Machine Learning for Web Proxy Analytics

Mark Maldonado, Ayad Barsoum*
St. Mary's University, San Antonio, TX (USA)
Emails: mmaldonado26@mail.stmarytx.edu; abarsoum@stmarytx.edu

Abstract — Proxy servers used around the globe are typically graded and built for small businesses to large enterprises. This does not dismiss any of the current efforts to keep the general consumer of an electronic device safe from malicious websites or denying youth of obscene content. The average internet user does not understand the importance of network security much less than a general computer, phone, tablet, and/or personal information security. With the emergence of artificial intelligence and/or machine learning we can utilize the power to have smart security instantiated around the population's everyday life. An improved security posture needs to be the first thought before accessing any networked device. In this work, we present a simple solution of providing a web proxy to each user of mobile devices or any networked computer powered by a neural network, which will support quick security decisions for all web requests. The idea is to have a proxy server to handle the functionality to allow safe websites to be rendered per request. When a website request is made and not identified in the pre-determined website database, the proxy server will utilize a trained neural network to determine whether or not to render that website. The neural network will be trained on a vast collection of sampled websites by category. Although, this looks like text processing to determine a category, this is only a single part of the entire process. The neural network needs to be trained constantly to improve decision making as new websites are visited, which will provide the best possible security or access controls for users.

*Keywords***:** Machine learning, neural networks, Web proxy, network security

I. Introduction

Over the past couple of decades, the use of machine learning or artificial intelligence is a term that has been coined as the next goal of smart business or providing help to people in everyday life. While the term "Artificial Intelligence" is older than a few decades since John McCarthy coined the term in 1956 [1, 19]. The general concept of artificial intelligence is discovering ways to have machines reason and perform intelligently driven by software and algorithms. This effort is closely related to how the human brain works since we want these machines capable of learning and to think rationally. This is the overarching mindset we must use when moving forward with a hardware or software-based design to implement a security product. Home network security is the number one issue any household is trying to overcome. With the sheer amount of malicious traffic generated, a general understanding of home network security must become mandatory. Ensuring security for a family has become a challenge due to the number of websites children and young teenagers can access. Without proper monitoring of network traffic, our youth can infect their devices, laptops, or other network components, which is just the tip of the iceberg. Now there are sufficient hardware devices and software products to help with these issues. Some issues with this approach are the cost of these hardware devices and the time or knowledge to implement. A software approach is feasible, but this will only protect the device it's installed on. Lastly, having someone understand the potential logs from a hardware device is overbearing or trying to fully understand what a piece of software is doing to protect is also a challenge. The general population needs something easy to use, install, and understand the information being generated from a product that can handle all these problem-sets. The idea is to have a proxy server [7,8,] to handle the functionality to allow the correct websites to be rendered per request. When a request is made and not identified in the pre-determined website database, the proxy server will utilize the neural network. This instance of the network will be tested against the already trained neural network to determine if the requested website is allowed or not.

II. Machine Learning

In the world of machine learning [9, 10, 11, 12, 13] and/or artificial intelligence, one must find the perfect starting point and should have an idea where the project will go or possibly evolve into. There are several things we need to consider when picking a neural network design and what we want to achieve. A neural network [14, 15, 16, 17, 18] is generally comprised of 3 basic parts known as neurons, layers, and bias.

Neurons deal with numerous types of information to be processed. Each individual neuron must know how to handle these types of information: input values, weights and bias, net sum, and an activation function. Although a neuron is just a small portion, it is critical to have accurate processing of data.

Layers are important component in a neural network. There is a minimum of three layers: input, hidden, and output. Each layer must handle information being fed forward to create an expected answer. Starting with the input layer, it is critical to have information prepared accurate and normalized properly. This will lower the chance of unexpected results. The hidden layer is particularly unique, as there can be multiple depending on the complexity of processing. When more than one hidden is present, each layer will feed forward as normal and additional processing for back-propagation is acceptable. After information has traversed through the neural network layers, the output is equally important to have accurate results.

Bias has enough worth to the input during execution. Each layer provides a heavier weight to the neurons if preprocessed data is activated as such. The bias is known as a constant in the network with a predefined value to allow accuracy towards a specific answer. Not all inputs to the network will require a bias to be active.

Neural networks can take on multiple forms and complexity. With the flexibility implemented into this network, we are capable of quick changes for testing purposes. This feature will allow modifying the number of hidden layers, several neurons per layer and the number of cycles (epochs) the network will iterate through for learning. Before testing or even training a neural network, having an idea of what the overall goal of the network is required. For instance, dealing with web proxy logs and requested websites, having a neural network determine whether if a website is categorized as good or bad will be the overall question.

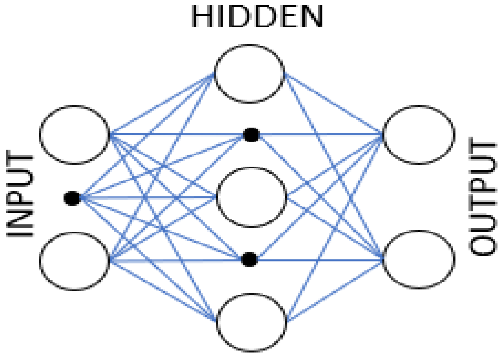

Figure 1. Neural Network with 3 layers

Figure 1 presents the layout design, which is the baseline for our analytics engine. The general scheme of the neural network used in our work has 3 layers: one input, one hidden, and one output. Each of these layers is handled differently as learning takes place. The input layer is predetermined with 32 neurons, a single hidden layer with 15 neurons and an output layer with two neurons or a calculated weight to determine if the suggested input was good or bad.

Analytics engines are a type of service which provides a complete picture of the information which is analyzed. With the amount of information collected and generated, we will utilize the analytical techniques known as search and knowledge discovery and stream analytics. These techniques will enable this system to define a robust information set to be displayed. On a broad scale, data analytics technologies and techniques provide a means of analyzing data sets and drawing conclusions about them to help organizations make informed business decisions. Business intelligence queries answer basic questions about business operations and performance. Big data analytics is a form of advanced analytics, which involves complex applications with elements such as predictive models, statistical algorithms and what-if analyses powered by high-performance analytics systems [2, 20, 21]. The analytics engine is the main process running for the environment. After the initial start of a users' account, it will begin to create a baseline of general data and information flow. This baseline is determined by a default timeframe and will auto update after a user invokes an update or the main analytics engine does the periodic neural network training. This same information will be used to create a general-purpose database to store heuristics information. Any type of information outside the parameters of normal determined by both analytics engine and user input will create alerts for users.

III. Neural Network Layout and Design

The overall question needs to be answered if a website is determined good or bad with some automation. This process is only a small piece of the entire analytics engine. Although the input can be viewed as one of the most important components as it's the main basis for how the network will determine the outputs. Before any machine learning can take place, there must first be some pre-processing of text. A total of 36 websites collected, with half being dedicated to determining what is identified as either known good or bad content. As training takes place, we will use 32 websites to be utilized while the additional four are strictly for testing once completed. Python was the programming language used to create the text parser and proxy server. There are a number of modules provided in python to handle language processing. One is the module called natural language toolkit "nltk". Due to the nature of this work, natural language processing is a critical aspect. This module creates a starting point for training the network. The bag of words approach will be taken to help determine the bias for input neurons. The bag of words will be handled by taking every single unique word from all training data. This dictionary will be categorized by good or bad type content. Here is a sample bag of words model:

```
bow = {
        “good”: [
                “word1”,
                “word2”,
                “word3”
        ],
        “bad”: [
                “1word”,
                “2word”,
                “3word”
        ],
        “count”: {
                “good”: 1323,
                “bad”: 234
        }
}
```

Each word category will contain the blueprints for determining the initial weight. Both the bag of words dictionary and each training document will have been normalized when processed through python's nltk module. Disregarding any word smaller than 3 and larger than 15 characters. After all training documents are parsed and the bag of words is created, the hidden layer will start to run through it's learning algorithms per neuron. A single neuron must have the functionality of knowing each connection to and from it, calculate weights, and the bias it's been presented. Weight calculation is also known as a summing function. Figure 2 shows the operations of a single neuron.

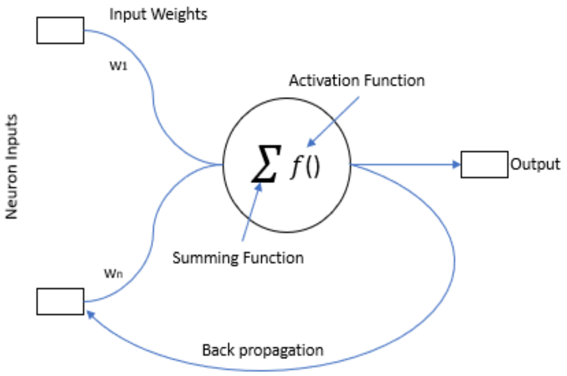

Figure 2. Operations of a single neuron

For each input connection, there is a weight associated with that specific connection. When a neuron is activated, the state is computed by adding the inputs multiplied by its corresponding connection's weight [4]. A detailed look at the summation function and adding bias before activation:

$$n = \sum_i w_i I_i + b \qquad (1)$$

The values are as follows:
n = current neuron.
w = weight from previous neurons input
I = previous set of inputs
b = bias towards a specific category

Researchers have found out that a neural network using Rectified Linear Unit (ReLU) function, trains faster than other non-linear functions like sigmoid and tanh without a significant drop in accuracy. So, the ReLU function is one of the most important activation functions [4]. A detailed look at the activation function which is ReLU:

$$f(x) = max(0, x) \qquad (2)$$

This function was selected to be the activation due to the speed and the lack of using negative numbers. After looking at the difference between ReLU and the sigmoid activation functions, it was determined ReLU to be faster in processing when the number of networks and neurons are increased.

During the training process there were many configurations used for the neural network. Finding the optimal number of neurons with epochs was a challenge. As defined earlier there was 1 input layer, 1 hidden layer and 1 output layer. This layer neural network had functionality to change the number of neurons in the hidden layer for testing purposes. The default output matrix will be 1 x 2. The output shows 1 neuron with two weights for each category. We have defined an error threshold of 0.2 which will drop any final output to 0 due to the low average. With the input matrix we have a 32 x 2246. This shows 32 neurons one per parsed document. Each neuron will have information regarding the binary sequence of values compared to the previously defined bag of words per category. After training we conducted testing against the set aside documents to check the accuracy of the neural network. Below are the results from tests:

<u>TEST #1</u>
Hidden layer neurons: 20
Epochs: 20000
Alpha (Error Threshold): 0.2
Processing Time: 29s

<u>Testing Category Weights:</u>
- GOOD: 0%
- BAD: 0%

This test group returned no value to the neural network and it was probably over trained with the number of epochs compared to the amount of hidden layer neurons.

<u>TEST #2</u>
Hidden layer neurons: 100
Epochs: 20000
Alpha (Error Threshold): 0.2
Processing Time: 148.45s

<u>Testing Category Weights:</u>
- GOOD: 0%
- BAD: 0%

This test group also returned no value. After picking up the number of hidden layer neurons, with the belief it might help during training, nothing valid from the test documents.

TEST #3
Hidden layer neurons: 10
Epochs: 5000
Processing Time: 11s

Testing Category Weights:
- GOOD: 81%
- BAD: 75%

This test group had a smaller number of neurons and epochs. This had better results during testing. After more research was conducted the conclusion to have a much larger training data set will provide a much better value during the testing phase.

IV. Service Components

There are several services created to assist the machine learning and enable users to view their content or settings. These services are defined as a web server, application program interface "API", mobile application, and proxy server. Each of these services has a critical role to achieve full capabilities.

The proxy server is the first interface users will interact with when requesting websites. A proxy server sits in between clients and external servers, essentially pocketing the requests from the clients for server resources and making those requests itself. The client computers never touch the outside servers and thus stay protected from any unwanted activity. A proxy server usually *does something* to those requests as well [6]. This is a normal proxy with added features to help the general home user. Almost all proxies come with a logging capability and a way to track which user requested which websites. The proxy also has these capabilities, with the intent to create analytics out of recorded logs of user activity. Figure 3 shows the general flow of information for the proxy.

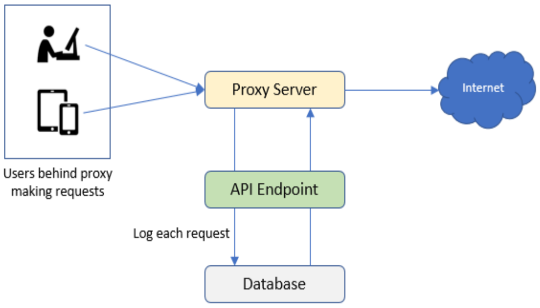

Figure 3. General Flow of Information for the Proxy Server

Along with the normal capabilities of a proxy, it will handle HTTP and HTTPS connections. The user will have some initial setup and configuration settings to do. There will be a default configuration created upon the first run, which has the rules to help secure a home network for children and young teenagers. After the setup is completed the user will be able to use either the mobile application or website to further configure the proxy and enable rules they wish to use within their household.

The web server is integrated with a server-side API for mobile and remote web requests. Think of the World Wide Web (and of any other RESTful API) as a technology stack. URLs are on the bottom; they identify resources. The HTTP protocol sits on top of those resources, providing read access to their representations and write access to the underlying resource state. Hypermedia sits on top of HTTP, describing the protocol semantics of one particular website or API [5]. It is designed to present users informational pages, which are engineered to grasp the users' attention and provide meaningful content. This content is dynamic due to the constant change of information submitted by users. Figure 4 shows the general layout for the webservers architecture.

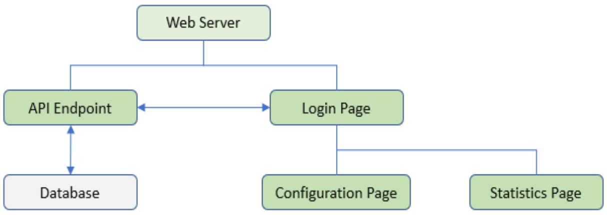

Figure 4. General Layout for the Webservers Architecture

An initial login page will be provided before any user can configure the proxy settings. Once completed there will be an admirative account defined for a household. The server will display traffic traversing the proxy and analyzed by a heuristics engine. The web server will have direct access to a database server through an API. Database changes will be submitted through HTTP POSTs only. Any information requested will be through HTTP GET requests. The web server will provide charts regarding proxy usage. The charts are categorized by:

- Websites visited
- Access by hour of day
- Access by day

Users will also be able to add, remove, or modify existing websites accessed. When machine learning is conducted on new websites and it has been determined as unknown or unable to categorize, the server will get a weighted sum of each category for users' viewing. A user can allow or deny suggested categories. These changes are taken into effect once updated. The API endpoint is an extension of the web server and will generate tokens to access user information. These tokens will be rotated out on a per-login basis. This endpoint is primarily used to provide information for any authenticated user browsing the website or on a mobile device.

Mobile applications are designed to provide the general population a way to view information while either at home or work. Some mobile applications are not elaborate by nature of the devices they are installed onto. Figure 5 is a basic layout and information flow.

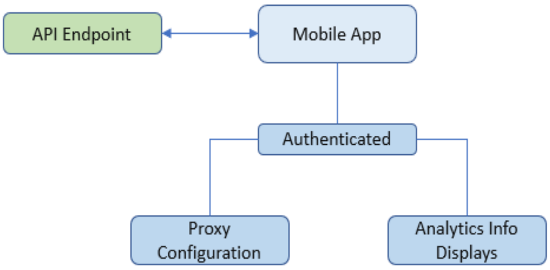

Figure 5. General Layout for the Mobile Architecture

The mobile application is an extension of the web server as it will show nearly the same information. Some of these differences do not require as much computing power to help the mobile device render with speed. All authenticated users must log in using OAuth with google play services. The server will generate time-based tokens which are unique to each session. To reduce the amount of traffic delivered, the analytics will only show small amounts of data at any given time. Just like the web server, charts will be displayed to the mobile application. With the limiting processing capability charts will be tailored to show:

- Activity by hour of day
- Activity by day of week

The administration section of the mobile application has the same capability as the web server. The main account will be able to modify what configurations were setup initially from login. The ability to make changes wherever and whenever is provided by the mobile application. Tailoring the account to a specific user allows for each individual approved device to be monitored. Each device allowed will follow the rules from allowed, denied or suggested tables within the API database. Having the same API creates stability across the project and there will be a smaller room for error when programming.

The last component to enable the web server and mobile application is a backend database server. Database services are designed to hold large amounts of relational data sets. There are two types of databases to be utilized: Relational and Non-Relational. A relational database (RDB) is a collective set of multiple data sets organized by tables, records and columns. RDBs establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data searchability, organization and reporting. RDBs are Structured Query Language (SQL), which is a standard user application that provides an easy programming interface for database interaction [3]. Non-Relational database provide information in a non-structured form for querying. Data-sets can be defined as a key, value pair. Figure 6 shows a general layout of the database services.

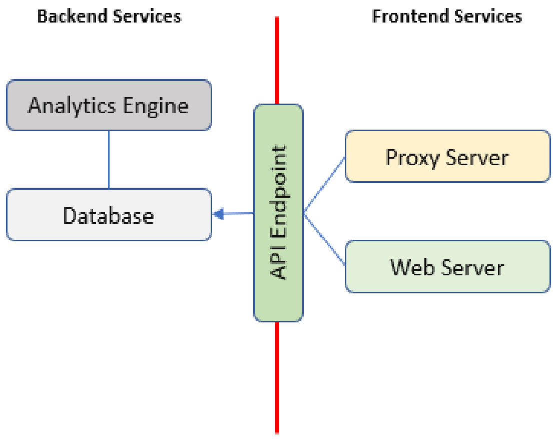

Figure 6. General Layout for the database services

The database will be used to store and retrieve user specific data and supply an interface. This user specific data is tailored to help configuration settings are saved and queried when needed. There are also non-user specific tables and databases which are required for the machine learning synapses. This data and other backend services are required to save their configurations to a more secured portion of the database. Only database administrators can access these tables.

V. Hardware Specifications

Cost effectiveness is the model for consumers, a Raspberry Pi 3 "Pi3" has enough processing power to execute the proxy server as a multi-module software package. Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1Amp. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally powered USB hub [4]. The use of computers or mobile devices are available for each customer to utilize. This will allow to use the older versions of hardware. Updating the system to minimum requirements for desktop or laptop users might be needed to render or execute all aspects of the accessed services. Users are encouraged to upgrade systems as problems may arise. If minimum requirements are not met, then not all aspects of the server services might be function. Although a Pi3 might not be able to handle a large households worth of traffic, but it will handle a small amount until a larger device is discovered to handle the network throughput.

VI. Conclusion

A proxy server powered not only by normal rules and regulations, but also by a machine learning neural network shows the capability to implement such a device for the average home user. With correct functionality built into each service of the proxy and back-end services, users are able to configure settings, save permissions, and provide a tailored profile for children or young teenagers to follow. With a machine learning aspect to this, any new or unknown website viewed will be checked by the neural network prior to allowing access. This will cut down the amount of time it will take to check every single website other users are trying to view. This service is meant to be a cost-effective approach to provide security for any household no matter the budget constraints that might be present.

References

[1] Richard Neapolitan, Xia Jiang. Contemporary Artificial Intelligence. August 2012. p. 4

[2] Margaret Rouse. Big Data Analytics – Definition "Whatis.com". http://searchbusinessanalytics.techtarget.com/definition/big-data-analytics. March 2017.

[3] Techopedia. What is a Relational Database (RDB)? – Definition from Techopedia. https://www.techopedia.com/definition/1234/relational-database-rdb

[4] Power Supply. (raspberrypi.org). https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md

[5] Mike Amundsen, Sam Ruby, Leonard Richardson. RESTful Web APIs. September 2013. ch. 11.

[6] Mike Meyers, Jonathan S. Weissman. Mike Meyers' CompTIA Network+ Certification Passport, Sixth Edition (Exam N10-007), 6th Edition. July 2018. obj. 2.3

[7] Luotonen, A., & Altis, K. (1994). World-wide web proxies. Computer Networks and ISDN systems, 27(2), 147-154.

[8] Weaver, N., Kreibich, C., Dam, M., & Paxson, V. (2014, March). Here be web proxies. In International Conference

on Passive and Active Network Measurement (pp. 183-192). Springer, Cham.

[9] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

[10] Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning. Neural and Statistical Classification, 13.

[11] Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

[12] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[14] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[15] Psaltis, D., Sideris, A., & Yamamura, A. A. (1988). A multilayered neural network controller. IEEE control systems magazine, 8(2), 17-21.

[16] Haykin, S. (1994). Neural networks (Vol. 2). New York: Prentice hall.

[17] Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (1996). Neural network design (Vol. 20). Boston: Pws Pub..

[18] Anthony, M., & Bartlett, P. L. (2009). Neural network learning: Theoretical foundations. cambridge university press.

[19] Göranzon, B., & Florin, M. (Eds.). (2012). Artifical Intelligence, Culture and Language: On Education and Work. Springer Science & Business Media.

[20] Zikopoulos, P. C., Eaton, C., DeRoos, D., Deutsch, T., & Lapis, G. (2012). Understanding big data: Analytics for enterprise class hadoop and streaming data (p. 176). New York: Mcgraw-hill.

[21] LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2011). Big data, analytics and the path from insights to value. MIT sloan management review, 52(2), 21.

Mark Maldonado is a graduate student at St. Mary’s University, San Antonio, Texas enrolled in the MS Cybersecurity degree program. He received his B.Sc. in Information Systems Security from American Military University, West Virginia. Mark is a United States Air Force veteran of 15 years working as an analyst for ten years. The last five years were dedicated to a software development work role

Ayad Barsoum is an Associate Professor in Computer Science Department at St.Mary's University, San Antonio, Texas. He is the Graduate Program Director of MS in Cybersecurity. Dr. Barsoum received his Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Waterloo (UW), Ontario, Canada in 2013. He is a member of the Centre for Applied Cryptographic Research at UW.

He received his B.Sc. and M.Sc. degrees in Computer Science from Ain Shams University, Cairo, Egypt, in 2000 and 2004, respectively.

At the University of Waterloo, Barsoum has received the Graduate Research Studentship, the International Doctoral Award, and the University of Waterloo Graduate Scholarship. Dr. Barsoum has received “Amazon Web Services in Education Faculty Grant” for funding his research and teaching through using Amazon cloud infrastructure