NORTHWEST NAZARENE UNIVERSITY

Development and Implementation of Neural Network on IT Audit Logs

THESIS
Submitted to the Department of Mathematics and Computer Science
In partial fulfillment of the requirements
For the degree of
BACHELOR OF SCIENCE

Jarrett M. Sheehan
2021

THESIS
Submitted to the Department of Mathematics and Computer Science
In partial fulfillment of the requirements
For the degree of
BACHELOR OF SCIENCE

by
Jarrett M. Sheehan
2021

Development and Implementation of Neural Network on IT Audit Logs

Author: Jarrett Sheehan

Jarrett Sheehan

Approved: Kevin S McCarty

Dr. Kevin McCarty, Department of Mathematics and Computer Science Faculty Advisor

Approved: Christopher Lloyd

Christopher Lloyd

Second Reader, Operations Manager of Information Technology NNU ITS

Approved: Barry Myers

Barry L. Myers, Ph.D., Chair,Department of Mathematics & Computer Science

ABSTRACT

Application developed to detect anomalies from data generated by a network.

SHEEHAN, JARRETT (Department of Mathematics and Computer Science),
MCCARTY, DR. KEVIN (Department of Mathematics and Computer Science)

Companies now are constantly flooded with data. Things such as breaches can get lost in that data flow and without an appropriate solution it could cause a breach to go unnoticed for up to six months. Northwest Nazarene University alone generates two gigabytes of data per day. One solution to this problem can be hiring people to sift through these logs manually. However, that process could take three people working full time to keep up with that flow of data. This project intends to train a neural network to automatically parse through the network logs and notify the network team of any anomaly it comes across.

Acknowledgments

I would like to thank my parents for the constant patience and support during the stressful times I spent working on this project. I would also like to thank Marshall Schultz from Northwest Nazarene University's IT Network Operations Team, working with me to get me all of the data I needed to train my neural network, especially during one of the most frantic times the network department has seen. Thirdly I want to thank Robert White and Gabriel Johnson for being my rubber duckies while I was struggling to get aspects of my project to work. Lastly, I would like Dr. Kevin McCarty for the guidance and instruction on this project.

Table of Contents

Table of Figures

Introduction

Cyber Security is one of the most important industries today. This is because every day governments, businesses, and people are being targeted by attackers. In 2019 the Maryland Department of Labor was breached[1], and the names and social security numbers of 78,000 people were taken by the attackers. The year before an exploit in Facebook's code caused 50 million user accounts to be compromised. Again, it is not just large businesses and governments, but people too being targeted. People are falling for phishing scams leading their bank account details to be exposed or causing ransomware to be installed on their computers. By this year it is expected that specialists covering Cyber Security will be predicting within five years, spending at over 10.5 trillion dollars annually[2]. Security specialists need all the help that they can get as, in most cases, it can take up to 6 months to detect that a breach has occurred. This is due to the immense amount of log files that are generated in a given network. Northwest Nazarene University alone generates about 2 gigabytes of data per day in log files. To physically parse through this amount of data, it would take three men working eight hours a day to keep up with the data flow. So oftentimes the logs are ignored until something goes wrong, causing a worker to go through those logs until they find where things went wrong. This is too much data for someone to reasonably sort through.

Background

In the past people have used neural networks to sort through this data and find an anomaly before it leads to something worse. A neural network is a series of

---

[1] "Response to Recent Cybersecurity Incident - Maryland Department ...." https://www.dllr.state.md.us/datahotline/. Accessed 6 May. 2021.
[2] "Cybercrime To Cost The World $10.5 Trillion Annually By 2025." 21 Feb. 2021, https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/. Accessed 6 May. 2021.

algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. These can be trained on data that has been pre-determined to be “clean” or free from anomalies. An anomaly is something that deviates from the given set. For example if you have a set of numbers and they all range from one to ten and inside that set is a number of three hundred. That three hundred would classify as an anomaly. Tensorflow is also commonly used, as a platform, when creating a model for a neural network. Tensorflow is an open-source library created by Google’s Brain team and it was built to implement machine learning and neural network models.

Implementation

In this project, several steps had to be taken. The first step in any project is research. The research ended up taking quite a long time, as the majority of this project were processes and programs that I was not familiar with. Several decisions were made in this step that were carried throughout the project. The first decision was deciding on what programming language the neural network would be made in. This project would not be made entirely from the ground up, only trained from an already built framework. Given the purpose of this project, there are only two programming languages commonly used in training neural networks, R and Python. Python was the easy choice for language, as R is mostly a language for computing statistics and is extremely inconsistent as far as algorithm implementation goes as each algorithm made in R is highly specialized for its given task. Several defining features went into the choice for Python but one of the main features was the number of imported libraries contained in the programming language.

The second step in this project was creating the metadata used in the development of the parser and the neural network. A total of nine gigabytes of logs were obtained from NNU's IT Network Operations Team. From that data, a consistent formula for how the data is broken up was created.

```
2020-09-01T13:19:18-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the running state.
2020-09-01T13:22:19-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the stopped state.
2020-09-01T14:19:18-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the running state.
2020-09-01T14:22:19-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the stopped state.
2020-09-01T14:48:40-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Background Intelligent Transfer Service service entered the running state.
2020-09-01T14:48:40-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The start type of the Background Intelligent Transfer Service service was changed from demand start to auto start.
2020-09-01T14:50:41-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The start type of the Background Intelligent Transfer Service service was changed from auto start to demand start.
2020-09-01T14:50:41-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Background Intelligent Transfer Service service entered the stopped state.
2020-09-01T15:19:18-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the running state.
2020-09-01T15:22:19-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the stopped state.
2020-09-01T16:19:18-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the running state.
2020-09-01T16:22:19-06:00 DC01.nampa.nnu Service_Control_Manager[604]: The Network Setup Service service entered the stopped state.
```

Figure 1 - Network Log Data Example

Several consistencies were found in the logs that helped break it up into portions. The data was broken up into four parts: date and time, network server, the service, and the service action.

The third part of this project was building out the parser. Taking the metadata that was formed by the logs a parsing program was created in Java using the IDE Intellij. Java was chosen because building the parser-based on the indexOf() function was an extremely simple solution as the parsed data would not need to be saved, but immediately printed or put into a database. The indexOf() method stores a character value or a substring. In this project frequently used identifiers were marked down. For example at the end of every date and time string in the network logs was "-6:00". This was determined to be the time zone of the servers and was used to mark the end of the date and time string. The parser would then search the line for the first mention of "-06:00" and store the location as a single number. The same process was followed for indexing where the data for the server was stored and service were stored. Inside the logs, the server name always ended with ".nnu" and the service name always ended with

“]: “. This information was stored and when run the parser would index the location of those identifiers. The next step for the parser was to break up the date and time. As the date and time had a standardized output it was fairly easy to develop the process. A substring was created for both pieces of metadata, where the date always went from character 0 to character 10 and time started at character 12 and went to character 19. This was the same for every log. Using the indexes created, more substrings were created.

When creating a substring it was important to understand that when indexing it will always take the location of the first character in your saved string. So when a specific substring is being created the length of the indexed string must be added to the index to start the substring at the end. Also, in some cases, it was necessary to add one to the end as there would always be a space between certain pieces of data. For example, when parsing out the server’s substring, the index of the time zone would be taken, having the length plus one being added to it, allowing the substring to start right where it needs to. Then the index for the server is called adding the length of the server’s index, allowing the substring to contain the entire piece of metadata. Shown below are Figure 2 and Figure 3. Figures 2 is the code that was made to parse all of the data. Figure 3 is a sample of what the code would output when run.

```
String timeZone = "-06:00";
String server = ".nnu";
String action = "]: ";

String dateTime = line.substring(0, 19); //datetime example 2020-09-11T11:59:53-06:00
String date = dateTime.substring(0, 10);
String time = dateTime.substring(11, 19);

int dateTimeSearch = line.indexOf(dateTime);
int timeZoneSearch = line.indexOf(timeZone);
int serverSearch = line.indexOf(server);
int actionSearch = line.indexOf(action);

String dateTimeSub = line.substring(dateTimeSearch + dateTime.length(), timeZoneSearch);
String serverSub = line.substring(timeZoneSearch + timeZone.length() + 1, serverSearch + server.length());
String actionSub = line.substring(serverSearch + server.length() + 1, actionSearch + action.length());
String msgSub = line.substring(actionSearch + action.length());



System.out.print(date + "\n");
System.out.print(time + "\n");
//System.out.print(dateTimeSub + "\n");
System.out.print(serverSub + "\n");
System.out.print(actionSub + "\n");
System.out.print(msgSub + "\n\n");
```

Figure 2 - Code for Log Parser

```
Date - 2020-09-01
Time - 13:19:18
Server - DC01.nampa.nnu
Service - Service_Control_Manager[604]
Action - The Network Setup Service service entered the running state.
```

Figure 3 - Example Parsed Data

The fourth step of this project was to build out the database. The database was built from Microsoft SQL Server Management Studio. The database houses a table and inside that table is a set of five columns. Each column for the table represents the parsed data from the logs.

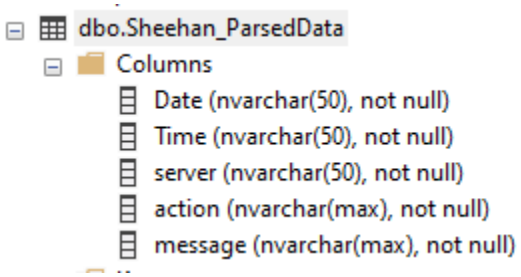

Figure 4 - Database Table

The fifth step was adding the ability to input data into the database that was created using the parser. The parser connects to the database through Java's SQL Development Kit. This allows someone to connect to a SQL database through a given Java IDE and create commands also known as stored procedures inside the Java code. The main function of this code was to create a connection, then create a stored procedure that stored the data that was parsed.

```
private static final String JDBC_DRIVER = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
private static final String JDBC_URL_START = "jdbc:sqlserver:";
private static String _serverName = "SQLServer";
private static String _databaseName = "JarrettSeniorProject";
private static String _userName = "Username";
private static String _password = "Password";

public static String getServerName() { return _serverName; }
public static void setServerName(String serverName) { _serverName = serverName; }
public static String getDatabaseName() { return _databaseName; }
public static void setDatabaseName(String databaseName) { _databaseName = databaseName; }
public static String getUserName() { return _userName; }
public static void setUserName(String userName) { _userName = userName; }
public static String getPassword() { return _password; }
public static void setPassword(String password) { _password = password; }
public static Connection GetConnection(String serverName, String databaseName, String userName, String password) {
    String jdbcDriver = JDBC_DRIVER;
    String jdbcURL = JDBC_URL_START + "//" + serverName + ":1433;databaseName=" + databaseName + ";UserName=" + userName + ";Password=" + password + ";";
```

Figure 5 - Connection Code

```
Statement statement = conn.createStatement();
statement.executeUpdate( sql: "INSERT INTO Sheehan_ParsedData" + "VALUES (date, time, serverSub, actionSub, msgSub)");
```

Figure 6 - Input Code

The sixth step was setting up the neural network and training it. Due to the research performed, TensorFlow is the best platform available to train a neural network. TensorFlow being open-sourced allows it to be changed to fit whatever purpose someone will need. Since there will be complex computations taking place, having a platform that can utilize the resources appropriately and efficiently is also important. Now that a platform was chosen the next step was to find the best model for machine learning. An autoencoder became the choice after research. An autoencoder (AE) is a model that takes data that has already been determined to not include any outliers and tries to replicate that data by following a list of attributes that it creates. It does this by encoding the data or compressing it down to a bottleneck. Once the data is compressed into the bottleneck it reconstructs that data, decoding the compressed data. Once an AE is built and trained it will know what the “good” data looks like, having built a model based on that data. The model can now be compared to data that could be “good” or “bad” and the neural network will be able to pick out anomalies that may present themselves. Seldon’s variational autoencoder (VAE) became the specific choice to train the model for this project with. A VAE is a type of autoencoder that instead of having the encoder output a single output, outputs two. One output is a median value and the other output is a standard deviation. This provides a wider range of values that can be compared to new data. This is important as the data taken from the network logs has a wide range of what classifies as normal, as it would be trained based on several different servers. Seldon’s VAE was chosen specifically because they have already made a lot of the code necessary to build a neural network.

Results

I was able to build an efficient parser to break up the data from logs and this parser is capable of uploading the parsed data to a database where it could later be pulled and tested against the model. The parser was able to break down the nine gigabytes of data in roughly seven minutes, that means the parser is far efficient enough to handle a steady stream of data coming from the network. Due to time constraints I wasn't able to complete the neural network, but the amount of work done and research done will serve as a jumping off point for a later project down the road.

Future Work

Two future steps can be taken from this project. The first step is the construction of the neural network. The second step that can be taken is building an alert for the network team so that when the neural network detects an anomaly it can send off a message specifying the server and the service that is sending anomalies.

Conclusion

The Network Operations Team provided a large dataset for training and parsing. I was able to build a parser and upload the results into a database and that parser is capable of handling the daily flow of data. Unfortunately I ran into difficulties along the way, such as not being able to connect my parser to the database I had built to upload the data and not completing the neural network due to time. Programming the parser and the ability to add to a database grew my knowledge of Java and programming immensely. Since much of this project was me working by myself it taught me a lot about self-motivation and pushed me to keep trying despite how difficult the task may have seemed. There are many research techniques that I learned while progressing in the project I would not have learned otherwise. Such as figuring out what is most important

for a project and finding what I need to fill those purposes. I was able to take concepts I had learned in classes and apply them appropriately to seemingly different applications.

References

Chen, J. (2020, December 23). *Neural Network*. Website. https://www.investopedia.com/terms/n/neuralnetwork.asp

Looveren, A.(2019, February 08). *Outlier Detection with Seldon*. Website. https://www.seldon.io/outlier-detection-with-seldon/

Maryland Department of Labor (2019). *Response to Recent Cybersecurity Incident*. Website. https://www.dllr.state.md.us/datahotline/

Sausalito, Calif. (2020, November 13). *Cybercrime to Cost the World $10.5 Trillion Annually by 2025.* Website. https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016

Stewart, M.(2019, April 14). *Comprehensive Introduction to Autoencoders*. Website. https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368