Detecção de anomalia através da mineração de fluxos contínuos de dados

Guilherme Cassales[1], **Hermes Senger**[1]

[1]Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
Rodovia Washington Luís, Km 310, São Carlos – SP – Brasil

{gwcassales, senger.hermes}@gmail.com

Abstract. *Cyberattacks are a constant threat to distributed systems. The implementation of a Network Intrusion Detection System (NIDS) is an alternative to mitigate this threat. However, traditional data mining techniques doesn't satisfy some conditions of data stream mining, such as the model evolution through time. This paper applies the algorithm ECSMINER, a stream designed technique, in a security dataset. The preliminary results presents low error rate (aproximately 3%) indicating the viability to employ this technique in NIDS.*

Resumo. *Ataques cibernéticos são uma constante ameaça para sistemas distribuídos. A implementação de um Sistema de Detecção de Intrusão pela Rede (NIDS) é uma alternativa para mitigar esta ameaça. Contudo técnicas de mineração de dados tradicionais não satisfazem algumas condições da mineração em fluxo de dados, como a evolução do modelo ao longo do tempo. Este trabalho aplica o algoritmo ECSMINER em um banco de dados de segurança. Os resultados preliminares apresentam baixa taxa de erros (aprox. 3%) e indicam a viabilidade de empregar esta técnica em NIDS.*

1. Introdução

Atualmente as aplicações e serviços prestados por meio da Internet vêm sendo expandidos com objetivo de atingir um público alvo cada vez maior. Contudo, como em todos os sistemas distribuídos a exposição à ameaças cibernéticas é prevalente e, muitas vezes, potencializada pelo desejo de prover primeiramente um sistema funcional e adicionar medidas de segurança em um segundo momento [Khan et al. 2017]. Uma das soluções comumente utilizadas para mitigar as ameaças de segurança é a utilização de Sistemas de Detecção de Intrusão pela Rede (*Network Intrusion Detection Systems* - NIDS) baseado em anomalia. A detecção baseada em anomalias é capaz de detectar ataques novos, pois baseia-se na estimação do comportamento normal do sistema e gera alertas sempre que identifica um tráfego que difere desse padrão [García-Teodoro et al. 2009].

Estes sistemas operam sobre o fluxo de dados que passa por um determinado local da rede. Geralmente o fluxo é contínuo, possui alta velocidade e tamanho ilimitado, impossibilitando seu armazenamento em memória e o controle sobre a ordem de chegada dos dados [Babcock et al. 2002]. Estas características, somadas ao fato dos dados evoluírem ao longo do tempo tornam necessária a utilização de algoritmos incrementais [Gama 2010]. Nos algoritmos incrementais o modelo de decisão, baseado em padrões, evolui incrementalmente com objetivo de adaptar-se às mudanças na natureza dos dados que chegam continuamente [Nguyen et al. 2015].

2. Trabalhos relacionados

Em [Masud et al. 2011] os autores apresentam o ECSMINER, um algoritmo para mineração em fluxo de dados com funcionamento em duas etapas. Em um primeiro momento há um treinamento offline que produz um modelo inicial. Em seguida, na etapa online, o modelo é utilizado para classificar o fluxo. Após classificados, os novos exemplos são redirecionados para um especialista, que retornará os exemplos rotulados após um intervalo de tempo. O modelo utiliza os exemplos rotulados para atualizar o modelo, o qual continua a classificar os novos exemplos, porém agora incorporando os dados rotulados pelo especialista.

Em [Lee et al. 2011] os autores utilizam uma combinação de *Self Organizing Map* (SOM) e K-means. O SOM atua como um redutor da dimensionalidade e o K-means detecta os novos padrões. Os resultados apresentados demonstram que métodos online detectaram com maior acurácia e menor FAR os ataques desconhecidos inseridos nos conjuntos de teste do que modelos gerados por algoritmos tradicionais. Este trabalho utiliza a base de Kyoto e o KDD99.

Em [Kumar and Venugopalan 2017] os autores propõem um algoritmo baseado em distância e na utilização do atributo protocolo para detecção de anomalias utilizando a base de Kyoto. Apresenta bons resultados, como 95% de *recall* e e 98% de *precision*.

3. Metodologia proposta

Muitos trabalhos da área de segurança utilizam o *dataset* da KDD Cup 99 [KDD-CUP 1999], contudo o cenário de segurança atual é muito diferente daquele retratado nestes dados. Sendo assim, é importante que um conjunto de dados mais recente seja utilizado. O Kyoto *Dataset* [SONG et al. 2017], possui muitos atributos baseados nos atributos do KDD99 e é constituído por ataques mais recentes. Além disso possui fluxos de 2006 a 2015, o que possibilita fazer experimentos em maior escala.

Para os experimentos em questão, foram utilizadas as entradas do mês de dezembro de 2015. Após concatenar todos arquivos, foi feito um pré-processamento básico para normalizar os atributos numéricos e binarizar (ou remover) os atributos nominais (serviço e protocolo) pelo método 1-para-n. Além disso, alguns atributos foram removidos de maneira empírica por não impactar nos resultados, sendo eles o atributo Flag de conexão e os atributos adicionais – com exceção do atributo Label – elencados na descrição dos dados.

Outra medida adotada foi o truncamento dos dados na normalização. Alguns atributos, como a duração da conexão, possuíam *outliers* muito elevados que faziam com que após a normalização, alguns exemplos acabassem ficando com valor 0, mesmo quando estes possuíam uma duração original diferente de 0. Sendo assim, foram utilizados os valores 25, 2250 e 2900 como máximos para os atributos duração, bytes enviados e bytes recebidos. Estes valores foram escolhidos de forma que no máximo 1% dos exemplos fosse modificado.

4. Experimentos e Resultados preliminares

Os experimentos foram realizados utilizando o ECSMINER e mudando alguns de seus parâmetros, bem como comparando o desempenho entre um conjunto de dados com atributos nominais removidos e binarizados.

4.1. Atraso na entrega do rótulo (Tl)

Uma das configurações possíveis do ECSMINER diz respeito ao tempo de espera necessário para que o modelo tenha acesso ao rótulo da instância, ou seja, o rótulo terá um atraso de *n* instâncias para ser acessado pelo modelo. Isto reflete uma situação real, pois rotular exemplos demanda tempo do especialista. Nos experimentos a seguir compara-se o comportamento do experimento com um tempo de espera de 50 mil instâncias com uma espera de 200 mil instâncias. Ao comparar figuras 1 e 2, é possível notar que ao atrasar a entrega do rótulo, a quantidade de padrões novidade encontrado é muito grande, isso ocorre porque o modelo não aprende novas ”classes”, apenas identifica padrões que são diferentes do conceito que ele havia aprendido. O resultado do Fnew (porcentagem de exemplos da classe normal incorretamente classificados como novidade) foi bem baixo, indicando que, apesar da grande quantidade de padrões novidade encontrados, poucos exemplos da classe normal foram identificados incorretamente como novidade. Ou seja, ainda que o modelo nunca tenha sido atualizado nesse período de atraso, a classe normal não sofre mudanças ao longo do tempo e o treinamento inicial foi suficiente para identificá-la. Outro ponto interessante é que o MNew (porcentagem de exemplos das novas classes incorretamente classificados como classe normal) é baixo, ou seja, poucos exemplos das novas classes foram identificadas como normal.

Quando os rótulos começam a chegar e o modelo precisa aprender o que é normal e o que é novidade ele começa a errar bastante, ou seja, aparecem picos de FNew porque o modelo aprendido ainda não é capaz de representar o novo cenário e os exemplos da classe novidade (que agora é classe conhecida) ainda continuam a serem identificados como novidade. Após um tempo, o modelo estabiliza e já não faz mais sentido falar em MNew, já que não teremos classes novidade (o experimento era classificação binária, ataque ou normal).

Figura 1. ECSMINER, J48, binário, nominais removidos, atraso de 50000 instâncias para rótulo

Comparando os resultados entre as bases com dados nominais binarizados (Figura 3) e removidos (Figura 1) é possível notar que *removê-los gera um resultado levemente melhor*. Uma diferença entre os experimentos é que quando os atributos nominais são removidos o Mnew é 0 durante toda execução, porém com os nominais binarizados o Mnew

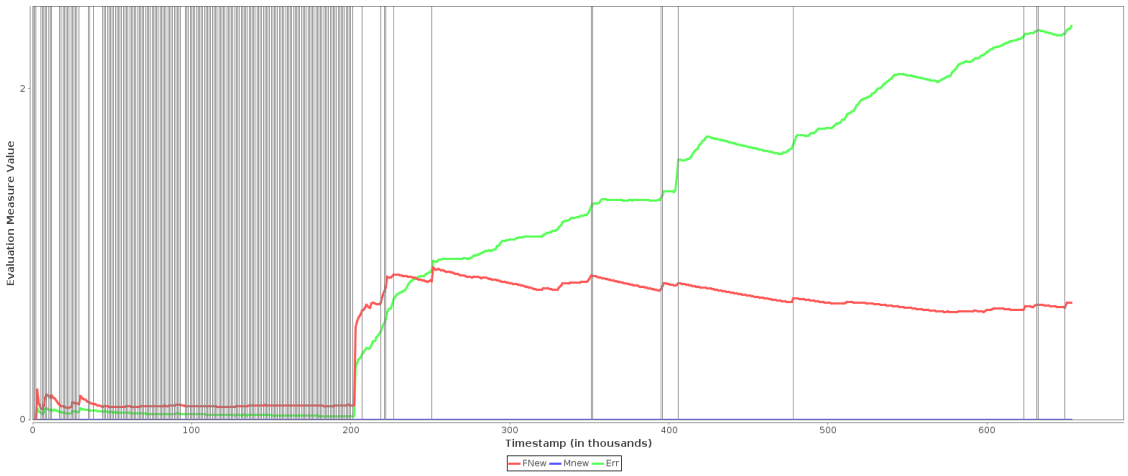

Figura 2. ECSMINER, J48, binário, nominais removidos, atraso de 200000 instâncias para rótulo

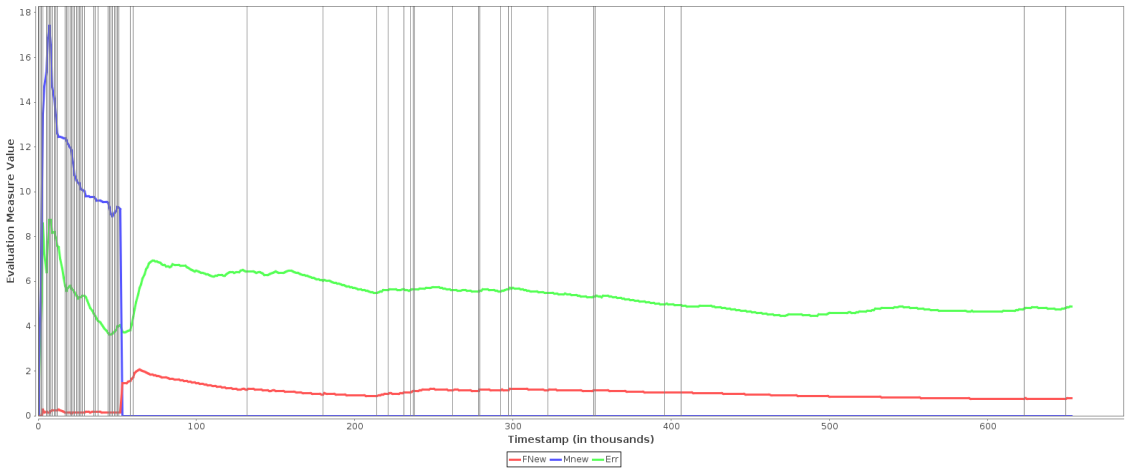

Figura 3. ECSMINER, J48, binário, nominais binarizados, atraso de 50000 instâncias para rótulo

é de aproximadamente 18% no início da execução, sendo que com a chegada dos rótulos ele vai para 0%. Este comportamento ocorre porque a informação extra, em um primeiro momento, faz com que exemplos de ataque (novidades) sejam erroneamente classificados como fluxo normal. Após a chegada dos rótulos o modelo é incrementado e consegue diferenciar melhor o que é normal do que é novidade, ou seja, o conceito normal aprendido na fase offline não foi suficientemente bom no caso dos nominais binarizados. Nota-se que este comportamento cessa no momento que os rótulos das instâncias começam a chegar, indicando que a partir deste momento o modelo é incrementado e começa a identificar corretamente o conceito normal. O Fnew possui comportamento semelhante nos dois casos, porém o erro é perceptivelmente maior no experimento com os nominais binarizados, indicando que a adição destes atributos gera uma complexidade maior e faz com que o modelo inicial não seja capaz de reconhecer corretamente o que é o conceito normal. Isto causa o alto Mnew no período de atraso e um erro mais alto durante todo experimento.

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou experimentos preliminares sobre a aplicação de técnicas de mineração em fluxos contínuos de dados utilizando o banco de dados de intrusão gerado pela Universidade de Kyoto. Os resultados apresentados possuem em média 3% de erro, mesmo quando nem todos os rótulos estão disponíveis para a atualização do modelo, indicando a viabilidade da utilização desta técnica em cenários de classificação binária.

Como trabalhos futuros, espera-se aplicar estudar o comportamento com experimentos multiclasse. Além disso, outra linha de trabalho proposta é a utilização de um conjunto de dados maior, que utilize os dados de fluxo de um ano inteiro.

6. Agradecimentos

Os autores agradecem à FAPESP pelo apoio (Processos 2015/24461-2 e 2018/00452-2). Hermes Senger agradece ao CNPQ (Processo 305032/2015-1). Guilherme W. Cassales agradece à CAPES-DS pelo apoio recebido na forma de bolsa de doutorado.

Referências

Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 1–16, New York, NY, USA. ACM.

Gama, J. (2010). *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.*, 28(1-2):18–28.

KDD-CUP (1999). The uci kdd archive. Disponível em `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

Khan, S., Parkinson, S., and Qin, Y. (2017). Fog computing security: a review of current applications and security solutions. *Journal of Cloud Computing*, 6(1):19.

Kumar, D. and Venugopalan, S. (2017). A distance based algorithm for network anomaly detection using initial classification of 'protocol type' attribute. *International Journal of Advanced Research in Computer Science*, 8(7):754–759.

Lee, S., Kim, G., and Kim, S. (2011). Self-adaptive and dynamic clustering for online anomaly detection. *Expert Syst. Appl.*, 38(12):14891–14898.

Masud, M., Gao, J., Khan, L., Han, J., and Thuraisingham, B. M. (2011). Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, 23(6):859–874.

Nguyen, H.-L., Woon, Y.-K., and Ng, W.-K. (2015). A survey on data stream clustering and classification. *Knowl. Inf. Syst.*, 45(3):535–569.

SONG, J., Takakura, H., and Okabe, Y. (2017). Kyoto 2006+ new version data. Acessado em Março de 2018. `http://www.takakura.com/Kyoto_data/new_data201704/`.