Identificação de Ataques em Redes de Computadores usando Comitê de Classificadores

Jorge L. de Castro e Silva[1], José Everardo B. Maia[1], Nelson Luis S. da Fonseca[2]

[1]Departamento de Estatística e Computação – Universidade Estadual do Ceará (UECE)
Av. Paranjana, 1700 – 60.740-903 – Fortaleza – CE – Brasil

[2]Instituto de Computação – Universidade de Campinas (UNICAMP)

jlcs@larces.uece.br, jmaia@uece.br, nfonseca@ic.unicamp.br

Abstract. *Monitoring the traffic profile of networks is a promising approach for detecting attacks on computer networks. The occurrence of attacks is usually detected when the traffic properties does not following its historical normality. In these cases, the identification of threats from some traffic properties is a challenging problem considering that: i) some attacks are masqueraded in the normal traffic; and ii) others types have a behavior very similar. In this paper the problem of attacks detection is formulated as a problem of networks traffic flow classification. The approach chosen uses classifiers committee. This approach was tested and the results are promising, reaching an average hit rate 94%, of success for identification of attacks.*

Resumo. *O monitoramento do perfil do tráfego é uma abordagem promissora para detectar ameaças nas redes. A ocorrência de ataques é percebida quando as propriedades do tráfego fogem da sua normalidade histórica. Neste contexto, a identificação de ameaças a partir de algumas propriedades do tráfego é um problema desafiador considerando que alguns ataques são mascarados no tráfego normal e outros têm perfis de comportamento muito parecidos entre si. Neste trabalho o problema de detecção de ataques é formulado como um problema de classificação de fluxos de tráfego. A idéia é utilizar um comitê de classificadores. Esta abordagem foi testada e os resultados obtidos na identificação de ataques são promissores, alcançando uma taxa média de acerto de 94%.*

1. Introdução

A identificação de ameaças e ataques as redes de computadores através da observação de fluxos de tráfego é importante para monitoração, gerenciamento e provisão de QoS das redes [Nguyen e Armitage, 2008]. Métodos populares de identificação baseados em análise do número da porta e no conteúdo dos pacotes estão de alguma forma limitados. O reconhecimento de ataques baseado no conteúdo de pacotes utiliza padrões específicos de *bytes* (assinatura), contidos nos pacotes, para identificar ameaças [Moore e Papagiannaki, 2005]. Este método pode identificar ataques somente quando as assinaturas são conhecidas. Leis de privacidade e criptografia podem tornar inacessível o conteúdo dos pacotes, limitando esta abordagem.

Aprendizagem de máquina (**ML**) [Mitchell, 1997] é uma alternativa promissora para identificar ataques com base nas estatísticas dos fluxos de tráfego da rede. As estatísticas de fluxos de tráfego são providas por atributos que independem do conteúdo dos pacotes, tais como tamanho do pacote e intervalo de tempo entre chegadas. Essa independência é a maior vantagem do emprego de ML. Cada fluxo de tráfego é caracterizado pelo mesmo conjunto de atributos. Um classificador ML é construído através do treinamento de um conjunto representativo de instâncias de fluxos que podem conter ataques conhecidos. Após seu treinamento, o classificador poderá ser usado para determinar classes de fluxos desconhecidos.

Técnicas de ML podem ser agrupadas em duas categorias: abordagem não supervisionada, que inclui os algoritmos EM [McGregor, Hall e Lorier, 2004], AutoClass [Zander, Nguyen e Armitage, 2005] e K-Means [Erman, Mahanti e Arlitt, 2007]; e abordagem supervisionada, que abrange os algoritmos Naive Bayes [Moore e Zuev, 2005], [Park, Tyan e Kuo, 2006], Redes Bayesianas [Nguyen e Armitage, 2006] e Árvore de Decisão [Bonfiglio, Mellia e Meo, 2007], entre outros. Existem outras abordagens como a semi-supervisionada e a aprendizagem por reforço [Duda, Hart e Stork, 2001].

Em geral, as técnicas de ML para classificação de tráfego e identificação de ataques empregam somente um classificador. Todavia, a acurácia conseguida por um único classificador está limitada, uma vez que o algoritmo de aprendizagem buscará somente em um espaço de hipóteses para identificar a melhor classe. Quando a quantidade de dados de treinamento disponível é muito pequena, comparada ao tamanho do espaço de hipóteses, o algoritmo de aprendizagem geralmente não encontra a melhor solução. Uma alternativa para o problema é empregar vários algoritmos de aprendizagem para construir múltiplos classificadores.

Comitê de Máquina (**MC**) é um paradigma de aprendizagem que obtém uma solução, para um dado problema, a partir de saídas individuais propostas por múltiplas soluções alternativas para o mesmo problema, chamadas de componentes do comitê [Haykin, 1999]. Este paradigma representa uma das principais direções da pesquisa em aprendizado de máquina e tem sido aplicado em vários problemas práticos, inclusive classificação de padrões.

Neste artigo, o problema de detecção de ataques é formulado como um problema de classificação de fluxos de tráfego de redes. A abordagem escolhida utiliza um comitê de classificadores, no qual diferentes algoritmos são empregados para solução do problema, e o resultado é obtido pela combinação dos resultados dos classificadores individuais. Este trabalho também analisa o desempenho computacional de técnicas de comitês de máquinas aplicadas em tráfego malicioso. A principal contribuição do artigo é o emprego de comitês de classificadores na identificação de ataques, utilizando uma combinação de algoritmos de aprendizagem não aplicada em outros trabalhos, e a avaliação de desempenho tanto dos comitês quanto dos algoritmos individualmente. O trabalho também compara o desempenho computacional dos comitês com o desempenho de algoritmos individuais de classificação.

Este artigo é estruturado como segue. A Seção 2 apresenta os trabalhos relacionados. Seção 3 sumariza os conceitos de aprendizagem de máquina, incluindo comitê de máquina. Seção 4 discute a abordagem empregada, incluindo detalhes sobre

conjunto de dados, atributos e algoritmos usados. A Seção 5 apresenta os principais resultados e a Seção 6 conclui e propõe futuros trabalhos.

2. Trabalhos Relacionados

A literatura apresenta algumas abordagens para classificação de tráfego e identificação de ataques, mas nenhuma delas trabalha bem para todos os tipos de tráfego. O sítio web da CAIDA [CAIDA, 2011] agrupa os artigos sobre classificação de tráfego em 5 categorias: descrição analítica (*survey*), análises, metodologias, ferramentas e outras. Trabalhos de análise buscam números confiáveis sobre a composição do tráfego, enquanto artigos sobre metodologias focam nos métodos de classificação.

Os seguintes artigos se enquadram na categoria *survey*. Zhang et al. (2009) mostram o estado da arte em classificação de tráfego; Nguyen e Armitage (2008) apresentam uma descrição analítica em classificação e identificação de tráfego Internet usando aprendizagem de máquina e Kim et al. (2008) analisam a desmistificação da classificação do tráfego Internet.

Os trabalhos a seguir são mais recentes e estão na categoria de análises. Maier et al. (2009) examinam as características dominantes no tráfego Internet de banda larga residencial e Pietrzyk et al. (2009) fazem uma análise do tráfego ADSL (*Asymmetric Digital Subscriber Line*) através de estatísticas de classificação.

Os principais artigos que tratam das metodologias em classificação utilizam modelos de Markov [Mitchell, 1997], heurísticas para classificar tráfego de *backbone* Internet [John e Tafvelin, 2008] e técnicas de GMM (*Gaussian Mixture Model*) [Dusi, Gringoli e Salgarelli, 2008] para identificar aplicações que rodam em sessões SSH (*Secure Shell*). Alguns novos trabalhos que estão na categoria ferramentas, tais como a plataforma TIE [Dainotti et al. 2009] e a ferramenta TDG [Iliofotou et al., 2007] usam grafos de dispersão para analisar tráfego de rede.

Outros trabalhos recentes empregam classificadores baseados na similaridade de fluxos de tráfego [Chung et al., 2009], classificadores que executam em tempo real [Wang e Yu, 2009] e classificações de tráfego baseadas em comitês de máquina [HaiTao et al., 2009].

Alguns artigos discutem a identificação de ataques nas redes através de técnicas de aprendizagem de máquina. O principal trabalho nesta área tem sido conduzido pela Universidade de Twente, Holanda, comparando o desempenho de algoritmos de aprendizagem de máquina para detecção de intrusão em redes [Jalil, Kamarudim e Masrek, 2010] e pesquisando a detecção de intrusão baseada em fluxos IP [Sperotto et al., 2010].

3. Comitê de Máquina e Classificação de Tráfego

O problema de classificação de tráfego será formulado neste trabalho a partir de um conjunto de dados *D*. Seja $D = \{(x_i, y_i)\}, i = 1,...,n,$ um conjunto rotulado de exemplos de treinamento, onde x_i é um vetor que representa os valores dos atributos da i-ésima instância e y_i é um índice indica a classe de x_i. Seja $U = \{(x_i)\}, i = 1,...,n.$ um conjunto não rotulado de exemplos. O objetivo da classificação de tráfego é definir uma

função de mapeamento $f : X \rightarrow Y$ treinada com o conjunto de dados *D*, que atribui cada $x_i \in U$ uma predição correta da classe predefinida y_i. Os valores de x_i são vetores da forma $(x_{i1}, x_{i2}, ..., x_{ij})$ com *j* atributos de um fluxo de tráfego e os valores de y_i são extraídos de um conjunto discreto de classes, $y_1, y_2, ..., y_k$. Seja *C* um classificador definido (modelado) pelo treinamento de um algoritmo *L* nos exemplos de treinamento de *D*. O classificador *C* atribui um valor de classe y_i para cada instância de *U*. Um número finito *M* de classificadores serão denotados neste trabalho por $C_1, ..., C_M$.

Em um comitê, *M* classificadores são combinados de alguma forma e agem como se fossem um único classificador. O modelo pode ser descrito como segue. Seja C_j, $j = 1, ..., M$, um classificador do conjunto de dados *U* em *k* classes. A combinação de *M* diferentes classificadores usados para classificar *U* em *k* classes compõe um comitê de classificadores. As saídas do comitê serão denotadas por $S(.)$. Portanto, a mistura de múltiplos componentes do comitê pode ser vista como a determinação de $S(U)$, através do uso das saídas de $C_j(U)$ de todos *M* componentes.

Duas abordagens usadas na composição de comitê de classificadores são as técnicas de Votação (*Voting*) e a de Empilhamento (*Stacking*) [Wolpert, 1992]. A técnica de Empilhamento trabalha com um classificador no nível meta (nível 1) para reunir as previsões de classe dos múltiplos classificadores do nível base (nível 0). O sucesso desta técnica está na capacidade de explorar as várias previsões dos classificadores de nível base para realizar a previsão da classe no nível meta com mais precisão. Em contrapartida, a técnica de Votação não realiza nenhum processo de aprendizagem para previsão da classe e é usada simplesmente como um *baseline* para comparação com a técnica de Empilhamento. Entretanto, outras abordagens de comitês, como Boosting [Freund e Schapire, 1996], não são tratadas neste trabalho.

3.1. Técnica de Votação

Seja $C_1, ..., C_M$ o conjunto de classificadores modelados pelo treinamento de *M* algoritmos de aprendizagem diferentes, $L_1, ..., L_M$, em um conjunto de dados *D*. A classificação de uma nova instância em tempo de execução requer uma consulta aos classificadores $C_1, ..., C_M$ para um valor de classe. A classe com maior quantidade de votos será selecionada. Esta técnica é conhecida como votação majoritária. Variações desta técnica incluem a votação majoritária ponderada e a votação usando distribuições de probabilidade das classes [Dietterich, 2000]. Na primeira abordagem, o voto de cada classificador é ponderado pela sua precisão. Na abordagem probabilística, cada classificador gera um vetor de distribuição de probabilidade de todas as classes relevantes. Para cada classe, é calculada a média dos valores de probabilidade individual de todos classificadores e a classe com valor máximo será selecionada.

3.2. Técnica de Empilhamento

A idéia central desta técnica proposta por Wolpert (1992) é usar um classificador no nível meta (nível 1) que consegue aprender com as saídas dos vários classificadores de nível base (nível 0), estimando as classes através da validação cruzada dos dados.

Seja D um conjunto de dados, chamado de dados de nível 0 (base), consistindo de vetores de atributos, e $L_1,...,L_M$ um conjunto de M algoritmos de aprendizagem diferentes. Durante um processo de validação cruzada, D é dividido aleatoriamente em J partes disjuntas, $D_1,...,D_j$, de igual tamanho. Em cada uma das partes dos exemplos de treinamento de D, $D_1,...,D_{j-1}$, aplicam-se M algoritmos de aprendizagem, $L_1,...,L_M$, para treinar os classificadores $C_1(j),...,C_M(j)$. Estes classificadores, após treinados, são aplicados na parte de teste de D, D_j. As classes previstas pelos classificadores treinados na parte de teste, em cada vetor de atributos x_i de D_j, concatenadas com o valor de classe original, $y_i(x_i)$, formam um novo conjunto de dados, chamado de dados de nível meta (nível 1), MD_j.

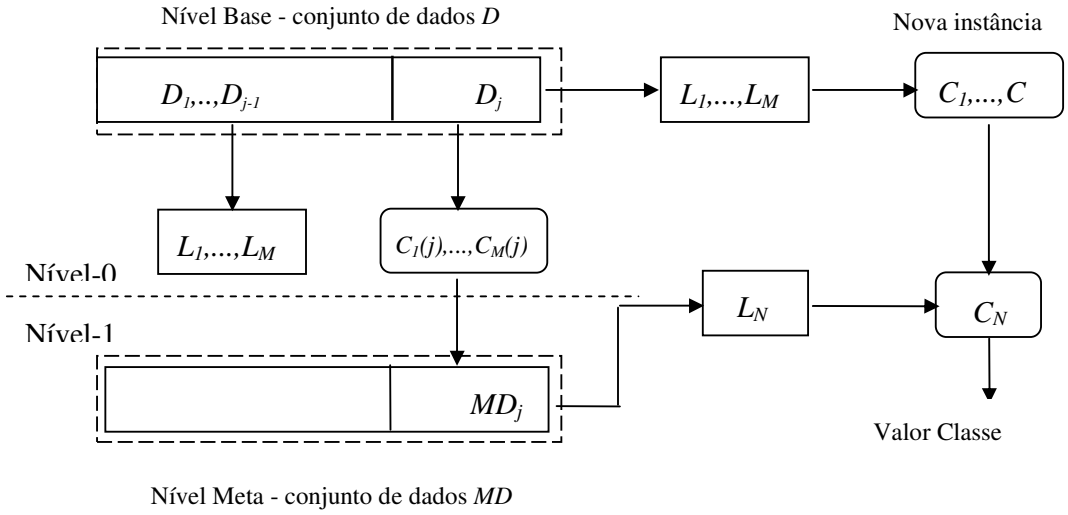

Figura 1. Nível-0 – Processo de validação cruzada para criação do conjunto de dados no nível meta. Nível-1 – Processo de Empilhamento no conjunto *MD*.

Ao fim do processo de validação cruzada, o conjunto de dados *MD*, formado pela união dos MD_j ($MD = \cup MD_j$) constitui o novo conjunto de dados no nível meta. Em seguida aplica-se no conjunto *MD* um algoritmo de aprendizagem L_N para gerar o classificador C_N (nível meta). O algoritmo L_N pode ser um dos algoritmos de $L_1,...,L_M$ ou um diferente. Por fim, os algoritmos de aprendizagem $L_1,...,L_M$ são aplicados ao conjunto inteiro de dados D para gerar os classificadores finais de nível base, $C_1,...,C_M$, que serão usados em tempo de execução. Para classificar uma nova instância, as previsões concatenadas de todos classificadores de nível base, $C_1,...,C_M$, formam um

vetor de nível meta que é atribuído um valor de classe pelo classificador de nível meta C_N. A Figura 1(Nível-0) ilustra a metodologia da validação cruzada e o Nível-1 mostra o processo de empilhamento aplicado ao conjunto *MD*.

4. Abordagem Experimental

Este artigo compara a acurácia e a taxa de erro de 5 tipos de ataques usando comitês de classificadores compostos pelos algoritmos C4.5 (Árvore de Decisão), RBF (Redes de Funções de Base Radial), BayesNet (Redes Bayesianas). Os comitês também são comparados com esses três algoritmos individualmente. O algoritmo C4.5 [Quinlan, 1993] constrói árvores de decisão usando o conceito de entropia da informação. O algoritmo RBF [Yee e Haykin, 2001] usa uma Gaussiana normalizada e o algoritmo de agrupamento K-Means para prover as funções de base. As Redes Bayesianas [Russel e Norvig, 2003] utilizam grafos acíclicos direcionados cujos nós representam variáveis aleatórias. A escolha desses 3 algoritmos para compor o comitê deve-se ao fato de que já foram bem testados em outros trabalhos, apresentando desempenho superior quando comparados com outros algoritmos de aprendizagem.

As técnicas de Votação e de Empilhamento são usadas na formação dos comitês de classificadores. Este artigo também compara o desempenho computacional, ou seja, o tempo de execução de cada comitê em relação aos tempos individuais de cada algoritmo. Pesquisas recentes [Wang e Yu, 2009] em classificação de tempo real procuram investigar o desempenho computacional de classificadores quando submetidos a milhares de pacotes simultâneos.

Este trabalho usou o MATLAB [Hanselman e Littlefield, 2001], versão 2010b, para teste e treinamento dos classificadores. Os programas (*scripts*) MATLAB criados chamam métodos Java do pacote Weka [Witten et al., 1999], versão 3-7-2.

4.1. Traços de Dados

Este trabalho usou um conjunto de dados de tráfego de ataque que pode ser obtido na url http://traces.simpleweb.org/netflow/netflow2/. Os dados ocupam 14,2 MB que corresponde a 14.170.132 instâncias de fluxos de tráfego, sendo 99% delas identificadas e rotuladas. Os dados, descritos em [Sperotto et al., 2009], foram gerados em um computador de acesso aberto (*honeypot*) instalado na Universidade de Twenty em setembro de 2008. O *honeypot* foi conectado diretamente a Internet para prover os seguintes serviços: FTP na portas 20 e 21, SSH na porta 22 e WWW-http na porta 80.

Os dados usados estão no formato de fluxo e foram capturados em um roteador com o serviço de NetFlow [Cisco, 2011] habilitado. Os dados consistem de incidentes de tráfego de ataque devidamente reconhecidos e rotulados. Informações sobre os fluxos e seus atributos estão disponíveis em http://traces.simpleweb.org/netflow/netflow2/dataset_description.txt.

Cada instância do conjunto de dados contém 28 atributos. Dentre esses atributos, este trabalho selecionou apenas 5 para fins de classificação. Os atributos selecionados foram: packts (número de pacotes), octets (número de octetos), flow duration (duração do fluxo), tcp_flag e prot (tipo de protocolo). Além destes, selecionou-se também o atributo que indica o tipo de ataque e corresponde à classe da instância, chamado de

alert_type (tipo de alerta). A literatura da área mostra que apenas esses 5 atributos são suficientes para obter um alto percentual de acerto.

Os seguintes valores são possíveis para as classes identificadas pelo atributo alert_type: ssh_conn, http_conn, authident_sideeffect, irc_sideefect, icmp_sideeffect e not_identified. O Quadro 1 mostra as classes que correspondem a incidentes identificados e suas descrições.

Classes	**Descrição**
ssh_conn	Tráfego de ataque de uma conexão *ssh*
http_conn	Tráfego de ataque de uma conexão *http*
authident_sideeffect	Tráfego direcionado para o serviço de autenticação a partir de tentativas de conexões *ssh* e *ftp*
irc_sideeffect	Tráfego gerado por um proxy *irc* instalado no *honeypot*
icmp_sideeffect	Tráfego gerado pelo serviço *icmp* a partir de tentativas de conexões *ssh* e *ftp*
not identified	Ataque não identificado ou tráfego normal

Quadro 1. Classes

A distribuição das amostras para treinamento e testes dos algoritmos pode ser proporcional as frequencias que cada tipo de classe aparece nos dados ou então igual (uniforme) para todos os tipos. A primeira alternativa parece mais razoável, embora o número de amostras tipo 'ssh_conn' seja dominante (98% do conjunto de dados utilizado). Neste caso, o resultado da classificação poderia ser tendencioso devido uma classe aparecer com mais frequencia nas amostras de treinamento. Consequentemente, classes com poucos exemplos poderiam não ser corretamente identificadas. A segunda alternativa, caracterizada por conter amostras com igual número de exemplos de tipo de classe, também chamada de amostra balanceada, não afeta a precisão dos algoritmos classificadores.

Baseado nessas considerações, este artigo utiliza as duas alternativas de distribuição de amostras mencionadas. Foram coletadas duas amostras de dados, cada uma com 35.968 instâncias. A primeira amostra foi escolhida aleatoriamente e consiste de 6.000 exemplos de cada classe, exceto a classe 'not_identified', que tem um total de apenas 5.968 exemplos. Esta é a amostra balanceada. A segunda amostra, também escolhida aleatoriamente, é composta de exemplos com quantidades proporcionais à quantidade total de cada classe do conjunto de dados.

28.774 e 7.194 instâncias foram usadas para treinamento e teste de cada amostra, correspondendo a 80% e 20% de amostras, respectivamente. A composição de percentuais diferentes para a amostra de treinamento e de teste (20%, 40%, 60% e 80%) foi também usada para verificar sua influência nos modelos de classificação. A média da taxa de acerto foi obtida pela média de 10 testes para cada algoritmo classificador.

4.2. Configuração de Parâmetros

Os seguintes parâmetros foram usados na configuração dos classificadores individuais. O classificador RBF utilizou 2 clusters para o algoritmo K-Means prover as funções de base e o centróide. A árvore de decisão C4.5 foi podada com fator de confiança 0,25 e com duas instâncias por folha. A aprendizagem das Redes Bayesianas usou o algoritmo subida da encosta (*Hill Climbing*) para busca das estruturas da rede e um algoritmo simples para encontrar as probabilidades condicionais.

A combinação desses três algoritmos individuais (RBF, C45 e BayesNet) foi usada para formar o comitê de classificadores e os parâmetros a seguir foram usados na parametrização dos comitês. A votação majoritária foi escolhida como regra de combinação para a técnica de Votação e o algoritmo de árvore de decisão rápida foi escolhido como meta-classificador para a técnica de Empilhamento.

5. Resultados e Análises

5.1. Comparação de Desempenho

A Tabela 1 mostra os percentuais médios de respostas corretas (classes corretas) obtidos pelo experimento para os algoritmos RBF, C45 e BayesNet e para os comitês de Votação e Empilhamento. O experimento avaliou amostras com 20%, 40%, 60% e 80% de rótulos não classificados para cada algoritmo e para cada comitê. Individualmente, o algoritmo C45 apresentou o maior percentual de respostas corretas e o RBF o menor. Coletivamente, o comitê de Votação apresentou percentuais abaixo do C45. Contudo, este comitê é usado apenas como *baseline* para comparar com o desempenho da técnica de Empilhamento. Observou-se que os percentuais de acertos do comitê de Empilhamento estão acima dos outros, mostrando aproximadamente um percentual 2,4% superior aos três outros algoritmos.

Embora esta diferença de percentual seja pequena, a literatura mostra que existe um nível de saturação quando a percentagem de acurácia está próxima de 100%. O uso de um conjunto de dados com muitas instâncias (14.170.132) e de uma amostra aleatória com 35.968 instâncias pode também ter contribuído para este aumento de percentagem. Souza, Campos e Campos (2008) mostram que a seleção simultânea de instâncias e atributos é uma forma de aumentar o desempenho de classificadores quando empregados em conjunto de dados com muitas instâncias.

Tabela 1. Taxas médias de acerto dos classificadores

Percentual não rotulado	Aprendizagem Supervisionada (%)			Aprendizagem por Comitê (%)	
	Algoritmos individuais			Comitê com os três algoritmos (RBF, C45 e BayesNet)	
	RBF	C4.5	BayesNet	Votação	Empilhamento
20%	87.2950	94.1201	92.7579	93.3278	94.0923
40%	86.7728	94.2448	92.8477	93.4038	94.2726
60%	90.1117	94.1384	92.8641	93.4896	94.1708
80%	89.0318	93.8938	91.8677	92.1526	93.9529

Esperávamos taxas de acerto menores para o conjunto de dados com mais instâncias não rotuladas, portanto menos instâncias para treinamento. No entanto, a taxa

de acerto do algoritmo RBF aumentou de 87% para 89%, aproximadamente. Este resultado inesperado pode ser devido à amostra escolhida.

Vale ressaltar que as variâncias das médias das taxas de acerto apresentaram valores próximos de zero, mostrando um comportamento homogêneo dos dados, e por isso este trabalho não apresenta os intervalos de confiança para as taxas médias da Tabela 1.

A Tabela 2 apresenta o desempenho dos algoritmos individuais e dos comitês usando a métrica ROC (*Receiver Operating Characteristic*). ROC é uma métrica para comparar desempenho de sistemas, representada pela área sob a curva ROC, que está baseada em taxas verdadeira positiva e falsa positiva. Quanto mais o valor desta métrica estiver próximo de 1, melhor o desempenho do classificador. Comparando com a tabela anterior, observa-se que o classificador baseado no comitê de Empilhamento e os algoritmos C45 e BayesNet apresentaram melhores medidas ROC.

Tabela 2. Comparação das valores ROC dos classificadores

Percentual não rotulado	Aprendizagem supervisionada			Aprendizagem com Comitês	
	Algoritmos individuias			Comitês com os três algoritmos	
	RBF	C4.5	BayesNet	Votação	Empilhamento
20%	0.971	0.990	0.989	0.960	0.989
40%	0.974	0.991	0.990	0.960	0.990
60%	0.978	0.990	0.990	0.961	0.990
80%	0.976	0.988	0.989	0.953	0.988

A Tabela 3 mostra a matriz de confusão da técnica de Empilhamento usada para classificar o conjunto de dados com 20% de instâncias não rotuladas. A diagonal principal da matriz mostra o número de instâncias classificadas corretamente. Observa-se que a classe 'not identified' (coluna f) tem um grande número de instâncias classificadas erroneamente, enquanto que outras classes apresentam poucos falsos positivos.

Tabela 3. Matriz de confusão da técnica de Empilhamento

a	b	c	d	e	f	← classificado como
1128	1	2	1	0	68	**a = ssh_conn**
1	1199	0	1	0	6	**b = http_conn**
1	0	1174	0	0	3	**c = authident_sideeffect**
0	0	0	1182	0	2	**d = irc_sideeffect**
0	0	0	0	1183	47	**e = icmp_sideeffect**
5	13	167	26	85	899	**f = not_identified**

A Figura 2 ilustra a comparação dos três algoritmos com os dois comitês. O gráfico mostra a taxa de acerto no eixo Y e o percentual de amostras não classificadas no eixo X. Observa-se no gráfico que o classificador RBF tem o pior desempenho, enquanto que o algoritmo C45 e o comitê de Empilhamento apresentam as maiores taxas de acerto. A linha do gráfico do algoritmo C45 está quase sobreposta a linha do comitê de Empilhamento devido a escala do eixo Y. O gráfico também mostra que a taxa de acerto diminui em todos classificadores quando se utiliza o conjunto de dados

com 80% de instâncias não classificadas. Houve aumento da taxa de acerto do algoritmo RBF quando se aumentou de 40% para 60% de instâncias não rotuladas. Esperava-se uma menor taxa de acerto quando os dados de treinamento diminuíssem. Este resultado inesperado pode tanto ser devido às amostras escolhidas quanto a aleatoriedade das instâncias.

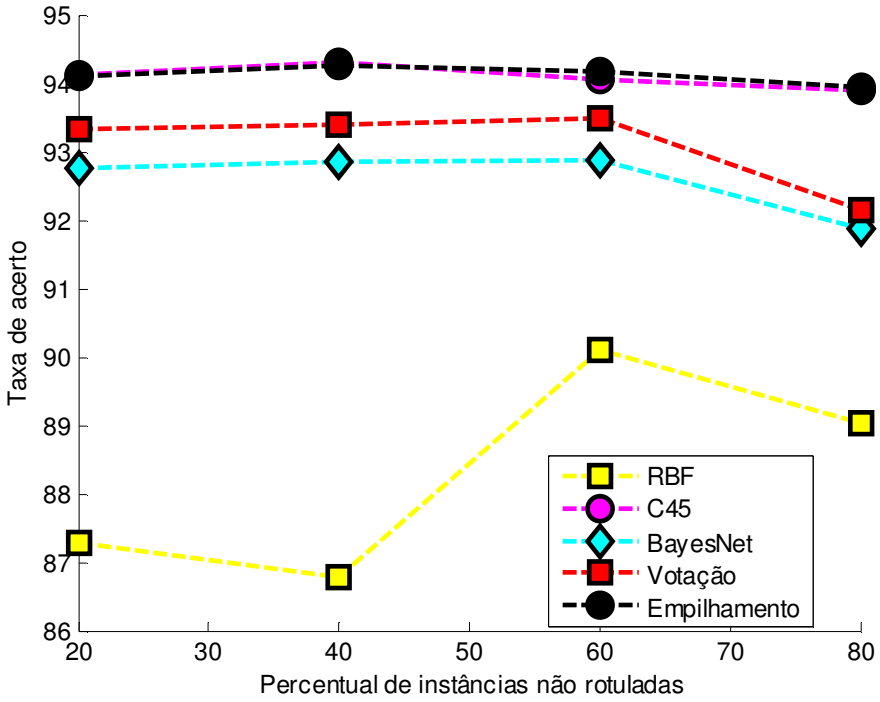

Figura 2. Comparação da acurácia dos algoritmos e comitês

5.2. Desempenho Computacional

Os testes foram realizados em um único computador com processador de 2 núcleos, 1,3 GHz de frequencia e 4 GB de memória RAM. A análise de desempenho computacional dos classificadores é importante quando se considera a classificação em tempo real. Este trabalho investigou os tempos de execução dos comitês e dos algoritmos de classificação quando submetidos a milhares de fluxos simultâneos.

A Figura 3 ilustra a velocidade (tempo) de cada algoritmo de classificação e de cada comitê, quando testado com 20%, 40%, 60% e 80% de instâncias não rotuladas. O gráfico mostra o tempo em segundos no eixo Y e o percentual de amostras não classificadas no eixo X. Embora existam pequenas diferenças nos resultados quando se considera o desempenho dos classificadores usando as medições de acurácia, as medidas de tempo de execução mostraram diferenças significativas. Por exemplo, o comitê de Empilhamento é o mais lento, tomando cerca de 12 segundos para classificar 20% das amostras sem rótulo e quase 50 segundos para classificar 80% das amostras não classificadas. Entretanto, o algoritmo C4.5 é mais rápido quando comparado com os outros, levando menos de 1 segundo para classificar. Os demais algoritmos também apresentaram desempenho semelhante ao C45.

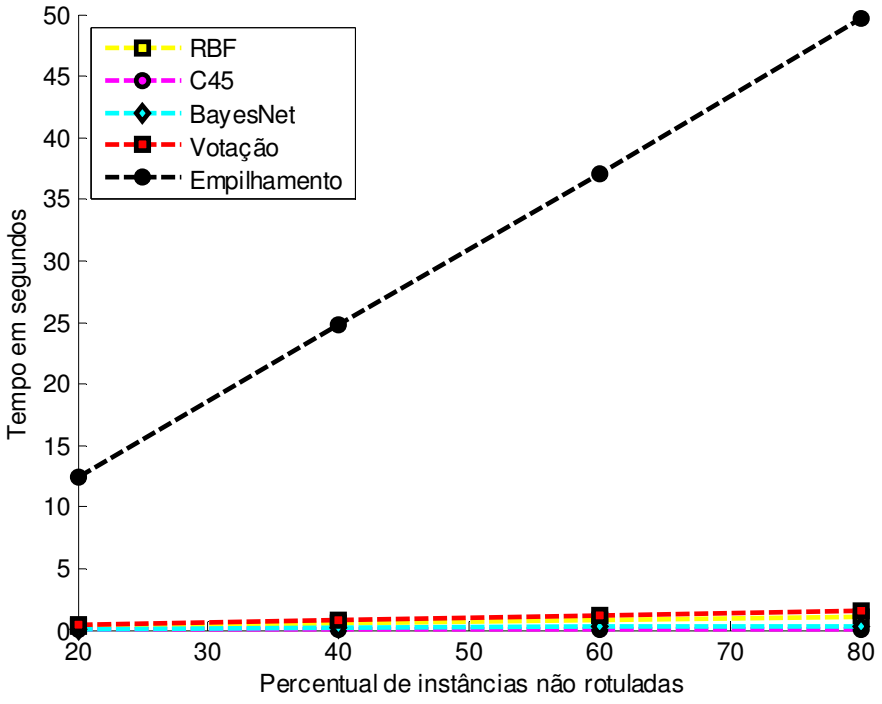

Figura 3. Desempenho computacional dos algoritmos e comitês

6. Conclusões

Este artigo propõe o uso de comitê de classificadores para identificação de ataques e compara o desempenho desses comitês com os algoritmos individuais (RBF, C4.5 e BayesNet). Os resultados mostram que a técnica de Empilhamento apresentou maior taxa de acerto. No entanto, embora o percentual médio de predições corretas obtidas pela técnica de Empilhamento seja somente 2,4% maior que os demais, esta técnica poderá atingir um melhor desempenho através da seleção de instâncias e atributos mais apropriados.

Este trabalho também analisou o desempenho computacional dos comitês de classificação aplicado em tráfego malicioso. Observou-se que o tempo de execução do comitê de Empilhamento é maior que os tempos das abordagens com um único algoritmo, estando 27 vezes maior que o tempo gasto pelo algoritmo C45. Não podemos afirmar que esta técnica é apropriada ou não para classificação em tempo real. As características do computador usado nos experimentos e as linguagens utilizadas nos testes (MATLAB e Java) não contribuíram para um melhor desempenho. Contudo, considerando os resultados do tempo de execução, o gerente de rede poderia fornecer um tempo limite (*threshold*) para decidir sobre o uso desta técnica.

Embora a técnica de Empilhamento tenha apresentado melhor desempenho que os algoritmos individuais, como RBF e BayesNet, alguns experimentos poderiam ser realizados em trabalhos futuros, tais como a seleção de atributos e instâncias adequadas para a classificação com base em comitês e testes de novos comitês e de novas técnicas de fusão de algoritmos.

Referências

Bonfiglio, D., Mellia, M., and Meo, M. (2007) "Revealing Skype Traffic: When Randomness Plays with You". In: SIGCOMM'07, New York, NY, USA, p. 37-338.

CAIDA - The Cooperative Association for Internet Data Analysis (2011). "An overview of traffic classification", http://www.caida.org/research/traffic-analysis/classification-overview.

Chung, J., Park, B., Won, Y.J., Strassner, J. and Hong, J.W. (2009) "Traffic Classification Based on Flow Similarity", IPOM 2009, LNCS 5843, p. 65–77.

Cisco.com (2011), "Cisco IOS NetFlow Configuration Guide, release 12.4", http://www.cisco.com, .november.

Dainotti, A., Donato, W., Pescape, A. and Ventre, G. (2009) "TIE: a Community-oriented Traffic Classification Platform". In: TMA'09, International Workshop on Traffic Monitoring and Analysis.

Dainotti, A., Donato, W., Pescape, A. and Rossi, P. S. (2008) "Classification of Network Traffic via Packet-Level Hidden Markov Models". In: IEEE GLOBECOM,.

Dietterich, T. G. (2000) "Ensemble Methods in Machine Learning". In: Proceedings International Workshop on Multiple Classifier Systems (MCS), LNCS 1857, p. 1-15, Italy, Springer.

Duda, R. O., Hart, P. and Stork, D. (2001) Pattern Classification. 2nd Edition, Wiley.

Dusi, M., Gringoli, F. and Salgarelli, L. (2008) "A Preliminary Look at the Privacy of SSH Tunnels". In: ICC'08, IEEE International Conference on Communications.

Erman, J., Mahanti, A. and Arlitt, M. (2007) "Identifying and discriminating between Web and Peer to Peer Traffic in the Network Core". In: WWW'07, Banff, Alberta, Canada.

Freund, Y. and Schapire, R., E. (1996) "Experiments with a new boosting algorithm". Machine Learning: In Proceedings of the Thirteenth International Conference, Bari, Italy, p. 148-156.

HaiTao, H., XiaoNan, L., FeiTeng, M., ChunHui, C. and JianMin, W. (2009) "Network Traffic Classification based on Ensemble Learning and Co-training", Science in China SeriesF: Information Sciences, Springer.

Hanselman D. and Littlefield B. (2001) Mastering MATLAB 6, A Comprehensive Tutorial and Reference, , Prentice-Hall.

Haykin, S. (1999). Neural networks – A comprehensive foundation, 2nd. Ed., Prentice Hall.

Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M. Singh, S., and Varghese, G. (2007) "Network Traffic Analysis using Traffic Dispersion Graphs (TDGs): Techniques and Hardware Implementation". In: IMC '07, Proceedings of the 7th ACM SIGCOMM conference on Internet measurement.

Jalil, K. A., Kamarudim, M. H. and Masrek, M. N. (2010) "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion". In: ICNIT'10, International Conference on Networking and Information Technology.

John, W. and Tafvelin, S. (2008) "Heuristics to Classify Internet Backbone Traffic based on Connection Patterns". In: ICOIN'08, International Conference on Information Networking.

Kim, H., Claffy, K.C., Formenkov, M., Barman, D., Faloutsos, M. and Lee, K. (2008) "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices". In: CoNEXT '08 Proceedings of the ACM CoNEXT Conference, New York, USA.

Maier, G., Feldmann, A., Paxson, V., and Allman, M. (2009) "On Dominant Characteristics of Residential Broadband Internet Traffic". In IMC'09.

McGregor, A., Hall, M. and Lorier, P. (2004) "Flow Clustering using Machine Learning Techniques". In: PAM'04 5th International Conference on Passive and Active Measurement, p. 205-214.

Mitchell, T. M. (1997) Machine Learning. McGraw-Hill Education (ISE Editions).

Moore, A. and Papagiannaki, K. (2005) "Toward the Accurate Identification of Network Applications". In PAM'05 6th International Conference on Passive and Active Measurement, p. 41-54.

Moore, A. W. and Zuev, D. (2005) "Internet Traffic Classification using Bayesian Analyses Techniques". In: ACM SIGMETRICS 2005, Banff, Alberta Canada, p. 50-60.

Nguyen, T. and Armitage, G. (2008) "A Survey of Techniques for Internet Traffic Classification using Machine Learning". IEEE Communications Surveys and Tutorials, 10 (4), p. 56-76.

Nguyen, T. and Armitage, G. (2006) "Training on Multiple Sub-Flows to Optimize the Use of Machine Learning Classifiers in Real-World IP Networks". In: LCN 2006, Tampa, Florida, USA, p. 369-376.

Park, J., Tyan, H-R. and Kuo, C-C (2006) "Internet Traffic Classification for Scalable Qos Provision". In: 2006 IEEE International Conference on Multimedia and Expo., Toronto, Ontario, Canada, p. 1221-1224.

Pietrzyk, M., Costeux, J-L., Urvoy-Keller, G. and En-Najjary, T. (2009) "Challenging Statistical Classification for Operational Usage: the ADSL Case". In IMC'09.

Quinlan, R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

Russell, S. and Norvig, P. (2003) Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall.

Sperotto, A., Sadre, R., van Vliet, F. and Pras, A. (2009) "A Labeled Data Set For Flow-based Intrusion Detection". In: IPOM´09, 9th IEEE International Workshop on IP Operations and Management.

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B. (2010) “An Overview of IP Flow-Based Intrusion Detection”. In: IEEE Communications Surveys & Tutorials, 12 (3). p. 343-356.

Souza, J. T., Campos, R. A. and Campos, G.A.L. (2008) “A Novel Approach for Integrating Feature and Instance Selection”. In Proceedings of the International Conference on Machine Learning and Cybernetics, p. 374-379.

Wang, Y. and S-Zheng Yu. (2009) “Supervised Learning Real-time Traffic Classifiers”. Journal of Networks, Vol. 4, No. 7, Academy Publisher.

Witten, I. H. , Frank, E., Trigg, L., Hall, M., Holmes, G. and Cunningham, S. (1999) “Weka: Practical Machine Learning Tools and Techniques with Java Implementations”. In: Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems. p. 192–196.

Wolpert, D. H. (1992) “Stacked generalization”. Neural Networks, vol 5, no 3, p. 241-259.

Yee, P. V. and Haykin, S. (2001) Regularized Radial Basis Function Networks: Theory and Applications. John Wiley.

Zander, Nguyen, T. and Armitage, G. (2005) “Automated Traffic Classification and Application Identification using Machine Learning”. In: LCN 2005, Sidney, Australia, p. 250-257.

Zhang, M., John, W., Claffy, K.C., and Brownlee, N. (2009) “State of the Art in Traffic Classification: A Research Review”. In: PAM '09: 10th International Conference on Passive and Active Measurement, Student Workshop.