Comparison of Shallow and Deep Neural Networks for Network Intrusion Detection

Daniel E. Kim
Deparment of Computer Science
California State University, Fullerton
Fullerton CA, 92831 USA
dan.kim@csu.fullerton.edu

Mikhail Gofman
Deparment of Computer Science
California State University, Fullerton
Fullerton CA, 92831 USA
mgofman@fullerton.edu

***Abstract*—The increasing complexity and malice of modern computer and network attacks drives a need and search for more adaptive and smarter intrusion detection methods. Neural networks can provide a useful, self-learning approach to threat detection for network intrusion. After testing a variety of simple shallow and deep neural networks on the well-known NSL-KDD dataset comprised of network traffic capture containing 148,000 observations and 41 features with 22 specific attacks, we confirm the findings of previous researchers [15] that shallow neural networks are better suited for network intrusion detection than deep neural networks. Shallow networks were able to more accurately classify network data and produced lower error rates compared to deep networks.**

Keywords—network security, machine learning, neural networks

I. Introduction

The increase in both the occurrence of large security breaches and the severity highlights the need for more intelligent network security solutions. The application of machine learning to network security is a burgeoning new field with a promise of providing tools and methodologies helpful in addressing many long-standing security problems. This paper examines the use of a specific category and form of machine learning, neural networks, and it explores the comparisons between shallow and deep neural networks regarding their efficacy of network intrusion detection.

Current widely used network security systems only offer basic, static, and reactionary solutions to security problems, which is similar to the “checklist” mindset of computer and network security. The most common network security components include firewalls and antivirus and antimalware programs. These solutions only offer a preventative and reactionary approach to network security, and cannot adapt to or learn from threats they encounter [9]. In addition, these systems must be constantly updated with new threat definitions.

Network security can operate at varying levels of the OSI and TCP/IP protocol stack, such as with firewalls that operate anywhere from the application layer to the transport layer [2]. Network security practitioners and service providers apply a myriad of methodologies to secure networks and systems. Most services employ a traditional method of checking against known attack libraries and signatures, whereas other services can be more creative and can employ heuristic approaches. Despite the variety of approaches, few have incorporated artificial intelligence (AI) into the many active tools of network intrusion detection.

There are various types of machine learning approaches, and many of these approaches have been applied to some form of network security application. One such approach is genetic algorithms, which use randomly generated sets of functions designed to output a solution to a problem, and these sets are altered by genetic operators that produce an evolved set of solutions [21]. The generated solutions are scored by a fitness function, and the best performing solution is selected. Not all implementations of genetic algorithms have performed well in network data classifications, such as [21], who achieved only around a 57% detection rate with their genetic algorithm; however, in [13], up to a 96% detection rate was achieved using a specially tailored genetic algorithm and fitness function. It is worth noting, that using this approach, only two attack types were existent in the network data and the feature set consisted of seven features, which allows for an easier classification task.

The approach proposed involves applying neural networks as the machine learning method of choice for network intrusion detection. Specifically, the efficacies of shallow networks compared to deep neural networks in network data classification between normal and malicious network data are explored.

II. Neural Networks

Neural networks are a popular form of machine learning in modern applications. The power of neural networks results from the interconnected, massively parallelized computing units structured into distinct layers [5]. The basic idea and motivation behind neural networks were derived from the biological systems of the human brain. Neurons, or nodes, receive weighted input signals which are then summed and

978-1-5386-4649-6/18/$31.00 ©2018 IEEE

passed through an activation function that bounds the output of the neuron, sending the output to nodes in the next layer [10].

A. Shallow vs Deep Neural Networks

Since their inception, various forms and types of neural networks have been developed, including shallow and deep neural networks. The terms "shallow" and "deep" refer to the number of layers in a neural network. Shallow neural networks refer to a neural network that has a small number of layers, usually regarded as having a single hidden layer, and deep neural networks refer to neural networks that have multiple hidden layers. Both types of networks perform specific tasks better than the other, and selecting the correct network depth is important in creating a successful model.

The simple structure of shallow networks enables them to learn important features independently from other features, which allows them to be well-suited for learning tasks that involve low-dimensional data [8]. Deep neural networks excel in highly complex tasks with large input data and a high dimensionality. The top layers learn higher-level features derived from the patterns and knowledge from the lower layers. This creates layer-dependent learning where features from lower layers are reused to solve more complex tasks [8].

B. Network Architecture

The shallow networks used in this study followed a similar architecture. All shallow networks included a simple multi-layer perceptron with an input layer, a single hidden layer, and an output layer. The number of hidden nodes in the hidden layer was the only varied hyperparameter. The networks utilize on-line learning, as opposed to batch learning, and use a standard backpropagation and gradient descent method. Specifically, the shallow networks used the scaled conjugate gradient descent (SCG) backpropagation algorithm. Scaled conjugate gradient descent allows for both quicker convergence and greater computational efficiency during gradient descent calculations. The activation function used for the layers was the hyperbolic sigmoid function, better known as a tanh function. A tanh function provides a more robust classification of the network data, bounding the output of the neurons between -1 and 1. All layers of the network were fully connected to ensure that all possible combinations of features were considered for the classification tasks.

The deep neural networks used in this research include many of the aspects of the shallow neural networks. Instead of a single hidden layer, the deep networks have two hidden layers. The layers of the deep networks were fully connected, utilized on-line learning, and used a tanh activation function. SCG backpropagation was also used by the deep networks. The number of hidden nodes in each hidden layer was the varied hyperparameter of the networks, which was used to identify an optimal performing network.

Fig. 1 and Fig. 2 visually depict the architectures of a shallow and deep network, respectively, that were used in the experimentation of this research. Both images were visualized using MATLAB 2016b.

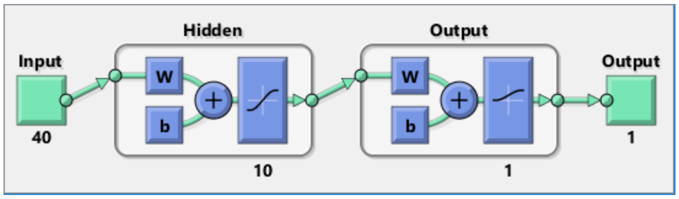

Fig. 1. Shallow neural network architecture, visualized by MATLAB

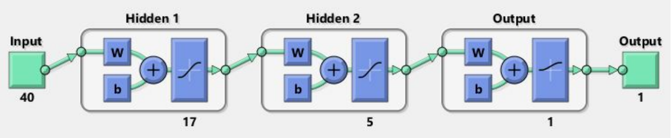

Fig. 2. Deep neural network architecture, visualized by MATLAB

III. EXPERIMENTATION

The experiments were performed using MATLAB version 2016b and the Neural Network Toolbox on a Windows 10 system. MATLAB offers both script-oriented and GUI-based methods for implementing specific neural networks, both of which were used for experimentation and development.

A. Dataset

Although there is an abundance of publicly available network traffic data, mainly in the form of pcap files from utilities such as tcpdump and Wireshark, there are only a limited number of labeled datasets. As neural networks are a form of supervised machine learning, datasets must be labeled for the networks to properly learn from a target output. This requirement narrowed the search for datasets to two specific datasets: the KDD 99 CUP dataset and the NSL-KDD dataset. The KDD 99 dataset is a highly popular and commonly used dataset for network attack and intrusion detection, but it includes several flaws [18]. The flaws of the KDD 99 dataset are extensively covered by [18] and include numerous duplicate and redundant network data records, which can cause a skew in the performance of classification algorithms related to frequently occurring data.

Thus, in [18], an improved dataset, the NSL-KDD dataset, was created to address and remedy the issues found in the KDD 99 dataset. The resulting dataset is much more challenging in the context of anomaly detection. The NSL-KDD dataset removes all duplicate observations and contains a subset of the original data with over 148,000 observations. The data consists of 41 features with 22 specific attacks. As the NSL-KDD dataset was used in the experiments, the data were preprocessed in preparation for neural network usage.

The first task required to preprocess the data was to enumerate all nominal data in the dataset. Neural networks are not able to process nominal data types, nor can they process data of different types. With most of the features already consisting of numerical values, enumerating the nominal data types was the most intuitive step. Next, features with no varying values, or features in which all observations had the same value, were removed to omit unnecessary and unhelpful data and to avoid causing classification difficulties and errors. Finally, the packet type feature in the dataset that differentiated between normal and specific attack type values was enumerated to a binary format, in which normal packets were represented by 0 and all attack packets were represented by 1. This forced the system to recognize all types of attacks instead of just a specific attack and to categorize them under a singular attack class.

B. *Results*

All tested networks were run multiple times to gather averaged performance values. All the neural nets used the preprocessed NSL-KDD dataset as the input, and the data was split in the following manner: 70% for training, 15% for validation, and 15% for testing. Rather than limiting the training phase to a specific number of epochs, where an epoch is a complete pass of all training data through the network, the training was set to expire when six validation checks were reached. The validation checks implemented in MATLAB are triggered when the non-training validation subset error rate continuously increases for more than the set number of epochs, which was six epochs in this case.

TABLE I. NEURAL NETWORK PERFORMANCE

Network Type	**Network Performance**		
	Number of Hidden Nodes	***Accuracy***	***Error***
Shallow Neural Networks	6	96.70%	3.40%
	10	97.85%	2.10%
	17	98.50%	1.40%
	20	98.45%	1.70%
Deep Neural Networks	10, 2	48.15%	53.20%
	10, 5	48.10%	52.00%
	17, 2	48.20%	51.90%
	17, 5	48.30%	51.10%
	20, 2	48.10%	51.90%
	20, 5	48.10%	52.10%

The results shown in Table 1 indicate that the network with 17 hidden nodes performed best of all other networks. This network also had the highest consistency in its maximum level of accuracy. Thus, the 17-hidden node neural network was chosen to represent the shallow neural network group for its performance and capability. The peak performance of the 17-hidden node shallow network was a 98.50% detection rate of malicious traffic and an error rate of 1.40% false classification. The highest performing deep network achieved a 48.30% detection rate, a decrease in performance of around 50%, and an error rate of 51.10%. Due to the consistent lower performance and lack of variance of accuracy for the deep neural network, the hyperparameters that were tested were limited.

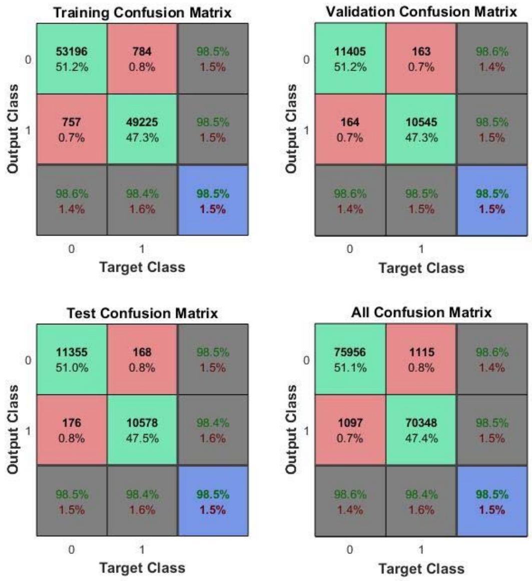

Fig. 3. Confusion matrix of shallow neural network

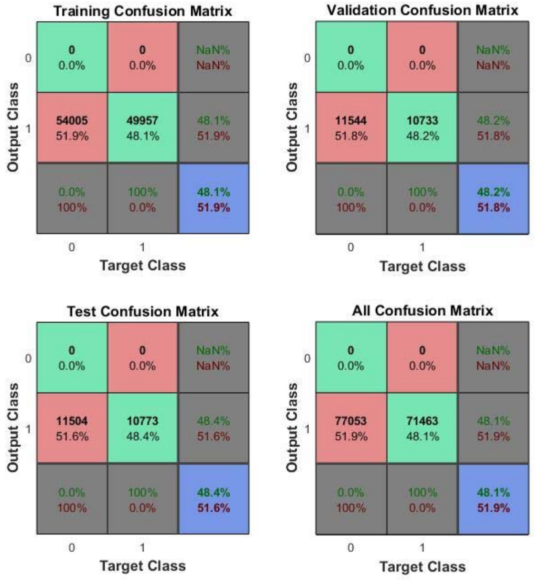

Fig. 4. Confusion matrix of deep neural network

The confusion matrices in Fig. 3 and Fig. 4 show a clear distinction in the performance of the shallow and deep neural networks. A closer look at the confusion matrix reveals that all observations were classified as malicious network packets in the deep neural network. The true negative and false negative rates for the deep network were both 0, and the false positive rate is over 50%, an extremely high error rate. The deep neural networks are only able to classify all the network data into a single class, resulting in the 50/50 split distribution of the observations between the true positive and false positive rates.

IV. Related Works

The high performance of up to a 98.50% accuracy for the shallow neural network and the figures derived from its execution show that the network presents a solid classification model and is optimized for network traffic data classification. Various researchers experienced a similar high performance and accuracy with simple shallow neural networks, such as a group of researchers who tracked and classified the system behaviors of different users and achieved a 96% accuracy in user classification [14]. Although not particularly equivalent to network data, the research and results of [14] provide a glimpse into the applicability of neural networks in computer activity classification and system intrusion detection. The researchers crafted a unique profile for each of the 10 users on a single machine by tracking the number of times a set of 100 commands were executed, such as "ls" and "chmod".

Researchers from [6] were able to achieve similarly high-performance rates using network data with a 99.4% level of accuracy in their multi-level perceptron model; however, their network was trained and designed to only detect a single type of attack: DoS and DDoS attacks, which only utilized UDP packets. In addition to only being tested against a single type of attack, modern DoS attacks do not only utilize UDP packets, which allows for a much clearer distinction between normal data and malicious DoS attack data. In [11], an artificial neuron network approach to network intrusion detection was also applied, which achieved a maximum performance of a 99.25% accuracy. Similarly, researchers from [4] utilized a simple neural network approach for network intrusion and achieved a 99.9% accuracy. The researchers from the last two studies, however, used the KDD 99 dataset, which is deeply flawed with a large number of duplicated data and unequal distribution of normal and attack data within subsets, as discussed in the experimentation chapter [18]. Although it was once a benchmark dataset commonly used for network intrusion applications, the KDD 99 dataset is widely regarded as unusable for reliable testing. Tests using our own shallow neural network with the KDD 99 dataset showed a similar extremely high performance, achieving a maximum of 99.9% accuracy.

The researchers from [17] were one of the few, if not only other team, that utilized the NSL-KDD dataset for testing their neural network approach to intrusion detection. They created and used an optimized shallow neural network for their artificial neural network-enabled intrusion detection system model. Results were collected for both binary class classification and a 5-attack type classification, achieving 98.86% and 95.05% accuracy respectively. However, they did not compare the performance of their shallow neural network to the deep neural networks like we did.

V. Conclusion

As the performance of the shallow networks achieved 98.50% accuracy for malicious network data detection and the deep networks only achieved a 48.30% accuracy, there is a clear performance advantage for shallow networks. Compared to the 1.5% false positive rate of shallow networks, extremely high false positive rates also disqualified the deep neural networks from practical network security applications due to the costly nature of such errors [15]; a fact that our evaluation deep neural networks on the NSL-KDD dataset has confirmed. The deep neural networks could only classify all network data into a single class, as malicious network data, suggesting a loss of information in the network data features during the higher-level transformations that occur in the higher hidden layers of the deep networks.

In the midst of various research studies being conducted related to system and network intrusion detection, the results and information gained from this research offers a unique perspective regarding the efficacy of shallow neural networks compared to deep neural networks. The usage of the NSL-KDD dataset offers a more disciplined neural network model that can accurately classify network data despite the increased difficulty in proper classification due to the corrected dataset. The superior performance of shallow networks is not the only advantage over deep networks; execution times, along with decreased computational resources due to the presence of only a single hidden layer, offers a more computationally lightweight machine learning approach to network intrusion detection.

References

[1] A. Bivens, C. Palagiri, and M. J. Embrechts (2002). Network-based intrusion detection using neural networks.

[2] M. Curtin (1997, March). Introduction to network security. Retrieved from http://www.interhack.net/pubs/network-security/

[3] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," presented at the Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy, 4-6 May 1992).

[4] L. P. Dias, J. J. F. Cerqueira, K. D. R. Assis, & R. C. Almeida, "Using artificial neural network in intrusion detection systems to computer networks," paper presented at the 2017 9th Computer Science and Electronic Engineering (CEEC), 27-29 Sept, 2017.

[5] S. Haykin, Neural networks and learning machines. Upper Saddle River, NJ: Pearson Education, Inc, 2009.

[6] E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," paper presented at the 2016 International Symposium on Networks, Computers and Communications (ISNCC), 11-13 May, 2016.

[7] J. Z. Lei and A. Ghorbani, "Network intrusion detection using an improved competitive learning neural network," presented at the Proceedings. Second Annual Conference on Communication Networks and Services Research, 19-21 May 2004.

[8] H. N. Mhaskar and T. Poggio (2016). Deep vs. shallow networks: an approximation theory perspective. Computing Research Repository. doi: arXiv:1608.03287

[9] A. Mishra, A. Agrawal, and R. Ranjan, "Artificial intelligent firewall.," presented at the Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, Rajpura/Punjab, India, 2011.

[10] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, no. 1, pp. 525-533, 1993.

[11] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," presented at the Neural

Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on Neural Networks, 2002.

[12] T. Rashid, Make your own neural network: a gentle journey through the mathematics of neural networks, and making your own using the Python computer language. San Bernardino, CA: CreateSpace Independent Publishing, 2016.

[13] G. Ren Hui, M. Zulkernine, and P. Abolmaesumi, “A software implementation of a genetic algorithm based approach to network intrusion detection,” presented at the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network, 23-25 May 2005.

[14] J. Ryan, M. J. Lin, and R. Miikkulainen, “Intrusion detection with neural networks,” presented at the Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10, Denver, Colorado, USA, 1998.

[15] R. Sommer and V. Paxson, “Outside the closed world: on using machine learning for network intrusion detection,” presented at the 2010 IEEE Symposium on Security and Privacy, 16-19 May 2010.

[16] D. Stiawan, A. L. Shakhatreh, M. Y. Idris, K. A. Bakar, and A. H. Abdullah, “Intrusion prevention system: a survey.” Journal of Theoretical and Applied Information Technology, vol. 40, no. 1, pp. 44-54, 2012.

[17] B. Subba, S. Biswas, and S. Karmakar, “A neural network based system for intrusion detection and attack classification,” presented at the 2016 Twenty Second National Conference on Communication (NCC), 4-6 March 2016.

[18] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, 8-10 July 2009). *A detailed analysis of the KDD CUP 99 data set.* Paper presented at the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications.

[19] University of California, Irvine. (1999). Knowledge discovery and data mining tools competition (KDD) 99 [Data file]. Retrieved from http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data

[20] University of New Brunswick. (2009). NSL-KDD [Data file]. Retrieved from http://www.unb.ca/cic/datasets/nsl.html

[21] L. Wei and I. Traore, “Detecting new forms of network intrusion using genetic programming,” presented at the Evolutionary Computation, 2003. CEC '03. The 2003 Congress on Evolutionary Computation, 8-12 Dec. 2003.

[22] Y. Yao, Y. Wei, F. X. Gao, and G. Yu, Anomaly intrusion detection approach using hybrid MLP/CNN neural network,” presented at the Sixth International Conference on Intelligent Systems Design and Applications, 16-18 Oct. 2006.