International Journal of Statistics and Probability; Vol. 12, No. 6; November 2023
ISSN 1927-7032 E-ISSN 1927-7040
Published by Canadian Center of Science and Education

A Predictive Model to Predict a Cyberattack Using Self Normalizing Neural Networks

Oluwapelumi Eniodunmo[1] & Raid Al-Aqtash[1]

[1] Department of Mathematics and Physics, Marshall University, One John Marshall Dr, Huntington, WV 25755

Correspondence: Raid Al-Aqtash, Department of Mathematics and Physics, Marshall University, One John Marshall Dr, Huntington, WV 25755, USA. E-mail: alaqtash@marshall.edu

Received: August 19, 2023 Accepted: November 3, 2023 Online Published: December 29, 2023

doi:10.5539/ijsp.v12n6p60 URL: https://doi.org/10.5539/ijsp.v12n6p60

Abstract

A cyberattack is an unauthorized access and a threat to information systems. Intelligent intrusion systems rely on advancements in technology to detect cyberattacks. In this article, the KDD CUP 99 dataset, from the Third International Knowledge Discovery and Data mining Tools Competition that was held in 1999, is considered, and a class of neural networks, known as Self-Normalizing Neural Networks, is utilized to build a predictive model to detect cyberattacks in the KDD CUP 99 dataset. The accuracy and the precision of the self-normalizing neural network is compared with that of the k-nearest neighbors and the support vector machines, in addition to other models in literature. The self-normalizing neural network appears to perform better than other models in predicting cyberattacks, while also being efficient in predicting a normal connection.

Keywords: Cyberattack, classification, data mining, neural networks, support vector machines, KDD CUP 99

1. Introduction and Motivation

Cyberattacks form a never-ending war that has posed a great threat to secured information systems. It is the use of technical shortcomings of a network security mechanism, to either gain access to unauthorized information or disrupt the elements of a network. The developments of automated and intelligent systems provide more computing power to hackers to steal information, destroy data or system resources, and have raised global security issues. Data mining tools have received continuous attention in research. These tools can be adopted to create sophisticated intrusion detection systems that help information systems mitigate and defend against cyberattacks. However, the advancement in technology and accessibility of information creates more identifiable elements that can be used to gain unauthorized access to systems and resources. Data mining and classification tools such as the k-nearest neighbor (KNN), support vector machines (SVM), decision tress, among others, improved over time to build models for intrusion detection systems, and to enable information systems, internet connected devices or devices running on some form of computer network, to gain immunity against cyberattacks. However, these classification models hit some limitations as the size of data increase. Neural networks are deep machine learning tool that can be used to handle big data and to understand complex relationships. Neural networks use artificial intelligence and interconnected nodes in hidden layers to create predictive models. Figure 1 is an example of a simple neural network to predict a binary response using three input features and two levels of hidden layers. Recent research proved to build better models by showing better accuracy for intrusion detection systems using neural networks.

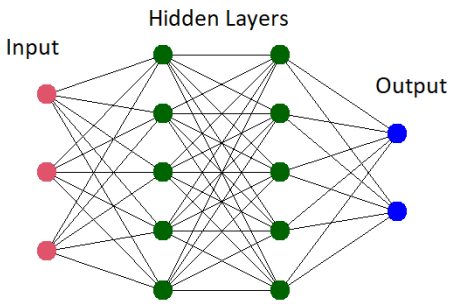

Figure 1. Example of a simple neural network

For the rest of this article, we describe the main attributes of a well known cyberattacks dataset in section 2. In section 3, we utilize a special case of neural networks, namely the self-normalizing neural network (SNN),introduced by Klambauer et al. (2017), to create a predictive model to predict cyberattacks and detect network intrusions. In section 4, we discuss and compare the SNN to other classification models, such as KNN and SVM. Finally, we conclude the article by summarizing the main findings and comparing the SNN to other models in literature. This article is based on Eniodunmo (2022) master's thesis in mathematics at Marshall University.

2. Data Preparation

A network enables connection and information exchange between devices, they are implemented with security in consideration. This network which powers communication, consists of 3 significant parameters: connection (or basic) features, content features, and traffic features. The KDD CUP 99 dataset contains a little less than 5 million records of connection vectors, over 41 features, labeled as either a normal connection or a form of network attack, with precisely one specific attack type. There are 22 different attack types that can be grouped into four major categories: Denial of Service (DOS), Probe, UserToRoot (U2R) and RemoteToLocal (R2L). The 10% KDD Cup 99 dataset used in this study, contains 494,021 records over 41 features, including 9 categorical features (6 of them are binary) and the remaining 32 are numeric features. The first step in preparing the data for building our predictive model, was to take a descriptive and graphical overview of the dataset. Each of the features was observed for missing values. No missing values were found, and there were no significant outliers from the dataset due to the domain of the data. Since an attack might involve manipulation of network parameters in a non-definite pattern, we expect that our data ranges through extreme values. We then categorize the specific attacks into their respective categories: DOS, U2R, R2L, and Probe.

Feature selection was performed, as it helps to learn meaningful features and to remove the unimportant or noisy features. The responses are label encoded and represented as integers. The categorical features in the dataset are scored with respect to the response using the mutual information method, which is a statistical measure of the dependence between two variables, by evaluating the reduction in entropy in transforming a dataset. The information gain for each random variable X is calculated as

$$I(X;Y) = H(X) - H(X|Y),$$

where $I(X;Y) \geq 0$ is the mutual information for X and Y, $H(X)$ is the entropy for X and $H(X|Y)$ is the conditional entropy for X given Y. The value of $I(X;Y)$ represents the strength of the relationship between feature X and the response Y. A significant positive number represents a very strong relationship, and a small positive value represents a weak relationship. If $I(X;Y) = 0$, then the variables are independent. Thus, the features with higher information gain $I(X;Y)$ are selected. The quantitative features are observed for a significant relationship with the categorized labels by plotting side-by-side boxplots of each numeric variable among the categorized response labels. While the graphs showing which features have a relationship with the categorized labels, we use the analysis of variance (ANOVA) method to obtain the essential numeric features in our dataset by scoring each feature with respect to the response. The ANOVA determines the existence of statistically significant differences among several group means. Hence, we test for statistically significant differences for a feature in our dataset among the response labels. Now that we have the features that matter to the response variable, we prepare the selected features for modeling. The categorical features are one-hot encoded, and the numeric features are standardized with a mean of 0 and a standard deviation of 1. In this step, 24 feature were selected for modeling. The selected features are displayed in Table 1.

Table 1. The 24 selected features; From (Eniodunmo, 2022)

1. service	9. srvcount	17. dsthostsrvserrorrate
2. loggedin	10. dsthostdiffsrvrate	18. serrorrate
3. protocoltype	11. dsthostsamesrcportrate	19. srvserrorrate
4. flag	12. dsthostsrvdiffhostrate	20. dsthostserrorrate
5. isguestlogin	13. hot	21. rerrorrate
6. rootshell	14. diffsrvrate	22. dsthostsrvrerror_rate
7. count	15. srvdiffhostrate	23. srvrerrorrate
8. dsthostcount	16. samesrvrate	24. dsthostrerrorrate

3. Predictive Modeling

The class of self-normalizing neural networks, introduced by Klambauer et al. (2017), is a simple class of neural networks that model structured data by using a set of linear layers and non-linear activation function, known as a *Scaled Exponential*

Linear Unit (SELU), which is given by

$$f(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \leq 0 \end{cases}$$

where $\lambda \approx 1.0507$ and $\alpha \approx 1.6733$ are predetermined. The assumptions of a self normalizing neural network are:

1. The inputs must have a mean of 0 and variance of 1.
2. The network weights should be lecun normalized, that is, they should have a mean of 0 and variance of $\frac{1}{n}$, where n is the number of neurons in the hidden layer.
3. The SELU activation function is used.
4. If dropout is used, then the alpha-dropout should be used.

The SELU activation function, proposed by Klambauer et al. (2017), is self-normalizing the weight, biases and activations of the neural network. This activation function keeps a constant variance in the network, the self normalizing property is realized by the parameters λ and α as they keep inputs from previous layers that already have their mean and variance predefined on an interval, in that same interval. The SELU function is special in that it does not face the vanishing or exploding gradient problem, and it does not have the 'dead SELU' problem unlike the ReLU activation function. For more details on SELU and SNN, we refer to Klambauer et al. (2017).

4. Discussion and Comparisons

After the variable selection process, the dataset was divided into 80% for training and the remaining 20% for testing, to enable training and checking the accuracy of our model.

A SNN consisting of 24 input features, 5 hidden layers of 30 neurons, and 5-class output, is constructed. We want to choose the least number of layers efficient for a model, as training time increases with the number of layers in a model. After several parameter tuning for our model, a 5-layer and 30 neurons model performs efficiently for our dataset. Each layer performs the linear operations of weight multiplication and bias addition. It then applies the SELU activation function and uses the AlphaDropout technique with a dropout probability of 0.0002, which is the best value for our model after several tuning. We use 500 epochs, a batch size of 10,000, and a learning rate of 0.05. Since this is a multi-class classification problem, we use cross-entropy loss to evaluate the loss in our model for each epoch and the Adam optimizer. The training data is then fed into the neural network model. The performance of the model is evaluated on the test data. The confusion matrix for the SNN model on the test data is shown in Figure 2. Three measures are used to assess the performance of the model: 1) Accuracy, which represent the correct classification rate, 2) Precision, which is the positive predictive value, and 3) Recall, or the sensitivity. Table 2 shows accuracy of the model as well as the precision and recall for each output category. The SNN has 99.97% accuracy in prediction with very high precision in detecting normal connections, DOS attacks, and Probe attacks. Thus, our model is excellent for an intrusion detection system that intends to prevent such attacks. The SNN model is also great for R2L attacks with a high precision of 94%, but as for U2R attacks, we have a 70% precision. This is not bad for intrusion detection systems that do not primarily have U2R attacks.

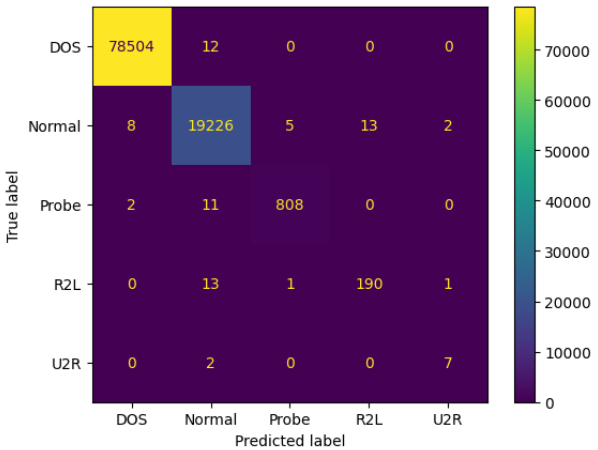

Figure 2. SNN Confusion Matrix for the 20% test data from the 10% KDD Cup 99 cyberattack dataset *

*From (Eniodunmo, 2022)

Table 2. SNN Model Accuracy, Precision and Recall; From (Eniodunmo, 2022)

	Class	Precision	Recall
SNN (99.97%)	Normal	100%	100%
	DoS	100%	100%
	Probe	99%	98%
	R2L	94%	93%
	U2R	70%	78%

Parameter tuning were made, such as selecting different combinations of 24 features that fall into the three significant network parameters. Batch sizes of 10,000, 20,000, and 60,000 were combined with the stochastic gradient descent and Adam optimizer. Table 3 shows some results of the different parameter tuning applied to the network.

Table 3. Our neural network results with some of the different parameter tuning; From (Eniodunmo, 2022)

Batch Size	**Learning Rate**	**Hidden Neurons**	**Dropout Probability**	**Optimizer**	**Accuracy**
10,000	0.05	30	0.0002	Adam	99.97%
10,000	0.01	30	0	Adam	99.95%
20,000	0.01	32	0	SGD	99.48%
20,000	0.01	32	0.05	SGD	99.95%
60,000	0.01	30	0.001	SGD	99.94%

The KNN algorithm is applied to the cyberattack dataset consisting of the 24 selected features and the 80% training and 20% testing datasets. Consequently, a 99.32% accuracy is achieved by choosing an odd k of 629, which is the square root of the number of records in the 80% training set of the 10% KDD Cup 99 dataset. Figure 3 shows the confusion matrix for the KNN model on the test data, while the measures of accuracy, precision and recall are listed in Table 4. From Table 4, we can conclude that the KNN has slightly less accuracy, precision and recall than the SNN model. Further, it can be observed that the KNN has a precision of 0% for the U2R class of attack. This implies that the KNN method fails to identify U2R attacks.

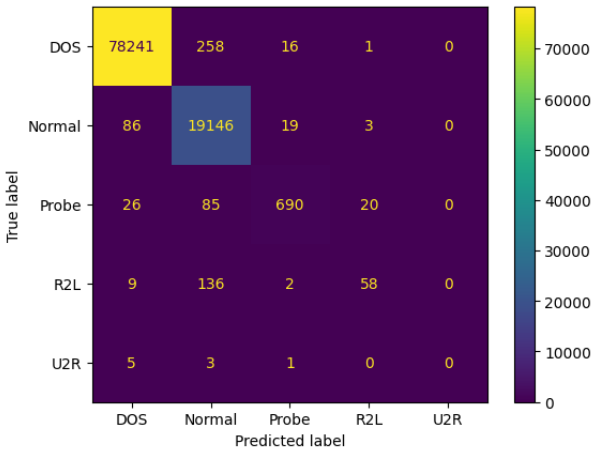

Figure 3. KNN Confusion Matrix for the 20% test data from the 10% KDD Cup 99 cyberattack dataset*

*From (Eniodunmo, 2022)

Table 4. KNN Model Accuracy, Precision and Recall; From (Eniodunmo, 2022)

	Class	Precision	Recall
KNN (99.32%)	Normal	98%	99%
	DoS	100%	100%
	Probe	95%	84%
	R2L	71%	28%
	U2R	0%	0%

The SVM was also considered. The confusion matrix for the SVM on the test data is displayed in Figure 4 and the accuracy, precision and recall are listed in Table 5. By observing Table 5, we can see that the SVM has 99.85% accuracy, which does not perform better than the SNN. The SVM can be seen in Table 5 to have less precision for the attack classes compared to the SNN model.

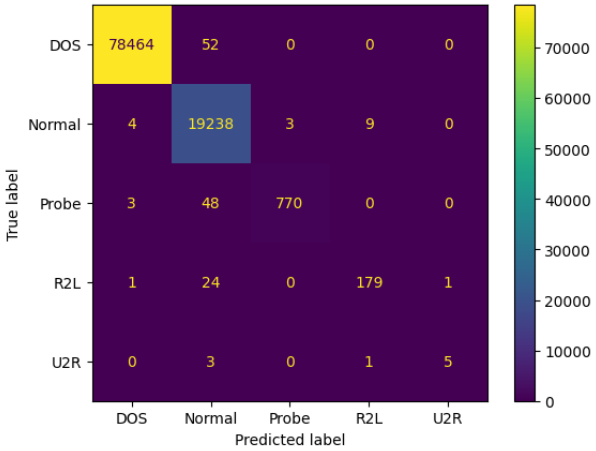

Figure 4. SVM Confusion Matrix for the 20% test data from the 10% KDD Cup 99 cyberattack dataset *

*From (Eniodunmo, 2022)

Table 5. SVM Model Accuracy, Precision and Recall; From (Eniodunmo, 2022)

	Class	Precision	Recall
SVM (99.85%)	Normal	99%	100%
	DoS	100%	100%
	Probe	100%	94%
	R2L	95%	87%
	U2R	83%	56%

5. Conclusion

In this article, a class of self-normalizing neural networks (SNN) is used to detect network intrusion from the 10% KDD CUP 99 dataset. The top 6 categorical predictors, with a strong relationship with the response variable were selected using the mutual information method. The top 18 predictors with a strong relationship with the response variable were selected using the ANOVA F-value method. Our predictive model using the SNN class of neural networks with a 99.97% accuracy appears to perform better than the KNN and SVM models, where the accuracy is 99.32% and 99.85%, respectively. We also look at the precision of our models in classifying the connections, and our SNN model appears to be a good predictor for normal connections, DOS, R2L, and Probe attacks with 100%, 100%, 94%, and 99% precision, respectively.

The self-normalizing neural network also appears to have better accuracy than Liu and Zhang (2020), which uses a convolutional neural network with an accuracy of 98.02%. It is also performing better than Adams et al. (2022), which used an artificial neural network with the hyperbolic tangent activation function, had an accuracy of 99.1%. The SNN also appears to perform better than TS, and Shrinivasacharya (2021), which used a Bi-directional LSTM and performs better than most traditional machine learning methods, with an accuracy of 99.73%.

A possible future development is that this class of self-normalizing neural network can be evaluated against other well-known cyberattack datasets such as the NSL-KDD, UNSW-NB15, and ISCX 2012. These datasets have been generated to simulate more recent network traffic scenarios, wide varieties of low-footprint intrusion, and the possible attack complexities that may occur on a network in the present time, which are not present in the KDD Cup 99 dataset as it was generated two decades ago. The KDD Cup 99 dataset also suffers from certain limitations, such as redundancy of data, as a result of which bias may occur.

References

Adams, S. O., Azikwe, E., & Zubair, M. A. (2022). Artificial neural network analysis of some selected KDD cup 99 dataset for intrusion detection. *Acta Informatica Malaysia*, *6*(2), 50-56.

AL-Shabi, M. (2021). Design of a network intrusion detection system using complex deep neuronal networks. *International Journal of Communication Networks and Information Security, 13*(3), 409-415.

Bebeshkoa, B., Khorolskaa, K., Kotenkoa, N., Kharchenkoa, O., & Zhyrova, T. (2021). Use of neural networks for predicting cyberattacks. *CEUR Workshop Proceedings*, *2923*, 213-223.

Eniodunmo, O. (2022). A Predictive Model to Predict Cyberattack Using Self-Normalizing Neural Networks (Order No. 29994170). Available from Dissertations & Theses @ Marshall University; ProQuest One Academic. (2746455893). Retrieved from https://marshall.idm.oclc.org/login?url=https://proquest.com/?url=https://www.proquest.com/dissertations-theses/predictive-model-predict-cyberattack-using-self/docview/2746455893/se-2

Ghanem, W., & Jantan, A. (2019). Training a neural network for cyberattack classification applications using hybridization of an artificial bee colony and monarch butterfly optimization. *Neural Processing Letters*, *51*, 905-946. https://doi.org/10.1007/s11063019-10120-x

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. https://doi.org/10.48550/arXiv.1706.02515

Liu, G., & Zhang, J. (2020). CNID: Research of network intrusion detection based on convolutional neural network. *Discrete Dynamics in Nature and Society*. https://doi.org/10.1155/2020/4705982

Priya, S., & Kumar, K. (2021). Performance analysis comparison on various cyber-attack dataset by relating a deep belief network model on an intrusion detection system. *Information Technology In Industry*, *9*(3), 608-613.

TS, P., & Shrinivasacharya, P. (2021). Evaluating neural networks using bi-directional LSTM for network IDS (intrusion detection systems) in cyber security. *Global Transitions Proceedings*, *2*(2), 448-454. https://doi.org/10.1016/j.gltp.2021.08.017

Copyrights