Optimal Hybrid Model of Classifiers for Intrusion Detection

Sanaa Kholfi
School of Information Technology
George Mason University
Fairfax, VA 22033
skholfi@gmu.edu

Muhammad Habib
School of Information Technology
George Mason University
Fairfax, VA 22033
mhabib@gmu.edu

Sultan Aljahdali
Department of Information Technology
College of Business Administration
Jeddah, Saudi Arabia
aljahdali@cba.edu.sa

Abstract

We present in this paper an intrusion detection software-system that we have built based on combined statistical and computational models to detect intrusions and classify them as attack or non-attack. More specifically, we build a computational machine to derive optimal parsimonious hybrid model of classifiers in intrusion detection. The classifiers are based on the following classification methods, Naïve Bayes-NB, Support vector machine-SVM, K-nearest neighbor-K-nn, and Neural networks-NN.

Keywords:

Intrusion detection, optimal hybrid model, network security, Naïve Bayes-NB, Support vector machine-SVM, K-nearest neighbor-K-nn, and Neural networks-NN.

1. Introduction

Information security is one of the cornerstones of the Information Society. Integrity of financial transactions, accountability for electronic signatures, privacy of personal information, dependability of critical infrastructure, all depend on the availability of strong, trustworthy security mechanisms. Internet connectivity has provided efficient access to large numbers of people and great benefits to the modern society, however as more computers are connected to each other and more applications are implemented (e.g. electronic commerce, online banking), internet connectivity has entailed security risks such as attacks, sabotage and malicious usage.

One subset of information security research that has been the subject of much attention in recent years is that of intrusion detection systems, or IDSs. "Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. Intrusion is defined as attempts to compromise the confidentiality, integrity, availability or to bypass security mechanism of a computer or network." [1]

The ninth annual Computer Crime and Security Survey released in June 2004 by the Computer Security Institute (CSI) and the FBI's Computer Intrusion Squad in San Francisco found that 97% of the respondents had detected computer security breaches. The 494 survey respondents willing to quantify their losses reported total damage at over $141 million. Table.1 shows that 53% of respondents have experienced unauthorized use of computer systems in the last 12 months, 59% reported insider abuse and only 39% indicated system penetrations.

Percentage of respondents who detected:	
Computer security breaches within the last 12 months	53%
Employee abuse of Internet access privileges	59%
Denial-of-service attacks	17%
System penetration from the outside	39%
Theft of transaction information	10%
Financial fraud	5%

Table.1 2004 CSI/FBI Computer Crime and Security Survey
Source: Computer Security Institute

This paper is a research in progress. Section 2 provides more insights about intrusion detection and a brief review of the statistical and computational models used to develop our computational machine to derive optimal parsimonious hybrid models of classifiers for intrusion detection. Section 3 describes our computational machine and the modeling approach illustrated by sample results in section 4. The paper concludes with observations on future research needed to improve both the model and the machine.

2. Background

2.1 Intrusion Detection Overview

Intrusion Detection is the art of detecting inappropriate or incorrect activity. Among other tools, an Intrusion Detection Systems (IDSs) can be used to determine if a computer network or server has experienced an unauthorized intrusion. They collect information from a variety of vantage points within computer systems and networks, and analyze this information for signs of intrusion and misuse.

Even though the intrusion detection system differs between statistical filters, their basic objective and functionality are similar. Essentially, an event is distilled into a set of features such as IP, flags, destination/source address, number of failed logins, protocol type, etc. This set of features can then be represented as a vector whose components are Boolean or real values. The reduction of both false positives and false negatives represents a critical objective in intrusion detection.

Two of the most common methods used in intrusion detection systems are rule based and statistics driven. Rule based methods classify documents based on whether or not they meet a particular set of criteria. Machine learning algorithms are primarily driven by the statistics that can be derived from the feature vectors. One of the most used methods is the Bayesian classification; it attempts to calculate the probability that an event is an intrusion based upon previous feature frequencies in attack/non-attack event [2]. Other famous learning algorithms used in intrusion detection systems are support vector machines [3], neural networks [3,4] and k-nearest neighbor [5].

Our computational machine for intrusion detection presented in this paper groups three statistical methods and one computational model to improve the accuracy of our intrusion detection machine, this will be accomplished by combining different classifiers to achieve the best possible detection performance:

1. Naïve Bayes, NB
2. Support vector machine, SVM
3. K-nearest neighbor, K-nn
4. Neural networks, NN

These methods are combined[1] with different technique of feature selection: chi-square χ^2, entropy and mutual information (MI). We have considered three linear combination methods (voting, averaging and recursive least square) to combine the statistical methods.

2.2 Statistical and Computational Models Overview

Naïve Bayes is a technique for estimating probabilities of individual feature values, given a class, from training data and to then allow the use of these probabilities to classify new records. Support vector machine, SVM, constructs a two class classifier function that divides the feature space into two subspaces, one for each class. Using training set, SVM specifies in advance which data should cluster together.

K-nearest neighbor, K-nn, is a technique that classifies each record in a data set based on a combination of the classes of the K records most similar to it in a historical data set. It has no distinct training (model-building) phase because the training data is actually the model. The neural network algorithm that we have considered in this paper is back propagation. Back propagation is the best known training algorithm for neural networks and the most useful. It is thoroughly described in most neural network textbooks (e.g., [7], [8]). It has lower memory requirements than most algorithms, and usually reaches an acceptable error level pretty quickly.

---

[1] In this paper, we have considered only 7 combinations in this paper: NB-χ^2, NB-entropy, SVM-entropy, NN-MI and K-nn-(χ^2, MI and entropy).

3. Computational Machine Description

The main aim of this computational machine is to create an optimal hybrid model of classifiers for intrusion detection to segregate between attack and n0n-attack events. We have developed this software using the Visual Basic programming language and S-Plus software for statistical analysis. The design of this software is described in the following sub-sections.

3.1 Supporting Machinery

We have built our machine using Visual Basic programming language to provide users with facilities to easily monitor the computational machine. This is accomplished by user interfaces which control the elements of an interactive system. Additional to Visual Basic language we have integrated to it the statistical software S-Plus to facilitate the statistical analysis.
We present in the following only the first version of our computational machine.

3.2 Machine User Interface

The optimal model for classification is data dependent [6], we have designed our computational machine to allow each user to save his profile and use it for classification in the future. Fig.1 shows the login interface, existing users can login by submitting their usernames and passwords, new users can create new profiles.

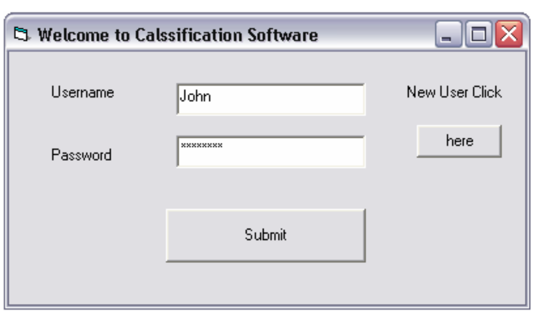


Fig.1. Login Interface

New users, when Login, they have to specify the parameters for the computational machine, these parameters include the guidance parameter λ, the significance level α, the combination method and the number of classifiers to be considered. The optimality criteria considered is based on the harmonic error $E_\lambda = 1 - F_\lambda$, where F_λ is Van Rijbergen's F-measure of accuracy [10], defined as a combination of both recall (R) and precision (P):

$$\mathbf{F}_\lambda = \frac{\mathbf{1}}{\lambda\, \mathbf{R}^{-1} + (\mathbf{1} - \lambda)\, \mathbf{P}^{-1}}$$

The user then uploads the training data and submits his preferences to obtain the optimal parsimonious hybrid model for his data. Fig.2. shows the parameter interface where the user enters all the required information. After then, the user uploads the data to be detected, at that time an output interface appears so the user can customize the output file. Fig.3. shows all the possible options.

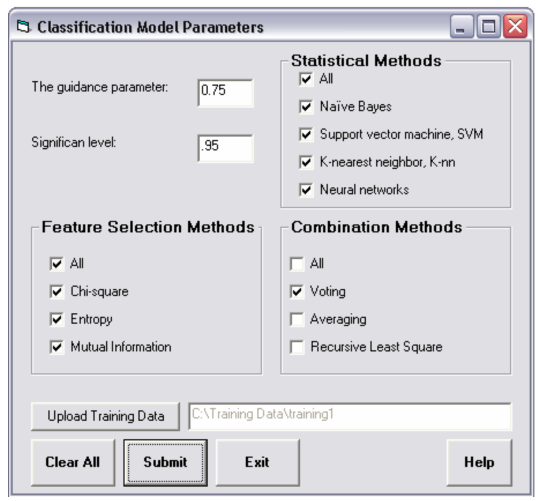


Fig.2. Parameters Interface

During the whole process, the computational machine interacts with the user through user interfaces that have been meticulously designed and compounded with a detailed help explaining all steps and terms in the computational machine.

The statistical analysis is done using S-Plus and the output is generated as a text file.
More features will be added to this machine in the future work.

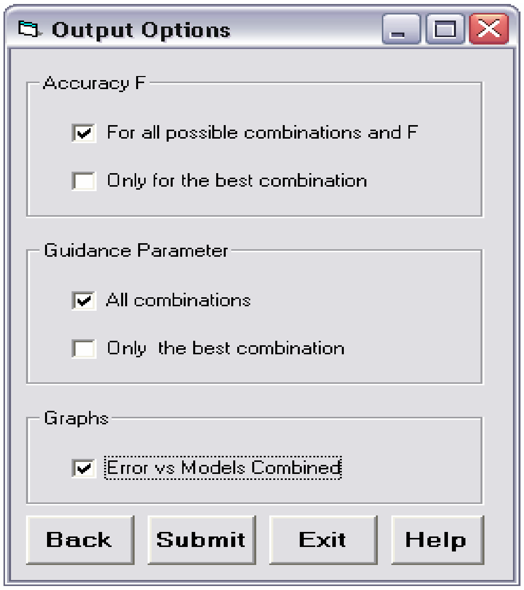


Fig.3.Output Options Interface.

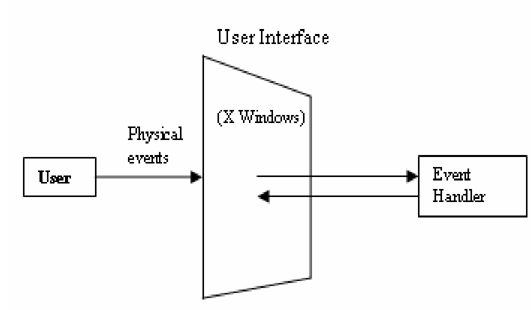
3.3 Computational Machine Model

The computational model that has been implemented is described in the figure below. Fig.4. shows the general architecture of the software model. It includes the different steps for training data: features selection, data transformation and finally data analysis in S-Plus software. The output is the best combination model that will be used to classify new data.

More features will be implemented to enhance our computational machine Fig. 5 and more testing data for intrusion detection will be collected and tested in our machine.

4. Software Model

(a)



(b)

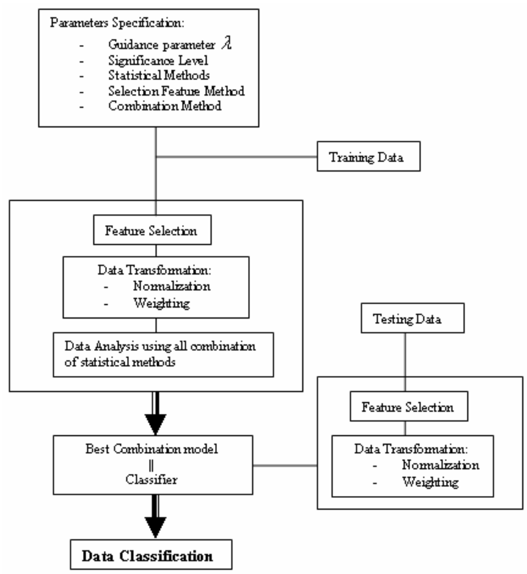


Fig.4 Software Model

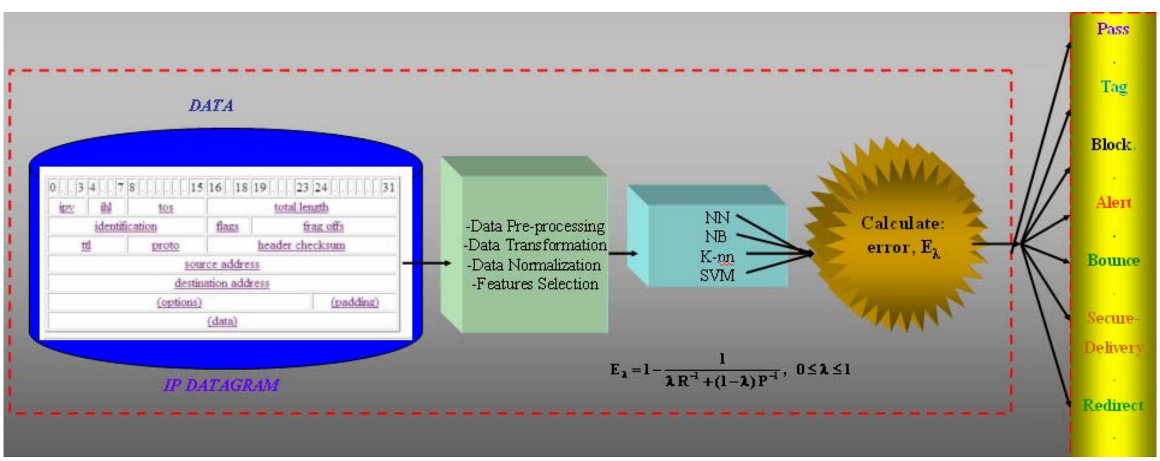


Fig.5 Sketch of the Computational Machine

5. Reference

[1] Rebecca B., Peter M. “Intrusion Detection Systems”, NIST Special Publication on IDS, Nov 2001.

[2] Nahla B. A., Salem B., Zied E. “Naive Bayes vs decision trees in intrusion detection systems”, 2004.

[3] Srinivas M., Guadalupe J., Andrew S. “Intrusion Detection Using Neural Networks and Support Vector Machines”.

[4] Jake Ryan , Meng-Jang Lin , Risto Miikkulainen, Intrusion detection with neural networks, Proceedings of the 1997 conference on Advances in neural information processing systems 10, p.943-949, July 1998, Denver, Colorado, United States.

[5] Y. Liao arid V. Rao Vemuri, "Use of K-nearest neighbor classifier for intrusion detection", *Computers & security*, vol. 21, no. 5, pp. 439-448, 2002.

[6] Alduhaiman, Khaled, Computational machine for optimal parsimonious hybrid models of e-text classification, GMU, 2004

[7] Patterson, D. (1996). Artificial Neural Networks. Singapore: Prentice Hall. Good wide-ranging coverage of topics, although less detailed than some other books.

[8] Fausett, L. (1994). Fundamentals of Neural Networks. New York: Prentice Hall.

[9] Whitaker C.J., L.I. Kuncheva, Examining the relationship between majority vote accuracy and diversity in bagging and boosting, Technical Report, 2003, School of Informatics, University of Wales, Bangor.