Comparação de Algoritmos Não Supervisionados para Detecção de Anomalias de Segurança da Informação

Luis Diniz[1], Fabricio Silva[1]

[1]Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa - *Campus Florestal* (UFV)

{luis.diniz, fabricio.asilva}@ufv.br

Resumo. *Levando-se em consideração a larga utilização de sistemas computacionais em diversos contextos e quantidade de informações sensíveis produzidas por eles, torna-se essencial dar atenção à sua segurança. O objetivo deste trabalho é explorar o uso de técnicas de clustering e análises estatísticas para detecção de anomalias em uma base de dados sobre segurança da informação. Será feita uma análise comparativa entre diferentes técnicas na NSL-KDD, em termos de precisão, atributos escolhidos, cenário de execução, entre outros. Cinco técnicas não supervisionadas foram escolhidas para comparação: Random Forests, Local Outlier Factor, Elliptic Envelope, K Means e HBOS. Os resultados obtidos variam consideravelmente segundo o cenário em que foi aplicado. Após a análise dos resultados obtidos ficou evidente que o desempenho das técnicas escolhidas é sensível ao contexto e à sua configuração de aplicação.*

1. Introdução

É fato que sistemas computacionais são usados em larga escala por pequenas e grandes empresas, nas mais diversas áreas. Tal utilização, em alguns casos, pode gerar um grande volume de informações relacionadas com a entidade que detém os dados, assim como pela entidade que os produziu. As relações das entidades envolvidas com dados podem ser críticas e, nesse caso, devem ser abordadas cautelosamente. Questões como a privacidade das informações são extremamente delicadas e têm relação direta com a segurança do sistema utilizado para armazená-las. A segurança torna-se uma questão ainda mais crítica se considerarmos o fato de que há um elevado número de sistemas computacionais nas nuvens, o que leva à maior vulnerabilidade.

Tendo isso em vista, destaca-se a importância do desenvolvimento de medidas contra a vulnerabilidade de sistemas capazes de armazenar grandes volumes de dados [Sivarajah et al. 2017]. A prevenção e detecção de intrusos tem um importante papel neste cenário. Considerando-se um sistema em rede é possível, por meio da análise dos dados produzidos no log do sistema, detectar atividades suspeitas que devem ser, posteriormente ou em tempo real, investigadas para que estas sejam classificadas ou não como reais ameaças.

Tendo como fonte de informações logs, há técnicas para a detecção de intrusos, caso de três categorias de ataques avaliados neste trabalho (U2R, R2L e probe). Um sistema capaz de realizar tal tarefa é conhecido com NIDS (*Network Intrusion*

Detection System). Há dois importantes métodos para detectar atividades suspeitas [Kemmerer and Vigna 2002]: *misuse detection* e *anomaly detection*.

Misuse detection busca o que não está previsto no conjunto de dados. Para que esta técnica funcione corretamente, é necessário que assinaturas (descrições) de ataques já conhecidos sejam disponibilizadas para que o método trabalhe baseado nelas. Por esta razão, pode-se dizer que esta técnica é limitada, já que não permite ao sistema identificar novos tipos de ataque. Por outro lado, a taxa de acertos com uso de assinaturas tende a ser alta.

Anomaly detection consiste na identificação de comportamento dos dados analisados com base em uma entrada, não são necessárias assinaturas para a identificação dos ataques. O método aprende quais atividades são suspeitas e quais não são. Sua principal vantagem é possibilitar a identificação de novos ataques devido ao fato de não precisar de assinaturas. Entretanto, a taxa de falsos positivos tende a ser alta.

Em aplicações reais, a técnica de detecção de anomalias se mostra mais interessante. Fato que se deve principalmente à possibilidade de detecção de ataques desconhecidos (novos tipos de ataque) sem a necessidade da inserção de suas assinaturas (descrições). Tendo isso em vista, há diversas maneiras de se implementar algoritmos e/ou sistemas capazes de identificar anomalias [Chandola et al. 2009, Garcia-Teodoro et al. 2009], dentre elas pode-se citar análise estatística, algoritmos de classificação, algoritmos de clusterização e grafos. O presente trabalho dá especial atenção para aprendizado não supervisionado aplicado à detecção de anomalias. *Clustering* se mostra um método vantajoso pois é implementado de forma não supervisionada, ou seja, não há necessidade de um conjunto de dados classificado como livre de anomalias (conjunto de dados de treinamento) para que o algoritmo aprenda a identificar uma atividade suspeita. É evidente que, dado um conjunto de dados livre de anomalias, a implementação de outras técnicas apresentará bons resultados. Todavia, em aplicações do mundo real, é pouco frequente a disponibilidade de um conjunto de dados de treinamento. [Ferrari and De Castro 2015] mostra que há diferentes maneiras de se aplicar clustering, cada um dos algoritmos mencionados por [Ferrari and De Castro 2015] apresenta suas particularidades. Técnicas baseadas em análises estatísticas também são exploradas, considerando que se mostram relevantes na literatura.

O objetivo deste trabalho é explorar o uso de técnicas de *clustering* e análises estatísticas para detecção de anomalias na base de dados NSL-KDD. Será feita uma análise comparativa entre diferentes técnicas na NSL-KDD, em termos de precisão, atributos escolhidos, cenário de execução, entre outros.

O texto é organizado em mais 4 seções. A Seção 2 consiste numa revisão de literatura. A Seção 3 trata especificamente de detalhes do conjunto de dados NSL-KDD. A Seção 4 discorre sobre a metodologia do trabalho, incluindo o processamento dos dados, as técnicas escolhidas, entre outros aspectos. Na Seção 5 os resultados do trabalho são apresentados e discutidos.

2. Trabalhos Relacionados

Diversas estratégias foram propostas para o desenvolvimento de sistemas de detecção de intrusos. Tais estratégias possuem inúmeras abordagens possíveis, cada uma

delas apresenta suas particularidades. Esta seção descreve trabalhos com as abordagens mais recorrentes na literatura.

Há uma série de soluções, ditas supervisionadas, para aplicar a detecção de anomalias. Uma possível implementação de um IDS (*Intrusion Detection System*) pode ser realizada por meio de redes neurais. [Shun and Malki 2008] mostra resultados promissores utilizando redes do tipo *feedforward* com base no algoritmo de *back propagation*. Outra estratégia recorrente é a aplicação de métodos estatísticos para identificação de comportamentos anômalos (dados que desviam do padrão esperado), [Kruegel and Vigna 2003] executa sólida análise baseada em logs de servidores apache e obtêm resultados satisfatórios. [Meng 2011] também implementa estratégias utilizando redes neurais, SVM e árvores de decisão em um ambiente uniforme e deixa claro que o uso dessas estratégias deve ter especial atenção já que não funciona de forma eficiente em todos os cenários de IDS. [Sommer and Paxson 2010] explicita algumas razões que tornam o uso de *machine learning* um desafio na detecção de intrusos, mas ressalta que tal estratégia não é totalmente ineficiente e que deve, apenas, ser usada em situações adequadas. Todas as técnicas mencionadas acima são ditas supervisionadas, pois necessitam de um conjunto de dados de treinamento para que possam ser executadas. A aplicação de estratégias supervisionadas é vantajosa pois apresenta alto nível de precisão. Todavia, necessita de um conjunto de dados de treinamento e um conjunto de teste, no caso específico um conjunto de dados livre de anomalias e outro com anomalias. Em aplicações reais, a obtenção de um conjunto de informações adequadas para treinamento não é tarefa trivial e nem sempre viável.

Há também, diversas soluções implementadas por meio de técnicas não supervisionadas. [Münz et al. 2007] utiliza o clássico algoritmo *K-means* para executar detecção de anomalias. [Leung and Leckie 2005] combina dois tipos de *clustering* (i.e. *grid e density*) e obtém resultados satisfatórios, todavia, a taxa de falsos positivos apresentada é alta. [Rajasegarar et al. 2014], por sua vez, implementa *clustering* hiperesférico para aplicar detecção de intrusos em redes sem fio. [Yassin et al. 2013] combina o *K-means* com o *Naïve Bayes Classification* e apresenta resultados que comprovam significativa precisão. [Almalawi et al. 2014] desenvolve um sistema baseado no algoritmo dos K vizinhos mais próximos para executar a detecção de intrusos em *Supervisory Control and Data Acquisition* (SCADA). Todas estas técnicas não supervisionadas são interessantes em cenários reais levando-se em conta que não há a necessidade de um conjunto de dados livre de treinamento, ou seja, são soluções práticas. Em contrapartida, a precisão de tais soluções deixa a desejar quando comparada com estratégias supervisionadas ou, em alguns casos, híbridas.

A combinação de estratégias supervisionadas e não supervisionadas é promissora, pois é capaz de identificar ataques desconhecidos e ainda sim manter uma taxa de acerto dentro de limites aceitáveis. Tendo isso em vista, tem-se os chamados sistemas híbridos. [Kim et al. 2014] utiliza *misuse detection* e detecção de anomalias combinadas para identificar intrusos, os resultados apresentados têm melhor tempo de execução do que os modelos convencionais. [Depren et al. 2005] segue a mesma estratégia que [Kim et al. 2014] e comprova que a combinação dos algoritmos é mais eficiente que cada um deles isolado. [Lin et al. 2015] combina *clustering* com a técnica dos vizinhos mais próximos e compara seus resultados com soluções recorrentes na literatura. [Peddabachigari et al. 2007] im-

plementa árvore de decisão combinada com *support vector machine* (SVM) e obtém bons resultados. [Zhang et al. 2008] utiliza o algoritmo *random-forests* tanto para aplicação de misuse detection quanto para detecção de anomalias, e mostra que *misuse detection* apresenta melhor desempenho do que detecção de anomalias mesmo que não seja possível a identificação de novos ataques.

É evidente que, IDSs capazes de detectar ataques conhecidos e desconhecidos são mais interessantes para aplicações do mundo real. Já que em sistemas supervisionados é necessário um conjunto de dados livre de anomalias para que seja aprendido a diferença entre um registro potencialmente danoso e um registro regular. E mesmo que haja uma parte não supervisionada em sistemas híbridos, há também parte implementada de forma supervisionada, além de ser claramente mais complexa a implementação de sistemas que combinam os dois tipos de técnicas.

Tendo isso em vista, o presente trabalho tem o objetivo de comparar estratégias não supervisionadas para detecção de anomalias. Modelos recorrentes na literatura, assim como novas propostas, serão comparados para que direções de pesquisa em detecção de anomalias de forma não supervisionada sejam estabelecidas. [Killourhy and Maxion 2009] também compara diversas estratégias e abordagens mas não foca em práticas não supervisionadas. [Lazarevic et al. 2003] realiza comparação similar mas com foco na base de dados DARPA 1998. Por sua vez, [Goldstein and Uchida 2016] compara 19 algoritmos não supervisionados em 10 diferentes bases de dados, todavia, a base de dados NSL-KDD (versão refinada do KDD99) não se encontra entre elas. Dessa forma, objetiva-se estudar diferentes algoritmos para detecção de anomalias não supervisionados na base de dados NSL-KDD fornecida pelo CIC (*Canandian Institute of Cybersecurity*) da Universidade de New Brunswick.

3. A base de dados NSL-KDD

Para a comparação dos algoritmos, uma base de dados de benchmark foi escolhida: NSL-KDD. Esta base de dados consiste no refinamento da KDD99, usada na Terceira Competição Internacional de Descoberta de Conhecimento e Ferramentas de Mineração de Dados (*Third International Knowledge Discovery and Data Mining Tools Competition*).

O objetivo desta competição é desenvolver um detector de intrusos em rede usando a base de dados disponibilizada como referência para testes. Após a competição, diversos trabalhos foram elaborados com base nos dados disponibilizados, como pode ser observado em [Tavallaee et al. 2009], de forma a evidenciar que a KDD99 apresenta características não desejadas para uma base de dados de benchmark.

Tendo isso em vista, a NSL-KDD foi escolhida, pois consiste numa versão refinada da KDD99. Segundo [Dhanabal and Shantharajah 2015], registros redundantes foram removidos, de forma a não influenciar nos resultados. [Tavallaee et al. 2009] faz uma detalhada análise da base de dados. Uma quantidade suficiente de registros está disponível na base de testes, permitindo a execução de testes na base completa. Além disso, o número de registros selecionados para cada nível de dificuldade é inversamente proporcional à porcentagem de registros na base de dados original (KDD99). Permitindo assim que as taxas de classificação de diferentes algoritmos variem num intervalo maior, dessa forma é possível avaliar com mais precisão a eficiência dos métodos analisados.

3.1. Categorias de Ataque

Há quatro categorias de ataques presentes na base NSL-KDD, são elas: *denial of service* (DoS), *probe*, *user to root* (U2R) e *remote to user* (R2L).

3.1.1. Denial of Service (DoS)

Ataques do tipo DoS objetivam deixar o sistema ou máquina inativos, tornando-os inacessíveis para os usuários. Tal objetivo é alcançado por meio de tráfego em excesso direcionado à máquina alvo, ou através de envio de informações específicas capazes de tornar a rede inoperante. Ou seja, algum ponto chave da rede torna-se sobrecarregado de modo a negar acesso de serviço à usuários.

3.1.2. Probe

Um ataque é categorizado como probe quando há uma tentativa de obter acesso a uma máquina e seus arquivos por meio de pontos fracos da rede previamente analisados pelo potencial invasor.

3.1.3. User to Root (U2R)

Quando um usuário começa com permissões normais e tenta explorar as vulnerabilidades da rede com o objetivo de ganhar privilégios de super usuário, diz-se que é um ataque do tipo *user to root*, ou seja, um usuário simples que tenta obter privilégios de root.

3.1.4. Remote to User (R2L)

Categoriza-se um ataque como *remote to user* (remoto) quando o possível intruso tem como alvo uma máquina ou conjunto de nós específicos da rede. O dispositivo do intruso não é afetado. Apenas o computador (ou rede) alvo que terá vulnerabilidades exploradas, dessa forma o intruso ganha acesso à outra máquina.

4. Metodologia

4.1. Algoritmos

Ao todo, cinco algoritmos foram selecionados para comparação. Este conjunto de técnicas foi escolhido devido à forma como são implementados. O objetivo foi reunir 5 técnicas que usassem diferentes caminhos para detectar registros não normais. Abaixo segue breve descrição de cada uma delas:

- *K Means*: essa é uma técnica de *clustering* e é conhecida por apresentar desempenho satisfatório em diversos cenários. Sua relevância na literatura foi um fator decisivo em sua escolha.
- *Local Outlier Factor* (LOF): este é um algoritmo não supervisionado para detecção de *outliers*. LOF calcula o desvio de densidade local de determinado ponto a partir de seus vizinhos. Quando um ponto apresenta baixa densidade em relação à sua vizinhança este é classificado como anomalia.

- *Elliptic Envelope*: assumindo que instâncias regulares (observações normais) vêm de uma distribuição conhecida, pode-se concluir que registros que não se encaixam nessa distribuição são instâncias anômalas.
- *Random Forest*: a técnica aplicada "isola" uma observação selecionando aleatoriamente uma *feature* e então seleciona aleatoriamente um *threshold* (entre o valor mínimo e máximo da *feature* escolhida). Como particionamento recursivo pode ser representado por uma árvore, o número de divisões necessários para isolar uma amostra é equivalente ao comprimento do caminho a partir da raiz. Quando uma floresta de árvores aleatórias produz caminhos relativamente curtos para amostras específicas há grandes chances dessa amostra ser uma anomalia.
- *Histrogram-Based Outlier Score* (HBOS): algoritmo baseado em análise estatística que apresentou bom desempenho em cenário específico segundo [Goldstein and Uchida 2016]. Mais detalhes podem ser encontrados em [Goldstein and Dengel 2012].

4.2. Preparação dos dados

Como todas as técnicas utilizadas são não supervisionadas, não há conjunto de treinamento para os algoritmos. Dessa forma, todas as instâncias fornecidas como conjunto de treinamento (*KDDTrain+*) e todas as instâncias fornecidas como conjunto de testes (*KDDTest+*) foram unidas em uma única base de dados. A configuração inicial após a união das bases de treinamento segue conforme Tabela 1:

Tabela 1. Configuração da base de dados original

Classe	KDDTrain+	KDDTest+	Total
Normal	67342	9711	77053
DoS	45927	7459	53386
Probe	11656	2421	14077
U2R	52	67	119
R2L	995	2885	3880
Todos	125972	22543	148515

4.3. Seleção de Features

Apenas *features* numéricas (não binárias) foram utilizadas durante a execução dos algoritmos, dessa forma 32 *features* foram selecionadas para alguns cenários de execução.

Para os ataques do tipo DoS [Khan et al. 2018] recomenda a seleção de 7 *features*, para os outros 3 tipos de ataque (U2R, R2L e Probe) [Dhanabal and Shantharajah 2015] sugere a seleção de 2 *features* para obtenção de resultados satisfatórios. Mais detalhes sobre a seleção das *features* podem ser encontrados em [Khan et al. 2018, Dhanabal and Shantharajah 2015].

Ao selecionar um subconjunto de *features* relevantes elimina-se a necessidade de usar todas as 32 colunas numéricas, ou seja, reduz-se a dimensionalidade dos dados e consequentemente o tempo de processamento. Fica evidente que tal redução pode ser proveitosa já que diminui o custo computacional de análise caso resultados satisfatórios sejam obtidos.

4.3.1. PCA

A técnica *Principal Component Analysis* (PCA) também foi utilizada a partir das subconjunto indicado em [Khan et al. 2018]. Dessa forma, conjuntos com 2 e 3 *features* foram obtidos para o conjunto com instâncias normais e do tipo DoS. Para os outros 3 tipos de ataque as dimensões foram reduzidas para 7 e 3.

4.3.2. Outras técnicas

Outras técnicas de seleção de*feature* (*e.g.: lower variance*) também foram utilizadas mas não houve resultados satisfatórios, ou seja, após a execução das técnicas nenhum subconjunto relevante do conjunto original foi obtido.

4.4. Cenários de execução

Há então 4 cenários de execução e comparação das técnicas não supervisionadas aplicadas para cada uma das categorias de ataque. A Tabela 2 explicita cada um desses cenários com a quantidade de registros, *features* e os tipos de ataques presentes.

Tabela 2. Cenários em que técnicas não supervisionadas foram aplicadas

# registros	# features	Categorias de ataques presentes
130439	{32, 7, 3, 2}	DoS
77172	{32, 7, 3, 2}	U2R
80433	{32, 7, 3, 2}	R2L
91130	{32, 7, 3, 2}	Probe

A Figura 1 ilustra todo o processo de preparação dos dados, desde a união dos conjuntos de teste e treinamento até a seleção de *features* e execução da PCA. É importante ressaltar que todos os valores foram normalizados.

4.5. Métrica Utilizada

Ao se tratar de técnicas não supervisionadas a métrica AUC (área sob a curva ROC) é amplamente usada na literatura e de grande valia para comparação.

A curva ROC (*Receiver Operating Characteristcs*) é uma curva em que o eixo X é representado pela taxa de falsos positivos (equação 1), também chamada de *fall-out* e o eixo Y é representado pela taxa de verdadeiros positivos (equação 2), também chamada de *recall*, revocação ou sensibilidade. Segundo [Fawcett 2006], tal curva representa o *tradeoff* entre benefícios (verdadeiros positivos) e custo (falsos positivos).

$$FPR = \frac{FP}{FP + TN} \tag{1}$$

Onde FPR é a taxa de falsos positivos, TN é a quantidade de verdadeiros negativos e FP é a quantidade de falsos positivos.

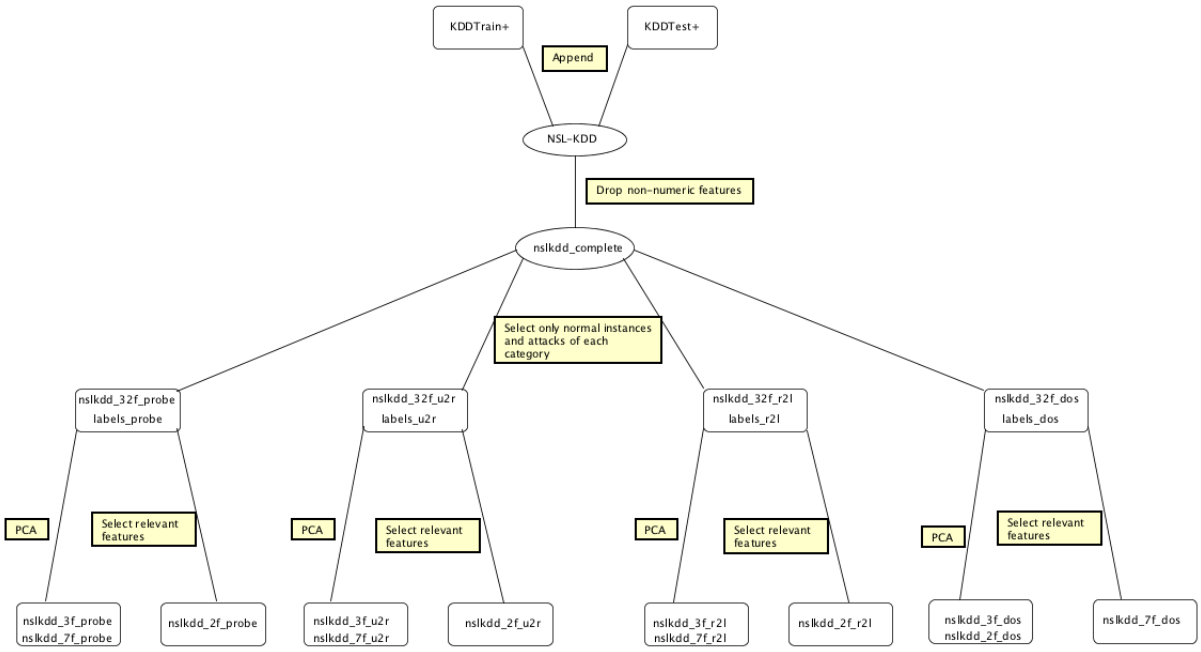

Figura 1. Fluxo para processamento dos dados originais

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

Onde TPR é a taxa de verdadeiros positivos, TP é a quantidade de verdadeiros positivos e FN é quantidade de falsos negativos.

Comparar diversas curvas ROC pode não ser uma tarefa muito simples, dessa forma utiliza-se a área sob as curvas de maneiras a transformá-las em uma única medida escalar. A AUC sempre estará contida no intervalo [0, 1], e valores menores ou iguais a 0.5 indicam classificadores não realistas. AUC igual à 0.5 indica que a curva ROC é uma diagonal x = y, o que significa que o classificador está fazendo o mesmo que predição aleatória. Chamamos essa curva de predição aleatória.

O valor absoluto da área representa a probabilidade de uma instância escolhida aleatoriamente ser classificada como positiva. Mais detalhes sobre a curva ROC e AUC podem ser encontrados em [Fawcett 2006].

5. Resultados e Discussão

Após a execução das 5 técnicas apresentadas na Seção IV nos cenários descritos na Tabela 2 obteve-se os resultados contidos nas Tabelas 3 à 10.

As Figuras 2, 3, 4 e 5 mostram os valores de AUC obtidos para as técnicas com as diferentes quantidades de *features*. Nessas figuras é possível realizar uma rápida comparação de desempenho das técnicas aplicadas para cada categoria de ataque.

A presente seção objetiva discutir os resultados obtidos para cada uma das categorias de ataques analisadas. Serão salientados para discussão os resultados que caem acima da curva de predição aleatória (apresentam AUC superior à 0.5). Também será

feita uma comparação com os resultados obtidos em [Goldstein and Uchida 2016] para os algoritmos HBOS, LOF e técnicas baseadas em clusterização. É importante ressaltar que a análise feita em [Goldstein and Uchida 2016] considera apenas instâncias normais e ataques, ou seja, as anormalidades não são divididas em quatro categorias.

5.1. *Denial of Service*

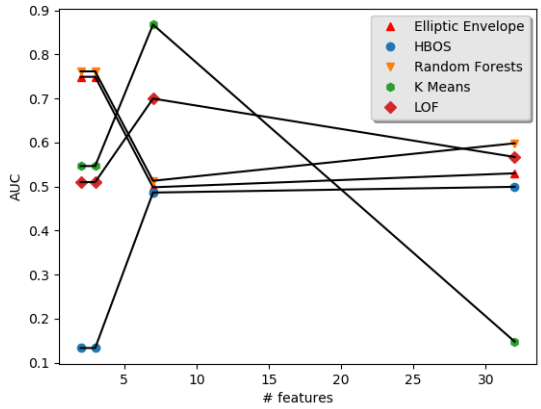

Figura 2. AUC para todas as técnicas nos 4 cenários para ataques DoS

Tabela 3. AUC para os 4 cenários e as 5 técnicas - DoS

	# features			
Técnica	32	7	3	2
Elliptic Envelope	0.5302	0.4984	0.7494	0.7494
HBOS	0.4994	0.4865	0.1333	0.1333
Random Forests	0.5984	0.5132	0.7618	0.7618
K Means	0.1478	0.8679	0.5467	0.5467
LOF	0.5672	0.6999	0.5103	0.5103

Tabela 4. Comparação de resultados DoS com [Goldstein and Uchida 2016]

		NSL-KDD			
Técnica	KDD99	32	7	3	2
K-Means	-	0.1478	0.8679	0.5467	0.5467
K-NN	0.9747	-	-	-	-
K(th)-NN	0.9796	-	-	-	-
LOF	0.5964	0.5672	0.6999	0.5103	0.5103
HBOS	0.9990	0.4994	0.4865	0.1333	0.1333

Os ataques do tipo DoS estão presentes em maior quantidade na base de dados, representam cerca de 40% das instâncias no conjunto que as engloba exclusivamente com os registros normais. A partir da Figura 2 fica evidente que a técnica baseada em histogramas apresenta desempenho insatisfatório (abaixo ou muito próximo da curva de predição aleatória) para todos os cenários considerados.

A técnica LOF apresenta insatisfatória ou baixa eficiência para 3 dos 4 cenários analisados. Apenas para o caso com 7 *features* seu desempenho é relativamente melhor. Tal fato é interessante pois este é o cenário de *features* mais relevantes indicado por [Khan et al. 2018] especificamente para ataques *Denial of Service*.

Ao se comparar os resultados com [Goldstein and Uchida 2016] é possível observar que apenas a técnica HBOS apresentou desempenho muito abaixo do esperado para todos os cenários. As técnicas baseadas em clusterização e o algoritmo LOF obtiveram resultados dentro do intervalo anteriormente observado em pelo menos uma das configurações avaliadas.

5.2. *User to Root*

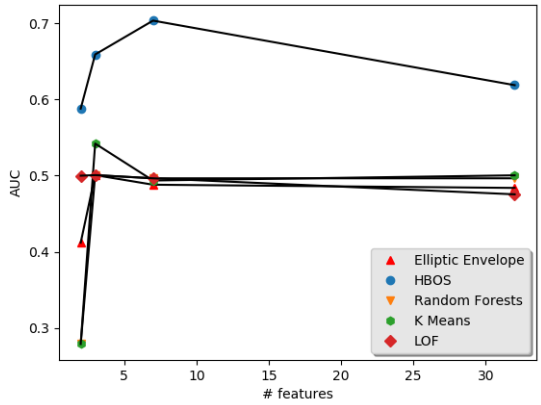

Figura 3. AUC para todas as técnicas nos 4 cenários para ataques U2R

Tabela 5. AUC para os 4 cenários e as 5 técnicas - U2R

	# features			
Técnica	32	7	3	2
Elliptic Envelope	0.4839	0.4881	0.5007	0.4120
HBOS	0.6188	0.7035	0.6591	0.5879
Random Forests	0.4965	0.4965	0.5007	0.2792
K Means	0.5004	0.4936	0.5421	0.2789
LOF	0.4755	0.4965	0.5007	0.4999

Tabela 6. Comparação de resultados U2R com [Goldstein and Uchida 2016]

		NSL-KDD			
Técnica	KDD99	32	7	3	2
K-Means	-	0.5004	0.4936	0.5421	0.2789
K-NN	0.9747	-	-	-	-
K(th)-NN	0.9796	-	-	-	-
LOF	0.5964	0.4755	0.4965	0.5007	0.4999
HBOS	0.9990	0.6188	0.7035	0.6591	0.5879

Os ataques do tipo U2R representam a categoria em menor quantidade na base de dados, são cerca de 0.15% das instâncias no conjunto que as engloba exclusivamente com os registros normais. De fato, uma representatividade tão baixa é característica muito forte de comportamento anômalo. Ou seja, resultados mais altos são esperados.

Todavia, apenas a técnica baseada em histograma apresentou resultados satisfatórios. Tal fato pode ser claramente observado na Figura 3. Todas as outras quatro técnicas obtiveram resultado abaixo ou pouco acima da curva de predição aleatória. O cenário de redução para 7 dimensões apresentou o melhor resultado ao passo que a seleção de *features* sugerida por [Dhanabal and Shantharajah 2015] apresentou o pior.

Dessa forma é possível concluir que a seleção sugerida por [Dhanabal and Shantharajah 2015] não é eficiente e que a redução para 7 dimensões mostra maior eficácia.

Comparando-se os resultados com [Goldstein and Uchida 2016] observa-se discrepância no desempenho obtido pela técnicas de clusterização e o algoritmos HBOS. Apenas a técnica LOF apresenta similaridades com os resultados previamente obtidos para todos os cenários em análise.

5.3. *Remote to User*

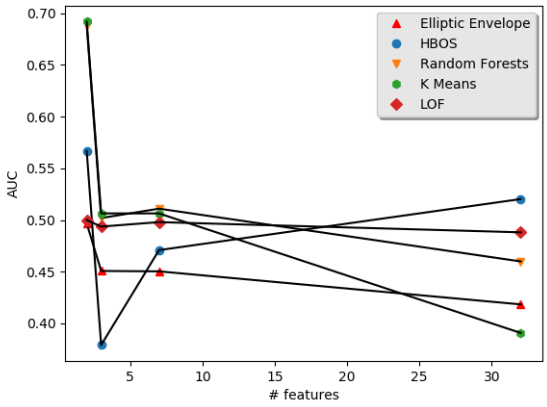

Figura 4. AUC para todas as técnicas nos 4 cenários para ataques R2L

Tabela 7. AUC para os 4 cenários e as 5 técnicas - R2L

	# features			
Técnica	32	7	3	2
Elliptic Envelope	0.4183	0.4503	0.4507	0.4967
HBOS	0.5202	0.4708	0.3790	0.5667
Random Forests	0.4599	0.5109	0.5018	0.6886
K Means	0.3907	0.5063	0.5063	0.6919
LOF	0.4880	0.4979	0.4936	0.5000

Os ataques do tipo R2L também representam uma pequena fração (4.79%) das instâncias presentes no conjunto que as engloba exclusivamente com os registros normais. Mais uma vez, espera-se dessa baixa representatividade bons resultados.

Tabela 8. Comparação de resultados R2L com [Goldstein and Uchida 2016]

		NSL-KDD			
Técnica	KDD99	32	7	3	2
K-Means	-	0.3907	0.5063	0.5063	0.6919
K-NN	0.9747	-	-	-	-
K(th)-NN	0.9796	-	-	-	-
LOF	0.5964	0.4880	0.4979	0.4936	0.5000
HBOS	0.9990	0.5202	0.4708	0.3790	0.5660

Entretanto, apenas duas técnicas apresentaram resultados satisfatórios, o que pode ser facilmente observado na Figura 4, são elas *K Means* e *Random Forests* para o cenário de seleção de features indicadas por [Dhanabal and Shantharajah 2015]. Tal fato é interessante pois mostra a relevância das *features* selecionadas assim como proporciona ganhos em eficiência, já que utiliza o cenário com a menor quantidade de dados.

Após a comparação de resultados com [Goldstein and Uchida 2016] observa-se o mesmo padrão observado para os ataques U2R. Ou seja, apenas a técnica LOF apresentou alguma similaridade com os resultados previamente obtidos por [Goldstein and Uchida 2016] para os cenários avaliados.

5.4. *Probe*

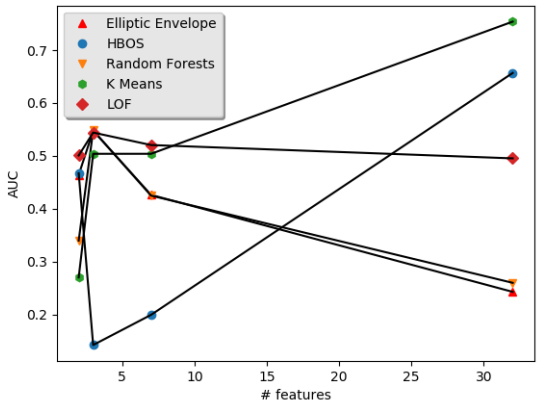

Figura 5. AUC para todas as técnicas nos 4 cenários para ataques Probe

Tabela 9. AUC para os 4 cenários e as 5 técnicas - Probe

	# features			
Técnica	32	7	3	2
Elliptic Envelope	0.2427	0.4261	0.5476	0.4639
HBOS	0.6568	0.1991	0.1421	0.4670
Random Forests	0.2598	0.4248	0.5482	0.3396
K Means	0.7539	0.5039	0.5039	0.2699
LOF	0.4952	0.5205	0.5441	0.5008

Tabela 10. Comparação de resultados Probe com [Goldstein and Uchida 2016]

		NSL-KDD			
Técnica	KDD99	32	7	3	2
K-Means	-	0.7539	0.5039	0.5039	0.2699
K-NN	0.9747	-	-	-	-
K(th)-NN	0.9796	-	-	-	-
LOF	0.5964	0.4952	0.5205	0.5441	0.5000
HBOS	0.9990	0.6568	0.1991	0.1421	0.4670

Ataques do tipo probe também têm baixa representatividade (15.45%) e os resultados obtidos não foram excelentes. Apenas para as técnicas HBOS e *K Means* no cenário com 32 *features* (todas as *features* numéricas) há resultados satisfatórios, ver figura 5. É importante salientar que este resultado é indesejável, pois é computacionalmente mais caro já que faz-se necessário o uso de um grande volume de *features*.

Mais uma vez, apenas a técnica LOF apresenta similaridades com os resultados obtidos em [Goldstein and Uchida 2016] para todos os cenários sob análise.

5.5. Visão Geral

Ao se comparar todos os cenários de execução é possível notar que a técnica *K Means* se sobressai em grande parte deles e que a seleção de *features* e a técnica de redução de dimensões (PCA) podem ou não ser eficientes. Ou seja, a eficiência do uso de técnicas de pré-processamento de dados e de detecção de anomalias é extremamente sensível ao contexto em que são aplicadas. O presente trabalho explicitou em quais contextos e configurações as melhores escolhas para as cinco técnicas analisadas. A Tabela 11 mostra quais as melhores escolhas e em quais contextos elas devem ser aplicadas.

Tabela 11. Melhor algoritmo por categoria de ataque

Categoria de Ataque	#features	Melhor Algoritmo
Denial of Service	7	K Means
User to Root	7	HBOS
Remote to User	2	K Means/Random Forests
Probe	32	K Means

Ficou evidente que os resultados alcançados não foram tão altos quanto outros encontrados na literatura [Goldstein and Uchida 2016]. Tal fato pode ser devido à exploração das particularidades da base de dados analisada. Separá-la em 16 diferentes cenários pode ter contribuído de forma negativa para a eficiência das estratégias aplicadas. É de extrema importância ressaltar a característica mais marcante de anomalias, sua baixa representatividade em relação à quantidade de instâncias consideradas normais. Dessa forma, imagina-se que a aplicação de técnicas supervisionadas nos contextos explorados neste trabalho podem apresentar melhores resultados.

Referências

Almalawi, A., Yu, X., Tari, Z., Fahad, A., and Khalil, I. (2014). An unsupervised anomaly-based detection approach for integrity attacks on scada systems. *Computers & Security*, 46:94–110.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.

Depren, O., Topallar, M., Anarim, E., and Ciliz, M. K. (2005). An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722.

Dhanabal, L. and Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Ferrari, D. G. and De Castro, L. N. (2015). Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 301:181–194.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28.

Goldstein, M. and Dengel, A. (2012). Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63.

Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

Kemmerer, R. A. and Vigna, G. (2002). Intrusion detection: a brief history and overview. *Computer*, 35(4):supl27–supl30.

Khan, S., Gani, A., Wahab, A. W. A., and Singh, P. K. (2018). Feature selection of denial-of-service attacks using entropy and granular computing. *Arabian Journal for Science and Engineering*, 43(2):499–508.

Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE.

Kim, G., Lee, S., and Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4):1690–1700.

Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 251–261. ACM.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 25–36. SIAM.

Leung, K. and Leckie, C. (2005). Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc.

Lin, W.-C., Ke, S.-W., and Tsai, C.-F. (2015). Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, 78:13–21.

Meng, Y.-X. (2011). The practice on using machine learning for network anomaly intrusion detection. In *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*, volume 2, pages 576–581. IEEE.

Münz, G., Li, S., and Carle, G. (2007). Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet*.

Peddabachigari, S., Abraham, A., Grosan, C., and Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. *Journal of network and computer applications*, 30(1):114–132.

Rajasegarar, S., Leckie, C., and Palaniswami, M. (2014). Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 74(1):1833–1847.

Shun, J. and Malki, H. A. (2008). Network intrusion detection system using neural networks. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 5, pages 242–246. IEEE.

Sivarajah, U., Kamal, M. M., Irani, Z., and Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 305–316. IEEE.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–6. IEEE.

Yassin, W., Udzir, N. I., Muda, Z., and Sulaiman, M. N. (2013). Anomaly-based intrusion detection through k-means clustering and naives bayes classification. In *Proc. 4th Int. Conf. Comput. Informatics, ICOCI*, number 49, pages 298–303.

Zhang, J., Zulkernine, M., and Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5):649–659.