An Analysis of Neural Network Performance at Malicious Traffic Detection

Aidan Price
MS Business Analytics Program
University of Colorado
Boulder, CO, USA
aipr5269@colorado.edu

Chris Blair
MS Business Analytics Program
University of Colorado
Boulder, CO, USA
chbl7904@colorado.edu

Mike McCormick
MS Business Analytics Program
University of Colorado
Boulder, CO, USA
mimc2735@colorado.edu

***Abstract*—This document reports preliminary results for a comparative analysis of multiple neural network models used to detect malicious network traffic in the CTU-13 data set [1]. A Streaming Analytics Machine (SAM) is used to generate features from the data, which are then used to train a series of neural network models. Each model is evaluated with standard classification metrics to measure performance at malicious traffic detection.**

I. Introduction

In today's world of rapid growth in internet use, more people and devices are connecting to the internet than ever before. This growth in the Internet has been matched by growth in malicious network activity, in terms of both frequency and impact [2]. In this paper, we expand upon the CU Heterogeneous Information Network and Streaming Analytics Machine/Streaming Analytics Language (SAM/SAL) projects by applying deep learning techniques to identify malicious network traffic in the CTU-13 data set [3][4]. Our method compares several deep learning techniques to model the complex patterns of network traffic data and classify malicious traffic. Our preliminary results are mixed. Several models had difficulty modeling the data while the Long Short Term Memory (LSTM) model showed that neural networks can successfully be used to identify malicious traffic in the CTU-13 data set. Our next steps will be to improve the performance of all models on the CTU-13 data set and evaluate the generalizability of our neural networks by applying them to other real world data sets.

II. Definitions

A. Neural Networks

- Artificial Neural Network (ANN) - a machine learning model inspired by the structure of the human brain that processes information by passing data inputs through multiple hidden layers to output a prediction [5].
- Convolutional Neural Network (CNN) - a neural network used primarily for image detection and classification, based on layers that mimic how the human brain perceives image features and patterns [6].
- Long Short Term Memory (LSTM) - a neural network commonly used for natural language processing tasks that can store data in both short and long term "memory" in order to predict sequences of data [7].

B. Metrics

- Accuracy - the overall fraction of predictions that a model gets right. It is measured as the number of correct predictions divided by the total number of predictions, or as true positives plus true negatives divided by all predictions.
- Precision - measures which proportion of positive identifications were actually correct, and is defined as true positives divided by true positives plus false positives. A precision of 1 means that no false positives occurred.
- Recall - measures which proportion of actual positives were identified correctly in the first place, and is measured as true positives divided by true positives plus false negatives. A recall of 1 means that there were no false negative identifications.
- Receiver Operating Characteristic (ROC) - a graph that shows the performance of a classification model at all classification thresholds, plotting False Positive Rate versus True Positive rate at each threshold.
- Area under the ROC Curve (AUC) - measures the entire two-dimensional area under the ROC Curve. It provides an aggregate measure of performance across all possible classification thresholds. A model whose predictions are completely right would have an AUC of 1, while a model whose predictions are completely wrong would have an AUC of .5. AUC is scale- invariant and it measures the quality of the model's predictions irrespective of what classification threshold is picked.

III. Related Works

Application of deep learning techniques to malicious network traffic classification is an area of active and wide-ranging study. The ever growing increase in the number and complexity of cyber threats has resulted in a growing interest in the application of deep learning techniques for the detection and classification of malicious network traffic. The ability of deep learning models to autonomalously learn complex patterns and features from data make them a promising approach. In this section, we review some of the existing literature on the use of deep learning for malicious traffic classification, focusing on ANN, CNN, and LSTM models.

In their 2020 paper, Yiming Zhang, Yujie Fan, Shifu Hou, YanFang Ye, Xusheng Xiao, and Pan Li detail how they used Deep Neural Networks (DNN) with observational cyber-guided knowledge modeled by structural heterogeneous information network (HIN) [8]. HIN was used to create domain knowledge on 5 social coding relationships. These include: user-fork repository, user-comment repository, user-star repository, user-contribute repository, and repository-have-file. When compared to BoW-DNN, M2V-DNN and M2V-SVM, their proposed method nicknamed GitCyber outperformed all of these models. Since the effectiveness of neural networks has been proven here we will combine techniques mentioned above with SAM/SAL to evaluate the potential capabilities for our project.

Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng explore malware traffic classification using neural networks in their 2017 paper [9]. They proposed a new taxonomy of internet traffic classification from an artificial intelligence perspective, leading to a new malware traffic classification method that used convolutional neural networks to analyze traffic data taken as images. They validated their method using two different scenarios.

In their 2018 paper, Nga Nguyen Thi, Van Loi Cao, and Nhien-An Le-Khac explored the application of LSTM Recurrent Neural Networks (RNN) to anomaly detection in computer network traffic [10]. Their findings demonstrated the viability of applying LSTM neural networks to detecting malicious network traffic but also highlighted the importance of data preparation and input structure to achieve useful results. Similarly, our approach will leverage SAM/SAL to develop features that can be effectively used with a LSTM neural network.

IV. Data Preparation

Features for model development are extracted from the CTU-13 dataset using the Streaming Analytics Language (SAL) and Streaming Analytics Machine (SAM), which creates text files for the thirteen scenarios [11].

Due to processing constraints, the five smallest scenarios (5, 6, 7, 11, 12) were selected for initial model development. Each scenario was split into stratified training (80%), validation (10%), and test (10%) sets to ensure an equal distribution of the minority class across all data sets. Individual neural network models were then trained and evaluated on each scenario.

V. Neural Network Development

Each model (ANN, CNN, LSTM) was developed using the Keras open-source software library, and written in Python in Jupyter Notebooks.

A. Artificial Neural Network (ANN)

The developed ANN model consists of an input layer, an embedding layer, 3 hidden layers with 20% neuron dropout after each, and a dense layer with a binary input. The input layer accepts variable length inputs of integer sequences that are passed to the subsequent embedding layer for further processing. The embedding layer converts the input integer sequences into fixed-size dense vector representation. The output of the embedding layer is then passed through three hidden layers, each with 8 neurons using ReLU activation, and dropout for each layer in order to mitigate potential overfitting of the training data. The output from the third hidden layer is then passed onto the dense layer with a Sigmoid activation function, which predicts the probability of the input sequence resulting in a benign (0) or malicious (1) output, with a probability of .5 or more of malicious output resulting in a malicious classification, and all others resulting in benign classification.

The model is trained using Adam optimization and a binary cross-entropy loss function, due to the binary output. Two epochs of training data are passed through the model with a batch size of 32 validated with the validation set before final evaluation on the test set.

B. Convolutional Neural Network (CNN)

While CNN's are most commonly used for computer vision and image classification, with one dimensional data it can be very effective for Time Series Classification and anomaly detection. The Convolutional Neural Network developed utilizes transfer learning by using an AutoEncoder with 9 layers: 5 Dense layers with a dropout layer between each. 4 of these layers (hidden) use a ReLU activation function while the last layer (output) uses a Sigmoid activation function for binary classification. The dropout layers are necessary to avoid overfitting within our model as it is a regularization technique that will randomly drop nodes so each update to a layer during training is performed with a different "view" of the configured layer.

This model also utilizes an Adam optimizer with Nesterov Momentum and uses binary cross-entropy for a loss function since we are looking at a classification problem. 8 epochs of training data are passed through the model with a batch size of 32 validated with the validation set before final evaluation on the test set.

C. Long Short Term Memory (LSTM)

Our LSTM model was modeled after an example neural network from the Keras website [9]. The developed LSTM model consists of four layers: (1) input, (2) embedding, (3) bidirectional LSTM, and (4) dense. The input layer accepts variable length inputs of integer sequences that are passed to the subsequent embedding layer for further processing. The embedding layer converts the input integer sequences into fixed-size dense vector representation. The output of the embedding layer is passed to two parallel bidirectional LSTM layers that capture the context and dependencies of the sequence in both directions. The first layer returns the sequence of outputs from each time step while the second layer returns only the final output. The final output from the second bidirectional LSTM layer is then passed to the dense layer with a Sigmoid activation function, which performs binary classification by predicting the probability of the input sequence being benign (0) or malicious (1).

The model is trained using Adam gradient-based optimization and a binary cross-entropy loss function. Two epochs of training data are passed through the model with a batch size of 32 and validated with the validation set before final evaluation on the test set.

VI. Preliminary Findings

The results of our development and application of ANN, CNN, and LSTM models to the CTU-13 data subset was decidedly mixed. The ANN and CNN models both performed poorly, with extremely low precision and recall values. The LSTM model performed extremely well with the exception of scenario 7, where the model failed to classify malicious traffic with any degree of precision or recall.

There are multiple potential reasons for this mixed performance. For the ANN and CNN models, our developed models were likely too simple and underfitted the data. Our future efforts will include exploration of additional layers and hyperparameter tuning to improve performance. In addition, for ANN specifically, we encountered issues with a known keras bug that resulted in our model having difficulty variable length sequences of integers. The low performance of LSTM on scenario 7 may be due to the parameters of that particular scenario. Scenario 7 had the second lowest number of netflows (114,078) with only a single bot. By contrast, LSTM performed well on Scenario 11, which had the smallest number of netflows (107,252) and three bots. There may not have been enough training data for the LSTM model to effectively train on scenario 7, resulting in the low evaluation metrics.

The below table highlights performance of each neural network on the selected scenarios:

Table 1 - Preliminary Results

Model (Scenario)	Accuracy	Precision	Recall	AUC
ANN (5)	0.99	0.00	0.00	0.50
ANN (6)	0.99	0.00	0.00	0.50
ANN (7)	1.00	0.00	0.00	0.50
ANN (11)	0.92	0.00	0.00	0.50
ANN (12)	0.99	0.00	0.00	0.50
CNN (5)	0.94	0.00	0.00	0.475
CNN (6)	0.94	0.00	0.00	0.475
CNN (7)	0.95	0.00	0.00	0.474
CNN (11)	0.88	0.00	0.00	0.475
CNN (12)	0.95	0.00	0.00	0.479
LSTM (5)	0.99	0.97	0.88	0.94
LSTM (6)	1.00	0.99	0.99	0.99
LSTM (7)	0.99	0.00	0.00	0.50
LSTM (11)	1.00	1.00	0.99	0.99
LSTM (12)	0.99	0.86	0.52	0.76

VII. Next Steps

The next steps in our analysis will be extensions of the work we have completed to date. We intend to train and evaluate our models on the full CTU-13 data set, both per individual scenario and by combining the scenarios into a single data set and rotating a hold out scenario for evaluation. We also intend to evaluate the generalizability of this method by applying our neural network models to other data sets like Aposemat IoT-23 and MTA-KDD-19.

VIII. Conclusion

The comparative analysis of ANN, CNN, and LSTM neural networks for detecting malicious network traffic in the CTU-13 data set provided valuable preliminary insights into the effectiveness of different deep learning approaches. The use of SAM and SAL to generate features from the data allowed for efficient training of the neural networks and the evaluation of each model using standard classification metrics revealed varying levels of performance. These preliminary results demonstrate the potential of neural networks for detecting malicious traffic and highlight the importance of continued research to improve the accuracy and reliability of these models with the ability to match the speed and scale of network traffic.

References

[1] Sebastian Garcia, Martin Grill, Jan Stiborek and Alejandro Zunino (2014). An empirical comparison of botnet detection methods. Computers and Security Journal, Elsevier. 2014. Vol 45, pp 100-123. http://dx.doi.org/10.1016/j.cose.2014.05.011

[2] IBM. (2022, July). Cost of a Data Breach Report 2022.

[3] Goodman, E. (2023). CU_HIN. Github Repository. https://github.com/elgood/CU_HIN

[4] Goodman, E. (2023). SAM. Github Repository. https://github.com/elgood/SAM

[5] IBM. What are Neural Networks? https://www.ibm.com/topics/neural-networks

[6] IBM. What is Deep Learning? https://www.ibm.com/topics/deep-learning

[7] Hochreiter, S. Schmidhuber, J. (1997). Long Short-term Memory. Neural Computation, 9(8), pages 1735-1780. https://www.researchgate.net/publication/13853244/

[8] Zhang, Y., Fan, Y., Hou, S., Ye, Y., Xiao, X., Li, P., … Xu, S. (2020). Cyber-guided Deep Neural Network for Malicious Repository Detection in GitHub. 2020 IEEE International Conference on Knowledge Graph (ICKG). https://doi.org/10.1109/ICBK50248.2020.00071

[9] Wang, W., Ming, Z., Zeng, X., Ye, X., Sheng, Y. (2017). Malware traffic classification using convolutional neural network for representation learning. 2017 International Conference on Information Networking. https://ieeexplore-ieee-org.colorado.idm.oclc.org/document/7899588/authorsauthors

[10] Nguyen Thi, N., Loi Cao, V. Nhien-An, L. (2018). One-class Collective Detection based on Long Short-Term Memory Recurrent Neural Networks. Cornell University. https://arxiv.org/abs/1802.00324

[11] Team, K. (n.d.). Keras Documentation: Bidirectional LSTM on IMDB. Keras. Retrieved March 7, 2023, from https://keras.io/examples/nlp/bidirectional_lstm_imdb/