Machine Learning and Cybersecurity
A little approach

Alan Silva - Solutions Architect
Agenda

• Motivation
• A little bit of history
• Some kind of problems
• Anomaly Detection
• Why Deep Learning?
• When Deep Learning meets Cybersecurity
• Try to make a little demo
Motivation

Cyber Threat Taxonomy Tree - “Machine Learning and Security by Clarence Chio and David Freeman (O'Reilly)”
A little bit of history

- In the beginning it was just SPAM
A little bit of history

A Bayesian Approach to Filtering Junk E-Mail

Mehran Sahami†  Susan Dumais†  David Heckerman†  Eric Horvitz†

†Microsoft Research
Redmond, WA 98052-6399
{sdumais, heckerman, horvitz}@microsoft.com

In addressing the growing problem of junk E-mail on the Internet, we examine methods for the automated construction of filters to eliminate such unwanted messages from a user’s mail stream. By casting this problem in a decision theoretic framework, we are able to make use of probabilistic learning methods in conjunction with a notion of differential misclassification cost to produce filters which are especially appropriate for the nuances of this task. While this may appear, at first, to be a straight-forward text classification problem, we show that by considering domain-specific features of this problem in addition to the raw text of E-mail messages, we can produce much more accurate filters. Finally, we show the efficacy of such filters in a real world usage scenario, arguing that this technology is mature enough for deployment.

contain offensive material (such as graphic pornography), there is often a higher cost to users of actually viewing this mail than simply the time to sort out the junk. Lastly, junk mail not only wastes user time, but can also quickly fill-up file server storage space, especially at large sites with thousands of users who may all be getting duplicate copies of the same junk mail.

As a result of this growing problem, automated methods for filtering such junk from legitimate E-mail are becoming necessary. Indeed, many commercial products are now available which allow users to hand-craft a set of logical rules to filter junk mail. This solution, however, is problematic at best. First, systems that require users to hand-build a rule set to detect junk assume that their users are savvy enough to be

A little bit of history (SPAM Detection)

- Matching using word blacklists
- Use of fuzzing hashing algorithms to find approximate matches of mails marked as SPAM
- Classification by Naive Bayes
  - SPAM / HAM
Machine learning use-cases for Security

**Pattern Recognition**

- SPAM Detection
- Malware and Botnet Detection
- User-authentication and Behavior Analysis
  - When Threat Model is known

**Anomaly Detection**

- Network-related problems
  - Malicious URL detection
  - Malicious Network activity
  - Network Outlier detection
- User-authentication and Behavior Analysis
  - When don't know the threat model
Some Limitations of Machine Learning in Security

- **Complexity of classification results**
  - Sometimes is more simple extract information using a simple rule compared with a decision made by a machine learning system

- **Use intensive of resources by Machine Learning Systems than others alternatives**
  - Seems like a dealbreaker for execution in constrained environments such as embedded systems

- **The importance of human decision-making process**
  - This is a very difficult process for machine learning systems to emulate

- **Machine Learning models that works perfectly with a training set but badly with another testing set**
  - Normally because of data overfit
Anomaly Detection

- Anomaly can be defined as detection of unexpected events in systems
  - Identification of unexpected intruders or breaches

- Nature of Attack normally is:
  - Data exfiltration
  - Ransomware
  - Adware
  - APT (Advanced Persistent Threats)

- Anomaly detection is not confined only to the context of security
  - Used for finding events that don't conform with an expectation
    - Identification signs of system failure (prevention)
    - Fraud detection
Anomaly Detection

*The study of anomaly detection depends on two factors:*

- **Time**
  - Need time series analysis to understand if a deviation of analyzed stream is normal or expected

- **Stream of Data**
  - Because when analyzing a time series we seem a bunch of data included on this period
Anomaly Detection

*Supervised Learning X Anomaly Detection*

**Supervised Learning**
- *Fraudulent Credit Card Transactions*
  - Look for specific patterns that can be extracted from a body of positive and negative training examples

**Anomaly Detection**
- *Server Breaches*
  - Method of intrusion cannot be predicted and difficult to build a profile of every possible method of intrusion
Intrusion Detection Systems

An Intrusion-Detection Model

DOROTHY E. DENNING


Abstract-A model of a real-time intrusion-detection expert system capable of detecting break-ins, penetrations, and other forms of computer abuse is described. The model is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. The model includes profiles for representing the behavior of subjects with respect to objects in terms of metrics and statistical models, and rules for acquiring knowledge about this behavior from audit records and for detecting anomalous behavior. The model is independent of any particular system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose intrusion-detection expert system.

Index Terms-Abnormal behavior, auditing, intrusions, monitoring, profiles, security, statistical measures.

I. INTRODUCTION

This paper describes a model for a real-time intrusion-detection expert system that aims to detect a wide range of security violations ranging from attempted break-ins by outsiders to system penetrations and abuses by insiders. The development of a real-time intrusion-detection system is motivated by four factors: 1) most existing systems have security flaws that render them susceptible to intrusions, penetrations, and other forms of abuse; finding and fixing all these deficiencies is not feasible for technical and economic reasons; 2) existing systems with known flaws are not easily replaced by systems that are more secure—mainly because the systems have attractive features that are missing in the more-secure systems, or else they cannot be replaced for economic reasons; 3) developing systems that are absolutely secure is extremely difficult, if not generally impossible; and 4) even the most secure systems are vulnerable to abuses by insiders who misuse their privileges.

Intrusion Detection Systems

*Used until today, it is a reliable way of detecting intrusions and anomalies*

*Based on:*

- Thresholds
- Heuristics
- Simple statistical profiles
Objectives for an optimal anomaly detection system

- Low false positives and false negatives
- Low complexity to configure, tune and maintain
- Adapts to changing trends in data
- Works very well across datasets of different nature
- Resource efficient and suitable for real-time application
- Efficient logging alerts
Classification of Network Anomaly Detection Systems

- **Host Intrusion Detection**
  - Based on informations about the host (users account, running process, kernel modules loaded)

- **Network Intrusion Detection**
  - Based on network traffic (layer 3 and layer 4 Stateful Packet Inspection (SPI)) and all layers (Deep Packet Inspection (DPI))

- **Web Application Intrusion Detection**
  - Based on informations about HTTP data payloads (request/response)
Why Deep Learning?

<ADVICE> Deep Learning is not a silver bullet that can solve all problems in this field because of datasets patterns needs to be well specified </ADVICE>

But, in other way:

- *Deep Learning systems using self-taught learning to detecting unknown network intrusions*
  - And it is an advantage when compared with traditional intrusion detection systems (based on rules and signatures)

- *The generalization power of DL-based techniques is better compared to traditional ML-based approaches*
A brief overview of Feed Forward Neural Network

Representation of biological neuron from a artificial neuron

Representation of a feed forward network with two hidden layers

When Deep Learning Meets CyberSecurity

A Deep Learning Approach for Network Intrusion Detection System

Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam
College Of Engineering
The University of Toledo
Toledo, OH 43606, USA
{quamar.niyaz, weiqing.sun, ahmad.javaid, mansoor.alam2}@utoledo.edu

ABSTRACT
A Network Intrusion Detection System (NIDS) helps system administrators to detect network security breaches in their organization. However, many challenges arise while developing a flexible and effective NIDS for unforeseen and unpredictable attacks. In this work, we propose a deep learning based approach to implement such an effective and flexible NIDS. We use Self-Taught Learning (STL), a deep learning based technique, on NSL-KDD - a benchmark dataset for network intrusion. We present the performance of our approach and compare it with a few previous works. Compared metrics include the accuracy, precision, recall, and F-measure values.

Categories and Subject Descriptors
1.2 [Artificial Intelligence]: Miscellaneous; C.2 [Computer-Communication Networks]: Security

Keywords
Network security, NIDS, deep learning, sparse auto-encoder, NSL-KDD

effective in the detection of known attacks and shows high detection accuracy and less false-alarm rates. However, its performance suffers during detection of unknown or new attacks due to the limitation of rules that can be installed beforehand in an IDS. ADNIDS, on the other hand, is well suited for the detection of unknown and new attacks. Although ADNIDS produces high false-positive rates, its theoretical potential in the identification of novel attacks has caused its wide acceptance among the research community. There are primarily two challenges that arise while developing an effective and flexible NIDS for the unknown future attacks. First, proper feature selections from the network traffic dataset for anomaly detection is difficult. As attack scenarios are continuously changing and evolving, the features selected for one class of attack may not work well for other classes of attacks. Second, unavailability of labeled traffic dataset from real networks for developing an NIDS. Immense efforts are required to produce such a labeled dataset from the raw network traffic traces collected over a period or in real-time and this serves as the reason behind the second challenge. Additionally, to preserve the confidentiality of the internal organizational network structures as well as the privacy of various users, network administrators are reluctant towards creating new traffic that

When Deep Learning Meets CyberSecurity

Application of Recurrent Neural Networks for User Verification based on Keystroke Dynamics

Paweł Kobojek and Khalid Saeed
Faculty of Mathematics and Information Sciences, Warsaw University of Technology, Warsaw, Poland

Abstract—Keystroke dynamics is one of the biometrics techniques that can be used for the verification of a human being. This work briefly introduces the history of biometrics and the state of the art in keystroke dynamics. Moreover, it presents an algorithm for human verification based on these data. In order to achieve that, authors’ training and test sets were prepared and a reference dataset was used. The described algorithm is a classifier based on recurrent neural networks (LSTM and GRU). High accuracy without false positive errors as well as high scalability in terms of user count were chosen as goals. Some attempts were made to mitigate natural problems of the algorithm (e.g. generating artificial data). Experiments were performed with different network architectures. Authors assumed that keystroke dynamics data have sequence nature, which influenced their choice of classifier. They have achieved satisfying results, especially when it comes to false positive free setting.

Keywords—biometrics, GRU networks, keystroke dynamics, LSTM networks, recurrent neural networks, user verification.

In order to describe mathematically a typing pattern we first need to acquire specific data from the user. This data consists of a timestamp of the moment of pressing and/or leaving the button. Next, different measurements out of this can be computed, e.g. [5]:

- dwell time – time between moment of pressing and moment of leaving the button,
- flight time – time between pressing (or leaving) subsequent keys.

A user who types the text can make mistakes, which means that vectors representing different samples may differ in length.

In the next step, data is passed to some kind of a model, which task is to answer the question whether examined user is the one who claims to be. This model may be anomaly detection system or classifier. Popular approach is to use algorithms based on database of samples. In this case, new


When Deep Learning Meets CyberSecurity

DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning

Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar
School of Computing, University of Utah
{mind, lifeifei, guineng, svivek}@cs.utah.edu

ABSTRACT
Anomaly detection is a critical step towards building a secure and trustworthy system. The primary purpose of a system log is to record system states and significant events at various critical points to help debug system failures and perform root cause analysis. Such log data is universally available in nearly all computer systems. Log data is an important and valuable resource for understanding system status and performance issues; therefore, the various system logs are naturally excellent sources of information for online monitoring and anomaly detection. We propose DeepLog, a deep neural network model utilizing Long Short-Term Memory (LSTM), to model a system log as a natural language sequence. This allows DeepLog to automatically learn log patterns from normal execution, and detect anomalies when log patterns deviate from the model trained from log data under normal execution. In addition, we demonstrate how to incrementally update the DeepLog model in an online fashion so that it can adapt to new log patterns over time. Furthermore, DeepLog constructs workflows from the underlying system log so that once an anomaly is detected, users can diagnose the detected anomaly and perform root cause analysis effectively. Extensive experimental evaluations over large log data have shown that DeepLog has outperformed other existing log-based anomaly detection methods based on traditional data mining methodologies.

CCS CONCEPTS
-Information systems → Online analytical processing; -Security and privacy → Intrusion/anomaly detection and malware mitigation;

KEYWORDS
Anomaly detection; deep learning; log data analysis.

challenging and many traditional anomaly detection methods based on standard mining methodologies are no longer effective.
System logs record system states and significant events at various critical points to help debug performance issues and failures, and perform root cause analysis. Such log data is universally available in nearly all computer systems and is a valuable resource for understanding system status. Furthermore, since system logs record noteworthy events as they occur from actively running processes, they are an excellent source of information for online monitoring and anomaly detection.

Existing approaches that leverage system log data for anomaly detection can be broadly classified into three groups: PCA based approaches over log message counters [39], invariant mining based methods to capture co-occurrence patterns between different log keys [21], and workflow based methods to identify execution anomalies in program logic flows [42]. Even though they are successful in certain scenarios, none of them is effective as a universal anomaly detection method that is able to guard against different attacks in an online fashion.

This work proposes DeepLog, a data-driven approach for anomaly detection that leverages the large volumes of system logs. The key intuition behind the design of DeepLog is from natural language processing: we view log entries as elements of a sequence that follows certain patterns and grammar rules. Indeed, a system log is produced by a program that follows a rigorous set of logic and control flows, and is very much like a natural language (though more structured and restricted in vocabulary). To that end, DeepLog is a deep neural network that models this sequence of log entries using a Long Short-Term Memory (LSTM) [18]. This allows DeepLog to automatically learn a model of log patterns from normal execution and flag deviations from normal system execution as anomalies. Furthermore, since it is a learning-driven approach,
When Deep Learning Meets CyberSecurity

Example of Neural Network to Analyze Network Traffic
Machine Learning Presentations: cloudera.com/ml
THANK YOU