International Journal of Application on Sciences, Technology and Engineering
(IJASTE)
Volume 1, Issue 3, 2023 ISSN:2987-2499

Enhancing Intrusion Detection System Performance With 1D-CNN And Bi-LSTM Combination

Amalia Nurain[1], Vian Satria M. Navalino[2]

[1] Cyber Defense Engineering, Indonesia Defense University, Bogor, Indonesia*
Email: amalia.nurain1297@gmail.com
[2] Cyber Defense Engineering, Indonesia Defense University, Bogor, Indonesia
Email: viansatria57@gmail.com

Submitted: 05-03-2023, Revised: 07-03-2023, Accepted: 08-03-2023

ABSTRAK

Karena lalu lintas jaringan yang terus meningkat, permintaan untuk sistem Network Intrusion Detection telah meningkat karena penggunaan teknologi cloud telah menyebar luas. Penelitian terbaru menunjukkan bahwa model pembelajaran mendalam efektif untuk berbagai aplikasi yang melibatkan pemrosesan bahasa alami pada bagian penting dari Deteksi Intrusi Jaringan seperti deteksi anomali. Arsitektur model pembelajaran mendalam memiliki dua lapisan: lapisan pencocokan pola dan lapisan yang terhubung sepenuhnya untuk melatih label serangan. Dalam penelitian ini, kami menggunakan model pembelajaran mendalam dengan arsitektur yang dimodifikasi yang mengintegrasikan pencocokan pola dengan jaringan saraf konvolusional (CNN) dan ingatan jangka pendek panjang dua arah (Bi-LSTM). Hasil penelitian menunjukkan bahwa CNN dan Bi-LSTM dapat mengklasifikasikan kategori serangan dalam dataset UNSW-NB15 dengan akurasi 82%.

Kata Kunci: *Intrusion Detection System, Deep Learning, Bi-LSTM, 1D-CNN*

ABSTRACT

Due to the ever-increasing network traffic, the demand for a system that contain Network Intrusion Detection systems have increased as cloud technology usage has become widespread. Recent research has shown that deep learning models are effective for a variety of applications involving natural language processing on a critical part of Network Intrusion Detection such as anomaly detection. The architecture of a deep learning model has two layers: a pattern-matching layer and a fully linked layer for training the label of attack. In this study, we use a deep learning model with a modified architecture that integrates pattern matching with convolutional neural networks (CNN) and bidirectional long short-term memories (Bi-LSTM). The results show that CNN and Bi-LSTM can classify attack categories in the UNSW-NB15 dataset with an accuracy of 82%.

Keywords: *Intrusion Detection System, Deep Learning, Bi-LSTM, 1D-CNN*

1. INTRODUCTION

Since the beginning of widespread Internet usage, cloud technologies have been disruptively adopted, and this has caused an exponential increase in the number of intrusion events. The expense of network security has risen since a single data center that belongs to an organization like Microsoft, Amazon, Google, etc. includes various on-demand servers, platforms, etc. to provide services to a variety of small, medium, or big organizations. In an attempt to protect data and avoid service interruption, firewalls are growing ever more common. These devices incorporate a range of strategies for Prevention and Incident Management. The intrusions include eavesdropping, probing assaults, and network infections. Network time series data-based prediction models are one of the techniques used by Network Intrusion Detection Systems. Most time-series data exhibit non-linear properties because different data points vary over time as a result of erratic oscillations. The use of deep learning techniques like Convolution Neural Networks (CNN) and Bidirectional Long-Short Term Memory (Bi-LSTM) for prediction models based on network time series data for Network Intrusion Detection Systems is currently very common.

In this research, the suggested model will be evaluated using the UNSW-NB15 datasets. The UNSW-NB15 dataset, published in 2015 by the University of New South Wales in Australia, exposed the vulnerabilities in the KDD98 and KDD99 data sets, specifically the exclusion of modern low-footprint assaults from these datasets. This study proposes a hierarchical model to perform on both of these interesting datasets by combining layers of 1D- CNN and Bi-LSTM. The Bi-LSTM layers, which are essentially a subcategory of RNNs, are used to learn the long-time-range temporal properties of the data and combine these to anticipate attacks. The spatial/high-level properties of a dataset are learned using CNN. The projections are prepared for Binary Classification, which entails predicting both the likelihood of an assault occurring and the specific category under which it will fall. For multi-category attack prediction under UNSW-NB15, 10 classes were employed, which include Normal, DoS, Exploits, Generic, Reconnaissance, Worms, Shellcode, Analysis, Backdoor, and Fuzzers.

2. LITERATURE REVIEW

Network intrusion detection issues have long been popular with deep learning techniques. Deep learning methods that have been proposed have progressively addressed the issue and the answers they offer using the KDD-99 cup data. Initially, methods for resolving the issue emphasized pattern recognition. Leveraging pattern recognition algorithms, researchers have used feature selection by applying machine learning and deep learning methodologies [2].

A. Machine learning

Traditional machine learning methods including Support Vector Machine (SVM), Random Forest, and Adaptive Boosting have been widely employed by researchers to develop Network Intrusion Detection classifiers, but these methods have always failed due to high False Positive Rate (FPR), overfitting, and lower accuracy on classes with a smaller percentage of available data than on classes with sufficient numbers of data. The reason is that traditional machine learning approaches concentrate on learning feature availability, feature importance, and dimensionality reduction techniques to find the most optimal correlation between data points that appear to have the most influence upon the final result, completely ignoring the significance of the correlation between the features and taking into account the time steps in order to predict the best possible result. As a result, deep learning methodologies were used to fill in the gaps left by the previous system.

B. Support Vector Machine

The supervised learning method includes the Support Vector Machine (SVM), which is taught using a variety of data types from diverse subjects. SVM generates one or more hyperplanes in a high-dimensional space [2]. A non-linear classifier employs multiple kernel functions to assess the margins between hyperplanes. SVM plays a major part in applications for pattern identification and image processing. Data are typically split into two sets for classification tasks: training datasets and testing datasets. Labels for that class will be referred to as "target variables" and attributes as "observed variables" or features.

C. Deep learning

The application of deep learning algorithms in network intrusion detection has been the subject of many studies. This paper focuses on three important publications, namely Convolutional Neural Networks, Bidirectional Long Short Term Memory, and one-dimensional convolutional neural network, which have all utilized deep learning algorithms. The study will go into extensively about these research and compare the results of the proposed model to those found in the aforementioned three papers.

Deep learning techniques have been the subject of several research looking at network intrusion detection. Three significant research that made use of convolutional neural networks, bidirectional long short term memories, and one-dimensional convolutional neural networks are included in this list. This essay will give a thorough review of these research' findings before contrasting them with those of the suggested model. By doing this, the article hopes to develop deep learning algorithms for network intrusion detection.

a. Using Convolutional Neural Network

A particular kind of neural network called a convolutional neural network (CNN) is made to evaluate visual input in a way that closely resembles how the human brain processes visual information. These networks are particularly good at locating spatial features, which are the lines dividing various objects in a picture. CNNs have been demonstrated to perform better in tasks requiring photo categorization and identification than other varieties of neural networks [3]. This is caused by the distinctive architecture of CNNs, which enables them to concentrate on important spatial elements while removing unimportant information.

CNNs have been extensively employed in a variety of image identification and classification applications because they can identify spatial characteristics. These networks are valuable in applications like self-driving vehicles, facial recognition, and medical image analysis because they can efficiently recognize patterns and accurately identify various objects in pictures [4]. Researchers and developers may design cutting-edge apps that can aid in resolving issues in a variety of sectors by harnessing the characteristics of CNNs.

b. Using Bidirectional Long Short Term Memory

A particular kind of neural network, called a Bidirectional Long Short-Term Memory (BiLSTM) network, can improve learning by processing input both forward and backward. As a result, while generating predictions, the model is able to take into consideration both past and future time steps. As a result, the model is better able to grasp time-dependent inputs and learn more efficiently at each timestep [5][6].

Also, the BiLSTM is beneficial in applications like speech recognition and natural language processing where the sequence of the input data is crucial. The BiLSTM captures context and long-term dependencies more efficiently than conventional LSTMs or other types of neural networks since the input is fed in both ways.

c. Using 1D-CNN

For supervised learning on time-series data, the one-dimensional convolutional neural network (1D-CNN) is a well-liked technique. This method trains the CNN on a time-series dataset by serializing TCP/IP packets across a predetermined time range to enable precise classification[7]. Due of this methodology's success in identifying temporal patterns in sequential data, it has been frequently employed.

Whenever serializing TCP/IP packets for the purpose of training a 1D-CNN, it is advised to employ a specified time period to ensure the accuracy of the categorization of time-series data. The effectiveness of this approach has been demonstrated in a number of applications, including sensor data analysis, voice recognition, and medical signal processing [8]. The 1D-CNN can discover and learn the temporal patterns in the data by employing this method, which enables it to produce precise predictions and classifications.

3. RESULTS AND DISCUSSIONS

In this study, a suggested model is presented that combines a 1-D CNN with several layers of Bi-LSTM. This section will give an overview of the layers of the neural network and cover the datasets, preprocessing methods, and layers of the model in depth. The objective is to offer a thorough grasp of the architecture and parts of the suggested model. By contrasting the model with other current models, the study seeks to show how well it may improve performance on a particular activity. The study's findings will shed light on the possible advantages of combining a Bi-LSTM and 1-D CNN in deep learning applications.

A. Proposed Method

Rectified Linear Unit (ReLU) has an appeal on an activation function used in deep learning. It has been successful in producing good results, but it has the drawback of zeroing out negative input values. To address this issue, a variant called LeakyReLU was introduced, which does not zero out negative input values. In convolutional neural networks (CNNs), the slope parameter for each layer or channel in each layer can be learned. However, the number of slope parameters to be learned is small compared to the number of weights and biases that need to be learned [9]. The contribution of this study is to modify the 1D-CNN model using the PRELU activation function for better accuracy than previous studies and simplify the layer for faster training time.

B. Dataset

In 2015, the University of New South Wales released its dataset called UNSW-NB15. The dataset of UNSW has been in use for a long time and provides The UNSW- NB15 dataset contains more attack types, features that were extracted, unique IP addresses that were used for the simulation, and data collecting. This data collection mixes real-world modern normal activity with recent synthetic network traffic assault activities. **Table 1** provides a list of the characteristics found in the UNSW-NB15 Datasets.

Table 1. Dataset of UNSW-NB15 Attack Categories

Category	Compute
Normal	93000
Analysis	2677
Backdoor	2329
DenialofService	16353
Exploits	44525
Fuzzers	24246
Generic	58871
Reconnaissance	13987
Shellcode	1511
Worm	174
Total	257673

C. Data pre-processing

In the previous research, several modifications were made to the UNSW-NB15 dataset. These modifications included the removal of certain columns, such as the "id" column which contained index data, and the "label" column which contained binary labels. The binary label data was not utilized in this study, as the multiclass available in the "attack_cat" column was utilized instead. Additionally, one-hot encoding was applied to the "proto", "state", and "service" columns. In our implementation of the SVM method, a slight deviation from the previous research was necessary. Specifically, it was necessary to convert the "attack_cat" column to a numerical value in order to facilitate its fitting to ThunderSVM. To achieve this, the Label Encoder was utilized to transform the "attack_cat" column.

D. Evaluation

In the previous research [10], a 10-epoch method was utilized. However, in order to facilitate a more efficient comparison within the scope of our investigation, it was necessary to deviate from this standard and reduce the epoch to 5. This decision was made due to the constraints imposed by utilizing a free resource such as Google Colaboratory, as following the original researcher's method would have required approximately three hours per run. As such, a modified standard was adopted for the purposes of our study.

E. Model Result

The following is the outcome of the IDS classification utilizing SVM with the ThunderSVM tool. It is worth noting that the results of this SVM approach are inferior in comparison to deep learning methods as shown in **Figure 1** below. The reason for using ThunderSVM was to take advantage of the GPU, as the k-split of 6 took approximately 1 hour and 30 minutes to complete using the GPU. It should be noted that the process would take significantly longer if the GPU was not utilized.

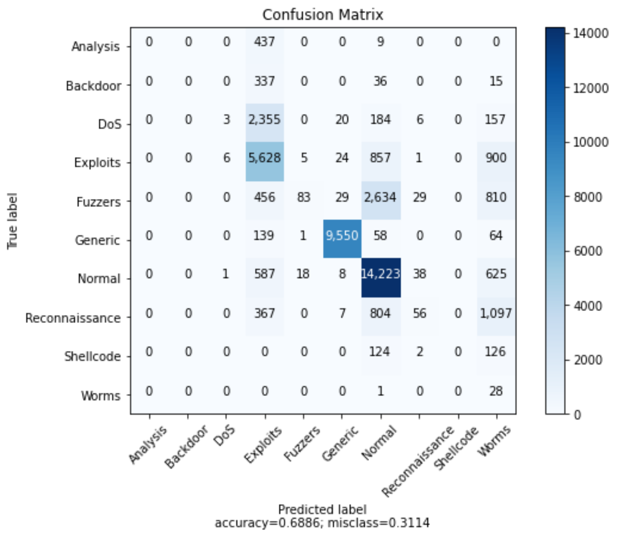


Figure 1. Confusion matrix of SVM

The results presented are derived from the Original Paper and incorporate a new standard that we have established as follows:

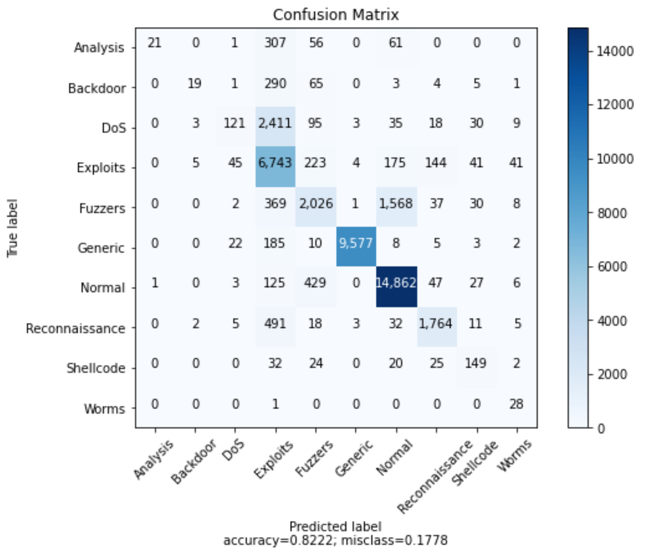
Table 2. UNSW-NB15 Dataset Attack Categories

Fold	Accuracy score
1	0.7957434918269455
2	0.8067573231500024

3	0.8154892190192334
4	0.8190709046454768
5	0.8186517638840377
6	0.8222144603562697

Table 3. Table of detection rate from original paper

Attack Category	Detection Rate (True Positive Rate)
Analysis	0.0470852
Backdoor	0.04896987
DoS	0.84440367
Exploits	0.90863765
Fuzzers	0.50136105
Generic	0.97604974
Normal	0.95883871
Reconnaissance	0.75675676
Shellcode	0.59126984
Worms	0.96551724



	Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Normal	Reconnaissance	Shellcode	Worms
Analysis	21	0	1	307	56	0	61	0	0	0
Backdoor	0	19	1	290	65	0	3	4	5	1
DoS	0	3	121	2,411	95	3	35	18	30	9
Exploits	0	5	45	6,743	223	4	175	144	41	41
Fuzzers	0	0	2	369	2,026	1	1,568	37	30	8
Generic	0	0	22	185	10	9,577	8	5	3	2
Normal	1	0	3	125	429	0	14,862	47	27	6
Reconnaissance	0	2	5	491	18	3	32	1,764	11	5
Shellcode	0	0	0	32	24	0	20	25	149	2
Worms	0	0	0	1	0	0	0	0	0	28

Figure 2. Confusion matrix of the original paper

The following is the result of the modification that has been made to the Original Paper:

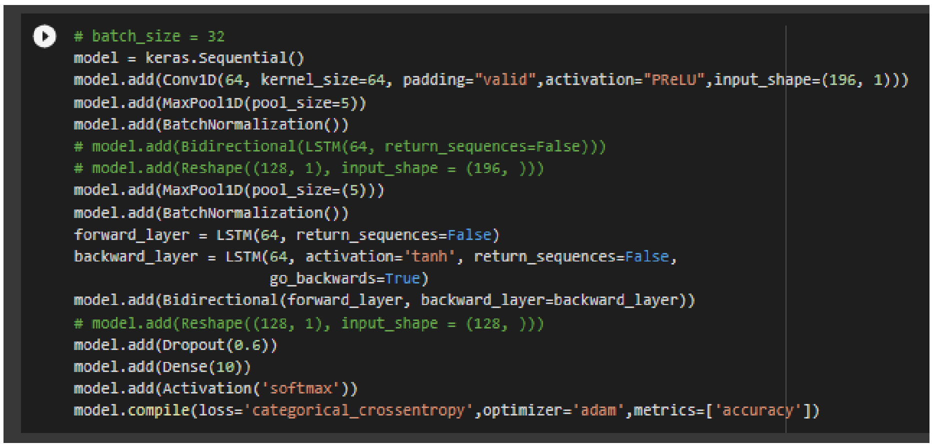
Table 4. Table of accuracy score modification

Fold	**Accuracy score**
1	0.8016811810180227
2	0.8096213849951102
3	0.811088343501141
4	0.8127605076260332
5	0.8127372220281756
6	0.81264408

Table 5. Table of detection rate modification

Attack Category	**Detection Rate (True Positive Rate)**
Analysis	0.03363229
Backdoor	0.02319588
DoS	0.2759633
Exploits	0.82320442
Fuzzers	0.53996536

Generic	0.97441908
Normal	0.93574194
Reconnaissance	0.71042471
Shellcode	0.37698413
Worms	0.55172414



```
# batch_size = 32
model = keras.Sequential()
model.add(Conv1D(64, kernel_size=64, padding="valid",activation="PReLU",input_shape=(196, 1)))
model.add(MaxPool1D(pool_size=5))
model.add(BatchNormalization())
# model.add(Bidirectional(LSTM(64, return_sequences=False)))
# model.add(Reshape((128, 1), input_shape = (196, )))
model.add(MaxPool1D(pool_size=(5)))
model.add(BatchNormalization())
forward_layer = LSTM(64, return_sequences=False)
backward_layer = LSTM(64, activation='tanh', return_sequences=False,
                      go_backwards=True)
model.add(Bidirectional(forward_layer, backward_layer=backward_layer))
# model.add(Reshape((128, 1), input_shape = (128, )))
model.add(Dropout(0.6))
model.add(Dense(10))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Figure 3. The modification of Bi-LSTM and 1D-CNN

This not only, fewer parameters in the model decrease the probability of overfitting, which is when the model performs well on training data but improperly on fresh, untested data. In machine learning applications, where generalization to new data is crucial, this is especially crucial. Our update offers a practical alternative that balances training time and model complexity without affecting the model's effectiveness.

Our modification's results show that it is feasible to accomplish original model performance with substantially fewer parameters with only required 80,522 parameters out of 214,670 parameters from the original paper, which has important practical relevance for real-world applications. Reducing the number of parameters in deep learning models can speed up inference time and use fewer resources, as mentioned in a study by Zhang, making it easier to use these models on devices with limited resources. Therefore, by using the GPU, the training process outlined in the original paper would take approximately three hours to complete, while our modification takes approximately 45 minutes which our update could make it possible to use the model in environments with constrained computing resources, such as embedded systems [11].

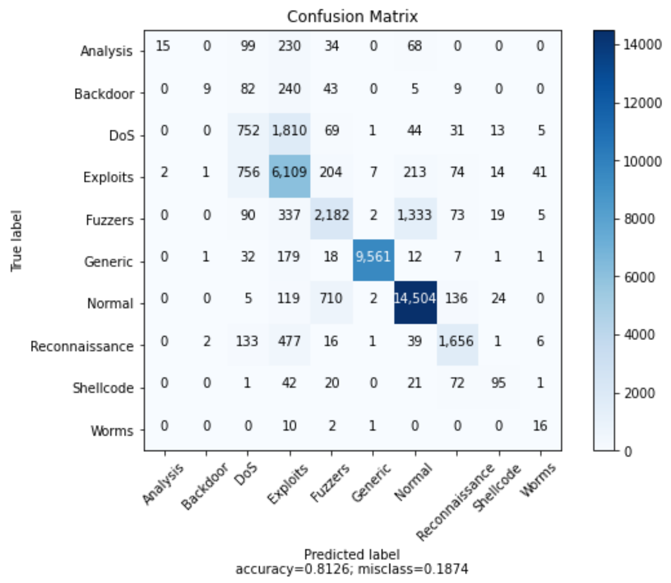


Figure 4. Confusion matrix epoch 5

4. CONCLUSIONS AND SUGGESTIONS

Since the beginning of widespread Internet usage, cloud technologies have been disruptively adopted, and this has caused an exponential increase in the number of intrusion events. Deep learning algorithms have been used in numerous studies on network intrusion detection. By using ReLU, It has been successful in producing good results, but it has the drawback of zeroing out negative input values. To address this issue, a variant called LeakyReLU was introduced, which does not zero out negative input values. The contribution of this study is to modify the 1D-CNN model using the PRELU activation function for better accuracy than previous studies and simplify the layer for faster training time. In the previous research, a 10-epoch method was utilized. However, in order to facilitate a more efficient comparison within the scope of our investigation, it was necessary to deviate from this standard and reduce the epoch to 5. Upon examination, it is evident that the results of our modification are not significantly different from those presented in the original paper. While the original paper utilized 214,670 parameters, our modification required only 80,522 parameters. By modifying the source code from the previous paper we managed to get an average accuracy of 0.81008, only 0.002907% less than the previous paper but has a significantly faster training speed.

REFERENCES

[1] Sinha, J., & Manollas, M. (2020, June). Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection. AIPR, Pages 223–231. 10.1145/3430199.3430224
[2] A, A. H., & Sundarakantham, K. (2019). MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM. International Conference on Trends in Electronics and Informatics, (3). 10.1109/icoei.2019.8862784
[3] Chandre, P., Mahalle, P., & Shinde, G. (2022, June). Intrusion prevention system using convolutional neural network for wireless sensor network. IAES International Journal of Artificial Intelligence, Vol. 11, No. 2, pp. 504~515. 10.11591/ijai.v11.i2.pp504-515

[4] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.

[5] JIANG, K., WANG, W., WANG, A., & WU, H. (2020). Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. IEEE Access, vol. 8,, pp. 32464-32476. 10.1109/ACCESS.2020.2973730

[6] Shende, S., & Thorat, S. (2020, June). Long Short-Term Memory (LSTM) Deep Learning Method for Intrusion Detection in Network Security. International Journal of Engineering Research & Technology, Vol. 9(Issue 06). 10.17577/IJERTV9IS061016

[7] Krishnan, A., & Mithra, S. T. (2021, June). A Modified 1D-CNN Based Network Intrusion Detection System. IJRESM, vol. 4, no. 6, pp. 291–294. http://journals.resaim.com/ijresm/article/view/921

[8] Wang, Z., Yan, Y., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 765-774.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. (arXiv). 10.48550/ARXIV.1502.01852

[10] Sinha, J., & Manollas, M. (2020, June). Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection. AIPR, Pages 223–231. 10.1145/3430199.3430224

[11] Laghrissi, F. E., Douzi, S., Douzi, K., & Hssina, B. (2021, May 07). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, Vol. 8, No. 65. https://doi.org/10.1186/s40537-021-00448-4