Project - Network Intrusion Detection System
CMPT469 Deep Learning w/ TensorFlow

Kai Wong

11/13/17

1 Abstract

This paper highlights the results of using a deep learning model implemented with an auto-encoder and LSTM-RNN architecture for a network intrusion detection system (NIDS).

2 Introduction

Cybersecurity is a major and pressing topic that all companies face in modern day. Building a reliable network intrusion detection system provides a great way to increase the security of a companys assets. By harnessing the power of deep learning through the use of neural networks, we can build a model that will accurately detect and classify the different types of attacks a network may face, and in turn better protect a company. For our data, we will use the NSL-KDDCup99 dataset, which is an improved version of the KDDCup99 dataset (cont. background/related work). The original KDDCup99 dataset, from the 1998 DARPA Intrusion Detection Evaluation Program, was created for the task of building a predictive model capable of distinguishing between bad connections (intrusions), and good connections (normal). The labeled dataset includes four main categories of attacks: denial-of-service (a flood of packets are sent such that computing/memory resources arent available to serve authorized requests), user-2-root (the attempt to exploit susceptibilities in a system to achieve root privileges), probing (the attempt to examine a machine to determine susceptibility for exploitation), and remote-to-user (the attack in which an intruder sends packets to a machine to expose a machines vulnerabilities and achieve local user privileges); the goal is to classify an incoming connection as either normal or an attack and the attack type. In this dataset, the test data contains attack types not in training data, making this task more realistic, as intrusion experts believe that most novel attacks are variants of known attacks, and knowing the signature of known attacks can be sufficient to catch these novel attacks. Thus, the dataset contains 24 training attack types, with an additional 14 types in the test data only. The dataset comes with derived features that help in distinguishing normal connections from attacks.

3 Background/Related Work in NIDs

The majority of previous works demonstrate that building a model using a long short-term memory recurrent neural network yields great results on the KDDCup99 dataset. In this paper, we will go over a few of the past approached techniques in order to provide reasoning behind the architecture for our model.

3.1 NSL-KDDCup99 Dataset

The NSL-KDDCup99 dataset is an improved and reduced version of the KDDCup99 dataset [1]. The original dataset contained a lot of redundant records, both in the training and testing data, and this makes learning algorithms biased towards frequent attack records (i.e. DOS attacks in the KDDCup99 dataset), and leads to poor classification results for infrequent but harmful records. The NSL-KDD dataset eliminates

these redundant records, and also partitions records into various classification difficulty levels (based on the number of learning algorithms that could correctly classify the records), then selecting records by random sample from each level. These steps of processing makes the number of records in the NSL-KDD dataset reasonable for the training of a learning algorithm [1].

3.2 Auto-Encoders and Deep Auto-Encoders

It can be seen how other deep learning approaches such as the use of auto-encoders can be beneficial in a network intrusion detection system (NIDS). In developing an effective and flexible NIDS for unknown future attacks, proper feature selection is key and difficult, as attack scenarios are continuously changing and evolving, and the features selected for one class of attack may not work well for other classes of attack [1]. In addition, producing a large labelled dataset from raw network traffic data collected over a period or in real-time is difficult. Therefore, the use of autoencoders can be helpful in providing automatic and meaningful feature extraction from unlabeled data [1,3] and also reducing the dimensionality of the data. A deep auto-encoder is simply an auto-encoder with multiple hidden layers, and has been used to train models on the NSL-KDD dataset [2, 5].

3.3 Deep Belief Networks

This notion of a neural network discovering patterns within data autonomously is a technique known as unsupervised learning, such as is done with the use of autoencoders. A deep belief network is made up of these autoencoders (or also with Restricted Boltzmann Machines (RBMs), and an algorithm greedily trains the network layer by layer using unsupervised learning [2]. Using this method provides a network that is successful in providing accurate classification for network intrusion detection [1,2].

3.4 LSTMs and RNNs

There are many previous works that have successfully built accurate learning algorithms using the feed-forward LSTM-RNN model [4,5,6,7]. It can be seen that anomaly detection models should be built to remember information from a number of previous events; a single data point in a collective data anomaly may not be considered to be an anomaly unless considered with the occurrences of other single points [4]. Thus, it can be seen that the ability to learn from historical data is important in a network intrusion detection model. This is what the LSTM-RNN model is capable of, and it is seen that extremely high classification accuracies are reported using this model [4,5,6,7].

4 Methodology

We can see that the problem of network intrusion detection is a classification problem. With this in mind, and building off research previously done with network intrusion models, we can build an effective classification model using a combination of auto-encoding and a long short term memory recurrent neural network.

We will prepend the network with an auto-encoding layer, which will reduce the dimensionality of the dataset and provide useful feature extraction. This will better improve the training of the LSTM-RNN section of our model, as this pre-training has been shown to decrease test error. Other transformations made to the data will include the transformation of the symbolic data columns into a one hot encoding vector. This data transformation will be the first step in implementing this model.

The auto-encoding section of this model will be unsupervised, while the LSTM-RNN will be supervised.

5 Experiments

6 Discussion and/or Analysis

7 Conclusion

8 References/Bibliography

[1] Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam, A Deep Learning Approach for Network Intrusion Detection System, http://www.covert.io/research-papers/deep-learning-security/A%20Deep%20Learning%20Approach%20for%20Network%20Intrusion%20Detection%20System.pdf

[2] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Christos Tachtatzis and Robert Atkinson, Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey, https://arxiv.org/ftp/arxiv/papers/1701/1701.02145.pdf

[3] Zhanyi Wang, The Applications of Deep Learning on Traffic Identification, http://www.covert.io/research-papers/deep-learning-security/Applications%20of%20Deep%20Learning%20On%20Traffic%20Identification.pdf

[4] Loc Bontemps, Van Loi Cao, James McDermott, and Nhien-An Le-Khac, Collective Anomaly Detection based on Long Short Term Memory Recurrent Neural Network, https://arxiv.org/ftp/arxiv/papers/1703/1703.09752.pdf

[5] Manoj Kuma Putchala, Deep learning approach for intrusion detection system (ids) in the Internet of Things (IoT) network using Gated Recurrent Neural Networks (GRU), https://etd.ohiolink.edu/!etd.send_file?accession=wright1503680452498351&disposition=inline

[6] Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim, Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection, http://www.covert.io/research-papers/deep-learning-security/Long%20Short%20Term%20Memory%20Recurrent%20Neural%20Network%20Classifier%20for%20Intrusion%20Detection.pdf

[7] Ralf C. Staudemeyer, Applying long short-term memory recurrent neural networks to intrusion detection, http://sacj.cs.uct.ac.za/index.php/sacj/article/view/248/150