
Srinivas Akkepalli\(^{1,\ast,†}\), 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.

---

\(\ast\)Corresponding author.

\(†\)These authors contributed equally.

\(\odot\) sress2020@gmail.com (S. Akkepalli); sagar.k@sreyas.ac.in (S. Kadapa)

\(\odot\) 0009-0004-4542-8542 (S. Akkepalli)

\(\odot\) © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
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]](image)

2.1. State-of-the-art methods

The proposed method for intrusion detection involves 3 phases:
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. 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: $F_0, M, V_{\text{validation}}, D_{t_{\text{test}}}$
Ensure: $F^*, M, P_{cm}$

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

Train $M$ on $D_{t_{\text{train}}}^{\text{train}}$ and $D_{t_{\text{validation}}}^{\text{validation}}$ with the features in $F_1$ as inputs
Test the trained $M$ on, $D_{t_{\text{test}}}^{\text{test}}$ to get the feature saliency $S_{(1,0)} = 1 - \text{Accuracy}$

$C_{F_1} = S_{(1,0)}$
for $i = 1$ to $n-1$ do
for each $f \in F_i$ do
$H = F_i - f$
Train $M$ on $D_{t_{\text{train}}}^{\text{train}}$ and $D_{t_{\text{validation}}}^{\text{validation}}$ with the features in $H$ as inputs
Test the trained $M$ on $D_{t_{\text{test}}}^{\text{test}}$ to get the feature saliency $S_{(I,f)} = 1$-accuracy
end for
$f^* = \arg\min_{f} S_{(I,f)}$
$F_{i+1} = F_i - f^*$
$C_{F_{i+1}} = \min S_{(I,f)}$
end for

$F^* = \arg\min_{F_{i}} |F_{i}|$ subject to $\max(C_{F_i}) - C_{F_i} \leq \varepsilon$
Train $M$ on $D_{t_{\text{train}}}^{\text{train}}$ and $D_{t_{\text{validation}}}^{\text{validation}}$ with the features in $F^*$ as inputs
Test the trained $M$ on $D_{t_{\text{test}}}^{\text{test}}$ to get $F^*, M$, 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 $D_f$. 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 $D_f$ (represented as $N_a$) over a predetermined value (signified as $N_0$). 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 $\theta$, is calculated using the Bienaymé-Chebyshev inequality, which may be described as follows:

while $N_a \geq N_0$ do
Read data from $D_f$

Train M using the features in F* 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 PemandQcm

\[ \delta = a_P - a_Q \]

if \( \delta > \theta \)
then
Update Dt and use the updated Dt to carry out the SBS-MLP operation.
Update \( \theta \)
end if
end while.

2.2. Comparison findings on NSL-KDD:

<table>
<thead>
<tr>
<th>Work</th>
<th>Detection Model</th>
<th>FS</th>
<th>Accuracy (%)</th>
<th>DR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In 2020, Meng Wang et al. [1]</td>
<td>MLP</td>
<td>SBS</td>
<td>97.66</td>
<td>94.88</td>
</tr>
<tr>
<td>SFS-MLP</td>
<td>MLP</td>
<td>SFS</td>
<td>97.61</td>
<td>94.71</td>
</tr>
</tbody>
</table>

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
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 $\alpha$ percent feature sequences. On the basis of threshold ($\alpha$) 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]

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

2.4. Comparison of results:

Table 3

Overall classification result of model on CIC-IDS2017 Dataset

<table>
<thead>
<tr>
<th>Labels</th>
<th>D.69OC+DOC(%)</th>
<th>Open Max (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port scan</td>
<td>81.69</td>
<td>78.86</td>
</tr>
<tr>
<td>Botnet</td>
<td>66.15</td>
<td>46.38</td>
</tr>
<tr>
<td>DDoS</td>
<td>51.47</td>
<td>30.94</td>
</tr>
</tbody>
</table>

**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 $S = \{x_i, y_i\}, i = 1, 2, \ldots, N$, $x_i$ is the network traffic sample, $y_i$ is the corresponding label

Output: Classification probabilities of the predicted class.

1: for $i \leftarrow 0$ 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

4. end for

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Accuracy (%)</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F1-score</th>
<th>Time in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTIDS</td>
<td>98.35</td>
<td>98.98</td>
<td>98.83</td>
<td>99.17</td>
<td>195.6</td>
</tr>
</tbody>
</table>

**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 transductive Federated transfer learning model.

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

$$\text{Attention}(Q.K.V) = \text{softmax} \left( \frac{Q K^T \cdot V}{\sqrt{d_k}} \right)$$  \hspace{1cm} (1)

$$\text{head}_i = \text{Attention} \left( QW^g \cdot KW^k \cdot VW^v \right)$$ \hspace{1cm} (2)

$$\text{Multi head}_i = \text{concat}(\text{head}1, \text{head}2, \ldots, \text{head}n)W_0$$ \hspace{1cm} (3)
3.1. Experimental setup

The setup was created and carried out with Python programming language, and all suggested methods make use of the Keras 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

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)
\]

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \quad (5)
\]

\[
\text{Specificity} = \frac{TP}{TP + FN} \quad (6)
\]

\[
\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)
\]

4. Experimental results and analysis

Overall Comparison of Results
### Table 5
Overall classification result of model on NSLKDD Dataset

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

5. Conclusion

We proposed a three phase intrusion detection model which is capable of recognizing multi-class assaults. For this, a Deep Transductive 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


