An Anomaly-based Network Intrusion Detection System Using Ensemble Feature Integration with Deep Learning Techniques

Srinivas Akkepalli[1,*,†], Sagar Kadapa[1,†]

[1]Osmania University, Hyderabad, Telangana, India,500007

Abstract

Present-day Internet consist of around half a million distinct networks. It might be challenging to categorize assaults in any network connection, since different attacks can have different connections and range in quantity from a few to hundreds of network connections. DS-based ML (Machine Learning) has been developed as a solution to this issue, monitoring and analyzing data packets to identify abnormal behaviors and novel assaults. The well-known NSLKDD datasets were utilized for this anomaly-based intrusion detection system. It comprises a significant number of computational time and features is more. The curse of dimensionality and data imbalance is the cause of the degradation in model accuracy that occurs with increased processing time, thus addressing these problems: *(i)* Using a feature selection method to include the features into the model and decrease their dimensionality which yields better results and requires less processing time than utilizing all the features,

Keywords

Feature selection methods, Deep learning techniques, NSL-KDD

1. Introduction

People's use of the Internet in daily life has significantly been increased. Secure communication is still an issue for Internet-based transactions, communication, and IOT applications. Network intrusion detection is a crucial part of network security. However, hackers constantly developing new methods to breach networks and steal data mean that despite several algorithms' best efforts, it is still difficult to identify new invaders. At present, the widely used detection method trains the intrusion samples using conventional ML techniques to produce the intrusion detection model. However, these algorithms have the disadvantage of low detection rates. A more advanced technique called Deep Learning (DL) automatically identifies characteristics from samples and effectively classifies invaders.

ACI'23: Workshop on Advances in Computational Intelligence at ICAIDS 2023, December 29-30, 2023, Hyderabad, India
*Corresponding author.
†These authors contributed equally.
sress2020@gmail.com (S. Akkepalli); sagar.k@sreyas.ac.in (S. Kadapa)
0009-0004-4542-8542 (S. Akkepalli)

2. Literature survey

In 2020, Meng Wang et al. [1], proposed a dynamical MLP-based detection method that combines a feedback mechanism and sequential feature selection to prevent DDoS attacks. Multi-layer perceptron (MLP) to illustrate and address the problems in IDS. In this paper wrapper feature selection is named SBS model to select the optimal features. MLP algorithm can not ensure finding the global optimal features, but a sub-optimal solution is also acceptable. This approach employed MLP and sequential feature selection to select the optimal features for the training phase. Also, generated a feedback system to reconstruct the detector when it experienced substantial dynamic detection failures. Finally, verified this technique's effectiveness and contrasted it with several relevant works. The outcomes demonstrated that this technology could produce equivalent detection performance and improve the detector's performance when necessary. However, the main drawbacks of this approach are it cannot guarantee finding the global optimal features thereby producing only sub-optimal results and the feedback mechanism may produce false-positive or false-negative results. In 2021, S. Krishnaveni et al. [2] used univariate ensemble feature selection technique. This approach is used for the selection of valuable reduced feature sets from given intrusion datasets. To improve accuracy ensemble method would replace it with a deep neural network model in the selection process. In 2021, Mahdi Soltani et al. [3] proposed an innovative approach to deep learning-based intrusion detection that may be used to adjust deep classification models that are vulnerable to zero-day attacks yet have low attack-wise accuracy.Machine learning methods [4, 5] are used to identify and predict the network attacks. In 2022, Zihan Wu and Hong Zhang [6] developed RTIDS, a three-module system with an inventive hierarchy self-attention design that is modelled after stacked encoders as well as decoders for feature extraction and contextual relationship learning. Self-attention mechanism is used to learn various feature representation weights. But incapable of recognizing multi-class assaults. Encryption over encryption techniques are proposed to secure the public networks [7]. Public surveillance systems [8] are common applications to prevent from intruders.

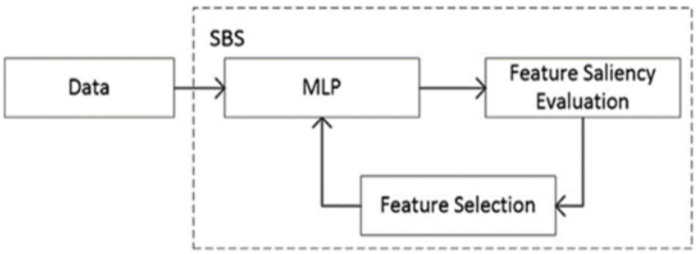

Figure 1: Proposed design of detection model [1]

2.1. State-of-the-art methods

The proposed method for intrusion detection involves 3 phases:

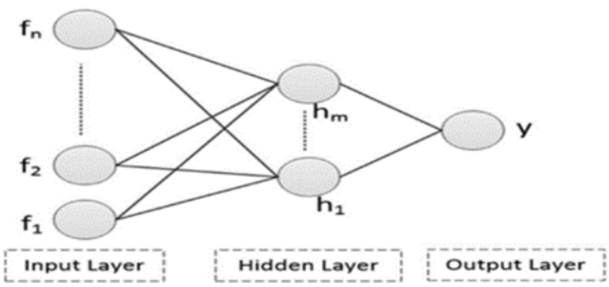

Figure 2: Neural network detection model [1]

2.1.1. Knowledge base

The training and feedback dataset, designated as Dt and Df, are two labelled datasets kept in the knowledge base. The samples applied to train the detection model makeup Dataset Dt, while the newly categorized and labelled samples from the detector's detection process are contained in Dataset Df. 2.Detection model: The MLP model was employed as a classifier in this work, and the best features were chosen using a wrapper feature selection technique called SBS.

2.1.2. Detection model

The MLP model was employed as a classifier in this work, and the best features were chosen using a wrapper feature selection technique called SBS.

Algorithm 1. SBS-MLP Algorithm:

Require: $F_0, M, V^{validation}, D_t^{test}$

Ensure: F^*, M, P_{cm}

$F_0 = \{f_1, f_2, \ldots\ldots\ldots f_n\}, F^| = \emptyset, F_1 = F_0$

Train M on D_t^{train} and $D_t^{validation}$ with the features in F_1 as inputs

Test the trained M on, D_t^{test} to get the feature saliency $S_{(1,0)}$ = 1 - Accuracy

$C_{F1} = S_{(1,0)}$

for $i = 1$ to $n - 1$ do

for each $f \in F_i$ do

$H = F_i - f$

Train M on D_t^{train} and $D_t^{validation}$ with the features in H as

Test the trained Mon D_t test to get the feature saliency

$S_{(I,f)}$ = 1-accuracy

end for

$f^* = \text{argmin}_f S_{(I,f)}$

$F_{i+1} = F_i - f^*$

$C_{Fi+1} = \min S_{(i.f)}$

End for

$F^* = \text{argmin}_{Fi} |F_i|$ subject to $\max(C_{Fi}) - C_{Fi} <= \varepsilon$

Train M on D_t^{train} and $D_t^{validation}$ with the features in F^* as inputs

Test the trained M on D^{test} to" F^*, M, and P_{cm}

Return F^*, M, and P_{cm}

2.1.3. Feedback mechanism

The feedback mechanism is in charge of identifying significant detection errors based on newly labeled samples that are entered into Df. It is only carried out if there are sufficient attack samples, which are indicated by the number (or proportion) of newly labeled attack samples in Df (represented as Na) over a predetermined value (signified as N0).The mechanism's basic hypothesis states that: if we retrain the detection model using the newly labeled samples during this time, after a certain amount of false-negative/positive errors in present detection have been accumulated, the retrained model's detection accuracy on test data will show a distinguishable decrease.

Algorithm 2 Error perceiving algorithm:

The crucial decision-making threshold, denoted as θ, is calculated using the Bienaymé-Chebyshev inequality, which may be described as follows:

while Na ≥ N0 do

Read data from Df

Train M using the features in F* as as inputs for Df trained Df validation; test the trained M using Dt-test to obtain the confusion matrix Qcm.

Calculate detection accuracy aPandaQas per PcmandQcm

δ = aP − aQ

if $\delta > \theta$

then

Update Dt and use the updated Dt to carry out the SBS-MLP operation.

Update θ

end if,end while.

2.2. Comparison findings on NSL-KDD:

Table 1

MLPMetrics [1]

Work	Detection Model	FS	Accuracy (%)	DR (%)
In2020, Meng Wang et al. [1]	MLP	SBS	97.66	94.88
SFS-MLP	MLP	SFS	97.61	94.71

Drawback:

1. The SBS-MLP method cannot guarantee the discovery of the global optimum features while a suboptimal solution is acceptable.
2. False-positive or false-negative responses might be produced by the feedback process.

An ensemble feature selection technique [2] was given based on univariate learning from given intrusion datasets to select valuable reduced feature sets. Five univariate filter techniques were utilized to provide features for intrusion detection due to their simplicity and speed, and an ensemble classifier was able to successfully fuse the separate classifiers to create a robust classifier that could be able to identify network assaults.

1. Proposed UEFFS method: Univariate Ensemble Filter Feature Selection

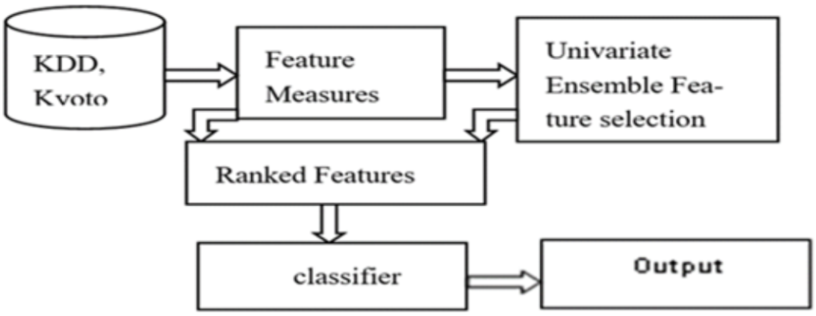

Figure 3: Proposed detection model [2]

Proposed Algorithm Steps:

1. The suggested method involves computing features from the subsequent three incursion datasets: Kyoto, NSL-KDD, and Honey Pot.
2. An incursion dataset's features were ranked using the five-univariate filter-based measures. The first filter measure Information Gain was used to scale all of the calculated rankings.
3. The approach modifies the scale's values (range between 0 to 1). The features that have the highest weights or ranks are ranked 1st. Each feature's priority value was calculated using its unique measure score and weight. The suggested technique computes a mean to determine the rankings and significance of each attribute.
4. The subset for optimal features is chosen from the rated top α percent feature sequences. On the basis of threshold (α) value, the top-ranked features from 80% of the datasets have been retained, while 20% of the lower-ranked features. were removed.

2.3. Comparison of results:

Drawback:

1. Multivariate measures to irrelevant feature selection.
2. Future work to the base models for the ensemble method would be to replace it with a deep neural network model in the selection process.

An Adaptable Deep Learning-Based Intrusion Detection System [**?**] was introduced to Zero-Day Attacks. The proposed novelty-based framework for deep learning-based intrusion detection to adapt the deep classification models with zero-day attacks in the real world's circumstances. This framework consists of four phases. first phase distinguishes the new attacks from the older ones. The second phase, a clustering module that links to a particular layer of the "deep classifier" model implements this phase by creating clusters out of the observed unidentified traffic.The third phase Supervised Labeling expert supervisor categorizes unknown traffic into four groups in the third phase: known harmful, new assault, undetected benign, and temporary anomalous traffic.The fourth phase updating the Model and collect results, The expert supervisor categorizes unknown traffic into four groups in the third phase: known harmful, new assault, undetected benign, and temporary anomalous traffic.

Table 2
Metrics for KDD. Kyoto, Honeypot Dataset [2]

Dataset	Feature selection Model	Number of features	Accuracy (%)	DR (%)	FAR	Pair wise T-test
NSL_KDD dataset	UEFFS	10	96.062	0.979	0.076	0.0224
	SFS and SVM	9	85.882	0.845	0.270	0.0323
Kyoto (2006)dataset	UEFFS	6	99.935	0.999	0.002	0.0118
	SFS and SVM	7	98.712	0.966	0.027	0.0124
Honeypot Dataset (2018)	UEFFS	7	98.892	0.965	0.028	0.0123
	SFS and SVM	10	96.854	0.978	0.084	0.0221

2.4. Comparison of results:

Table 3
Overall classification result of model on CIC-IDS2017 Dataset

Labels	D.69OC++ %	DOC(%)	Open Max (%)
Port scan	81.69	78.86	95.6
Botnet	66.15	46.38	55.41
DDos	51.47	30.94	28.0

Drawback: Open set recognition, Supervised labeling, Clustering/post-training, and updating take more time complexity, and lack of accuracy.

A Robust Transformer-Based Approach [6] for Institution Detection Systems refers a positional embedding technique to associate sequential information between features, then a variant stacked encoder-decoder neural network RTIDS consists of three modules and features and innovative hierarchy self-attention design Transformer model Specifically, we apply input and positional embedding to convert input network traffic into fixed-dimension vectors as input representations. Then stacked encoders and decoders are used for feature extraction and learning the contextual relations between inputs. Since the input features have different impacts on the classification result, we use the self-attention system to learn the different weights of the feature "representations.

RTIDS Algorithm [4]

Input: Training set S = (x_i, y_i) , i = 1, 2, ... N, x_i is the network traffic sample, y_i is the corresponding label

Output: Classification probabilities of the predicted class.

1: for i ← 0 until num Of Epochs do

2: for Sample s: Batch do

get its vectorized representation sr

put sr into encoder and decoder stacks for feature extraction and selection"

use the transformer Model.MultiHead Attention function to compute the attention scores of features

use transformer Model. SoftMax function to obtain classification probabilities

3. end for

use stochastic gradient descent (SGD) algorithm to minimize the loss function

4. end for

Table 4
Overall classification result of model on NSL KDD Dataset[4]

Algorithm	Accuracy (%)	Precision (%)	Recall (&)	F1-score	Time in sec
RTIDS	98.35	98.98	98.83	99.17	195.6

Drawback: RNN-based methods have certain limitations in step-by-step processing. Their feature extraction at any given point in time only relies on the hidden state of previously observed information, possibly resulting in missing features in the context vector.

3. Proposed model

To handle above mentioned issues, we proposed a Deep transudative Federated transfer learning model.

Self-attention: self-attention relates the words to each other and sequence = m rows and dmodel = d_k = m. The input Attribute values in the form of matrices Q.K.V.

$$\mathit{Attention}(\mathrm{Q.K.V}) = \mathrm{softmax}\left(\mathrm{QKT^T}/\sqrt{\mathrm{d_k}}\right)\mathrm{V} \tag{1}$$

$$\mathrm{headi} = \mathrm{Attention}\left(\mathrm{QW}^{q} \cdot \mathrm{KW}^{\mathrm{k}} \cdot \mathrm{VW}^{v}\right) \tag{2}$$

$$\mathrm{Multi\ headi} = concat(head1, head2.......headn)Wo \tag{3}$$

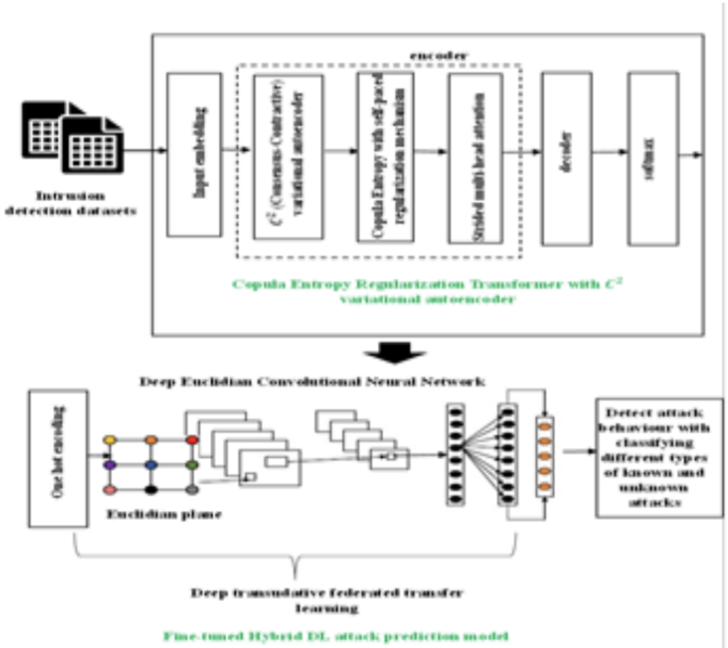

Figure 4: Proposed detection model

3.1. Experimental setup

The setup was created and carried out with Python programming language, and all suggested methods make use of the Kearas with Tensor flow backend framework. **Experimental Environment** Operating System Windows 10 pro 64-bit , Memory 64 GB CPU Intel(R) UHD Graphics 620, Anaconda 4.9.2, python 3.7.0, keras 2.4.2, Tensor flow 2.2.0

3.2. The evaluation metrics

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Specificity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \tag{7}$$

4. Experimental results and analysis

Overall Comparison of Results

Table 5
Overall classification result of model on NSLKDD Dataset

Work	Detection Model	FS	Accuracy (%)	Precision (%)	Recall(%)	DR (%)	FAR (%)
In 2020, Meng Wang et al.1	MLP	SBS	97.66	NA	NA	94.88	0.62
In 2021, S.Krishnaveni et al.2	UEFFS	NA	96.062	NA	NA	97.9	7.6
In 2021, Mahdi Soltani et al.3	Open Max	NA	98.66	NA	NA	NA	NA
In 2022 Zihan Wu and Hong Zhang	RTIDS	NA	98.35 Per 195sec	98.98	98.83	97.8	NA
Proposed DCNN Model	DCNN	CNN	98.0 per 160sec	99.0	98.0	98.4	NA

5. Conclusion

We proposed a three phase intrusion detection model which is capable of recognizing multi-class assaults. For this, a Deep Transudative Federated Transfer learning model was referred. Our proposed CNN model achieved accuracy= 98%, precision = 99%, Recall= 98%, F1_score= 99% and is efficient to detect zero-day attacks.

References

[1] M. Wang, Y. Lu, J. Qin, A dynamic mlp-based ddos attack detection method using feature selection and feedback, Computers & Security 88 (2020) 101645.

[2] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, S: Efficient feature selection and classification through ensemble method for networkintrusion detection on cloud computing, Cluster Computing 24 (2021) 1761–1779.

[3] M. Soltani, B. Ousat, M. J. Siavoshani, A. H. Jahangir, Anadaptable deep learning-based intrusion detection system to zero-day attacks, Journal of Information Security and Applications 76 (2023).

[4] R. Shanker, P. Agrawal, A. Singh, M. W. Bhatt, Framework for identifying network attacks through packet inspection using machine learning, Nonlinear Engineering 12 (2023) 20220297.

[5] R. Shanker, V. Madaan, P. Agrawal, Fss-part: Feature grouping subset model for predicting network attacks, SN Computer Science 5 (2023) 94.

[6] S. P. Rm, P. K. R. Maddikunta, M. Parimala, S. Koppu, T. R. Gadekallu, C. L. Chowdhary, M. Alazab, An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture, Computer Communications 160 (2020) 139–149.

[7] V. Madaan, D. Sethi, P. Agrawal, L. Jain, R. Kaur, Public network security by bluffing the intruders through encryption over encryption using public key cryptography method, in: Advanced Informatics for Computing Research: First International Conference, ICAICR 2017, Jalandhar, India, March 17–18, 2017, Revised Selected Papers, Springer, 2017, pp. 249–257.
[8] N. Mohod, P. Agrawal, V. Madan, Human detection in surveillance video using deep learning approach, in: 2023 6th International Conference on Information Systems and Computer Networks (ISCON), 2023, pp. 1–6. doi:10.1109/ISCON57294.2023.10111951.
[9] A. Heidari, M. A. Jamali, Internet of Things intrusion detec-tion systems: A comprehensive review and future directions, Cluster Computing, 2022.
[10] I. A. Kandhro, S. M. Alanazi, F. Ali, A. Kehar, K. Fatima, M. Uddin, S. Karuppayah, Detection of Real-Time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures, IEEE Access 11 (2023) 9136–9148.
[11] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detec-tion system, Ieee Access 7 (2019) 41525–41550.
[12] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, A. Razaque, Deep recurrent neural network for IoT intrusion detection system, Simul. Model. Pract. Theory 101 (2020) 102031.
[13] F. Laghrissi, S. Douzi, K. Douzi, B. Hssina, Intrusion detection sys-tems using long short-term memory (LSTM), Journal of Big Data 8 (2021).
[14] Y. G. Yang, H. M. Fu, S. Gao, Y. H. Zhou, W. M. Shi, Intrusion detec-tion: A model based on the improved vision transformer, Transactions on Emerging Telecommunications Technologies 33 (2022).
[15] T. P. Nguyen, H. Nam, D. Kim, Transformer-Based attention net-work for In-Vehicle intrusion detection, IEEE Access (2023).
[16] V. Hnamte, J. Hussain, DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system, Telematics and Informatics Reports 10 (2023).
[17] K. R. Karthikeyan, A. Indra, Intrusion detection tools and techniques -A survey, International Journal of Computer Theory and Engineering 2 (2010) 901–906.
[18] R. Bhargavi, V. Vaidehi, P. T. V. Bhuvaneswari, P. Balamuralidhar, M. G. Chandra, Complex event processing for object tracking and intrusion detection in wireless sensor networks, in: 2010 11th International Conference on Control Automation Robotics & Vision, IEEE, 2010.