moves down and the process repeats until all components of the input data are included; eventually the compression function will create a new matrix (i.e. feature map). The clustering function is a paradigm that greatly improves the computational speed of CNN and effectively prevents overcorrection. Usually there are 2 different grouping modes, namely maximum grouping and average grouping. The maximum group used in this study was better than the average group [25]. The implementation layers, the Softmax layer and the fully integrated layer are similar to the CNN 2-D standard, as described in Appendix [26].

The below are the steps required to extract the spatial features from the data:

The input of the C layer is the matrix of traffic data, where this layer is the CNN architecture's core. The same convolution kernel is shared by all neurons as the local perception concept is introduced and the number of weights is determined by using the number of convolution kernels. The computational efficiency is increased, when the number of weights is reduced greatly. The function for convolution is given in the Eq. (2):

$$h_j = f(h_{j-1} \otimes w_j + b_j) \qquad (2)$$

Where is the feature map of layer , . is the bias of layer , and ⊗ is the seizure function and is the activation function. This document uses the rectified linear unit (ReLU) drive function. The grouping layer integrates small neighborhood feature points to get new features and works after the confusion layer. This can reduce the size of the feature map and avoid overfitting. Let's write like this :

$$h_j = pool(h_{j-1}) \qquad (3)$$

After several grouped scrolls and layers, should be redrawn into a vector. So the output can be obtained through a FC layer. Therefore, we can derive spatial characteristics from the network traffic data extracted from CNN. However, it does not work well for learning serial sequence communication information. Therefore, the COA should improve the accuracy of the IDS in CNN's network, which is described below.

3.4.1. Chimp optimization algorithm

The chimpanzee colony is a fission civilization. The size of the group or colony changes over time and the members move through the environment. This form of community group structure is a dynamic characteristic for chimpanzees living in fusion colonies [27]. In light of these issues, feedback is presented to the Autonomous Committee. Each simulation team tries to find the search space individually with their own approach with this technique. Chimpanzees are not the same for all groups in terms of ability and intelligence, but they all do their duty as members of the colony.

In the chimpanzee colony there are four types of chimpanzees called drivers, obstacles, pursuers, and attackers. We all have unique abilities, but this diversity is important to our success. The drivers are still not trying to catch the beast. Barriers have been placed on the tree to build a dam across the dam path. Predators change quickly to catch prey. Finally, the attackers predict that the prey will enter the lower canopy or the path that will lead to it (prey). The attackers are supposed to need more cognitive efforts to predict the next move of the prey. This vital work (attack) is strongly related to age, intelligence and physical ability. Furthermore, during the same hunt, chimpanzees may change roles throughout the process or perform similar tasks [28].

Chimpanzees have been shown to hunt for the socially beneficial meat trade, such as supporting alliances, sex, or bathing [29]. So ingenuity will have an unexpected effect on hunting by opening up a new privileged sector. As far as we know, this "social incentive" was only for chimpanzees. Therefore, chimpanzees and other social predators make a fundamental difference based on their cognitive ability. In the final stage of the hunting process, chimpanzees chaotically cause all chimpanzees to abandon their unique tasks and become frantic and prey. The chimpanzee hunting method is often divided into two main stages: "inspection" which involves driving, blocking and chasing prey, and "exploitation". All these COA principles are expressed mathematically in the next section.

3.4.1.1. Mathematical idea and algorithm

This section presents mathematical prototypes of independent groups, driving, blocking, pursuing, and assaulting. Following that, the corresponding COA algorithm is described.

Driving and Chasing the Prey

The prey is hunted throughout the exploitation stages, as previously stated. Eqs. (4) and (5) are presented to statistically model driving and pursuing the prey.

$$d = |c.X_{prey}(t) - m.X_{chimp}(t)| \quad (4)$$

$$X_{chimp}(t+1) = X_{chimp}(t) - a.d \quad (5)$$

Where designates the sum of current iteration, and are the coefficient vectors, is represented as the vector of prey location and is represented as the position vector of a chimp a, m, and c vectors are considered by the Eq.s (6), (7) and (8), respectively.

$$a = 2.f.r_1 - f \qquad (6)$$

$$c = 2.r_2 \qquad (7)$$

$$m = Chaotic\_value \qquad (8)$$

which f through the iteration process is lowered nonlinearly from 2.5 to 0. The random vectors of r1 and r2 are inside [0,1] range.

Finally, m is a confusing vector based on a different confusion map that captures the effect of chimpanzee sexual preference on predatory activity. Because all particles have comparable behavior in local demographics and global research, people can be viewed as a single group with a shared search policy, subject to consistent population-based optimization. However, in theory, different independent groups with shared goals can be used simultaneously to obtain direct and random search results in each population-based optimization algorithm. Later groups of autonomous chimpanzees are mathematically designed with various update mechanisms. Any ongoing activity can be used to renew independent groups. These functions must be selected so that f is minimized with each iteration [30].

These four distinct groups use their unique ways to search for problem areas both locally and globally. Two alternative versions of COA with different version groups called COA1 and COA2 were selected from different techniques that tried to work better on benchmarking optimization problems.

Attacking Method (Exploitation Phase)

There are two ways to accurately model chimpanzee behavior: Chimpanzees can explore prey and move around (drive, block, and chase). In general, chimpanzees are carried out in the practice of hunting. From time to time, drivers, obstacles, and pursuers participate in the hunt. Unfortunately, there is no information on the best site in a concise search area (prey). The first attacker (driver), driver, obstacle, and pursuer is expected to be aware of the potential prey location to mathematically follow Chimp's behavior. Therefore, the four best solutions that are still available are in storage, forcing other chimps to upgrade their seats to make them better chimps. Equations (9), (10) and the relation are expressed (11).

$$d_{attacker} = |c_1 X_{attacker} - m_1 X|, d_{Barrier} = |c_2 X_{Barrier} - m_2 X|$$
$$d_{chaser} = |c_3 X_{chaser} - m_3 X|, d_{driver} = |c_4 X_{driver} - m_4 X|,$$
$$X_1 = X_{Attacker} - a_1(d_{attacker}), X_2 = X_{Barrier} - a_2(d_{Barrier}) \quad (9)$$

$$X_3 = X_{chaser} - a_3(d_{chaser}), X_4 = X_{driver} - a_4(d_{driver}) \quad (10)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \quad (11)$$

As can be seen, the ultimate location is randomly situated in a circle that is determined by the position of the assailant, obstacle, chaser and driver. In other words, four best groups guess the position of the prey and the chimps update their positions randomly in their proximity.

Prey Attacking (Utilization)

Chimpanzees attack the prayer and end the hunt in the final stages because the prey stops moving. The value of f must be lowered to properly model the attack process. Note that the range of variations is defined from f. In other words, in the interval of [-2f,2f] an is a random variable, whereas in the iterations, in f, the value decreases from 2,5 to 0. Where the random values of a chimpanzee lie within the range of [-1,1]

The COA allows chimpanzees to update their locations based on the status of the attacker, barrier, pursuer and driver chimpanzee and, based on those already issued, attack the prey. COAs may still be subject to local minimal conditions; however other operators should avoid this problem. Although the proposed methodology for motivation, prevention and monitoring somehow demonstrates the inspection process, the COA needs more operators to explore the pressure.

Searching for Pray (Exploration)

As mentioned above, chimpanzees are primarily explored by considering invader, obstacle, hunter, and chimpanzee status. You are different from looking for the beast and attacking the beast. The diversity of vectors with a random value greater or less than -1 is used to measure behavior, so investigating agents stay away from distractions and predictions. This protocol shows the scanning process and allows COA to scan globally.

The value of C is another element of the COA, which affects the study phase. As in the equation (7), c vector elements are random variables [0,2]. This component is used to increase (c> 1) or decrease (c <1) (8) the random weight in the distance estimate. This helps the COA improve its random behavior during the optimization process and reduces the risk of local cheating. c is required not only in initial rebuilds, but also in final reviews to generate random values and implement the analysis process. This factor is especially useful, especially in final iterations, to avoid local limitations. Vector C is also seen as the influence of barriers to access chimpanzee prey.

Social Incentive

The third step is to get social incentives (sex and care) that, as mentioned above, lead the Simps to release their hunting duties. So they are trying to mess up the meat by force. This chaotic behavior in the final stage helps to further reduce the two locally optimal traps and boring coordination difficulties for the animal.

We assume that 50 percent of the regular update location mechanism or fuzzy model can be picked and selected during simultaneous optimization of chimpanzee status. Same. (12) Reveal the mathematical mode.

$$X_{chip}(t+1) = \begin{cases} X_{prey}(t) - a.d & if\ \mu < 0.5 \\ Chatic_{value} & if\ \mu > 0.5 \end{cases} \quad (12)$$

Where □ is a random number in [0,1].

3.5. Hierarchical Network Model

In extracting the feature of the NIDS, not only the law of change over time should be measured, but the connection of the feature at the spatial level should also be considered. Therefore, this document uses 1D-CNN to extract features with COA, ultimately creating an in-depth HNM. The spatial and temporal aspects of the traffic data can be extracted simultaneously by creating a HNM that integrates 1D-CNN with COA networks. The output of the CNN's FC layer is a 1 * 128 aspect vector. The time step is set to 2 and 64 is set as input size, when the network's input layer has COA. By therefore, the input size for the COA corresponds to the CNN output size. The end of each COA operation is a combination of all previous functions and current functions. Finally, after the 1D-CNN output layer is added, a fully integrated layer is added, the previously extracted features are integrated, and finally sent to softmax for fully integrated layer output value classification.

4. RESULTS AND DISCUSSION

Simulation system environment includes Windows 10 operating system, i3-7100U CPU, 12G memory and Keras 2.2 framework. In this article, the learning rate for the network model is set to 0.001. In the regularization mode, the dropout weight loss rate is set to 0.5, and the retest is set to 100 times, each volume_ size is set to 128.

4.1. Evaluation Metrics

In this document, four parameters such as accuracy (AC), precision (P), recall (R), and the F1-measure (F1) are used to validate the model performance. The confusion matrix are normally formed by using the following four indicators:

1) True Positive (TP) – Correctly classified the data as Attack.

2) False Positive (FP) – Wrongly classified the normal data as attack.

3) True Negative (TN) – Accurately classified the normal data.

4) False negative (FN): Wrongly classified the attack data as normal.

We will use the following steps to evaluate the performance of our proposed solution:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Recall(True\ positive\ Rate) = \frac{TP}{TP+FN} \quad (15)$$

$$F1Score = \frac{2TP}{2TP+FP+FN} = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (16)$$

4.2. Performance Analysis of Proposed Method

There are two processes occurred in this study, one is training and other is testing. This study uses the KDDTest+ to train the model and the classification effect of each category is shown in Table 2 and Figure 3.

Table 2: Validation of Proposed CNN-COA on different categories of NSL-KDD dataset

Category	P (%)	R (%)	F1 (%)
Normal	88.17	**95.60**	91.47
DoS	**94.16**	86.47	**91.94**
Probe	65.81	69.62	67.28
U2R	61.94	62.47	63.42
R2L	62.32	59.04	61.15

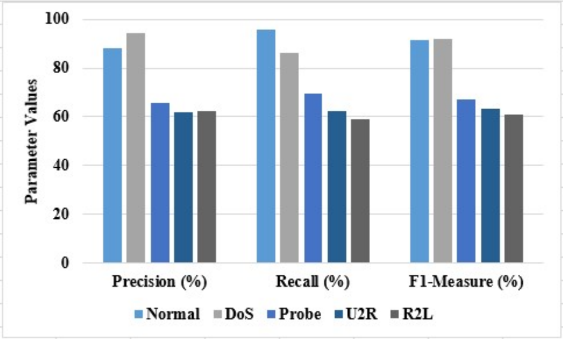


Figure 3: Graphical Representation of Proposed CNN-COA for different categories on NSL-KDD Dataset.

While in the normal category, the proposed method achieved 88.17% of precision, 95.60% of recall and 91.47% of F1-measure. While comparing with other categories on recall experiments, the proposed CNN-COA achieved high performance on normal category only. As like, the proposed method achieved high precision (i.e. 94.16%) and high F1-measure (i.e.91.94%) only on DoS category. In other categories like Probe, U2R, R2L, the proposed method achieved nearly 61% to 69% of precision, recall and F1-measure, where CNN-COA achieved less recall value (i.e.59.04%) on R2L category only.

In NIDS, various ML/DL techniques are used for detection, therefore, RF [18, 21] and classic CNN [16, 17, 19] along with Bidirectional Long Short-Term Memory (B-LSTM) [17] networks are considered as existing techniques for comparison performance, which is shown in Table 3 and Figure 4.

Table 3: Performance Analysis of various existing classifiers with Proposed CNN-COA on dataset

Classifier	Acc (%)	P (%)	R (%)	F1 (%)
RF	79.71	80.33	80.01	80.45
CNN	81.75	82.71	82.43	82.57
BLSTM	82.43	82.95	79.65	83.34
Proposed CNN-COA	87.19	88.28	89.49	91.19

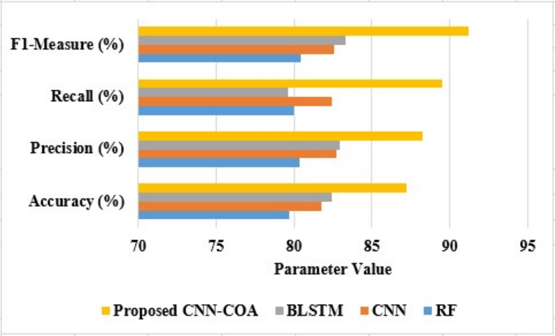


Figure 4: Graphical Representation of Proposed CNN-COA with existing classifiers

In the accuracy experiments, the RF, CNN, BLSTM and CNN-COA achieved 79.71%, 81.75%, 82.43% and 87.19%, where the same techniques achieved 80.33%, 82.71%, 82.95% and 88.28% of precision. In the F1-measure experiments, the RF achieved 80.45%, CNN achieved 82.57%, BLSTM achieved 83.34% and CNN-COA achieved 91.19%, where these existing techniques and proposed CNN-COA achieved 80.01%, 82.43%, 79.65% and 89.49% of recall. In NSL-KDD dataset, the performance of all classification techniques are not really good, the major reason is that number of samples is very less in the attacks categories during training sets. These classifiers give less importance for these attacks, while training. However, the proposed CNN-COA improves the detection rate than existing techniques, which shows that the problem of imbalance data is solved and minimized the low minority detection rate.

In Table 4, the simulation results are compared to the original data set, greatly reducing the training time for the classification models when all the experiments generated by the hybrid model run 50 epochs.

Table 4: Training Time on existing classifier with proposed CNN-COA

Data	Classifier	Training Time (s)	AC (%)	P (%)	R (%)
Original Data	RF	12900.93	79.73	78.26	79.15
	CNN	2929.62	77.01	81.20	77.61
	BLSTM	6949.43	76.37	79.64	76.83

	Proposed CNN-COA	4260.99	81.42	81.84	79.60
Hybrid Sampling Dataset	RF	1017.65	80.57	81.31	80.53
	CNN	219.33	81.95	84.14	82.56
	BLSTM	571.07	75.08	75.63	76.49
	Proposed CNN-COA	343.67	83.75	85.72	86.70

Among the all DL/ML models, the proposed CNN-COA achieved better detection results. When the hybrid was sampled from the dataset, the training time for RF, CNN, BLSTM, and CNN-COA decreased to 11883.25s, 2710.29s, 6378.36s, and 3917.32s, respectively. Even though the CNN-COA's training time is high, it is in the acceptable range that needs to be reduced in the future works.

5. CONCLUSION

This research study developed the hybrid sampling and HNM for NIDS, which is deeply studied and discussed. To construct the balanced dataset, OSS and SMOTE are combined to form the hybrid sampling. This process greatly solves the common problem of inadequate training and minimized the model's training time from the uneven samples. In addition, the network data pre-processing system is installed for complex and multidimensional Internet threats, which is compatible with the specific depth network model. Categorize the input data using a HNM generated by 1D-CNN using COA. Automatically extract sample features through repeated multi-level study using specific in-depth study features. The validation of the CNN-COA is carried out by using NSL-KDD dataset and tested with existing classifiers such as RF, CNN and BLSTM. The proposed method gives excellent results in accuracy, precision and extraction rate. The time performance is acceptable for practical implementation, because the network classification is trained only once and therefore, the proposed model can be used as off-line anomaly detection tool. The proposed model can be tested with various intrusion datasets by modifying the 1D-CNN model with hybrid DL techniques.

REFERENCES:

[1] K. Grahn, M. Westerlund, G. Pulkkis, "Analytics for network security: A survey and taxonomy, in: Information fusion for cyber-security analytics", Vol. 691, *Springer*, 2017, pp. 175–193.

[2] Q. Chen, R. A. Bridges, Automated behavioral analysis of malware: A case study of wannacry ransomware, in: 2017 16th IEEE *International Conference on Machine Learning and Applications (ICMLA*), IEEE, Cancun, Mexico, 2017, pp. 454–460.

[3] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, K. J. Kim, A survey of deep learning-based network anomaly detection, *Cluster Computing* 22 (1) (2017) 1–13.

[4] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407 (2019) 1–50.

[5] M. Panda, M. R. Patra, Network intrusion detection using naive bayes, *International journal of computer science and network security* 7 (12) (2007) 258–263.

[6] N. B. Amor, S. Benferhat, Z. Elouedi, Naive bayes vs decision trees in intrusion detection systems, in: *Proceedings of the 2004 ACM symposium on Applied computing, SAC* '04, ACM, Nicosia, Cyprus, 2004, pp. 420–424.

[7] J. Zhang, M. Zulkernine, A. Haque, Random-forests-based network intrusion detection systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (5) (2008) 649–659.

[8] M. A. Ambusaidi, X. He, P. Nanda, Z. Tan, Building an intrusion detection system using a filter-based feature selection algorithm, *IEEE transactions on computers* 65 (10) (2016) 2986–2998.

[9] Z. Li, Z. Qin, K. Huang, X. Yang, S. Ye, Intrusion detection using convolutional neural networks for representation learning, in: *International Conference on Neural Information Processing, Springer,* Guangzhou, China, 2017, pp. 858–866.

[10] A. Chawla, B. Lee, S. Fallon, P. Jacob, Host based intrusion detection system with combined cnn/rnn model, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, *Springer*, Dublin, Ireland, 2018, pp. 149–158

[11] Kumar, K.P.M.; Saravanan, M.; Thenmozhi, M.; Vijayakumar, K. Intrusion detection system based on GA-fuzzy classifier for

detecting malicious attacks. *Concurr. Comput. Pr. Exp.* 2021, 33, 5242.

[12] Y. Yang, K. Zheng, C. Wu, Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, *Sensors* 19 (11) (2019) 2528.

[13] Z. Liu, J. Wang, G. Liu, and L. Zhang, ''Discriminative low-rank preserving projection for dimensionality reduction,'' *Appl. Soft Comput.,* vol. 85, Dec. 2019, Art. no. 105768.

[14] Z. Liu, Z. Lai, and W. Ou, ''Structured optimal graph based sparse feature extraction for semi-supervised learning,'' *Signal Process.,* vol. 170, May 2020, Art. no. 107456.

[15] X. Zhang, J. Chen, and Y. Zhou, ''A multiple-layer representation learning model for network-based attack detection,'' *IEEE Access,* vol. 7, pp. 91992–92008, 2019.

[16] De Teyou, Gael Kamdem, and Junior Ziazet. ”Convolutional Neural Network for Intrusion Detection System In Cyber Physical Systems.” *arXiv preprin arXiv*:1905.03168 (2019).

[17] Sun P, Liu P, Liu C, Lu X, Hao R, Chen J. DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. *Security and Communication Networks* 2020; 2020:11 pages.

[18] Souza CA, Westphall CB, Machado RB, Sobral JMB, Vieira GS. Hybrid approach to intrusion detection in fog-based IoT environments. *Computer Networks* 2020; Volume 180 (ISSN 1389-1286).

[19] Jallad Al, K Aljnidi, M Desouki MS. Anomaly detection optimization using big data and deep learning to reduce false-positive. *J Big Data* 2020; 7:68

[20] Albahar MA, Binsawad M, Almalki J, El-etriby S, Karali S. Improving Intrusion Detection System using Artificial Neural Network. *International Journal of Advanced Computer Science and Applications* 2020; 11(6).

[21] Carneiro J.,Oliveira N., Sousa N., Maia E., Praça I., “Machine Learning for Network-based Intrusion Detection Systems: an Analysis of the CIDDS-001 Dataset”, *arXiv:*2107.02753, 2021

[22] Karatas G, Demir O, Sahingoz OK. Increasing the performance of machine learn- ing-based IDSs on an imbalanced and up-to-date dataset. *IEEE Access* 2020; 8:32150–62.

[23] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, ''A study of the behavior of several methods for balancing machine learning training data,'' SIGKDD Explor. *Newsl.*, vol. 6, no. 1, p. 20, Jun. 2004.

[24] Teng, S., Chen, G., Liu, Z., Cheng, L. and Sun, X., 2021. Multi-Sensor and Decision-Level Fusion-Based Structural Damage Detection Using a One-Dimensional Convolutional Neural Network. *Sensors,* 21(12), p.3950.

[25] Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In *Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece*, 15–18 September 2010.

[26] Teng, S.; Chen, G.; Gong, P.; Liu, G.; Cui, F. Structural damage detection using convolutional neural networks combining strain energy and dynamic response. *Meccanica* 2019, 55, 945–959.

[27] Khishe, M. and Mosavi, M.R., 2020. Classification of underwater acoustical dataset using neural network trained by chimp optimization algorithm. *Applied Acoustics,* 157, p.107005.

[28] Dhiman, G. (2021). SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications. *Knowledge-Based Systems*, *222*, 106926.

[29] Kaur, M., Kaur, R., Singh, N., & Dhiman, G. (2021). Schoa: a newly fusion of sine and cosine with chimp optimization algorithm for hls of datapaths in digital filters and engineering applications. *Engineering with Computers*, 1-29.

[30] Houssein, E. H., Emam, M. M., & Ali, A. A. (2021). An efficient multilevel thresholding segmentation method for thermography breast cancer imaging based on improved chimp optimization algorithm. *Expert Systems with Applications*, *185*, 115651.