POLVO-IIDS: Um Sistema de Detecção de Intrusão Inteligente Baseado em Anomalias

Paulo M. Mafra[1], **Joni da Silva Fraga**[1], **Vinícius Moll**[1], **Altair Olivo Santin**[2]

[1]Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – CEP 88040-900 – Florianópolis – SC – Brasil

[2]Pontifícia Universidade Católica do Paraná (PUC-PR)
R. Imaculada Conceição, 1155 – CEP 80215-901 – Curitiba – PR – Brasil

{mafra,fraga,vmoll}@das.ufsc.br,santin@ppgia.pucpr.br

Abstract. *The intrusion detection systems (IDS) identify attacks and threats to computer systems. Additionally, the IDSs can perform other functions like intrusion prevention (IPS), including proactive functions. A recurrent problem in intrusion detection systems is the difficulty to identify legitimate access from attacks. A lot of conventional systems are signature based, although they do not identify variations of these attacks nor new attacks.*
This paper presents an intrusion detection system model based on the behavior of network traffic through the analysis and classification of messages. Two artificial intelligence techniques named support vector machine (SVM) and Kohonen neural network (KNN) are applied to detect anomalies. These techniques are used in sequence to improve the system accuracy, identifying known attacks and new attacks, in real time.

Resumo. *Os sistemas de detecção de intrusão (IDS) têm como atribuição a identificação de ataques e ameaças aos sistemas computacionais. Adicionalmente, os IDSs podem desempenhar funções de prevenção a intrusão (IPS), incluíndo-se ações pro-ativas às intrusões. Um problema recorrente destes sistemas de detecção de intrusão é a dificuldade de diferenciar ataques de acessos legítimos. Muitos sistemas utilizam assinaturas de ataques conhecidos, contudo não conseguem identificar variações destes ataques nem novos ataques.*
Este artigo apresenta um modelo de sistema de detecção de intrusão que classifica mensagens por análise comportamental como normal ou anômala. Para detecção de anomalias são utilizadas duas técnicas de inteligência artificial chamadas support vector machine (SVM) e redes neurais de Kohonen (KNN). O uso destas técnicas em conjunto visa melhorar a taxa de acerto do IDS desenvolvido, identificando ataques conhecidos ou novos em tempo real.

1. Introdução

O número de computadores conectados à Internet cresce a cada dia. Estes computadores são usados para trabalho, estudo, comércio eletrônico, etc. Em todas estas áreas, os programas utilizados tais como leitores de emails, editores de texto e o próprio sistema operacional podem apresentar problemas de segurança. Para detectar e prevenir os eventuais usos destas vulnerabilidades nos complexos ambientes computacionais dos dias

atuais, é necessário que se faça uso de sistemas de detecção de intrusão (IDS). Os IDSs, tão essenciais na geração de alertas sobre comportamentos indesejáveis em sistemas, normalmente se apresentam divididos em duas grandes classes: os baseados em assinaturas e os baseados em anomalias [Allen et al. 2000].

Os sistemas baseados em assinaturas tentam identificar ataques que seguem padrões previamente reconhecidos e reportados por especialistas (cada assinatura identifica um determinado ataque). Nos IDSs baseados em anomalias, são representados comportamentos normais de operação do sistema onde o IDS está instalado ou da rede que o IDS monitora e, quando acontece um evento que não segue estes comportamentos identificados como normais, uma sinalização de anomalia é gerada no sistema.

Um ponto fraco dos sistemas de detecção de intrusão baseados em assinaturas é a impossibilidade de identificar novos tipos de ataques ou variações de ataques já conhecidos. Por outro lado, o número de falsos positivos gerados por IDSs baseados em anomalias é maior do que nos baseados em assinaturas. Uma questão importante nestes IDSs baseados em anomalias é como estes sistemas devem ser "treinados", ou seja, como os IDSs devem aprender o comportamento normal de um ambiente computacional e de seus usuários.

O processo de treinamento exige uma grande quantidade de dados e pode empregar diversas técnicas de inteligência artificial na automação do mesmo. As técnicas de inteligência artificial têm sido usadas tanto para detecção baseada em assinaturas quanto nas detecções baseadas em anomalias. Dentre essas técnicas, podemos citar o uso de sistemas especialistas [Lunt 1993]. Estes últimos usam um conjunto de regras que representam padrões conhecidos de ataques ou vulnerabilidades para detectar intrusões. Alguns trabalhos como [Bridges and Vaughn 2000, Lee et al. 1998, Luo 1999, Mukkamala et al. 2000] usam técnicas de mineração de dados para identificar padrões normais de funcionamento dos sistemas.

Redes neurais artificiais (RNA) também vêm sendo empregadas em IDSs [Ghosh et al. 1998, Giacinto et al. 2003, Cannady 1998, Lee 2001]. A maioria dessas redes neurais são compostas por um conjunto de entrada, algumas camadas intermediárias e uma saída. Essas redes possuem grande capacidade de identificação de padrões, sem que estes sejam idênticos, o que não acontece em sistemas baseados em assinaturas. Ou seja, é possível detectar variações de um mesmo ataque com o uso de redes neurais.

Este artigo apresenta um modelo desenvolvido para detecção de intrusão que faz uso de técnicas de inteligência artificial, visando alta taxa de acerto, podendo ser aplicado com eficiência a um tráfego de rede real. O artigo está organizado da seguinte forma: na Seção 2 apresentamos os principais trabalhos relacionados a detecção de intrusão que fazem uso de redes neurais. Na Seção 3, descrevemos o modelo de IDS desenvolvido. Na seqüência, Seção 4, detalhamos os testes efetuados com o modelo desenvolvido e os resultados obtidos. Ao final, traçamos algumas considerações sobre o modelo desenvolvido, bem como possíveis trabalhos futuros.

2. Trabalhos Relacionados

O emprego de redes neurais artificiais (RNAs) em sistemas de detecção de intrusão (IDSs) aparece em diversos trabalhos, como [Zanero and Savaresi 2004], [Kayacik et al. 2003] e

[Lei and Ghorbani 2004]. Estes trabalhos usam SOM (*Self Organizing Maps*) e algumas variações para armazenar os dados provenientes do treinamento da rede neural. A idéia principal da abordagem de redes neurais artificiais é a disponibilização de um método de classificação sem supervisão, rápido e eficiente para uma entrada de dados com muitas variáveis (IP de origem, IP destino, porta de origem, porta de destino, tamanho do pacote, protocolo, etc). Um problema presente em redes neurais é o tempo de treinamento destas redes, que é normalmente efetuado "*off-line*". Contudo, uma vez treinadas, os tempos de análise são consideravelmente eficientes.

Os trabalhos envolvendo redes neurais na detecção de intrusão apresentam resultados promissores, como diminuição na taxa de falsos positivos e melhora na taxa de detecção quando comparado a outros IDSs baseados em anomalias. Contudo, os IDSs que utilizam redes neurais, presentes na literatura, enfrentam o problema da dificuldade de treinamento com tráfego real e ataques reais e atuais. Amostras de tráfego real, podem apresentar algum tipo de tráfego malicioso (ruído) não identificado. A aplicação desse tráfego malicioso no treinamento da rede neural pode afetar o valor dos pesos dos neurônios, ocasionando erros na detecção. É difícil classificar uma quantidade grande de tráfego real, identificando todos os ataques existentes como pacotes mal formados, pacotes fragmentados, etc.

Em [Wang et al. 2004] foi desenvolvido um sistema de detecção baseado em anomalias chamado PAYL. O principal objetivo desse sistema é fazer detecções rápidas, preferencialmente em um *gateway* ou em um *front-end*, evitando a disseminação nas demais máquinas da rede. O método é baseado em anomalias e cobre todos os serviços de rede. Esse método analisa e gera modelos matemáticos (baseado na freqüência dos *bytes* contidos no *payload*) dos *payloads* que devem ser entregues às aplicações. O sistema PAYL "aprende" e gera um perfil dos *payloads* esperados para cada serviço. O detector captura os *payloads* de entrada de um serviço e compara com o perfil que foi gerado para aquele serviço na fase de treinamento.

Um IDS que faz a análise baseada em anomalias utilizando o cabeçalho e o *payload* dos pacotes é o POSEIDON [Bolzoni et al. 2006]. Porém, este sistema utiliza apenas as informações (contidas no cabeçalho da mensagem) do endereço destino e da porta para construir um perfil de cada serviço. O sistema é composto por duas camadas: SOM em conjunto com o modelo PAYL. O SOM é usado para classificar os dados. O modelo PAYL possui apenas uma camada de nós (neurônios) onde cada neurônio n tem um vetor de pesos w_n associado. A dimensão dos vetores de pesos é igual ao tamanho do maior dado de entrada para um determinado serviço de rede.

Outra abordagem que utiliza redes neurais para detecção de intrusão foi proposta por [Shyu et al. 2003]. Foi aplicada a técnica PCA (*Principal Component Analysis*), uma técnica de redes neurais com aprendizado sem supervisão, que busca reduzir a dimensão dos dados (reduzir a quantidade de variáveis) e facilitar a análise posterior dos mesmos.

Em [Xiang and Lim 2005] foi proposto um classificador híbrido com múltiplas camadas para detecção de intrusão, onde os ataques são classificados em três categorias: negação de serviço (DoS), scan (PROBE) e "outros". A categoria "outros" é dividida em duas sub-categorias: usuários locais que tentam obter privilégios de super usuário (U2R) e ataques a serviços através da rede (R2L).

Um outro trabalho recente que aborda o uso de inteligência artificial na detecção de intrusão é descrito por [Chen et al. 2005]. Nesse trabalho são usados *Support Vector Machines* (SVMs) para mapear a freqüência de chamadas de sistema efetuadas por cada processo na máquina. SVM é uma poderosa técnica para resolver problemas relacionados a aprendizagem, classificação e predição [Mukkamala et al. 2002]. O sistema desenvolvido em [Chen et al. 2005] compara a freqüência de chamadas de sistema com as freqüências mapeadas nos SVMs e procura por discrepâncias. Em um ambiente dinâmico, é quase impossível criar perfis de usuários que determinam o comportamento normal. É melhor observar o comportamento dos processos ao invés dos usuários. Neste trabalho, foram efetuados testes usando os dados do DARPA 1998 do MIT com outras redes neurais artificiais e com SVMs. Os resultados com SVMs apresentaram maior taxa de detecção e menor taxa de falsos positivos.

Em outro trabalho recente [Haijun et al. 2007], foi elaborado um estudo comparativo entre o uso de *support vector machine* SVM e outras técnicas de mineração de dados e redes neurais para detecção de intrusão. Nos testes efetuados, o uso de SVM apresentou uma taxa maior de detecção do que as técnicas de mineração de dados e ligeiramente melhor do que com o uso de redes neurais. Todos os sistemas apresentados nesta seção fazem uso de apenas um tipo de rede neural para classificação, alguns sistemas usam múltiplas camadas (múltiplas redes).

3. Modelo do POLVO-IIDS

Os sistemas de detecção de intrusão inteligentes (IIDSs) atuais aplicam apenas um tipo de rede neural e não conseguem atingir uma boa precisão na detecção. Tais sistemas também enfrentam problemas no treinamento de suas redes neurais devido a grande variância do comportamento do tráfego presente na Internet. Observando as características das redes de Kohonen, podemos ressaltar a habilidade em classificar dados, de forma genérica. Por outro lado, *Support Vector Machine* (SVM) conseguem uma precisão maior quando treinadas para separar os dados em apenas duas classes. Desta forma, desenvolvemos um sistema multi-camadas, chamado POLVO-IIDS, utilizando esses dois tipos de redes neurais (Kohonen e *Support Vector Machine*).

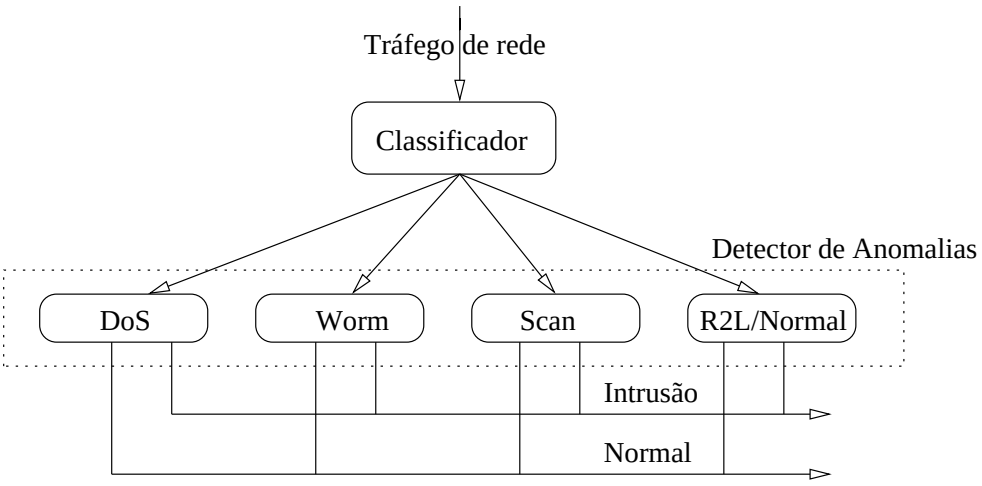


Figura 1. Arquitetura do POLVO

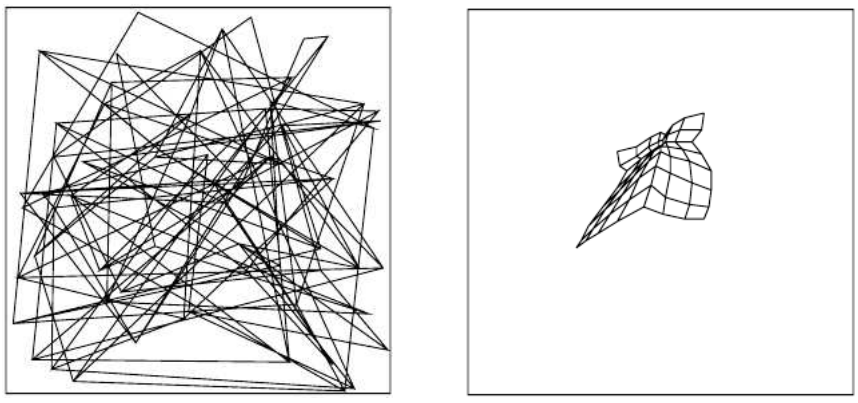
O objetivo principal do POLVO-IIDS é prover um sistema de detecção de intrusão inteligente (IIDS) que seja preciso (baixa taxa de falsos positivos e falsos negativos), flexível, tolerante a variações de ataques, adaptativo a variações do ambiente, modular e que atue em tempo real. O POLVO-IIDS gera modelos de comportamento do tráfego de rede e faz sua detecção baseada nesses modelos. A Figura 1 mostra a arquitetura do sistema proposto.

Esse sistema é composto por duas camadas. A primeira, chamada **classificador**, é responsável por coletar o tráfego de rede, analisá-lo e classificá-lo em quatro categorias: **DoS**, **Worm**, **Scan** ou **R2L/Normal**. Na segunda camada estes tráfegos separados passarão por detectores específicos a suas classes (Camada Detector de Anomalias), ou seja, esta segunda camada é a responsável pela detecção de intrusão propriamente dita. Essas camadas são descritas com detalhes nas seções 3.1 e 3.2 respectivamente.

3.1. Classificador

A ideia principal do classificador é fazer uma pré-seleção do tráfego de entrada, através da análise de características contidas nos pacotes em um determinado período de tempo. Esse classificador é semelhante a alguns IDSs baseados em anomalias presentes na literatura (um destes é o sistema apresentado em [Xiang and Lim 2005]). Contudo, por esses IDSs apresentarem alta taxa de falsos positivos, decidimos enviar a saída do classificador para uma outra rede neural, mais especializada, que tem a função de analisar determinado tipo de ataque e identificar com mais precisão o que é anomalia e o que é tráfego normal. Desta forma, pretendemos reduzir a taxa de falsos positivos dos IDSs convencionais e melhorar a taxa de acerto.

O Classificador é composto por uma rede neural de Kohonen [Kohonen 1988]. A escolha deste tipo de rede foi motivada pela característica das redes de kohonen em aprender padrões de forma automática (sem supervisão), pela facilidade em separar padrões conhecidos (treinados) e pela generalização na detecção de padrões (detecta variações de padrões conhecidos).



(a) Rede de Kohonen sem treinamento

(b) Rede de Kohonen iniciando o treinamento

Figura 2. Rede de Kohonen ao longo do tempo

A Figura 2 apresenta a organização dos neurônios ao longo do tempo. Em 2(a), a rede ainda não recebeu o treinamento. A Figura 2(b) apresenta a rede após um treinamento

inicial de 200 iterações [Kröse and van der Smagt 1996]. Podemos observar a mudança na distância entre os neurônios e a formação de um pico. Em um gráfico 3D, após a etapa de treinamento, cada padrão identificado forma um pico nesse plano (Figura 3), caracterizando mapas auto-organizados, na qual a localização espacial (coordenadas) dos neurônios representa as características contidas nos padrões de entrada dos dados durante o treinamento.

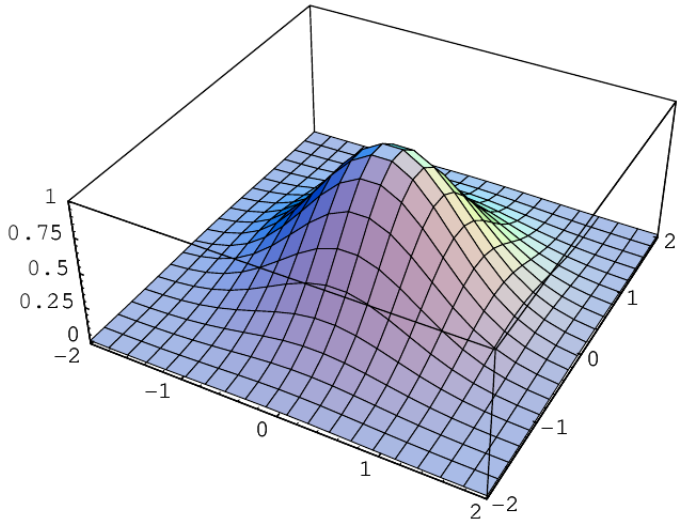


Figura 3. Rede de Kohonen após treinamento

Foram criadas quatro categorias (padrões) de comportamento de tráfego para detecção de anomalias. A rede neural de Kohonen, uma vez treinada para esses quatro padrões (categorias DoS, Worm, Scan e R2L/Normal), é capaz de separar o tráfego de rede suspeito. Para a construção desta rede, foi gerada uma rede neural com 41 entradas e 4 saídas. Para o aprendizado da rede (cálculo da distância entre neurônios) foi aplicado o cálculo da distância euclidiana $d = \sqrt{\sum_i (v_i - w_i)^2}$, onde v_i é o vetor de entradas e w_i é o vetor de pesos.

O emprego da distância euclidiana é ideal para entrada de dados aleatórios, como é o caso dos tráfegos a serem analisados em redes de comunicação reais. Na fase de aprendizado, os dados de entrada da rede devem ser aplicados diversas vezes pois o aprendizado (ajuste dos pesos dos neurônios) se dá por reforço.

Para cada entrada de dados do classificador, apenas um neurônio de saída é ativado. O valor de cada neurônio de saída pode variar de 0 até 1, sendo que se estiver acima de 0,8 está apto a ser ativado. Caso o neurônio esteja abaixo de 0,2 não ocorre a ativação do mesmo. Se nenhum neurônio estiver apto a ser ativado (com valor acima de 0,8), uma função força a vitória de algum neurônio que esteja entre 0,2 e 0,8. Os valores geralmente empregados em redes de Kohonen são de 0,1 e 0,9. Contudo, com tais valores a rede tende a ser mais restritiva (no sentido de identificar menos ocorrências de ataques mas o que for identificado tem maiores chances de ser um ataque). Preferimos estabelecer os valores em 0,2 e 0,8 para evitar erros na classificação de algum possível ataque. Se o mesmo for direcionado para o detector de anomalias correto, este se encarregará da detecção. Em testes com valores em 0,3 e 0,7, os resultados de classificação foram piores, classificando

um número maior de ataques em classes erradas (no caso de ter mais de um neurônio apto a ser ativado).

3.2. Detector de Anomalias

O detector de anomalias é composto por quatro redes neurais do tipo *Support Vector Machines* (SVM), que recebem o tráfego do classificador. Essas redes neurais tratam de quatro categorias distintas:

- **DoS**: Esta categoria é responsável pela identificação de ataques de negação de serviço, caracterizada pelo envio de múltiplas solicitações de abertura de conexão a um mesmo *host* em uma mesma porta, em um curto período de tempo.
- **Worm**: Esta classe corresponde a ataques gerados por *worms*, onde, geralmente, há a abertura de milhares de conexões para diversos destinos em um curto período de tempo.
- **Scan**: Estes ataques são caracterizados pela tentativa de abertura de conexões em diversas portas de um mesmo destino, com o intuito de descobrir quais serviços e versões estão instaladas no *host* destino.
- **R2L/Normal**: Esta categoria recebe fluxos de tráfego que são ou que aparentam ser normais. Estes fluxos podem ser ataques remotos a serviços específicos ou acessos legítimos aos serviços.

Para cada categoria de ataque citada acima, existe uma rede neural (SVM) especializada no tipo correspondente de ataque que terá duas opções como saída: tráfego normal ou atividade maliciosa. Optamos pelo uso de redes SVM porque os resultados obtidos em [Chen et al. 2005] e [Haijun et al. 2007] mostram que estas redes são mais eficientes do que outros tipos de redes neurais na identificação de anomalias. As SVMs suportam também uma certa quantidade de ruído na entrada da rede (alguns ataques incluídos no tráfego de treinamento) sem prejudicar o treinamento da mesma. Na escolha de parâmetros de configuração, as redes neurais SVM são menos complexas do que outros modelos no estabelecimento do número de camadas escondidas, do número de nós (neurônios) para cada camada e das funções de transferência. A escolha errada de alguns dos citados parâmetros pode causar uma degradação no desempenho de uma rede neural.

Cada categoria é representada por um hiperplano, conforme apresentado na Figura 4, definido por um número de vetores de suporte onde os dados de treinamento são separados em duas classes, uma para tráfego normal e outra para atividade maliciosa (classes A e B na figura). Esses vetores formam um sub conjunto dos dados de treinamento usados para definir limites entre as duas classes (vetores preenchidos na figura). Os limites podem ser expressos matematicamente como: $w^T x + b = 0$, onde w é o vetor de pesos, b é a âncora (*bias*) e x é um vetor de entrada.

Acreditamos que a detecção de intrusões, usando duas redes neurais em seqüência, uma para classificação e outra para tomada de decisão é viável e produz um aumento na taxa de detecção e redução na taxa de falsos positivos, quando comparado com outros sistemas de detecção baseados em anomalias presentes na literatura. Defendemos desta forma o possível uso do POLVO-IIDS na análise de tráfego real.

3.3. Protótipo

Foi desenvolvido um protótipo, seguindo o modelo proposto do POLVO-IIDS para a detecção de intrusões utilizando redes neurais artificiais. O protótipo está centrado no

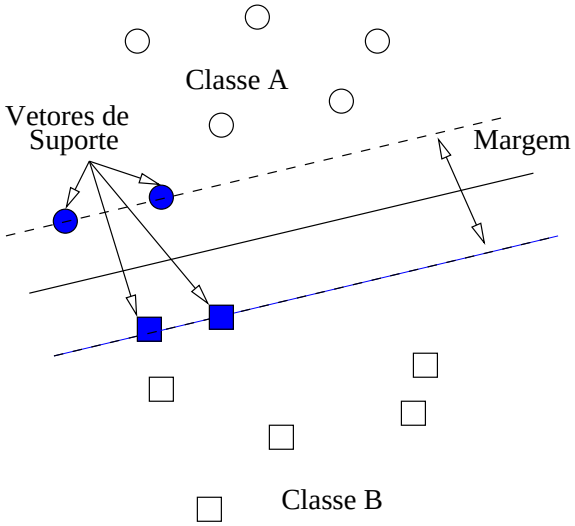


Figura 4. Separação de duas classes pelo SVM

Classificador e no Detector de Anomalias, ambos implementados fazendo uso da linguagem de programação Java, na qual as redes neurais de kohonen e SVMs foram implementadas no *framework* Joone (Java Object Oriented Neural Engine)[1] versão 2.0.

No presente protótipo, os dados de entrada do classificador (41 variáveis) são guardados e repassados para a rede SVM especializada, indicada pelo classificador. Essa rede SVM tem duas saídas possíveis: intrusão ou tráfego normal.

Neste protótipo, desenvolvemos um conversor de formatos que adapta os dados de entrada, no formato PCAP[2], para valores que possam ser mapeados nas variáveis de entrada das redes neurais. Na seção 4, avaliamos o desempenho do modelo proposto para detecção de intrusão, fazendo uma comparação dos resultados obtidos com os resultados de outras abordagens presentes na literatura.

4. Testes e Resultados Obtidos

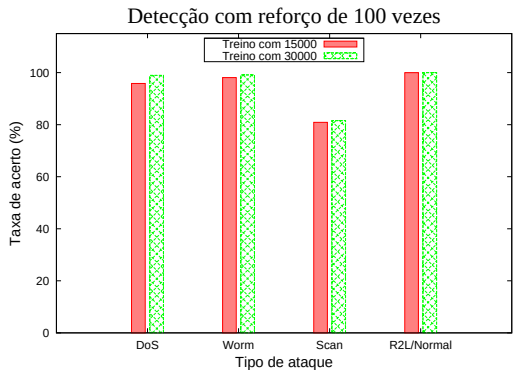
Os testes com o protótipo foram executados em uma máquina com 4GB de memória RAM e dois processadores Intel Xeon de 1.6GHz, cada qual com quatro núcleos. Para a realização de testes com o modelo apresentado, foi usado o tráfego *KDD Cup 1999 Data* disponível na Internet [Stolfo et al. 1999]. Pelos experimentos, a configuração mínima exigida para execução dos testes seria um computador com um processador de 1.6GHz e 1GB de memória RAM.

Primeiramente a rede neural do classificador foi treinada com 4 categorias de ataques: DoS, Worm, Scan, e ataques a serviços (R2L) ou tráfego normal. Na seqüência, as redes SVM do detector de anomalias foram treinadas com tráfego específico para cada tipo de ataque. Os vetores de peso das redes treinadas foram armazenados em arquivos para uso posterior (sem a necessidade de novos treinamentos das mesmas).

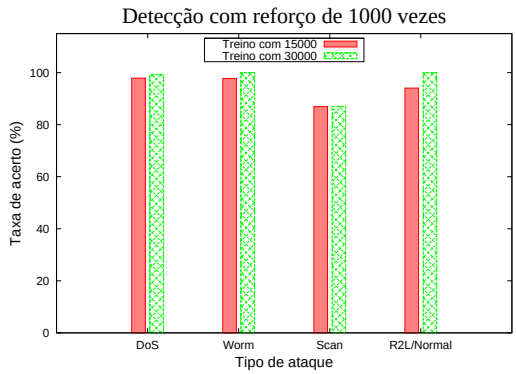
[1]Framework gratuito para a criação de redes neurais `http://www.jooneworld.com/index.html`

[2]Formato padrão para captura de pacotes de rede. `http://www.tcpdump.org/pcap/pcap.html`

Após as fases de treinamento das redes neurais, foram efetuados quatro testes para medir as taxas de acerto (porcentagem do número de detecções corretas pelo número de dados de entrada) do sistema desenvolvido, aplicando os dados do KDD Cup 1999. Os testes 1 e 2 contaram com 15000 e 30000 dados de entrada, respectivamente, para treinamento da rede com reforço de 100 vezes para cada entrada. Os testes 3 e 4 também contaram com 15000 e 30000 dados de entrada, respectivamente, para treinamento da rede com reforço de 1000 vezes para cada entrada. Nesses testes, foram aplicados 97143 entradas (dados presentes no KDD Cup 99) contendo tráfego normal e ataques dos tipos: DoS, Worm, Scan ou R2L.



(a) Testes 1 e 2 com reforço no treinamento de 100 vezes



(b) Testes 3 e 4 com reforço no treinamento de 1000 vezes

Figura 5. Taxas de acerto para cada tipo de ataque

Os resultados destes quatro testes estão representados na Figura 5. O gráfico 5(a) mostra os resultados obtidos em cada categoria (DoS, Worm, Scan, R2L ou Normal) para o treinamento da rede neural com reforço de 100 vezes para cada entrada (testes 1 e 2). O gráfico 5(b) apresenta os resultados obtidos em cada categoria (DoS, Worm, Scan, R2L ou Normal) para o treinamento da rede neural com reforço de 1000 vezes para cada entrada (testes 3 e 4).

Tabela 1. Taxas de acerto médio

Tipo de Teste	Taxa de Acerto Médio	Desvio Máximo
Teste 1: treino com 15000 entradas X 100	93,70 %	12,80 %
Teste 2: treino com 30000 entradas X 100	94,15 %	7,20 %
Teste 3: treino com 15000 entradas X 1000	94,89 %	13,36%
Teste 4: treino com 30000 entradas X 1000	96,55 %	9,53 %

A Tabela 1 apresenta a taxa de acerto médio e o desvio máximo em uma categoria de ataque (DoS, Worm, Scan, R2L) para cada teste realizado com o POLVO-IIDS. Nesta tabela, podemos observar que treinando a rede neural com uma quantidade maior de dados obtemos uma taxa de acerto maior em todas as categorias. Com menos treinamento, a variação na taxa de acerto em cada categoria é maior. O número de reforços (número de vezes que um dado é apresentado para a rede neural) também é importante para a consolidação de uma categoria de ataque. Acreditamos que, para os dados do KDD Cup

99, o treinamento das redes com 30000 entradas mostradas 1000 vezes seja ideal, embora tenha levado cerca de 40 minutos.

Tabela 2. Comparação de resultados entre IIDSs

IIDS	Acerto Médio	Desvio Máximo
Anomalous Payload-based IDS [Bolzoni et al. 2006]	58,80 %	41,20 %
HPCANN [Liu et al. 2006]	77,49 %	22,53%
MADAM ID [Lee and Stolfo 2000]	77,97 %	17,97%
Multi-level Hybrid Classifier [Xiang and Lim 2005]	89,19 %	22,52%
POLVO-IIDS	96,55 %	9,53 %

A Tabela 2 faz uma comparação entre os resultados obtidos pelo POLVO-IIDS e alguns sistemas presentes na literatura. Todos os sistemas citados utilizaram os dados do KDD CUP 99. Podemos observar que o POLVO-IIDS obteve um bom desempenho em todas as categorias (baixo desvio máximo). Os sistemas citados não apresentaram algumas informações como o número de dados usados no treinamento das redes nem maiores detalhes sobre os ajustes nos valores de ativação dos neurônios de suas redes.

5. Conclusões e Trabalhos Futuros

Esse artigo descreve o desenvolvimento de um modelo de detecção de intrusão baseado em anomalias (POLVO-IIDS) que faz uso de redes neurais artificiais. O uso destas técnicas visa identificar atividades maliciosas, através da análise do tráfego de rede, alcançando alta taxa de acerto.

Os sistemas baseados em anomalias são caracterizados pela rapidez nas consultas (em tempo linear ao tamanho do *payload* dos pacotes). Tais sistemas não funcionam corretamente na análise de tráfego cifrado e o POLVO-IIDS não é exceção. Os resultados iniciais com o POLVO-IIDS demonstram a viabilidade do modelo e uma considerável melhora na taxa de acerto quando comparado a outros sistemas presentes na literatura.

Os dados do DARPA 98 e do KDD Cup 99 são freqüentemente usados nos testes de IIDSs baseados em anomalias e servem como parâmetro para comparação entre estes IIDSs, porém a taxa de ataques nestes dados não é natural (não correspondem a realidade atual da Internet). Cerca de 80% de todas as instâncias correspondem a ataques, geralmente com apenas uma conexão. Os dados para tráfego normal são gerados por simuladores, sem a presença de fragmentos de pacotes, pacotes desordenados, etc. O modelo do POLVO-IIDS pode ser aplicado na análise de tráfego real pois sua estrutura permite a presença de algum ruído. Em trabalhos futuros pretendemos elaborar testes com tráfego real, embora haja dificuldade em comparar os resultados obtidos com os de outras abordagens.

No modelo desenvolvido, o treinamento das redes pode acontecer de maneira constante, mesmo com algum tráfego malicioso, desde que este tráfego seja pequeno. Desta forma é possível manter o IDS atualizado na evolução do comportamento das diversas aplicações que fazem uso da Internet.

Referências

Allen, J., Christie, A., Fithen, W., McHugh, J., and Pickel, J. (2000). State of the practice of intrusion detection technologies. In *CMU/SEI-99-TR-028*, Carnegie Mellon Software Engineering Institute.

Bolzoni, D., Etalle, S., and Hartel, P. (2006). Poseidon: a 2-tier anomaly-based network intrusion detection system. In *Fourth IEEE International Workshop on Information Assurance*, pages 220–237.

Bridges, S. M. and Vaughn, R. B. (2000). Fuzzy data mining and genetic algorithms applied to intrusion detection. In *National Information Systems Security Conference (NISSC)*, Baltimore, MD.

Cannady, J. (1998). Artificial neural networks for misuse detection. In *Proceedings of the 1998 National Information Systems Security Conference (NISSC'98)*, pages 443–456, Arlington, VA.

Chen, W.-H., Hsu, S.-H., and Shen, H.-P. (2005). Application of svm and ann for intrusion detection. *Comput. Oper. Res.*, 32(10):2617–2634.

Ghosh, A., Wanken, J., and Charron, F. (1998). Detecting anomalous and unknown intrusions against programs. In *Proceedings Annual Computer Security Applications (ACSAC)*, Los Alamitos, CA.

Giacinto, G., Roli, F., and Didaci, L. (2003). Fusion of multiple classifiers for intrusion detection in computer networks.

Haijun, X., Fang, P., Ling, W., and Hongwei, L. (2007). Ad hoc-based feature selection and support vector machine classifier for intrusion detection. In *Proceedings of 2007 IEEE Conference on Grey Systems and Intelligent Services*, Nanjing, China.

Kayacik, H. G., Zincir-Heywood, A. N., and Heywood, M. I. (2003). On the capability of an som based intrusion detection system. In *Proceedings of the International Joint Conference on Neural Networks*, volume 3, pages 1808–1813.

Kohonen, T. (1988). Self-organized formation of topologically correct feature maps. *Journal of the American Society for Information Science and Technology*, pages 509–521.

Kröse, B. and van der Smagt, P. (1996). *An introduction to neural networks*. URL ftp://ftp.informatik.uni-freiburg.de/papers/neuro/ann_intro_smag.ps.gz, The University of Amsterdam.

Lee, H. D. (2001). Training a neural-network based intrusion detector to recognize novel attacks, systems, man and cybernetics. In *IEEE Transactions on IEEE Computer Press 31*, pages 294–299.

Lee, W. and Stolfo, S. (2000). A framework for constructing features and models for intrusion detection systems. 3(4):227–261.

Lee, W., Stolfo, S., and Mok, K. (1998). Mining audit data to build intrusion detection models. In *Proceedings of the fourth international conference on knowledge discovery and data mining*, New York.

Lei, J. Z. and Ghorbani, A. (2004). Network intrusion detection using an improved competitive learning neural network. In *Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR)*, pages 190–197.

Liu, G., Yi, Z., and Yang, S. (2006). A hierarchical intrusion detection model based on the pca neural networks. *Journal of the American Society for Information Science and Technology*, pages 1561–1568.

Lunt, T. (1993). Detecting intruders in computer systems. In *Proceedings of 1993 Conference on Auditing and Computer Technology*.

Luo, J. (1999). Integrating fuzzy logic with data mining methods for intrusion detection. In *M.S. Thesis*, Mississippi.

Mukkamala, R., Gagnon, J., and Jajodia, S. (2000). Integrating data mining techniques with intrusion detection methods. In *Research Advances in Database and Information Systems Security*, Boston, MA.

Mukkamala, S., Janoski, G., and Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN '02*, volume 2, pages 1702–1707.

Shyu, M., Chen, S., Sarinnapakorn, K., and Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. In *Proceedings of ICDM'03*, pages 172–179.

Stolfo, J. S., Wei, F., Lee, W., Prodromidis, A., and Chan, P. K. (1999). Kdd cup data - knowledge discovery and data mining competition (1999).

Wang, H., Huang, J. Z., Qu, Y., and Xie, J. (2004). Web services: problems and future directions. *J. Web Sem.*, 1(3):309–320.

Xiang, C. and Lim, S. M. (2005). Design of multiple-level hybrid classifier for intrusion detection system. In *Proceedings of 2005 IEEE Workshop on Machine Learning for Signal Processing*, pages 117–122.

Zanero, S. and Savaresi, S. M. (2004). Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the ACM symposium on Applied computing*, pages 412–419, Nicosia, Cyprus.