A Reconfigurable Architecture for Network Intrusion Detection using Principal Component Analysis*

David Nguyen, Abhishek Das, Gokhan Memik, and Alok Choudhary
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208
{dnguyen, ada829, memik, choudhar}@ece.northwestern.edu

ABSTRACT

In this paper, we develop an architecture for principal component analysis (PCA) to be used as an outlier detection method for high-speed network intrusion detection systems (NIDS). PCA is a common statistical method used in multivariate optimization problems in order to reduce the dimensionality of data while retaining a large fraction of the data characteristic. First, PCA is used to project the training set onto eigenspace vectors representing the mean of the data. These eigenspace vectors are then used to predict malicious connections in a workload containing normal and attack behavior. Our simulations show that our architecture correctly classifies attacks with detection rates exceeding 99% and false alarms rates as low as 1.95%. For next generation NIDS, anomaly detection methods must satisfy the demands of Gigabit Ethernet. FPGAs are an attractive medium to handle both high throughput and adaptability to the dynamic nature of intrusion detection. Using hardware parallelism and extensive pipelining, our architecture is implemented on FPGAs to achieve Gigabit link speeds.

1. INTRODUCTION

Traditionally, intrusion detection techniques fall into two categories: signature detection and anomaly detection. Signature detection, or misuse detection, searches for well-known patterns of attacks and intrusions by scanning for pre-classified signatures in TCP/IP packets. On the other, hand anomaly detection can detect new intrusions while misuse detection may not. However, a drawback is that anomaly detection methods suffer from false alarms.

Reconfigurable hardware solutions are an attractive implementation choice for anomaly detection due to their inherent parallelism, pipelining characteristics, and adaptability. While our previous work show the development of hardware architecture for FPGAs that is effective at capturing network characteristics [2], in this paper we introduce a novel architecture for Principal Component Analysis (PCA) [1] which is used as an outlier detection technique.

2. PCA AS AN OUTLIER DETECTION TECHNIQUE

PCA reduces the amount of dimensions required to classify new data and produces a set of principal components, which are orthonormal eigenvalue/eigenvector pairs[1]. In other words, it projects a new set of axes which best suit the data. In our implementation, these set of axes represent the normal connection data. Outlier detection occurs by mapping live network data onto these 'normal' axes and calculating the distance from the axes. If the distance is greater than a certain threshold, then the connection is classified as an attack. The principal components derived from the covariance matrix are usually different from the principal components generated from the correlation matrix. When some values are much larger than others, then their corresponding eigenvalues have larger weights.

First, each eigenvalue of a principal component corresponds to the relative amount of variation it encompasses. The larger the eigenvalue, the more significant its corresponding projected eigenvector. Therefore, the principal components are sorted from most to least significant. If a new data item is projected along the upper set of the significant principal components, it is likely that the data item can be classified without projecting along all the principal components. Secondly, eigenvectors of the principal components represent axes which best suit a data sample. Points which lie at a far distance from these axes would exhibit abnormal behavior. Outliers measured using the Mahanobolis distance are presumably network connections that are anomalous. Using a threshold value (t), any network connection with a distance greater than the threshold is considered an outlier. In our work, an outlier is implied to be an attack.

3. FRAMEWORK AND IMPLEMENTATION

All anomaly detections require an offline training or learning phase. Principal component analysis clearly separates the offline and online detection phases. This property is an advantage for hardware implementation. Figure 1 outlines the steps involved in PCA. In the offline phase, labeled training data is taken as input and a mean vector of the whole sample is computed. Ideally these data sets are a snapshot of activity in a real network environment. Secondly, a correlation matrix is computed from the training data. A correlation matrix normalizes all the data by calculating the standard deviation. Next, eigenanalysis is performed on the correlation matrix to extract independent orthonormal eigenvalue/eigenvector pairs. These pairs make up the set of principal components used in online analysis. Lastly, the sets of principal components are sorted by eigenvalue in descending order. The eigenvalue is a relative measure of the variance of its corresponding eigenvectors. Using PCA to extract the most significant principal components is what makes it a dimensionality reducing method because only a subset of the most important principal components are needed to classify any new data.

The online portion takes q major principal components and r minor principal components and maps online data into the eigenspace of those principal components. There are two parallel pipelines, one for calculating the major component

*This work is supported in part by the National Science Foundation Grants CNS-0551639 and IIS-0536994.

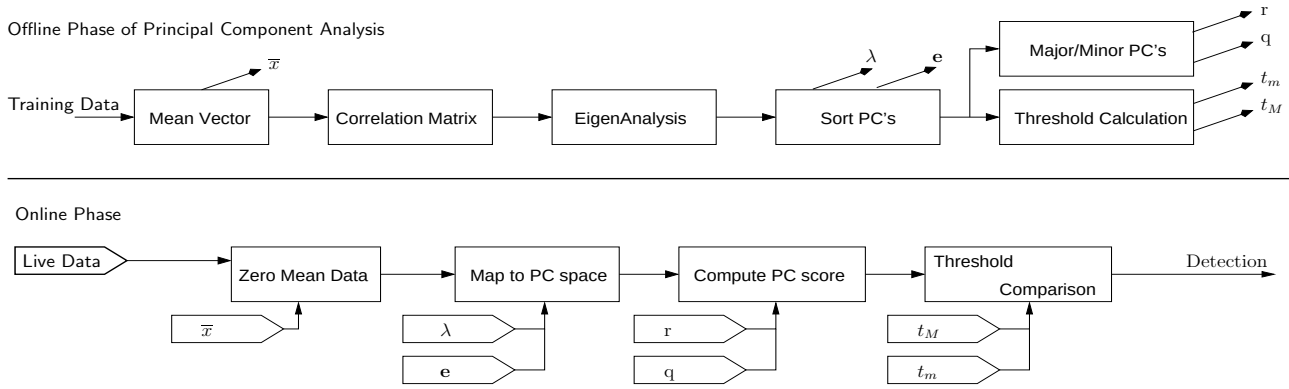

Figure 1: Principal Component Analysis for Network Intrusion Detection

variability score (MajC) and one for the minor (MinC). The simulations show that adding the MinC pipeline increases the detection ability and decreases the false alarm rate of using PCA for anomaly detection. For hardware design, the most computationally expensive portion of PCA is performing eigenvector calculations and sorting. The process of calculating eigenvectors is sequential and difficult to parallelize. Fortunately, this task is part of the offline phase.

The reprogrammability of FPGAs is an important advantage in our framework because our architecture tracks different types of network characterisrics and modify itself according to the selected features; it would be extremely costly to develop a fixed architecture that tracks the same type of network characteristics. Using RocketIO multi gigabit transceivers (MGT) available on new Virtex FPGA chips, it is possible to stream packet straight into the FPGA without suffering any slowdown. The RocketIO tranceivers can be used in conjunction with Gigabit Ethernet ports. As packets stream through the MGT in 1,2, or 4 byte chunks, a state machine is used to extract the related header fields from the packet.

4. SIMULATION RESULTS

For the FPGA implementation, a single principal component score pipeline was implemented to study the impacts of parallelizing PCA. The target device XC2VP100 (speed grade: -5) was chosen from the Xilinx Virtex II Pro family [4]. Synplify Pro 7.2 was used for synthesis and Xilinx ISE 5.2i for place and route statistics. We also vary the number of principal components to calculate a principal component score between four and eight. This workload is feasible for a real world implementation of PCA. In our simulations, each input data contained 28 fields for which we extracted 2 to 7 principal components.

In addition to classifying network connections based on their Major Principal Component score (MajC), another score based on the minor principal components (MinC) was calculated [3]. The number of major principal components (q) accounts for the majority of the data's correlation structure while the minor principal components (r) account for a small portion of the variation. This way, when attacks are detected, there is additional information if attacks do not conform to the normal correlation structure. For this study, we used minor components with eigenvalues less then 0.20. Figure 2 plots the detection and false alarm rates for different numbers of principal components used (q). The results show that PCA detects a high percentage of attacks (over 99.2%) with low false alarm rates (under 12.5%) even though the ratio of attack to normal connection is high.

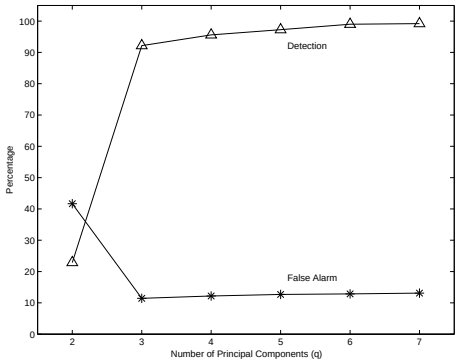

Figure 2: Detection & False Alarm Rate vs. q

Our FPGA design for PCA exploits parallelism on multiple levels. First, MajC and MinC scores are calculated in parallel. Secondly, within each pipeline, element by element matrix operations execute concurrently. The structure and layout of FPGAs lend well to matrix operations. And third, subtracting the mean from any new data tuple is performed outside the pipeline. The results from this operation are distributed to the MajC and MinC pipelines in data parallel manner. As a result, for a representative workload, our implementation outputs at a link speed of 23.76 Gbps; enough to support Gigabit line rates.

5. REFERENCES

[1] Jolliffe, I. T. *Principal Component Analysis.* Springer-Verlag, NY, 2002.

[2] Nguyen, D., Memik, G., Memik, S., Choudhary A. Real-Time Feature Extraction for High Speed Networks. In *Intl. Conf. on Field Programmable Logic and Applications (FPL)*, 2005.

[3] Shyu, M., Chen, S., Sarinnapakorn, K., and Chang, L. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In *IEEE Foundations and New Directions of Data Mining Work., in conjunction with the Third IEEE Intl. Conf. on Data Mining (ICDM'03)*, pages 172–179, 2003.

[4] XilinxVirtex-IIPro-datasheet. http://direct.xilinx.com/bvdocs/publications/ds083.pdf.