Detecção de Negação de Serviço Utilizando Algoritmo de Aprendizado por Quantização Vetorial para Base de Dados Rotulados

Daniel T. Bastos, Halana R. P. Camuci, Kelton A. P. da Costa

Curso de Tecnologia em Redes de Computadores - Faculdade de Tecnologia de Bauru (FATEC)
Rua Manoel Bento da Cruz, nº 30 Quadra 3 - Centro - 17.015-171 - Bauru, SP - Brasil

{daniel.bastos, halana.camuci, kelton.costa}@fatec.sp.gov.br

Abstract. *Security in networks has became essential from users with mobile devides to large corporations with WANs. This research aims provide another method for anomaly detection in networks using an artificial neural network. The method used is know as Learning Vector Quantization (LVQ) which has been applied in several areas, working with a low computational cost. In this study, the LVQ algorithm did not achieve significant results.*

Resumo. *A segurança em redes de computadores se tornou imprescindível desde ao usuário com um dispositivo móvel até as grandes corporações com redes WANs. Este estudo tem como objetivo apresentar mais um método para detecção de anomalia em redes utilizando uma rede neural artificial. O método utilizado é conhecido como Aprendizado por Quantização Vetorial (em inglês, Learning Vector Quantization, LVQ) que vem sendo aplicado em diversas áreas, por apresentar um desempenho na análise dos resultados, gerando baixo custo computacional. Para este referido estudo, o algoritmo LVQ não alcançou resultados relevantes.*

1. Introdução

Com o passar dos anos, mais pessoas adquirem o privilégio de acessar a internet, aumentando excessivamente o fluxo de informação na rede. Consequentemente esse uso em massa da internet traz várias desvantagens a segurança e a privacidade dos usuários.

De acordo com Daher Neto (2014), a maior parte dos ataques cibernéticos são motivados por dinheiro, roubo de informações confidenciais de grandes corporações, ou até mesmo por defesa de uma causa política ou social. Estes crimes ocorrem pelo fato das empresas e usuários comuns utilizarem dispositivos moveis conectados na internet para preservar dados sigilosos, como por exemplo: transações por aplicativos de bancos, armazenamento de arquivos em nuvem, tem-se então a atenção a segurança pessoal e empresarial dos usuários conectados na rede.

Atualmente muitas empresas tem investido em detecção de anomalias, por tanto muitas pesquisas estão sendo realizadas. Neste contexto, similares pesquisas foram realizadas utilizando o algoritmo.

Naoum e Al-Sultani (2012) realizaram uma pesquisa utilizando uma variante do algoritmo LVQ, com 23 neurônios, em cima da base de dados NSL-KDD, obtendo uma porcentagem de classificação de 89%.

Naoum e Al-Sultani (2013) no ano seguinte realizaram uma similar pesquisa. Dessa vez aprimorando o algoritmo LVQ e utilizando em conjunto com outro algoritmo também modificado, trabalhando com a base de dados NSL-KDD, resultando numa taxa de detecção de 97%.

A proposta deste trabalho foi a aplicar técnicas inteligentes já utilizadas em outras áreas da computação e demais áreas como saúde, economia entre outros. A referida aplicação, mais precisamente a técnica inteligente conhecida como Aprendizado por Quantização Vetorial presente neste estudo, utiliza e verifica a possibilidade de sua utilização na área de segurança em redes de computadores para identificar anomalias que trafegam entre os dispositivos, caso seja positivo os resultados, esta predita técnica poderá ser utilizada como mais um método para aprimorar a taxa de detecção de anomalias auxiliando as empresas que necessitam de alto percentual de segurança. Portanto o objetivo desse trabalho será demonstrar o teste de eficiência e eficácia do algoritmo LVQ e analisar se o mesmo será apropriado para ser aplicado em detecção de anomalias em Redes de Computadores.

Este estudo apresenta na Seção 2 a sua fundamentação teórica, abordando a área de segurança em redes com detecção de intrusão. Na Seção 3 é abordado conceitos sobre redes neurais e uma breve descrição dos recursos utilizados para o desenvolvimento da aplicação proposta. Na Seção 4, estão relatados os materiais e métodos utilizados, incluindo o método utilizado, a criação de uma base de dados e os resultados obtidos. Na Seção 5 é feita a conclusão e em seguida as referências.

2. Conceitos de Segurança em Redes de Computadores

Cada vez surgem mais oportunidades no mercado, e essas estão de muito se aproveitando das novas tecnologias. As tecnologias mais utilizadas e importantes são as relacionadas a interação a distância. Exemplos comuns seria o uso de comunicação entre pessoas a distância, por meio de mensagens ou voz. Outro seria a interação com equipamentos ou softwares.

De acordo Nakamura e Geus (2007), a área de tecnologia é notável pela sua evolução contínua, por este motivo novas conexões são estabelecidas a cada momento em servidores ao redor do mundo. Os autores ainda ressaltam que essa mesma conectividade abre uma porta para várias oportunidades maliciosas. Invasões de indivíduos mal-intencionados podem causar danos a qualquer rede que for penetrada. Roubo, exposição ou perda de dados importantes seriam um dos possíveis resultados de uma invasão bem-sucedida.

Qualquer rede, pessoal ou principalmente empresarial, deve ter um cuidado mínimo de proteção contra esse tipo de lesão. Com o avanço da tecnologia, cada vez mais tudo está se tornando automatizado. Uma automatização da segurança responsável pela barragem desse tipo de infiltração é uma realidade possível nos dias de hoje.

Alguns invadem somente com o objetivo de conseguir encontrar alguma fragilidade no sistema. Segundo Maia (2005), existem alguns métodos para identificar e evitar determinadas invasões pela rede de internet, que são eles: sistema de detecção de intrusão, detecção de prevenção de intrusão, detecção por uso indevido e por anomalia.

2.1. Método Sistemas de Detecção de Intrusão

Segundo Maia (2005), os Sistemas de Detecção de Intrusão (SDI) são elementos passivos da rede, coletam e analisam o tráfego, procurando por evidências de uso indevido geralmente baseados em regras ou comportamentos.

Essas regras são muitas das vezes preestabelecidas, resultadas de análises de ataques anteriores, que foram armazenados e rotulados, tornando-se um modelo comum a ser reconhecido.

O SDI leva um tempo para detectar algo anormal trafegando na rede. Ele nunca para sua função de ler os dados, e após coletar informações suficientes, consegue classificar esse tipo de ação, tendo a certeza se é ou não algo anormal.

2.2. Método Sistema de Prevenção de Intrusão

Sistemas de Prevenção de Intrusão (SPI) podem ser considerados como uma extensão do SDI.

Esses sistemas são responsáveis por tomar ações contra as intrusões detectadas. As ações podem variar de somente avisar sobre uma intrusão como também agir contra ela, desviando pacotes maliciosos manualmente, reiniciando a conexão ou bloqueando o IP do possível invasor.

O SPI assim, como o SDI, possui variações. Ele pode ser instalado na rede inteira, como também somente em um dispositivo. A rede pode ser cabeada ou sem fio.

2.3. Método por Uso Indevido

Segundo Maia (2005), esse método é o mais utilizado em conjunto com SDI.

Nesse método, existem regras de assinatura anterior estabelecidas. O sistema detectara uma intrusão não somente limitando exatamente a esses padrões, mas ira um pouco mais além, detectando intrusões com uma mínima taxa de variação comparado com seu relacionado no banco de dados.

Isso ainda impede que esse método detecte práticas novas ou desconhecidas de suas regras.

2.4. Método de Detecção por Anomalia

Nesse método, a detecção é focada em padrões de assinatura normais já conhecidos. Tudo o que for diferente do que normalmente seria transmitido, será apontado como anomalia.

A grande vantagem é que esse método consegue reconhecer novos ataques, pois tudo o que não for permitido, será uma anomalia.

A desvantagem é que essa técnica pode ter uma alta taxa de detecção, sendo elas investidas reais, chamadas de falso negativo, ou ações normais, chamadas falso positivo.

Segundo Silva (2007), os problemas mais comuns nesse tipo de abordagem seriam o treinamento extensivo de dados históricos, que leva bastante tempo e deverá ser feito com muita frequência. E um imenso gasto financeiro em máquinas para realizar esse processo, pois a cada novo diferencial do padrão, é preciso atualizar-se, criando uma demanda muito forte em cima dos dispositivos utilizados.

3. Conceitos em Redes Neurais Artificiais

De acordo com Mackay (1994 apud Moreira 2014) redes neurais são paradigmas matemáticos inspirados na estrutura neural de organismos inteligentes e que adquirem conhecimento por meio de experiência para resolver impasses de predição, reconhecimento de padrões e classificação.

A rede neural trabalha transmitindo dados entre neurônios por conexões chamadas sinapses. Cada sinapse é representada por um valor de peso ou força, que corresponde a informação memorizada no neurônio. Este processo acontece na camada de entrada.

Os estímulos obtidos pela camada de entrada são processados pela função soma. Os dados desta camada podem ser conectados em muitos neurônios, o que resulta uma série de saídas onde todo neurônio irá representar uma saída.
O conhecimento é obtido do ambiente por um procedimento conhecido como aprendizado.

3.1. Aprendizado por Quantização Vetorial

Aprendizado por Quantização Vetorial é uma das opções de algoritmos a serem utilizados no método de detecção de anomalias.

Conforme Silva, Spatti e Flauzino (2010), este algoritmo possui uma camada de neurônios que classifica e agrupa os dados apresentados, utilizando um vetor quantizado posicionado na distância mínima entre os dados, normalmente onde existe mais densidade. Os autores ainda demonstram em um experimento que no resultado final, uma das vantagens do LVQ foi o seu curto tempo de análise dos resultados, gerando um baixo custo computacional. O algoritmo trabalha classificando um conjunto de dados, geralmente conhecido como banco de dados. Tudo isso em cima de outro *software*, que irá ser a base para quaisquer experimentos realizados.

3.2. *WEKA Tools*

Waikato Environment for Knowledge Analysis (WEKA) é um *software* desenvolvido pela Universidade de Waikato, Nova Zelândia. Este é utilizado neste trabalho e possui várias ferramentas para visualização e algoritmos de análise de dados e modelagem preditiva, contendo uma interface gráfica facilitando o manuseio. Com esses recursos, o *software* é capaz de realizar tarefas envolvendo dados, incluindo mineração de dados, como esta apresentado no *site* do WEKA.

O WEKA utiliza um formato de arquivo ferramentas chamado *Attribute-Relation File Format* (ARFF). Este arquivo de texto em ASCII, quando for uma base de dados, descrevera vários dados, cada um em uma linha composta por os próprios dados e seus respectivos atributos, delimitadas por vírgulas. Algoritmos são capazes de ler estes arquivos para aprender.

4. Materiais e Métodos

Nesta seção, é apresentada a metodologia para validar os resultados da técnica LVQ descritos na seção 3.1 a fim de analisar e quantificar as anomalias em redes de computadores.

Um conjunto de dados foi duplicado, dividido e carregado na ferramenta WEKA usando o formato de arquivo ARFF já mencionado. Passados os treinos e testes, os resultados foram analisados de acordo com o nosso critério de avaliação. A Figura 1 ilustra o processo do trabalho.

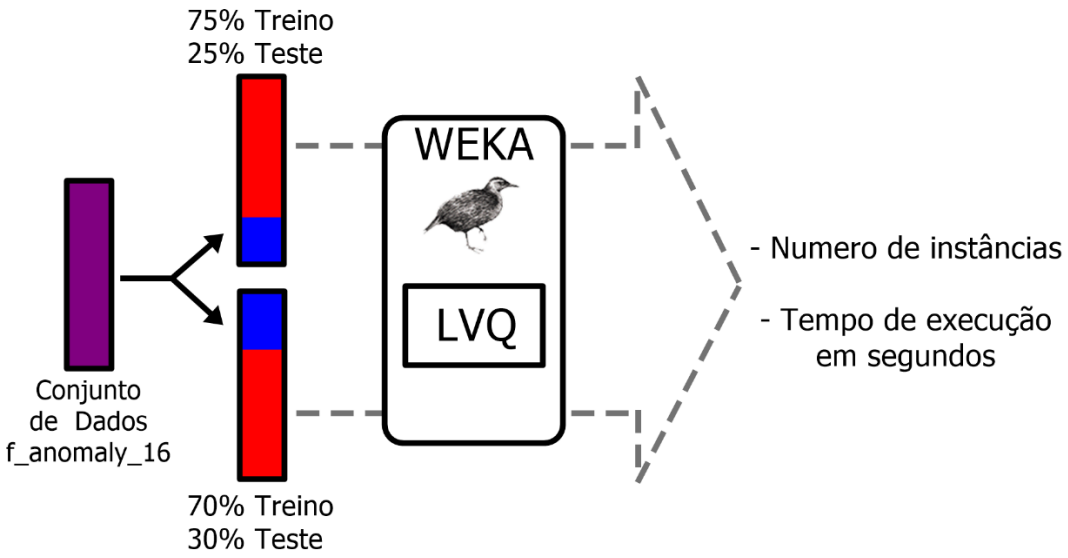

Figura 1. Representação do processo escolhido

Fonte: Daniel Bastos e Halana Rubia

4.1. Conjunto de dados

Utilizando um Notebook ASUS S64C (sistema operacional Windows 8.1, processador Intel Core I7 e 6GB de memória RAM), a base de dados foi capturada em dias úteis durante 7 dias seguidos na própria FATEC Bauru. Todos os dias a partir das 08:30 da manhã até a captura diária alcançar o número de 1000 pacotes, o que levou de 2 a 5 minutos cada.

A criação da base de dados teve início com uma captura em uma rede fechada para alunos na própria FATEC Bauru utilizando o *software TCPDump*, software na qual é possível ver os dados trafegados na rede no momento de uso, também permitindo a opção de salvar esse histórico.

A partir das 08:30 da manhã, foi iniciada uma captura com o objetivo de alcançar o número de 1000 pacotes diários, o que levou de 2 a 5 minutos. Durante 7 dias, essa rotina foi repetida nos dias uteis. O resultado final desta etapa foram 5 arquivos de 1000 pacotes. Tendo os dados em prontos, o próximo passo foi trata-los.

O processo de tratamento envolveu principalmente 2 softwares: *Wireshark* e *Microsoft Excel*.

Com o *Wireshark* foi possível visualizar em uma interface amigável o conteúdo salvo das capturas. Nele os arquivos foram em unidos, transformando em um grande arquivo de 5000 pacotes. Agora trabalhando em cima da captura total, foi realizada a seleção das colunas relevantes. Estas colunas foram em sua maior parte definidas pelas assinaturas de ataques descritas em um documento sobre anomalias disponível no site do Laboratório Lincoln do Instituto de Tecnologia de Massachusetts (em inglês, *Massachusetts Institute of Technology*, MIT[1]) e apresentadas na Tabela 1 abaixo.

Tabela 1. Colunas utilizadas

time	Tempo decorrido durante toda captura
delta_time	Tempo entre um pacote e outro
source_address	Endereço de envio (IP)
destination_address	Endereço de destino (IP)
source_port	Porta de envio
destination_port	Porta de destino
protocol	Tipo de protocolo
packet_lenght	Tamanho do pacote (bytes)
time_to_live	Tempo de vida de um pacote
ip_checksum	Checksum do endereço IP
tcp_checksum	Checksum do protocolo tcp
udp_checksum	Checksum do protocolo udp
hardware_src_addr	Endereço físico da máquina de envio
hardware_dest_addr	Endereço físico da máquina de destino
label_anomaly	Área de rotulamento dos pacotes

Finalizando as colunas, foi utilizado então um recurso de coloração de linhas do *Wireshark*, podendo definir regras e equações para "pintar" cada linha de acordo com o que foi definido, também utilizando as definições de anomalia do MIT.

Finalizando a parte com o *Wireshark*, a base foi exportada como uma planilha e editada no software *Microsoft Excel*. Uma nova coluna com a marcação das possíveis anomalias foi adicionada e as linhas coloridas pelo *Wireshark* foram marcadas nessa coluna. O título de cada coluna foi reescrito em um padrão, sem a utilização de espaços para evitar incompatibilidades futuras. Para ficar em um formato reconhecível pela próxima etapa, foi necessário remover exceções de células que continham virgulas, pois virgulas são as delimitações de cada campo de uma linha. E por fim o preenchimento de células vazias. O *Wireshark* em alguns casos apresentava somente uma alternativa dentro de duas possíveis, como por exemplo, exibia somente *False* (Falso) quando era falso,

[1] **https://www.ll.mit.edu/index.html**

deixando as células restantes (*True* (Verdadeiro)) em branco; nesse exemplo, as células vazias restantes foram preenchidas com *True*.

A planilha então foi exportada para o formato do *Excel* conhecido como csv, na qual transforma a tabela em linhas, separando as células por ponto e vírgula. Este arquivo foi aberto em um editor de texto (no caso, *Notepad++*). Com esse editor, foi possível substituir os pontos e virgulas de todo o arquivo por somente virgula. Alguns números estatísticos contados para a criação de uma futura tabela de resultados.

A esse ponto, só restava ser feita a adição do cabeçalho no arquivo. Voltando ao *Excel* em versões anteriores da planilha, foram removidas duplicatas de valores de cada coluna que não apresentasse valores numéricos, restando assim somente o número mínimo de valores. Estes então foram copiados e escritos no cabeçalho da base.

Finalizada, a extensão do arquivo da base foi alterada para ARFF. Para certificar de que não continha nenhum erro, ela foi aberta pelo visualizador de base de dados do WEKA. Caso houvesse algum erro, ela não seria aberta e mostraria as linhas aonde estão os erros na base.

4.2. Resultados Obtidos

O método utilizado foi duplicar a base e então dividir em porcentagens de 75% e 25% uma e 70% e 30% a outra. A porção com porcentagem mais alta foi utilizada para treinar a base e a com menor para ser testada pelo algoritmo.

O número de possíveis anomalias em cada porção é conhecido, também como a porcentagem. De posse destes valores, foi possível comparar se os resultados foram aproximados do que foram manualmente definidos. Os treinos e testes foram realizados no mesmo notebook.

Tabela 2. Dados Definidos

Porções	100%	A 75%	A 25%	B 70%	B 30%
Num. de Pacotes	5000	3750	1250	3500	1500
Num. de Possíveis Anomalias	607	449	158	460	147
Porcentagem Anomalia/Porção	12.14%	11.97%	12.64%	13.14%	9.8%

A Tabela 2 reúne as informações definidas anteriormente e que se espera conseguir com os testes. As 5 colunas apresentam o total, o primeiro treino e teste (marcados com A) e o segundo treino e teste (marcados com B), respectivamente. Abaixo são os números de pacotes que cada porção possui. Na linha abaixo são os números de possíveis anomalias de cada porção e por fim, a porcentagem de anomalia por porção.

Tabela 3. Resultados Obtidos

Porções	75%		25%		70%		30%	
Clusters	0	1	0	1	0	1	0	1
Anomalias Detectadas	808	2942	259	991	1098	2403	790	709
Porcentagem de grupo			21%	79%			53%	47%
Tempo gasto em segundos	8.19				12.26			

Já a Tabela 3 mostra o resultado pôs teste. As divisões estão apresentadas na linha superior. Na segunda linha são mostrados os *clusters*, que são pontos de referência que o algoritmo utiliza para classificar os dados. O dado que se assemelha mais a um determinado *cluster*, será marcado como sendo parte daquele *cluster*. Anomalias Detectadas são os números de pacotes que o algoritmo definiu sendo pertencente de cada *cluster*. Porcentagem de grupo são as porcentagens finais exibidas pelo *WEKA* após somente os testes. Por fim o tempo gasto em segundos durante todo o processo de treino e teste em cada uma das etapas.

Os valores finais em porcentagem da Tabela 2 não foram próximos dos apresentados pela tabela Tabela 3. O teste com 75-25 obteve um melhor resultado, marcando 259 pacotes como anomalia em poucos segundos, mas ainda sim a mais dos 158 verdadeiros. Já o segundo teste resultou em números altos, classificando aproximadamente metade da base como anomalia em um tempo também alto em comparação com o teste anterior.

5. Conclusão

Conclui-se o desempenho do algoritmo não atingiu resultados relevantes para este estudo. Era esperado resultados próximos a 70% nas duas porções, mas somente obtivemos ele em uma das porções. Já o tempo em que levou para serem processados foi rápido, o que ainda pode deixar mais caminhos a serem pesquisados no futuro.

A maior influência no resultado é aonde o algoritmo está sendo treinado, no caso, em qual porção ele será treinado. Alguns dos possíveis motivos para tais resultados pode ser o tamanho da base, pois quanto maior, a tendência é o algoritmo ser treinado com maior precisão, contando com todas as possibilidades possíveis de anomalia na base; e também relacionado a essa complicação, os campos de Protocolo de Internet (*Internet Protocol*, IP) e Endereço MAC (*Media Access Control*, MAC Address), pois muitas anomalias estariam ligadas a um desses dois e somente a eles, o que resultaria no algoritmo ser treinado a associar um especifico *MAC Address* a uma anomalia. Sendo assim, em bases pequenas, aumentam as chances de que o trafego gerado por uma máquina qualquer seja sempre tratada como anomalia, mesmo que não esteja realizando nenhuma ação maliciosa no momento.

Referências

Daher Neto, J (2014) "Hipercaixas Delimitadoras da Detecção de Intrusos em Redes de Computadores", Itajubá.

Maia, R. B. (2005) "Detecção de Intrusão Utilizando Classificação Bayesiana", Rio de Janeiro.

MIT Lincoln Laboratory, "DARPA Intrusion Detection Evaluation", https://www.ll.mit.edu/ideval/docs/attackDB.html, Mar, 2016.

Moreira, E. (2014) "Redes neurais artificiais e análise discriminante linear como alternativas para seleção entre famílias de cana-de-açúcar", Minas Gerais.

Nakamura, E. T.; Geus, P. L. (2007) "Segurança de redes em ambientes cooperativos", Novatec.

Naoum, R. S., Al-Sultani, Z. N. (2012) "Learning Vector Quantization (LVQ) and k-Nearest Neighbor for Intrusion Classification", Iraque.

_____. (2013) "Hybrid System of Learning Vector Quantization and Enhanced Resilient Backpropagation Artificial Neural Network for Intrusion Classification", Jordânia.

Silva, I.; Spatti, D. e Flauzino, R. (2010) "Redes Neurais Artificiais para Engenharia e Ciencias Aplicadas - Curso Prático", São Paulo.

Silva, L. S. (2007) "Uma Metodologia para Detecção de Ataques no Tráfego de Redes Baseada em Redes Neurais", São José dos Campos.

Weka 3, "Data Mining Software in Java", http://www.cs.waikato.ac.nz/ml/weka, Nov, 2015.