Comparação de Algoritmos Não Supervisionados para Detecção de Anomalias

Fabrício A Silva[1] e Luis G A Diniz[2]

***Abstract*—**

I. 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 computcionais 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 [23]. 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. Um sistema capaz de realizar tal tarefa é conhecido com NIDS (Network Intrusion Detection System). Há dois importantes métodos para detectar atividades suspeitas [10]: *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 [2], dentre elas pode-se citar análise estatística, algoritmos de classificação, algoritmos de clusterização e grafos. Estas são apenas algumas estratégias utilizadas, mais sobre elas e outras pode ser encontrado em [7].

O presente trabalho dará 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. [6] mostra que há diferentes maneiras de se aplicar clustering, cada um dos algoritmos mencionados por [6] 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, especificamente para ataques da categoria *Denial of Service* (DoS). Resultado esperado: uma análise comparativa entre diferentes técnicas na NSL-KDD, em termos de desempenho computacional, precisão, atributos escolhidos, entre outros.

O texto é organizado em mais 6 seções. A seção II consiste numa revisão de literatura. A seção III trata especificamente de detalhes da NSL-KDD. A seção IV discorre sobre o funcionamento das técnicas escolhidas e seu funcionamento. Na seção V detalhes da execução do trabalho são discutidos. Por fim, as seções VI e VII apresentam, respectivamente, os resultados e conclusões sobre a comparação das técnicas.

II. Trabalhos Relacionados

Diversas estratégias foram propostas para o desenvolvimento de sistemas de detecção de instrusos. Tais estratégias possuem inúmeras possíveis abordagens, 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. [22] mostra resultados promissores utilizando redes do tipo *feed-forward* 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), [14] executa sólida análise baseada em logs de servidores apache e obtêm resultados satisfatórios. O uso de técnicas de *machine learning* também é uma alternativa, [18] deixa claro que o uso dessa estratégia deve ter especial atenção já que não funciona de forma eficiente em todos os cenários de IDS. [24] 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, no caso específico um conjunto de dados livre de 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, [19] utiliza o clássico algoritmo *K-means* para executar detecção de anomalias. [16] combina dois tipos de clustering (i.e. *grid e density*) e obtém resultados satisfatórios, todavia, a taxa de falsos positivos apresentada é alta. [21], por sua vez, implementa clustering hiperesférico para aplicar detecção de instrusos em redes sem fio. [26] combina o *K-means* com o *Naïve Bayes Classification* e apresenta resultados que comprovam significativa precisão. [1] 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. [13] utiliza *misuse detection* e detecção de anomalias combinadas para identifcar intrusos, os resultados apresentados têm melhor tempo de execução do que os modelos convencionais. [3] segue a mesma estratégia que [13] e comprova qua a combinação dos algoritmos é mais eficiente que cada um deles isolado. [17] combina clustering com a técnica dos vizinhos mais próximos e compara seus resultados com soluções recorrentes na literatura. [20] implementa árvore de decisão combinada com *suport vector machine* (SVM) e obtem bons resultados. [27] 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 anoamlias 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. [12] também compara diversas estratégias e abordagens mas não foca em práticas não supervisionadas. [15] realiza comparação similar mas com foco na base de dados DARPA 1998. Por sua vez, [9] 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.

III. 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 (citar artigos), 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 [4] registros redundantes foram removidos, de forma a não influenciar nos resultados. [25] 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.

A. 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).

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 à negar acesso de serviço à usuários.

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

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 de user to root, ou seja, um usuário simples que tenta obter privilégios de root.

4) Remote to User (U2L): 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.

IV. METODOLOGIA

A. 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 [9]. Mais detalhes podem ser encontrados em [8].

Das quatro categorias de ataque listadas na seção III apenas a *Denial of Service* foi analisada. Essa escolha foi baseada na grande quantidade de instâncias desse tipo na base de dados. Dessa forma, para que apenas dados relevantes fossem usados como entrada nas execuções uma pré processamento específico foi realizado.

B. 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. Após o processamento a seguinte configuração foi estabelecida:

TABLE I
CONFIGURAÇÃO DA BASE DE DADOS APÓS PRÉ PROCESSAMENTO

Classe	KDDTrain+	KDDTest+	Total
Normal	67342	9711	77053
DoS	45927	7459	53386
Normal+DoS	113269	17170	130439

Para fins de comparação a base de dados original com todas as categorias de ataque e as instâncias classificadas como normais também foi utilizada, abaixo segue tabela com configuração original:

TABLE II
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

C. Feature Selection

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

Como o foco do trabalho são ataques da categoria DoS, 7 features numéricas específicas e ditas relevantes, segundo [11], foram selecionadas como outro cenário de execução. É importante salientar que houve normalização de todos os dados utilizados para que estes respeitassem a mesma escala.

O objetivo de selecionar um subconjunto de features relevantes é eliminar a necessidade de usar todas as 32 colunas numéricas e ainda sim obter resultados satisfatórios.

1) PCA: A técnica *Principal Component Analysis* (PCA) também foi utilizada a partir das subconjunto indicado em [11]. Dessa forma, conjuntos com 2 e 3 features foram obtidos.

2) Outras técnicas: Outras técnicas de feature selection (*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.

D. Cenários de execução

Há então 4 cenários de execução e comparação das técnicas não supervisionadas aplicadas. A tabela 3 explicita cada um desses cenários com a quantidade de registros, features e os tipos de ataques presentes.

E. 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.

TABLE III

CENÁRIOS EM QUE TÉCNICAS NÃO SUPERVISIONADAS FORAM APLICADAS

# registros	# features	Categorias de ataques presentes
148515	32	Todos as categorias
130439	7	DoS
130439	3	DoS
130439	2	DoS

A curva ROC (*Receiver Operating Characteristcs*) é uma curva em que o eixo X é representado pela taxa de falsos positivos (equação 1) e o eixo Y é representado pela taxa de verdadeiros positivos (equação 2), também chamado de recall, revocação ou sensibilidade. Segundo [5], tal curva representa o tradeoff entre benefícios (verdadeiros positivos) e custo (falsos positivos).

$$FPR = \frac{FP}{N} \tag{1}$$

Onde FRP é a taxa de falsos positivos, FP é a quantidade de falsos positivos e N é a quantidade de instâncias negativas.

$$TPR = \frac{TP}{P} \tag{2}$$

Onde TRP é a taxa de verdadeiros positivos, TP é a quantidade de verdadeiros positivos e P é quantidade de instâncias positivas.

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 [5].

V. RESULTADOS

Após a execução das 5 técnicas definidas na seção IV nos 4 cenários descritos na tabela 3 obteve-se os resultados contidos na tabela 4.

A figura 1 mostra os valores de AUC obtidos para as técnicas com as diferentes quantidades de features.

TABLE IV

AUC PARA OS 4 CENÁRIOS E AS 5 TÉCNICAS

	# features			
Técnica	32	7	3	2
Elliptic Envelope	0.4051	0.5690	0.5817	0.6379
HBOS	0.5336	0.4846	0.2291	0.1821
Random Forests	0.4991	0.5059	0.6933	0.7150
K Means	0.1927	0.8679	0.8679	0.8679
LOF	0.5534	0.6954	0.6812	0.6776

Nessa figura é possível realizar uma rápida comparação de desempenho das técnicas aplicadas.

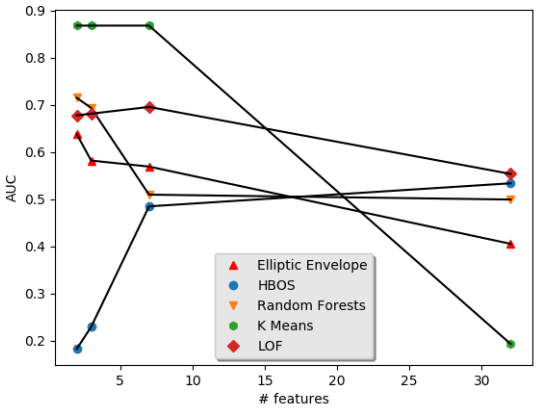

Fig. 1. AUC para todas as técnicas nos 4 cenários

VI. DISCUSSÃO DE RESULTADOS

Os valores obtidos para os cenários estabelecidos não atingiram, na maioria dos casos, desempenho satisfatório. A partir da figura 1 é possível observar que todas as técnicas exceto HBOS têm bons resultados (melhores que predição aleatória) para o cenário em que apenas 2 features são consideradas. A mesma observação pode ser feita para o cenário com 3 features. É importante, no entanto, chamar atenção para o valor encontrado para o Elliptic Envelope, que apesar de estar acima da curva de predição aleatória ainda está muito próximo.

Para o cenário sugerido por [11] LOF e K Means se destacam com valores mais elevados, HBOS e Random Forests são duas técinas que não devem ser consideradas para esta configuração já que estão muito próximas da curva de predição aleatória. Para o cenário com todas as features numéricas todas as técnicas obtiveram valores abaixo de 0.56, o que nos permite concluir que nenhuma delas deve ser aplicada nesta configuração pois o desempenho médio é ligeiramente pior que predição aleatória.

De fato, não era esperado que bons resultados fossem atingidos ao se utilizar todas as características numéricas disponíveis. É evidente que este cenário apresenta muita informação irrelevante para análise. É interessante o fato de que o cenário com a menor quantidade de features apresentou os melhores resultados. Tal fato deve-se à sucessiva aplicação de seleção de features [11] e redução de dimensões (PCA).

Ficou claro que K Means apresentou melhores resultados em todos os cenários relevantes (2, 3 e 7 features) e que HBOS apresentou resultados insatisfatórios em todas as configurações. Dessa forma, para detecção de ataques da categoria *Denial of Serivce* na base de dados NSL-KDD recomenda-se o uso da técnica não supervisionada K Means. Vale ressaltar que, a técina *Local Outlier Factor* também é uma boa alternativa para o problema. LOF não apresenta resultados tão precisos quanto K Means mas ainda sim mostra desemprenho satisfatório.

VII. Trabalhos Futuros

Tendo em vista ampliar a comparação das técnicas aplicadas no presente trabalho propõe-se como trabalho futuro o estudo específico de cada uma das outras três categorias de ataque.

References

[1] Abdulmohsen Almalawi, Xinghuo Yu, Zahir Tari, Adil Fahad, and Ibrahim Khalil. An unsupervised anomaly-based detection approach for integrity attacks on scada systems. *Computers & Security*, 46:94–110, 2014.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[3] Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.

[4] L Dhanabal and SP Shantharajah. 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, 2015.

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

[6] Daniel Gomes Ferrari and Leandro Nunes De Castro. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 301:181–194, 2015.

[7] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1):18–28, 2009.

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

[9] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[10] Richard A Kemmerer and Giovanni Vigna. Intrusion detection: a brief history and overview. *Computer*, 35(4):supl27–supl30, 2002.

[11] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, and Prem Kumar Singh. Feature selection of denial-of-service attacks using entropy and granular computing. *Arabian Journal for Science and Engineering*, 43(2):499–508, 2018.

[12] Kevin S Killourhy and Roy A Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE, 2009.

[13] Gisung Kim, Seungmin Lee, and Sehun Kim. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4):1690–1700, 2014.

[14] Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 251–261. ACM, 2003.

[15] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. 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, 2003.

[16] Kingsly Leung and Christopher Leckie. 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., 2005.

[17] Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai. Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, 78:13–21, 2015.

[18] Yu-Xin Meng. 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, 2011.

[19] Gerhard Münz, Sa Li, and Georg Carle. Traffic anomaly detection using k-means clustering. In *GI/ITG Workshop MMBnet*, 2007.

[20] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan, and Johnson Thomas. Modeling intrusion detection system using hybrid intelligent systems. *Journal of network and computer applications*, 30(1):114–132, 2007.

[21] Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 74(1):1833–1847, 2014.

[22] Jimmy Shun and Heidar A Malki. Network intrusion detection system using neural networks. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, volume 5, pages 242–246. IEEE, 2008.

[23] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[24] Robin Sommer and Vern Paxson. 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, 2010.

[25] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. 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, 2009.

[26] Warusia Yassin, Nur Izura Udzir, Zaiton Muda, and Md Nasir Sulaiman. 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, 2013.

[27] Jiong Zhang, Mohammad Zulkernine, and Anwar Haque. Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5):649–659, 2008.