# Poster: Automated Neural Network Structure Selection for IoT Botnet Detection

Kashif Naveed[*], Hui Wu[†]

School of Computer Science and Engineering, UNSW Sydney, Australia

Email: [*]mkashifn@gmail.com, [†]huiw@unsw.edu.au

***Abstract*—IoT botnet attacks are a major concern these days and their detection is an active area of research. Artificial Neural Networks (ANNs) have proven their power and capabilities to detect botnets effectively. However, the process of ANN structure selection and training has been iterative and experimental where one starts with a random number of layers containing an arbitrary number of neurons within them. Experimental results reveal that this work provides massive gains in terms of computational efficiency over the manually selected network structures.**

***Index Terms*—ANN, OBS, OBD, Deep Learning, Perceptron, Pruning**

## I. Introduction

Neural Networks (NNs) have been widely used in almost every industry [1], ranging from education, engineering, healthcare, medicine, business, marketing, finance and agriculture, to oil & gas. Such networks consist of layers of artificial neurons whose input and output connections are determined by the assigned weights. The structure of a NN plays a crucial role in the performance and is determined by the number of layers and the number of neurons within each layer.

Many open research problems exist in the field and one of such problems is to determine the structure of a NN and the number of epochs required to train the network. Existing techniques, including trial-and-error, heuristics and pruning, are not efficient, either incurring large overheads or failing to achieve the desired acuracy.

Botnets attacks are growing in size and numbers as their source code is available publicly. Such botnet devices are costing an annual loss of up to 2 billion dollars and are capable of generating huge traffic in the order of 1Tbps as published by Symantec and [2]. Neural networks have proven their capabilities to detect such botnets reliably if the neural network structure is selected correctly.

In this work, we investigate the problem of finding an efficient neural network structure for the botnet detection problem for IoT devices such that the botnet detection time is minimized and propose a novel analytical approach. The proposed work analyzes the input data and the preliminary neural network structure to come up with an efficient NN structure. By the word *efficient* here we mean: (1) a neural network with a fewer number of neurons and (2) a layer structure with the minimum computation time.

### A. Incompleteness of Existing Approaches

The design approaches of neural networks can be divided into three categories: trial-and-error approaches, heuristic-based approaches, and pruning-based approaches. The trial-and-error approaches lay out a structure with randomly selected hidden layers and neurons and keep revising until the desired goals have been achieved. The heuristic-based approaches apply some rules to determine the number of hidden layers and their neuron counts. The pruning-based approaches start with an over-sized NN and then remove some layers, neurons and weights which do not contribute significantly.

These approaches, however, are not efficient as more resources are consumed than needed during network training, optimization and operation. Therefore, there is a need for a complete framework that administers the whole learning process starting from devising the network architecture until the completion of the training to achieve the target accuracy and performance goals.

### B. Our Contribution

We perform analysis on the data to determine the network structure, while others either use heuristics [3], [4], [5], [6] without considering all aspects of the data or make use of pruning after the network training [7], [8], [9].

## II. Our Approach

This section presents various functions performed by our proposed framework to compute the best suitable network structure for IoT botnet detection.

### A. Key Philosophy

Andoni et al. [10] conducted a study on the effectiveness of gradient descent functions to approximate polynomial functions. They made the following observations for a function $f$ comprising of a polynomial of degree $d$ over $n$-dimensional variable $x \in \mathbb{R}^n$:

1) The number of hidden neurons required to approximate the function is directly proportional to both $d$ and $n$ [11].
2) The existence of local optima will not stop the neural network to converge to the global optima.

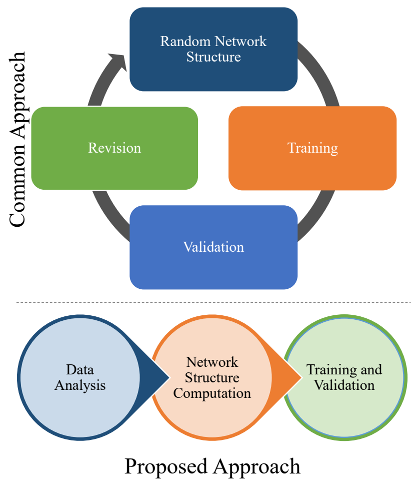

Fig. 1: Common vs proposed approaches.

3) A larger-than-needed network will still achieve similar results if it is compressed as long as it still contains a sufficient number of neurons to approximate the function.

We make use of these observations as key rules to compute the neural network structure to detect botnets in IoT devices using deep neural networks.

## B. Conceptual Overview

The general network structure selection is an iterative process starting with a random layout computed and taking it through training and validation cycles and then revising the design until the required computational and accuracy requirements are met. Our proposed work automates the whole process and computes the required network structure by analyzing the input data alone. This results in a reduction in the computing cost during the training and the operation as presented in the evaluation section as shown in Figure 1.

## C. Network Structure Selection

We start by measuring the normalized frequency $\omega_f$ of the output vector $y$ for each of the feature $f$ in the input matrix $X$. Since the output vector $y$ only contains binary values, $\omega_f$ is measured by counting the number of *edges* in the output against the sorted $f^{th}$ column. The column sorting is an important step because without sorting, the output will be affected by the order in which the input data is presented to the system. Once the normalized frequency is calculated, the degree of sensitivity $\zeta_f$ is computed as $\lceil \log \omega_f \rceil$. The ascending chain $\varpi$ is then computed by sorting the $\zeta_f$ in non-decreasing order and keeping only the unique values. The network structure vector $L$ is computed as a function of $\varpi$ and $i^{th}$ entry is computed as $\lfloor \frac{\sqrt{\frac{\xi^0}{2\xi^{\nu+1}}}}{\varpi_i} \rfloor$, where $\xi^0$ is the number of rows in the input matrix $X$ and $\xi^{\nu+1}$ is the number of neurons in the output layer.

Each entry in the list $L$ gives us the number of neurons in each layer. Since the training and execution time complexity depends upon the sizes of the matrices and the order they are multiplied, we need to compute the optimal structure that minimizes the term $\Upsilon_{NN}$. We then re-arrange the list $L$ to compute a sequence $\xi^1, \xi^2, \cdots, \xi^\nu$ such that $\xi^1\xi^2 + \xi^2\xi^3 + \cdots + \xi^{\nu-1}\xi^\nu$ is minimized.

## III. Evaluation

In this section, we will present the details and results of different experiments that were carried out to quantify the effectiveness of our proposed framework. We made use of Kaggle to run the tests and the Jupyter notebooks can be accessed at GitHub[1].

## A. Dataset Description

In our evaluation, we made use of dataset contributed by Meidan et al. [12]. This dataset contains traffic data for nine commercially available IoT devices infected with botnets. The dataset can either be accessed in a ready-to-use format from Kaggle[2] or the originally published source at University of California Irvine's Machine Learning Repository [13].

*Attacking IoT Botnets:* Botnets are Internet-connected IoT devices that are infected with malicious software enabling them to perform Distributed Denial-of-Service (DDoS) attacks. Two of the most commonly open-source botnets are BASHLITE[3] and Mirai[4]. Most of the current botnets make use of these botnets underneath. The dataset used in the evaluation contains attacks from both of these botnets.

*Gathered Statistics:* The dataset contains both the benign and attack traffic data coming from nine commercially available IoT devices. The infected traffic contains (1) SCAN attacks; (2) TCP, UDP, ACK and SYN flooding; and (3) COMBO attacks attempting to open connections and sending spam data.

*Comparison:* We compare the performance of our proposed work against (1) A **Heuristic** method making use of a 2-layer network [14], (2) A **Genetic** algorithm with an 18-bit chromosome and (3) A **Random** selection method [15].

## B. Neural Network Structure Selection

We have compared different neural network structures generated by the proposed method and the other techniques that are previously described. The first column contains the name of the device whose data was used for the test. The name of the method used for a particular structure is present in the second column. The third column contains the number of layers that were produced by each algorithm. The last column contains several smaller columns indicating the number of neurons within each layer.

*Training and Validation Accuracy:* Figure 2 plots the precision, recall, F1 and accuracy scores for both the training and validation phases. The experiments reveal that the selected neural network structures produce high scores.

[1]https://github.com/mkashifn/dahlia-exp
[2]https://www.kaggle.com/mkashifn/nbaiot-dataset
[3]https://github.com/anthonygtellez/BASHLITE
[4]https://github.com/jgamblin/Mirai-Source-Code

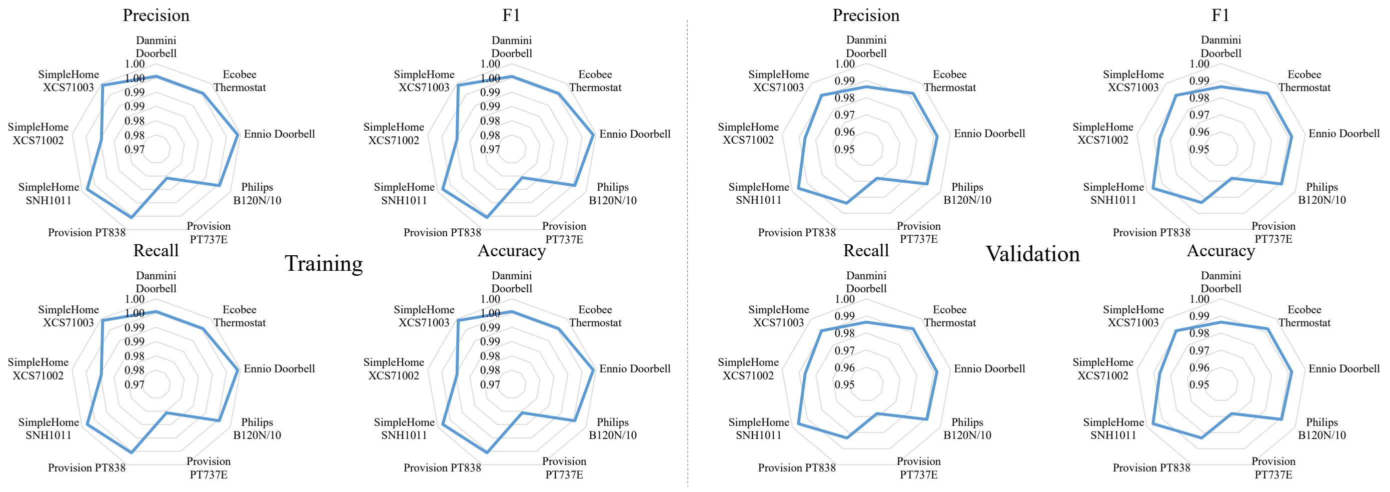

Fig. 2: Precision, recall, F1 and accuracy scores during training and validation.

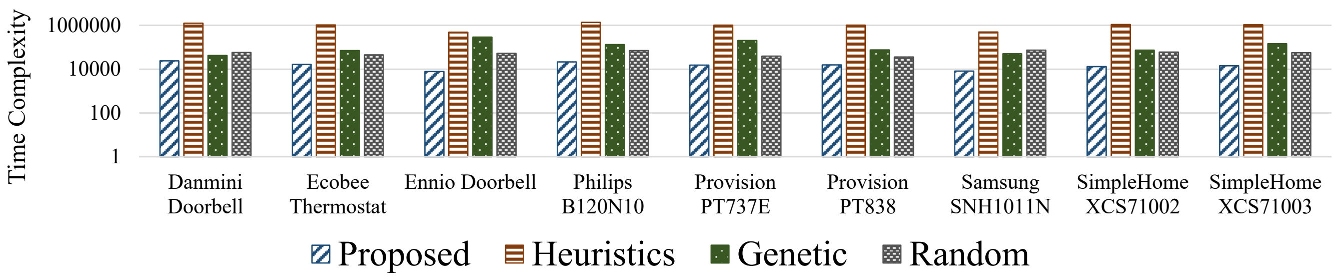

Fig. 3: Comparison of time complexity for network structures generated by different methods.

*Time Complexity Comparison:* Each of the neural networks incurs training and execution time complexities based upon its structure as analyzed in section IV. Figure 3 provides a graph of the resulting time complexities for the network structures generated by different methods for each of the nine devices. As you can see, the proposed work achieves the lowest time complexity compared to all other techniques.

## IV. Conclusion and Future Work

Our work automates the neural networks design process and experimental results reveal that our proposed framework produces a neural network structure that incurs the lowest amount of training and execution time complexity compared to other approaches. Since this work is focused on botnet detection for IoT devices, there is a need to extend the concept to provide solutions for (1) broader anomaly detection; (2) multi-class classification and (3) regression problems.

## References

[1] G. Taylor, *Neural networks and their applications*. John Wiley & Sons, Inc., 1996.

[2] C. Symantec, "Internet security threat report: Volume 24," 2019. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-enf

[3] R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem," in *Proceedings of the international conference on Neural Networks*, vol. 3. IEEE Press New York, 1987, pp. 11–14.

[5] B. D. Ripley, "Statistical aspects of neural networks," *Networks and chaos-statistical and probabilistic aspects*, vol. 50, pp. 40–123, 1993.

[4] F. Wang, "The use of artificial neural networks in a geographical information system for agricultural land-suitability assessment," *Environment and planning A*, vol. 26, no. 2, pp. 265–284, 1994.

[6] D. R. Hush, "Classification with neural networks: a performance analysis," in *Proceedings of the IEEE international conference on systems engineering*, 1989, pp. 277–280.

[7] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[8] V. Tresp, R. Neuneier, and H.-G. Zimmermann, "Early brain damage," in *Advances in neural information processing systems*, 1997, pp. 669–675.

[9] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in neural information processing systems*, 1993, pp. 164–171.

[10] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, "Learning polynomials with neural networks," in *International conference on machine learning*, 2014, pp. 1908–1916.

[11] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information theory*, vol. 39, no. 3, pp. 930–945, 1993.

[12] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot-network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[13] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[14] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274–281, 2003.

[15] M. Naveed and H. Wu, "Celosia: An Immune-Inspired anomaly detection framework for IoT devices," in *2020 IEEE 45th Conference on Local Computer Networks (LCN) (LCN 2020)*, Sydney, Australia, Nov. 2020.