Uma pesquisa sobre mineração de dados e métodos de aprendizado de máquinas para detecção de intrusão de segurança cibernética

Anna L. Buczak, *Membro, IEEE*, e Erhan Guven, *Membro, IEEE*

Resumo *- Este* **documento de pesquisa descreve uma pesquisa bibliográfica focada na aprendizagem de máquinas (ML) e métodos de mineração de dados (DM) para a ciberanálise em apoio à detecção de intrusão. São fornecidas curtas descrições tuto- riais de cada método ML/DM. Com base no número de citações ou na relevância de um método emergente, os artigos que representam cada método foram identificados, lidos e somados. Como os dados são tão importantes nas abordagens ML/DM, alguns conjuntos de dados cibernéticos bem conhecidos usados em ML/DM são descritos. A complexidade dos algoritmos ML/DM é abordada, a discussão dos desafios para o uso de ML/DM para segurança cibernética é apresentada, e algumas recomendações sobre quando usar um determinado método são fornecidas.**

Termos *de índice* - Análise *cibernética*, mineração de dados, aprendizado de máquinas.

I. Introdução

TO documento HIS apresenta os resultados de uma pesquisa bibliográfica sobre aprendizagem de máquinas (ML) e mineração de dados (DM).
ods para aplicações de segurança cibernética. Os métodos ML/DM são descritos, assim como várias aplicações de cada método para problemas de detecção de intrusão cibernética. A complexidade dos diferentes algoritmos ML/DM é discutida, e o documento fornece um conjunto de critérios de comparação para os métodos ML/DM e um conjunto de recomendações sobre os melhores métodos a serem usados, dependendo das características do problema cibernético a ser resolvido.

A segurança cibernética é o conjunto de tecnologias e processos projetados para proteger computadores, redes, programas e dados contra ataques, acessos não autorizados, mudanças ou destruição. Os sistemas de segurança cibernética são compostos de sistemas de segurança de redes e sistemas de segurança de computadores (host). Cada um deles tem, no mínimo, um firewall, software antivírus e um sistema de detecção de intrusão (IDS). Os IDSs ajudam a descobrir, determinar e identificar o uso não autorizado, duplicação, alteração e destruição dos sistemas de informação [1]. As violações de segurança incluem intrusões externas (ataques de fora da organização) e intrusões internas (ataques de dentro da organização).

Há três tipos principais de análise cibernética em sup- porta de IDSs: baseada em uso indevido (às vezes também chamada de baseada em assinatura), baseada em anomalias, e híbrida. As técnicas baseadas em abuso são projetadas para detectar ataques conhecidos, utilizando assinaturas desses ataques. Elas são eficazes para detectar tipos conhecidos de ataques sem gerar um número avassalador de falsos ataques.

Manuscrito recebido em 9 de março de 2015; revisado em 31 de agosto de 2015; aceito
16 de outubro de 2015. Data de publicação 26 de outubro de 2015; data da versão atual 20 de maio de 2016. Este trabalho foi apoiado pela Divisão de Implantação de Segurança de Redes do Departamento de Segurança Nacional sob contrato HSSA01-13- C-2709 (Chris Yoon, Gerente de Programa).

Os autores estão com The Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723 USA (e-mail: anna.buczak@jhuapl.edu).

Identificador de Objeto Digital 10.1109/COMST.2015.2494502

alarmes. Eles exigem atualizações manuais freqüentes do banco de dados com regras e assinaturas. Técnicas baseadas em erros não podem detectar ataques novos (dia zero).

Técnicas baseadas em anomalias modelam o comportamento normal da rede e do sistema, e identificam anomalias como desvios do comportamento nor- mal. Elas são atraentes por causa de sua capacidade de detectar ataques de dia zero. Outra vantagem é que os perfis de atividade normal são personalizados para cada sistema, aplicação ou rede, tornando assim difícil para os atacantes saber quais atividades podem realizar sem serem detectados. Além disso, os dados sobre quais técnicas baseadas em anomalias alertam (novos ataques) podem ser usados para definir as assinaturas dos detectores de uso indevido. A principal desvantagem das técnicas baseadas em anomalias é o potencial de altas taxas de falsos alarmes (FARs), pois os comportamentos do sistema anteriormente invisíveis (mas legítimos) podem ser categorizados como anomalias. As técnicas híbridas combinam o uso indevido e a detecção de anomalias. Elas são empregadas para aumentar as taxas de detecção de intrusões conhecidas e diminuir a taxa de falsos positivos (FP) para ataques desconhecidos. Uma revisão profunda da literatura não descobriu muitos métodos de detecção de anomalias puras; a maioria dos métodos eram realmente híbridos. Portanto, nas descrições dos métodos ML e DM meth- ods, os métodos de detecção de anomalias e híbridos são descritos juntos.

Outra divisão dos IDSs se baseia em onde eles procuram comportamento intrusivo: baseado em rede ou baseado em host. Um IDS baseado em rede identifica intrusões através do monitoramento do tráfego através de dispositivos de rede. Um IDS baseado em host monitora processos e atividades de arquivos relacionados ao ambiente de software associado a um host específico.

Este trabalho de pesquisa se concentra nas técnicas ML e DM para segurança cibernética, com ênfase nos métodos ML/DM e suas descrições. Muitos artigos descrevendo estes métodos foram publicados, incluindo várias revisões. Em contraste com as revisões prévias, o foco de nosso trabalho está nas publicações que atendem a certos critérios. As consultas do Google Scholar foram realizadas usando "machine learning" e cyber, e usando "data mining" e cyber. Foi dada ênfase especial a artigos altamente citados porque estes descreviam técnicas populares. No entanto, também foi reconhecido que esta ênfase poderia ignorar técnicas novas e emergentes significativas, portanto, alguns destes trabalhos também foram escolhidos. De modo geral, os trabalhos foram selecionados de modo que cada uma das categorias ML/DM listadas posteriormente tivesse pelo menos um e, de preferência, alguns trabalhos representativos.

Este artigo é destinado aos leitores que desejam iniciar pesquisas no campo do ML/DM para a detecção de intrusão cibernética. Como tal, grande ênfase é colocada em uma descrição completa dos métodos ML/DM, e referências a trabalhos seminais para cada ML

e o método DM são fornecidos. Alguns exemplos são fornecidos sobre como as técnicas foram utilizadas na segurança cibernética.

Este documento não descreve todas as diferentes técnicas de detecção de anomalias de rede, como fazem Bhuyan et al. [2]; em vez disso, ele se concentra apenas nas técnicas de ML e DM. No entanto, além da detecção de anomalias, são descritos métodos híbridos e baseados em assinaturas. As descrições dos métodos na presente pesquisa são mais profundas do que em [2].

Nguyen et al. [3] descrevem as técnicas ML para a classificação do tráfego na Internet. As técnicas descritas não se baseiam em números de portas bem conhecidos, mas em características estatísticas de tráfego - tiques. Sua pesquisa cobre apenas trabalhos publicados em 2004 a 2007, onde nossa pesquisa inclui trabalhos mais recentes. Ao contrário de Nguyen et al. [3], este trabalho apresenta métodos que funcionam em qualquer tipo de dados cibernéticos, não apenas fluxos do Protocolo Internet (IP).

Teodoro et al. [4] concentram-se em técnicas de intrusão de rede baseadas em anomalias. Os autores apresentam abordagens estatísticas, baseadas no conhecimento e na aprendizagem por máquina, mas seu estudo não apresenta um conjunto completo de métodos de aprendizagem por máquina de última geração. Em contraste, este trabalho descreve não apenas a detecção de anomalias, mas também métodos baseados em assinaturas. Nosso trabalho também inclui os métodos para reconhecimento do tipo de ataque (uso indevido) e para a detecção de um ataque (intrusão). Por último, nosso documento apresenta a lista completa e mais recente dos métodos ML/DM que são aplicados à segurança cibernética.

Sperotto et al. [5] concentram-se nos dados de fluxo de rede (NetFlow) e apontam que o processamento de pacotes pode não ser possível nas velocidades de fluxo devido à quantidade de tráfego. Eles descrevem um amplo conjunto de métodos para detectar o tráfego anômalo (possível ataque) e o uso indevido. Entretanto, ao contrário de nosso trabalho, eles não incluem explicações sobre os detalhes técnicos dos métodos individuais.

Wu et al. [6] concentram-se nas metáforas de Inteligência Computacional - ods e suas aplicações à detecção de intrusão. Métodos tais como Redes Neurais Artificiais (ANNs), Sistemas Fuzzy, Computação Evolutiva, Sistemas Imune Artificiais e Inteligência de Enxame são descritos em grande detalhe. Como somente os métodos de Inteligência Computacional são descritos, os principais métodos ML/DM, tais como agrupamento, árvores de decisão e mineração de regras (que este artigo aborda) não estão incluídos.

Este documento se concentra principalmente na detecção de intrusão cibernética, uma vez que se aplica a redes com fio. Com uma rede cabeada, um adversário deve passar por várias camadas de defesa em firewalls e sistemas operacionais, ou obter acesso físico à rede. No entanto, uma rede sem fio pode ser direcionada a qualquer nó, portanto é naturalmente mais vulnerável a ataques maliciosos do que uma rede cabeada. Os métodos ML e DM cobertos neste documento são totalmente aplicáveis aos problemas de detecção de intrusão e uso indevido tanto em redes com e sem fio. O leitor que deseja uma perspectiva focada apenas na proteção de redes sem fio é referido a documentos como Zhang et al. [7], que se concentra mais na topologia de rede em mudança dinâmica, algoritmos de roteamento, gerenciamento descentralizado, etc.

O restante deste documento está organizado da seguinte forma: A Seção II focaliza os principais passos na ML e DM. A Seção III discute os conjuntos de dados de segurança cibernética usados no ML e DM. A Seção IV descreve os métodos individuais e documentos relacionados para ML e DM em segurança cibernética. A Seção V

discute a complexidade computacional de diferentes methods. A Seção VI descreve as observações e recomendações. Finalmente, a Seção VII apresenta conclusões.

II. Principais Passos em ML e DM

Há muita confusão sobre os termos ML, DM e Knowledge Discovery in Databases (KDD). KDD é um processo completo que lida com a extração de informações úteis, anteriormente desconhecidas (ou seja, conhecimento) de dados [8]. DM é uma etapa parcial neste processo - a aplicação de algoritmos específicos para extrair padrões a partir de dados. As etapas adicionais no processo KDD (preparação dos dados, seleção dos dados, limpeza dos dados, incorporação de conhecimento prévio apropriado, e a correta pretação dos resultados da DM) garantem que o conhecimento útil seja extraído dos dados disponíveis. Entretanto, há muitos cátions publi- cations [por exemplo, Cross Industry Standard Process for Data Mining (CRISP-DM) [9]] e participantes da indústria que chamam todo o processo KDD de DM. Neste artigo, seguindo Fayyad et al. [8], o DM é usado para descrever uma etapa particular do KDD que trata da aplicação de algoritmos específicos para extrair padrões dos dados.

Há uma sobreposição significativa entre ML e DM. Estes dois termos são comumente confundidos porque freqüentemente empregam os mesmos métodos e, portanto, se sobrepõem significativamente. O pio- neer do ML, Arthur Samuel, definiu o ML como um "campo de estudo que dá aos computadores a capacidade de aprender sem ser explicado-elemente programado". O ML se concentra na classificação e predição, com base nas propriedades conhecidas previamente aprendidas com os dados de treinamento. Os algoritmos ML precisam de um objetivo (formulação do problema) do domínio (por exemplo, variável dependente para prever). DM concentra-se na descoberta de propriedades previamente desconhecidas nos dados. Ele não precisa de um objetivo específico do domínio, mas, em vez disso, concentra-se em encontrar novos e interessantes conhecimentos.

Pode-se ver a ML como o irmão mais velho da DM. O termo mineração de dados foi introduzido no final dos anos 80 (a primeira conferência KDD foi realizada em 1989), enquanto o termo aprendizagem de máquinas está em uso desde os anos 60. Atualmente, o irmão mais novo (ou seja, o uso do termo DM) é mais popular que o mais velho, o que pode ser a razão pela qual alguns pesquisadores realmente rotulam seu trabalho como DM ao invés de ML. Esta poderia ser a razão pela qual, quando consultas "machine learning" E cyber e "data min- ing" E cyber foram realizadas no Google Scholar, o primeiro obteve 21.300 resultados e o segundo 40.800 resultados. Os métodos usados nos papéis recuperados pela primeira consulta não foram significativamente diferentes dos usados nos papéis recuperados pela segunda consulta. Portanto, como este trabalho se concentra em métodos, chamaremos estes métodos de métodos ML/DM.

Uma abordagem ML geralmente consiste de duas fases: treinamento e testes. Muitas vezes, são realizadas as seguintes etapas:

- Identificar os atributos (características) e as classes a partir dos dados do trem.
- Identificar um subconjunto dos atributos necessários para a classificação (ou seja, redução da dimensionalidade).
- Aprenda o modelo usando dados de treinamento.
- Use o modelo treinado para classificar os dados desconhecidos.

Em caso de detecção de uso indevido, na fase de treinamento cada

A aula de mau uso é aprendida usando exemplos apropriados de

o conjunto de treinamento. Na fase de teste, novos dados são passados através do modelo e o exemplar é classificado quanto a se pertence a uma das classes de uso indevido. Se o exemplar não pertencer a nenhuma das classes de mau uso, ele é classificado como normal.

No caso de detecção de anomalias, o padrão normal de tráfego é definido na fase de treinamento. Na fase de testes, o modelo aprendido é aplicado aos novos dados, e cada exemplar do conjunto de testes é classificado como normal ou anômalo.

Na realidade, para a maioria dos métodos ML, deveria haver três fases, não duas: treinamento, validação e testes. Os métodos ML e DM freqüentemente têm parâmetros como o número de camadas e nós para uma ANN. Após a conclusão do treinamento, geralmente há vários modelos (por exemplo, ANNs) disponíveis. Para decidir qual deles usar e ter uma boa estimativa do erro que irá atingir em um conjunto de teste, deve haver um terceiro conjunto de dados separado, o conjunto de dados valida- tion. O modelo que melhor executa os dados de validação deve ser o modelo utilizado, e não deve ser aperfeiçoado dependendo de sua precisão no conjunto de dados de teste. Caso contrário, a precisão relatada é otimista e pode não refletir a precisão que seria obtida em outro conjunto de teste semelhante, mas ligeiramente diferente, do conjunto de teste existente.

Existem três tipos principais de abordagens ML/DM: não-considerada, semi-supervisionada e supervisionada. Em problemas de aprendizagem sem supervisão, a tarefa principal é encontrar padrões, estruturas ou conhecimentos em dados não rotulados. Quando uma parte dos dados é rotulada durante a aquisição dos dados ou por especialistas humanos, o problema é chamado de aprendizagem semi-supervisionada. A adição dos dados etiquetados ajuda muito a resolver o problema. Se os dados forem completamente etiquetados, o problema é chamado aprendizagem supervisionada e geralmente a tarefa é encontrar uma função ou modelo que explique os dados. As abordagens tais como ajuste de curvas ou métodos de aprendizagem por máquina são usados para modelar os dados para o problema subjacente. A etiqueta é geralmente a variável de negócio ou problema que os especialistas assumem ter relação com os dados coletados.

Uma vez que um modelo de classificação é desenvolvido utilizando dados de validação e treinamento, o modelo pode ser armazenado para que possa ser utilizado posteriormente ou em um sistema diferente. O Predictive Model Markup Language (PMML) é desenvolvido e proposto pelo Data Mining Group para ajudar no compartilhamento do modelo preditivo [10]. Ela é baseada em XML e atualmente suporta regressões logísticas e classificadores de redes neurais (NN) feed-forward. A última versão (4.2) suporta os classificadores Naïve Bayes, k-Nearest Neighbor (k-NN), e Support Vector Machine (SVM). O modelo suporta vários metadados comuns de DM, tais como um dicionário de dados (por exemplo, discreto, booleano, numérico), normalização, nome do modelo, atributos do modelo, esquema de mineração, tratamento e saída. Algumas plataformas populares de mineração de dados, como Weka [11], R [12] e RapidMiner [13] suportam modelos PMML.

O modelo CRISP-DM [9] ilustra (ver Fig. 1) as fases e paradigmas comumente utilizados pelos especialistas em DM para resolver problemas. O modelo é composto das seguintes seis fases:

- *Entendimento comercial*: Definição do problema de DM moldado pelas exigências do projeto.
- *Compreensão dos dados*: Coleta e exame dos dados.
- *Preparação dos dados*: Todos os aspectos da preparação dos dados para chegar ao conjunto de dados final.

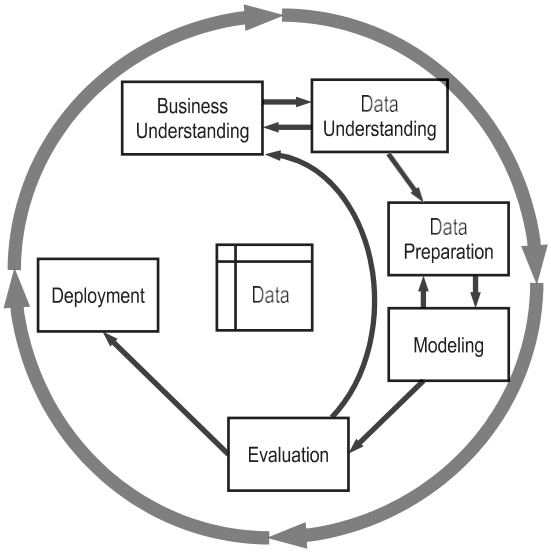


Fig. 1. Diagrama do Processo CRISP-DM.

TABELA I
MATRIZ DE CONFUSÃO
BINÁRIA

□TP, TN, FP, e FN representam, respectivamente, Verdadeiro Positivo, Verdadeiro Negativo, Falso Positivo, e Falso Negativo

	Actual class: X	**Actual class: not X**
Predicted class: X	TP*	FP*
Predicted class: not X	FN*	TN*

- *Modelagem*: Aplicando os métodos DM e ML e otimizando os parâmetros para se adequar ao melhor modelo.
- *Avaliação*: Avaliar o método com rics apropriados para verificar se as metas comerciais foram atingidas.
- *Implantação*: Varia desde a apresentação de um relatório até uma implementação completa da coleta de dados e da estrutura de modelagem. Normalmente, o analista de dados engaja as fases até a implantação, enquanto o cliente executa a fase de implantação.

Há várias métricas de classificação para os métodos ML/DM. Certas métricas são chamadas por dois ou mesmo três nomes diferentes. Na Seção IV, os artigos são descritos com os nomes métricos usados pelos autores dos artigos correspondentes. Para facilitar a compreensão dessa seção, as métricas com seus diferentes nomes são descritas a seguir. Para um problema de classificação binária, as métricas são computadas a partir da matriz de confusão (ver Tabela I).

As métricas freqüentemente utilizadas para a classificação binária (aprendizagem super-visual) são problemas:

- Precisão ou Proporção Correta: (TP + TN)/ (TP + TN + FP + FN). Quando as classes são equilibradas, esta é uma boa medida; no entanto, quando as classes são desequilibrado (por exemplo, 97% dos itens pertencem à classe X e 3% à classe Y, se todos os itens forem classificados como X, a precisão seria de 97%, mas todos os itens da classe Y seriam mal classificados), esta métrica não é muito útil.
- Valor Predictivo Positivo (PPV) ou Precisão: TP/(TP + FP). Razão de itens corretamente classificados como X para todos os itens classificados como X.
- Sensibilidade ou Rechamada ou Taxa Positiva ou Probabilidade de Detecção Verdadeira (P_D) ou Taxa de Detecção: TP/(TP + FN).

Razão de itens corretamente classificados como X para todos os itens que pertencem à classe X.

- Valor Preditivo Negativo (VPL): TN/(TN + FN). Razão de itens corretamente classificados como negativos (não X) para todos os itens classificados como não X.
- Especificidade ou taxa TN: TN/(TN + FP). Razão de itens corretamente classificados como negativos (não X) para todos os itens que pertencem à classe não X.
- FAR ou FP Rate ou Fall-out: FP/(TN + FP). FAR = 1-Específica. Razão de itens classificados incorretamente como pós-itivos (X) para todos os itens que pertencem a uma classe não X.

Em problemas de classificação, há um trade-off entre Sensibilidade e FAR (1-Specificidade). Este trade-off é ilustrado por uma curva da Característica Operacional do Receptor (ROC). A ROC tem FAR no eixo x e a Sensibilidade no eixo y. Como a debulha - antiga para classificação é alterada, um ponto diferente na ROC é escolhido com FAR diferente e Sensibilidade diferente. Uma FAR superior resulta em uma Sensibilidade superior e uma FAR inferior em uma Sensibilidade inferior. O ponto sobre o ROC que proporciona uma melhor classificação depende da aplicação. Muitas vezes, a FAR não pode ser superior a um determinado número, e é assim que se escolhe o classificador final.

Para um problema de várias classes (classificação em mais de duas classes), geralmente são utilizadas as seguintes métricas:

- *Precisão geral*: Exemplares classificados corretamente, todos os exemplos.
- *Taxa de detecção de classe*: Exemplares de uma determinada classificação de classe - todos os exemplos de uma determinada classe.
- *Taxa de FP de classe FAR ou de classe FP*: Exemplares de uma determinada classe classificada incorretamente, todos os exemplos não de uma determinada classe.

Também é possível calcular o VPP e o VPL por classe, mas nos trabalhos revisados e descritos na Seção IV, essas métricas não foram utilizadas.

Há dois tipos de métricas para métodos não supervisionados: interna e externa. As métricas internas são usadas nos dados que foram agrupados, e as etiquetas de classe (porque são desconhecidas pelo algoritmo de agrupamento) não são usadas para computar essas métricas. Métricas como a distância entre clusters (distância entre dois clusters diferentes, poderia ser entre seus centróides), distância intra-agrupamento (distância entre membros de um mesmo cluster, poderia ser a distância média ou distância entre membros mais distantes), e índice Dunn (identifica clusters densos e bem separados) são freqüentemente utilizados.

As métricas externas operam em um conjunto de dados para o qual as etiquetas de classe são conhecidas. As métricas utilizadas se assemelham às métricas de aprendizagem supervisionada. Vários dos trabalhos descritos na Seção IV utilizam métodos não supervisionados, mas as métricas finais fornecidas são a Taxa de Detecção de Classe e a FAR de Classe. Isto significa que, embora o método tenha sido desenvolvido de forma não supervisionada, havia etiquetas disponíveis para os dados de teste, de modo que foi possível calcular as métricas de classificação.

III. Conjunto de dados de ciber-segurança para ML e DM

Para as abordagens ML e DM, os dados são de grande importância. Como estas técnicas aprendem com os dados disponíveis, é necessário ter uma compreensão dos dados que eles utilizam para entender como diferentes autores aplicaram diferentes ML e DM.

Algoritmos DM. Esta seção descreve em detalhes os diferentes tipos de dados usados pela captura de pacotes de abordagens ML e DM (pcap), NetFlow e outros dados da rede. Portanto, a seção IV, que descreve os métodos em detalhes, cita apenas se um método usa pcap, NetFlow ou outros dados de rede e não descreve os dados em detalhes. As seguintes subsecções cobrem os detalhes de baixo nível dos conjuntos de dados.

A. Dados em nível de pacote

Existem 144 IPs listados pela Internet Engineering Task Force (IETF), incluindo protocolos amplamente utilizados como o Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), Internet Gateway Management Protocol (IGMP), etc. Os programas dos usuários que executam estes protocolos geram o tráfego de pacotes da rede da Internet. Os pacotes de rede recebidos e transmitidos na interface física (por exemplo, porta Ethernet) do com- puter podem ser capturados por uma interface específica de programação de aplicação (API) chamada pcap. Libpcap e WinPCap (as versões Unix e Windows, respectivamente) são as bibliotecas front-end de software de captura de pacotes para muitas ferramentas de rede, incluindo analisadores de protocolo, sniffers de pacotes, monitores de rede, IDSs de rede e geradores de tráfego. Alguns programas populares que usam dados pcap são tcpdump [14], Wireshark [15], Snort [16], e Nmap [17].

Na camada física da rede, um quadro Ethernet é composto pelo cabeçalho Ethernet (ou seja, endereço MAC (Media Access Control)), e até 1500 bytes (Unidade Máxima de Transmissão [MTU]) de carga útil. Esta carga útil contém o pacote IP, que é composto pelo cabeçalho IP (ou seja, camada de transporte), e a carga útil IP. A carga útil IP pode conter dados ou outros protocolos encapsulados de nível superior, tais como Network File System (NFS), Server Message Block (SMB), Hypertext Transfer Protocol (HTTP), BitTorrent, Post Office Protocol (POP) Versão 3, Network Basic Input/Output System (NetBIOS), telnet e Trivial File Transfer Protocol (TFTP).

Como o pacote inteiro é capturado por uma interface pcap, as características dos dados variam em relação ao protocolo que o pacote transporta. A Tabela II lista os subconjuntos de características capturadas para TCP, UDP e ICMP. Os endereços IP estão no cabeçalho IP, que são tratados na Camada de Rede.

B. Dados NetFlow

Originalmente, o NetFlow foi introduzido como um roteador pela Cisco. O roteador ou switch tem a capacidade de coletar o tráfego de trabalho da rede IP ao entrar e sair da interface. A versão 5 do NetFlow da Cisco define um fluxo de rede como uma seqüência unidirecional de pacotes que compartilham exatamente os mesmos sete atributos de pacotes: interface de entrada, endereço IP de origem, endereço IP de destino, protocolo IP, porta de origem, porta de destino e tipo de serviço IP. A arquitetura lógica NetFlow consiste em três componentes: um NetFlow Exporter, um NetFlow Collector, e um Console de Análise. Atualmente, existem 10 versões do NetFlow. As versões 1 a 8 são similares, mas a partir da versão 9, o NetFlow difere significativamente. Para as versões 1 a 8, a característica definida na Tabela III

QUADRO II
CABEÇALHOS DE PACOTES DE DADOS DE CIBER-SEGURANÇA

IP Header (IPv4)	
Internet Header Length	The number of 32-bit words in the header
Total Length	The entire packet size, including header and data, in bytes
Time To Live	This field limits a datagram's lifetime, in hops (or time)
Protocol	The protocol used in the data portion of the IP datagram
Source address	This field is the IPv4 address of the sender of the datagram
Destination address	This field is the IPv4 address of the receiver of the datagram
TCP Packet	
Source port	Identifies the sending port
Destination port	Identifies the receiving port
Sequence number	Initial or accumulated sequence number
Acknowledgement number	The next sequence number that the receiver is expecting
Data offset	Specifies the size of the TCP header in 32-bit words
Flags (control bits)	NS, CWR, ECE, URG, ACK, PSH, RST, SYN, FIN
UDP Packet	
Source port	Identifies the sending port
Destination port	Identifies the receiving port
Length	The length in bytes of the UDP header and UDP data
ICMP Packet	
Type	Control (e.g., ping, destination unreachable, trace route)
Code	Details with the type
Rest of Header	More details

QUADRO III
NETFLOW CABEÇALHO DE PACOTES DE DADOS DE SEGURANÇA CIBERNÉTICA

NetFlow Data – Simple Network Management Protocol (SNMP)	
Ingress interface (SNMP ifIndex)	Router information
Source IP address	
Destination IP address	
IP protocol	IP protocol number
Source port	UDP or TCP ports; 0 for other protocols
Destination port	UDP or TCP ports; 0 for other protocols
IP Type of Service	Priority level of the flow
NetFlow Data – Flow Statistics	
IP protocol	IP protocol number
Destination IP address	
Source IP address	
Destination port	
Source port	
Bytes per packet	The flow analyzer captures this statistic
Packets per flow	Number of packets in the flow
TCP flags	NS, CWR, ECE, URG, ACK, PSH, RST, SYN, FIN

apresenta o conjunto mínimo de variáveis de dados NetFlow para uma seqüência unidirecional de pacotes (ou seja, um fluxo).

Os dados NetFlow incluem uma versão comprimida e pré-processada dos pacotes de rede reais. As estatísticas são características derivadas e, com base em certos parâmetros como duração da janela, número de pacotes, etc., definem as configurações do NetFlow no dispositivo.

C. Conjuntos de dados públicos

Os conjuntos de dados da Defense Advanced Research Projects Agency (DARPA) 1998 e DARPA 1999 [18], [19] são amplamente utilizados em experimentos e freqüentemente citados em publicações. O conjunto DARPA 1998 foi criado pelo Cyber Systems and Technology Group do Laboratório Lincoln do Massachusetts Institute of Technology (MIT/LL). Foi construído um trabalho de simulação de rede e os dados foram compilados com base em dados de rede TCP/IP, dados de registro do Módulo de Segurança Básica do Solaris, e dados de registro do sistema de arquivos Solaris para usuário e raiz. Efetivamente, o conjunto de dados montado foi composto de dados de rede e sistema operacional (SO). Os dados foram coletados durante 9 semanas, sendo os primeiros 7 designados como conjunto de treinamento e os últimos 2 designados como conjunto de teste. Simulações de ataque foram organizadas durante as semanas de treinamento e testes.

Da mesma forma, o conjunto de dados da DARPA 1999 foi coletado por um total de 5 semanas, sendo as 3 primeiras designadas como o conjunto de treinamento e as 2 últimas designadas como o conjunto de testes. Este conjunto de dados tinha substancialmente mais tipos de ataque do que o conjunto de dados da DARPA 1998. Em ambas as le-ções col, os conjuntos de dados foram processados e curados para serem usados nos experimentos. As lixeiras TCP e os logs foram combinados em um único fluxo com muitas colunas.

Um dos conjuntos de dados mais utilizados é o conjunto de dados KDD 1999 [20], que foi criado para o desafio da Copa KDD em 1999. O conjunto de dados é baseado nos dados do DARPA 1998 TCP/IP e tem características básicas capturadas pelo pcap. Características adicionais foram derivadas da análise dos dados com o tempo e a seqüência de vitórias. O conjunto de dados tem três componentes - básico, conteúdo e características de tráfego - que produzem um total de 41 atributos. O conjunto de dados KDD 1999 é semelhante aos dados NetFlow, mas tem características mais derivadas e detalhadas porque os ataques foram simulados. A lista completa pode ser encontrada na Tabela IV.

O conjunto de dados do KDD 1999 (com cerca de 4 milhões de registros de tráfego nor- mal e de ataque) foi analisado de forma abrangente por Tavallaee et al. [21] e descobriu que tem alguns limites sérios - tiões. Alguns problemas inerentes foram observados, tais como sintetizar a rede e os dados de ataque (após a amostragem do tráfego real) devido a preocupações com a privacidade, um número desconhecido de pacotes descartados causados pelo excesso de tráfego, e definições vagas de ataque. Tavallaee et al. também realizaram avaliações estatísticas e seus próprios experimentos de classificação. Eles relataram um número enorme de registros redundantes (78% nos dados de treinamento e 75% nos dados de teste) causando viés. Além disso, nos experimentos de classificação realizados pelo grupo, eles indicaram que, ao selecionar aleatoriamente subconjuntos dos dados de treinamento e testes, muitas vezes é possível obter uma precisão muito alta e irrealista. Eles propuseram um novo conjunto de dados, o NSL-KDD, que consiste em registros selecionados do conjunto de dados KDD completo e não apresenta as deficiências acima mencionadas.

O conjunto DARPA 1998 define quatro tipos de ataques:

Negação de Serviço (DoS), Usuário para Raiz (U2R), Remoto para Local (R2L), e Sonda ou Escaneamento. Um ataque DoS é uma tentativa de negar aos usuários visados recursos computacionais ou de rede. Um ataque U2R garante o acesso root ao atacante. Um ataque R2L garante o acesso à rede local ao atacante. Ataques de Sonda ou Scan coletam informações sobre os recursos da rede. A DARPA 1999 acrescentou

QUADRO IV
CARACTERÍSTICAS DA CONEXÃO
TCP

Basic Features		
duration	integer	duration of the connection
protocol_type	nominal	protocol type of the connection: TCP, UDP, and ICMP
service	nominal	http, ftp, smtp, telnet... and other
flag	nominal	connection status
src_bytes	integer	bytes sent in one connection
dst_bytes	integer	bytes received in one connection
land	binary	if src/dst IP address and port numbers are same, then 1
wrong_fragment	integer	sum of bad checksum packets in a connection
urgent	integer	sum of urgent packets in a connection
Content Features		
hot	integer	sum of hot actions in a connection such as: entering a system directory, creating programs and executing programs
num_failed_logins	integer	number of incorrect logins in a connection
logged_in	binary	if the login is correct, then 1, else 0
num_compromised	integer	sum of times appearance “not found” error in a connection
root_shell	binary	if the root gets the shell, then 1, else 0
su_attempted	binary	if the su command has been used, then 1, else 0
num_root	integer	sum of operations performed as root in a connection
num_file_creations	integer	sum of file creations in a connection
num_shells	integer	number of logins of normal users
num_access_files	integer	sum of operations in control files in a connection
num_outbound_cmds	integer	sum of outbound commands in an ftp session
is_hot_login	binary	if the user is accessing as root or admin
is_guest_login	binary	if the user is accessing as guest, anonymous, or visitor
Traffic Features – Same Host – 2-second Window		
duration	integer	duration of the connection
protocol_type	nominal	protocol type of the connection: TCP, UDP, and ICMP
service	nominal	http, ftp, smtp, telnet... and other
flag	nominal	connection status
src_bytes	integer	bytes sent in one connection
dst_bytes	integer	bytes received in one connection
land	binary	if src/dst IP address and port numbers are same, then 1
wrong_fragment	integer	sum of bad checksum packets in a connection
urgent	integer	sum of urgent packets in a connection
Traffic Features – Same Service – 100 Connections		
duration	integer	duration of the connection
protocol_type	nominal	protocol type of the connection: TCP, UDP, and ICMP
service	nominal	http, ftp, smtp, telnet... and other
flag	nominal	connection status
src_bytes	integer	bytes sent in one connection
dst_bytes	integer	bytes received in one connection
land	binary	if src/dst IP address and port numbers are same, then 1
wrong_fragment	integer	sum of bad checksum packets in a connection
urgent	integer	sum of urgent packets in a connection
urgent	integer	sum of urgent packets in a connection

um novo tipo de ataque em que o atacante tenta exfiltrar arquivos especiais que têm de permanecer no computador da vítima.

IV. MÉTODOS ML E DM PARA CYBER

Esta seção descreve os diferentes métodos ML/DM para segurança cibernética. Cada técnica é descrita com alguns detalhes, e são fornecidas referências a trabalhos seminais. Também, para cada método, são apresentados dois a três trabalhos com suas aplicações ao domínio cibernético.

A. Redes Neuronais Artificiais

As ANNs são inspiradas pelo cérebro e compostas de neurônios artificiais interconectados capazes de certos cálculos em suas entradas [22]. Os dados de entrada ativam os neurônios na primeira camada da rede cuja saída é a entrada para a segunda camada de neurônios na rede. Da mesma forma, cada camada passa sua saída para a camada seguinte e a última camada produz o resultado. As camadas entre as camadas de entrada e saída são chamadas de camadas ocultas. Quando uma ANN é usada como classificadora, a camada de saída gera a categoria de classificação final.

Os classificadores ANN são baseados no perceptron [23] e foram muito populares até os anos 90, quando as SVMs foram inventadas. Em comparação com a otimização convexa quadrática aplicada em uma SVM, as ANNs freqüentemente sofrem de mínimos locais e, portanto, de longos tempos de duração durante o aprendizado. Ao contrário de uma SVM, à medida que aumenta o número de fea- tures em uma ANN, seu tempo de execução do aprendizado aumenta. Com uma ou mais camadas ocultas, a ANN é capaz de gerar modelos sem orelha. O recurso de retropropagação da ANN torna possível a modelagem da lógica EX-OR.

Com desenvolvimentos neste campo, tais como NNs recorrentes, feed- forward e convolucionais, as ANNs estão ganhando novamente em ularidade pop e, ao mesmo tempo, ganhando muitos prêmios em concursos recentes de reconhecimento de padrões (estes concursos ainda não estão relacionados à detecção de intrusão cibernética). Como as versões avançadas das ANNs requerem ainda mais poder de processamento, elas são implementadas comumente em unidades de processamento gráfico.

1) Detecção de uso indevido: Cannady [24] usou ANNs como classificador multicategorias para detectar o mau uso. Ele usou a geração de dados - com a ajuda de um monitor de rede RealSecure™ , que tem as assinaturas de ataque incorporadas no sistema. Dez mil eventos foram lecionados pelo monitor, dos quais 3000 vieram de ataques simulados. Os ataques foram simulados pelos programas Internet Scanner [25] e Satan [26].

A etapa de pré-processamento de dados resultou na seleção de nove características: identificador de protocolo (ID), porta de origem, porta de destino, endereço de origem, endereço de destino, tipo ICMP, código ICMP, comprimento de dados brutos, e dados brutos. Dez por cento dos dados foram selecionados aleatoriamente para testes. O estudo então utilizou os dados normais e de ataque remanescentes para treinar uma ANN, que aprendeu as assinaturas combinadas. O documento expressou que os resultados foram preliminares e relatou as taxas de erro para treinamento e testes, que foram de 0,058 e 0,070 erros de meios-raiz (RMS), respectivamente. Embora os detalhes não tenham sido divulgados, a saída da ANN foi um número entre 0 e 1 representando cada uma das duas categorias (ataque e normal). Portanto, um RMS de 0,070 pode ser considerado aproximadamente como 93% de precisão para a fase de teste. Cada pacote ou instância de dados foi categorizada como um grupo normal ou um grupo de ataque.

2) Detecção de Anomalias e Detecção Híbrida: Lippmann e Cunningham [27] propuseram um sistema que utiliza seleção de palavras-chave e redes neurais artificiais. A seleção de palavras-chave foi realizada em transcrições de sessões telnet e as estatísticas foram computadas do número de vezes que cada palavra-chave (de uma lista pré-determinada) ocorreu. A estatística de palavras-chave consti- tui a entrada para uma rede neural que fornece uma estimativa da probabilidade posterior de um ataque. Uma segunda rede neural opera nas instâncias que foram marcadas como um ataque, e tenta classificá-las (ou seja, fornecer um nome de ataque). Ambas as redes neurais consistiram de perceptrons multicamadas sem unidades ocultas. O sistema atinge 80% de detecção com aproximadamente 1 falso alarme por dia. Esta taxa de falso alarme representa uma melhoria de duas ordens de magnitude em relação ao sistema de base com a mesma precisão de detecção.

Bivens et al. [28] descrevem um IDS completo que emprega um estágio de pré-processamento, agrupando o tráfego normal, a normalização, um estágio de treinamento ANN e um estágio de decisão ANN. O primeiro estágio utilizou um Mapa de Auto-Organização (SOM), que é um tipo de ANN não supervisionado, para aprender o pat- terns de tráfego normal ao longo do tempo, como os números de porta TCP/IP comumente utilizados. Desta forma, o primeiro estágio quantificou os recursos de entrada em silos, que foram então alimentados para o segundo estágio, uma ANN Multilayer Perceptron (MLP). Os parâmetros da rede MLP, tais como o número de nós e camadas, foram determinados pela SOM do primeiro estágio. Uma vez concluído o treinamento do MLP, ele começou a prever as intrusões. O sistema pode ser reiniciado para que um novo SOM aprenda um novo padrão de tráfego e uma nova classificação de ataque MLP - a ser treinada. O estudo utilizou dados TCP/IP do desafio DARPA 1999 [18], [19], onde o conjunto de dados consistia de dados em nível de pacote de trabalho em rede. Ao contrário do estudo anterior da Cannady [24], que classificou cada dado em nível de pacote separadamente, este sistema usou janelas de tempo para realizar a detecção e classificou um grupo de pacotes. Assim, o sistema foi capaz de detectar tipos de ataques de maior duração. Como a entrada era de dados de pacotes de baixo nível de rede (em oposição aos dados NetFlow), a granu- laridade é alta e as previsões produzidas ainda correspondem a durações curtas. Bivens et al. [28] relataram com sucesso pré-ditar 100% do comportamento normal. Sua abordagem geral é promissora, mesmo que alguns ataques não tenham sido totalmente previstos e as FAR para alguns ataques tenham atingido 76%.

B. Regras da Associação e Regras da Associação Fuzzy

O objetivo da Association Rule Mining é descobrir, a partir dos dados, as regras da associação previamente desconhecidas. Uma regra de associação descreve uma relação entre diferentes atributos: SE (A E B) ENTÃO C. Esta regra descreve a relação que quando A e B estão presentes, C também está presente. As regras de associação têm métricas que dizem com que freqüência uma determinada relação ocorre nos dados. O suporte é a probabilidade anterior (de A, B e C), e a confiança é a probabilidade condicional de C dado A e B. A regra de associação Mineração foi introduzida por Agrawal et al. [29] como uma forma de descobrir co-ocorrências interessantes em dados de super-mercado. Ela encontra conjuntos freqüentes de itens (isto é, combinações de itens que são comprados juntos em pelo menos N transações no banco de dados), e dos conjuntos de itens freqüentes como {X, Y}, gera regras de associação do formulário: $X \rightarrow Y$ e/ou $Y \rightarrow X$.

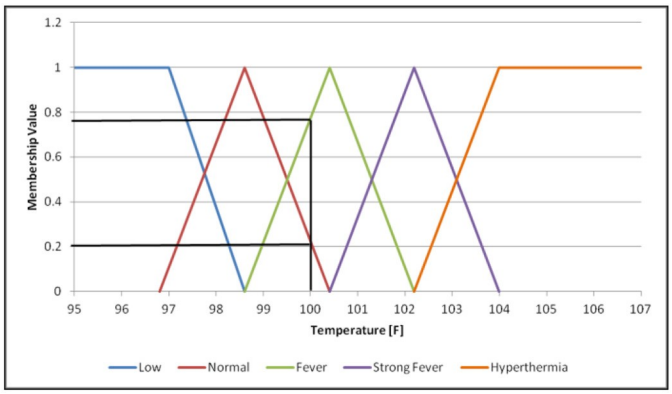


Fig. 2. Funções dos membros para a Variável Fuzzy de Temperatura do Corpo Humano: Baixa, Normal, Febre, Febre Forte e Hipotermia.

Um exemplo simples de uma regra de associação relacionada aos itens que as pessoas compram juntos é:

$$\text{SE } (\text{Pão E Manteiga}) \rightarrow \text{Leite} \quad (1)$$

Esta regra estabelece que se uma pessoa compra pão e manteiga, ela também compra leite.

Uma limitação da tradicional Regra de Associação de Mineração é que ela só funciona com dados binários [ou seja, um item foi adquirido em uma transação (1) ou não (0)]. Em muitas aplicações do mundo real, os dados ou são categóricos (por exemplo, nome IP, tipo de intervenção de saúde pública) ou quantitativos (por exemplo, duração, número de logins falhados, temperatura). Para atributos numéricos e categóricos, as regras booleanas são insatisfatórias. Uma extensão que pode processar variáveis numéricas e categóricas é chamada de Fuzzy Association Rule Mining [30].

As regras da associação Fuzzy são da forma:

$$\text{SE } (\text{X é } \boldsymbol{A}) \rightarrow (\text{Y é } \boldsymbol{B}) \quad (2)$$

onde X e Y são variáveis, e ***A*** e ***B*** são conjuntos difusos que caracterizam X e Y, respectivamente. Um exemplo simples de regra de associação fuzzy para uma aplicação médica poderia ser o seguinte:

SE (Temperatura é ***Febre Forte***) E (Pele é ***Amarelada***) E (Perda de Apetite é ***Profunda***) → (Hepatite é ***Aguda)***

A regra diz que se uma pessoa tem ***febre forte***, pele ***amarelada*** e ***perda profunda*** do apetite, então a pessoa tem Hepatite ***Aguda. Febre Forte***, Pele ***Amarelada***, ***Profunda*** e ***Aguda*** são funções de membro das variáveis Temperatura, Pele, Perda de Apetite e Hepatite, respectivamente. Como exemplo de funções de membresia felpuda, as funções de membresia da variável Temperatura são mostradas na Fig. 2.

De acordo com a definição no diagrama, uma pessoa com uma temperatura 100° F tem uma temperatura ***Normal*** com um valor de membro - navio de 0,2 e tem uma ***Febre*** com um valor de membro de
0.78. O uso de funções de associação fuzzy permite o uso de termos lingüísticos. Esses termos lingüísticos para temperatura corporal humana são ***Baixa***, ***Normal***, ***Febre***, ***Febre Forte*** e ***Hipotermia***. Mais informações sobre a lógica fuzzy e funções de associação fuzzy podem ser encontradas em [31].

1) Detecção de uso indevido: O estudo de Brahmi [32] é um bom exemplo das regras de associação aplicadas ao conjunto de dados DARPA 1998 para capturar as relações entre os parâmetros TCP/IP e os tipos de ataque. Nas regras, os antecedentes são extraídos do conjunto de dados do DARPA 1998 e os consequentes são os tipos de ataque. O trabalho explica a Mineração de Regras de Associação multidimensional, na qual existe mais de um antecedente nas regras, tal como (IF (serviço AND src_port AND dst_port AND num_conn) THEN attack_type), que é um exemplo de uma regra tetradimensional. O trabalho inclui encontrar as regras com elevado apoio e alta confiança. O melhor desempenho é alcançado utilizando regras de seis dimensões com taxas de detecção de 99%, 95%, 75%, 87% para os tipos de ataque DoS, Probe ou Scan, U2R, e R2L, respec- tively. As experiências não foram realizadas com dimensões mais elevadas devido ao custo computacional. Uma das principais vantagens de Association Rule Mining é que as regras descobertas explicam claramente as relações. A abordagem é promissora para a construção de assinaturas de ataque.

Zhengbing et al. [33] propuseram um novo algoritmo baseado no algoritmo Signature a priori [34] para encontrar novas assinaturas de ataques a partir de assinaturas de ataques existentes. Compararam o tempo de processamento do seu algoritmo com o de Assinatura a priori e descobriram que o seu algoritmo tem um tempo de processamento mais curto, e a diferença no tempo de processamento aumenta com o aumento do tamanho da base de dados. A contribuição desse documento é a sua descrição de um novo método para encontrar novas assinaturas de ataques a partir de assinaturas existentes. Tal algoritmo poderia ser utilizado para a obtenção de novas assinaturas para inclusão em sistemas de detecção de uso indevido, como o Snort [16].

2) Detecção de Anomalias e Detecção Híbrida: A estrutura NETMINE [35] realiza o processamento de fluxo de dados, análise de refinamento (através da captura de regras de associação a partir de dados de tráfego), e classificação de regras. A captura de dados é realizada em simultâneo com a análise do fluxo de dados em linha. Os pacotes de tráfego são capturados por ferramentas de captura de rede desenvolvidas no Politecnico di Torino [36], funcionando numa ligação de espinha dorsal da rede do campus. Os dados capturados são dados NetFlow com atributos tais como endereço de origem, endereço de destino, porto de destino, porto de origem, e tamanho do fluxo (em bytes).

NETMINE realiza extracção generalizada de regras de associação para detectar anomalias e identificar padrões recorrentes. As regras de associação individuais (por exemplo, para um endereço IP específico) podem ser demasiado detalhadas e ter um apoio muito baixo. As regras de associação generalizadas (por exemplo, para um tráfego de subrede) permitem elevar o nível de abstracção em que as correlações estão representadas. A extracção de regras generalizadas é realizada pelo novo algoritmo Genio [37], que é mais eficaz do que as abordagens anteriores para regras generalizadas de mineração. O processo de extracção de regras generalizadas de associação não está concebido para ser realizado em tempo real durante a captura de dados. No entanto, algumas experiências mostram a viabilidade da abordagem para frequências adequadas de actualização de janelas deslizantes.

A classificação das regras organiza as regras de acordo com a sua interpretação semântica. São definidas três classes básicas de regras:

- As regras de fluxo de tráfego envolvem os endereços de origem e de destino.
- As regras dos serviços prestados consistem em porto de destino (ou seja, o serviço) e endereço de destino (ou seja, o prestador de serviços).

- As regras de utilização do serviço têm o porto de destino e o endereço de origem (ou seja, o utilizador do serviço).

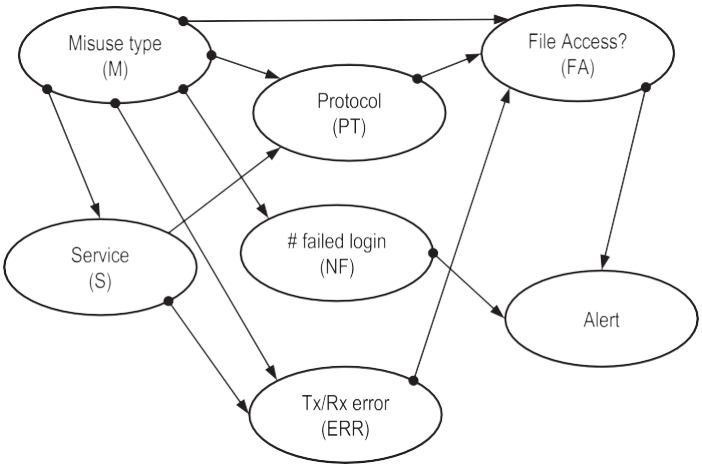
As regras extraídas destinam-se a ajudar o analista da rede a determinar rapidamente os padrões que valem a pena investir mais - tigação. Não existe uma classificação automática em categorias normais e anómalas.

O trabalho de Tajbakhsh et al. [38] utilizou o conjunto de dados KDD 1999 para realizar a Fuzzy Association Rule Mining para descobrir padrões de relacionamento com- mon. O estudo utilizou a versão corrigida do conjunto KDD (ver Secção III) com aproximadamente 300.000 casos. Utilizaram uma abordagem de agrupamento para definir as funções de associação fuzzy dos atributos, afirmando que o seu desempenho é melhor do que as abordagens baseadas em histogramas. Para reduzir os *itens* (de acordo com o papel, existem 189 itens diferentes: 31 atributos numéricos- icais com 3 valores fazem 93 itens e 10 atributos nominais com um total de 96), o estudo utilizou um método de associação hiper-edge. Por exemplo, o conjunto de itens {a, b} é con- margem de hiper-fiança se a confiança média das regras (a → b e b → a) for superior a um limiar. O trabalho utiliza os limiares de 98% para a redução de hiper-edge de associação e 50% para a confiança. O desempenho da detecção da anomalia é relatado como 100% exacto com uma taxa de FP de 13%. O perfor- mance cai rapidamente à medida que a taxa de FP é reduzida. O documento também declara os benefícios da abordagem de Mineração da Regra de Associação, tais como regras humano-compreensíveis, tratamento mais fácil de atributos sym- bolic (nominais), e classificação eficiente em grandes conjuntos de dados.

Luo e Bridges [39] tentaram integrar a lógica fuzzy com episódios de frequência para encontrar os episódios de frequência fuzzy que representam as sequências fuzzy frequentes nos dados. A fuzzificação (isto é, quantificação de dados com caixas de sobre-volagem) ajuda a lidar com as variáveis numéricas e a explicar as variáveis ao ciberanalista de segurança. Os episódios de fre- quência são determinados por um limiar fornecido pelo utilizador. Efectivamente, a abordagem global é semelhante à mineração de sequências. A experiência utilizou dados recolhidos pelo tcpdump de um servidor de um campus universitário. As principais características dos dados continham bandeiras TCP e números de portas, e são quantificados por lógica fuzzy. As simulações de intrusão foram conduzidas por gramas personalizadas. O estudo relata as medições de similaridade entre os conjuntos de dados de formação e teste. A maior laridade-similar relatada é de 0,82. Embora o trabalho não tenha relatado medidas de desempenho, a abordagem global é promissora, alargando e melhorando assim as abordagens anteriores na literatura.

C. Rede Bayesiana

Uma rede Bayesiana é um modelo gráfico probabilístico que representa as variáveis e as relações entre elas [40], [41], [41]. A rede é construída com nós como as variáveis aleatórias discretas ou contínuas e bordas direcionadas como as relações entre elas, estabelecendo um gráfico acíclico dirigido. Os nós de criança são dependentes dos seus pais. Cada nó mantém os estados da variável aleatória e a forma de probabilidade condicional. As redes Bayesianas são construídas utilizando conhecimentos especializados ou utilizando algoritmos eficientes que realizam inferências. A figura 3 dá um exemplo de uma rede Bayesiana para a detecção de assinaturas de ataque. Cada estado (ou variável de rede) pode ser uma



File Access state input variables and values	P(FA = True)	P(FA = False)
M=R2H, PT=NSF, ERR=0	0.95	0.05
M=R2H, PT=FTP, ERR=0	0.99	0.01
M=Probe, PT=none, ERR=50%	0.80	0.20
M=Probe, PT=PING, ERR=0	0.50	0.50
M=DoS, PT=POP, ERR=100%	0.80	0.20
M= DoS, PT=HTTP, ERR=50%	0.90	0.10

Fig. 3. Exemplo Rede Bayesiana de Detecção de Assinaturas.

entrada para outros estados com determinado conjunto de valores de estado. Para exame- ple, o estado do **protocolo** pode escolher valores a partir dos números de protocolo disponíveis. Cada um dos valores de estado que podem ir de um estado para outro tem uma probabilidade associada, e a soma dessas probabilidades somar-se-á a 1, representando todo o conjunto de valores de estado. Dependendo da aplicação, a rede pode ser utilizada para explicar a interacção entre as variáveis ou para chamar um resultado provável para um estado alvo (por exemplo, **alerta** ou **acesso a ficheiros**) utilizando os estados de entrada.

As tabelas de probabilidade podem ser calculadas a partir dos dados de formação disponíveis. A infirmação de variáveis não observadas, a aprendizagem de parâmetros e a aprendizagem de estruturas estão entre as principais tarefas de formação das redes Bayesianas.

1) Detecção de uso indevido: Em geral, a detecção de anomalias pode ser considerada como reactiva porque o sistema responde a um input quando o input é inesperado. Inversamente, num problema de detecção de uso indevido, o sistema é proactivo porque as assinaturas extraídas do input estão a ser verificadas continuamente em relação a uma lista de padrões de ataque. Tomar a abordagem proactiva como na detecção de uso indevido requer a classificação dos fluxos da rede. No seu trabalho, Livadas et al. [42] compararam vários métodos aplicados ao problema de DoS. O seu trabalho tentou resolver o tráfego de botnet no tráfego filtrado de Internet Relay Chat (IRC), determinando assim a existência de botnet e as suas origens. O estudo utiliza dados de nível TCP recolhidos em 18 locais na rede de campus sem fios da Universidade de Dartmouth ao longo de um período de 4 meses. Os dados TCP (botnets baseados no IRC não utilizam dados UDP ou ICMP) são utilizados para gerar os fluxos de rede ou dados NetFlow. Uma camada de filtro é utilizada para extrair dados IRC de todos os dados da rede.

Uma vez que a verdade do terreno é difícil de obter, o estudo utilizou aspectos cer- tos da comunicação do IRC para marcar os dados com etiquetas de classe IRC. Para etiquetar os dados gerados pela rede IRC é ainda mais difícil, pelo que o estudo utiliza dados simulados para as experiências. A per- formância da rede Bayesiana é relatada como 93% de precisão com uma taxa de FP de 1,39%. Os outros dois classificadores, Naïve Bayes e árvore de decisão C4.5, ambos atingem uma precisão de 97%. No entanto, as suas taxas de PF são superiores a 1,47% e 8,05%, respectivamente.

Num outro estudo, Jemili et al. [43] sugeriram um trabalho de enquadramento do IDS - utilizando classificadores de rede Bayesianos. O trabalho utilizou nove características dos dados do KDD 1999 na rede de inferência. Em

a fase de detecção da anomalia, a decisão normal ou de ataque é tomada por um módulo de inferência de árvore de junção com um desempenho de 88% e 89% nas categorias normal e de ataque, respe- tivamente. Na fase seguinte, os tipos de ataque foram reconhecidos a partir dos dados rotulados como dados de ataque pelo módulo de detecção de anomalias. Desempenhos de 89%, 99%, 21%, 7%, e 66% são relatados para o DoS, Sonda ou Scan, R2L, U2R, e outras classes, respec- tivamente. O estudo sugere que o baixo desempenho das categorias R2L e U2R se deve ao facto de o número de instâncias de formação ser muito mais baixo do que para as outras categorias.

2) Detecção de Anomalias e Detecção Híbrida: Quando a plataforma informática recebe pacotes TCP/IP, a pilha de rede do SO subjacente processa estes pacotes. A pilha de rede gera vários registos e chamadas do kernel do sistema, e ultimamente os dados dos pacotes são processados ao nível da aplicação invocada pelo kernel. Kruegel et al. [44] utilizaram uma rede Bayesiana - trabalho para classificar eventos durante chamadas de SO abertas e executivas. O conjunto de dados DARPA 1999 é utilizado para excitar o kernel do SO através de pacotes TCP/IP. Depois, um conjunto de atributos baseados nestas chamadas de sistema, tais como o comprimento da cadeia de argumentos da chamada de sistema e a distribuição e distribuição de caracteres, são comparados utilizando um teste Pearson. Além disso, a estrutura da string em relação à gramática (sintaxe de comando) e fichas é pesquisada nos parâmetros das chamadas de sistema. Estas características são utilizadas numa rede Bayesiana para calcular a probabilidade de um estado normal ou de um estado de anomalia. Como o limiar de detecção é utilizado para controlar o FAR, o sistema é flexível e pode fazer auto-ajustes contra demasiados alarmes falsos; 75% de precisão, 0,2% FAR e 100% de precisão, e 0,1% FAR são alcançados através da utilização de diferentes valores de limiar.

A determinação de ataques complexos requer frequentemente a observação das anomalias e das correlações entre elas para descobrir nários de sce ou planos de ataque. Para atingir este objectivo, foi estudada a correlação de alertas para reduzir o volume de alertas gerados pelo sistema. A agregação, as medidas de semelhança, ou o saber-fazer especializado são utilizados para correlacionar as anomalias e depois revelar os complexos padrões de cenários de ataque.

Um detector de intrusão DoS que utiliza uma rede Bayesiana é descrito por Benferhat et al. [45]. Existe apenas um nó pai que representa a variável oculta (ou seja, classe - {normal,

anomalia}), e as variáveis observadas são nós de criança. Assume-se que os nós crianças são estatisticamente independentes. O principal objectivo desta abordagem é realizar uma correlação de alerta utilizando dados históricos com uma utilização mínima de conhecimentos especializados. O conjunto de dados do DARPA 2000 é utilizado nas experiências. Para cada objectivo intru- sion, é construída uma rede. A configuração do estudo tem dois cenários diferentes extraídos do conjunto de dados DARPA; o sistema detectou um dos cenários com sucesso, mas não conseguiu detectar o outro. Infelizmente, o estudo não comunica quaisquer resultados numéricos.

D. Clustering

Clustering [46] é um conjunto de técnicas para encontrar padrões em dados de alta dimensão não etiquetados. É uma abordagem de descoberta não supervisionada, em que os dados são agrupados com base numa medida de similaridade. A principal vantagem do agrupamento para a detecção de intrusão é que pode aprender com os dados de auditoria sem exigir ao administrador do sistema que forneça descrições explícitas de várias classes de ataque.

Existem várias abordagens para agrupar os dados introduzidos. Nos modelos de conectividade (por exemplo, agrupamento hierárquico), os pontos de dados são agrupados pelas distâncias entre eles. Em modelos centróides (p. ex., k-means), cada cluster é representado pelo seu vector médio. Em modelos de distribuição (por exemplo, Expectation Maximization algo- rithm), os grupos são assumidos como aquiescentes a uma distribuição estatística. Os modelos de densidade agrupam os pontos de dados como regiões densas e ligadas (p. ex., Density-Based Spatial Clustering of Applications with Noise [DBSCAN]). Finalmente, modelos gráficos (por exemplo, clique) definem cada cluster como um conjunto de nós ligados (pontos de dados) onde cada nó tem uma borda para pelo menos um outro nó no conjunto.

Uma aprendizagem baseada em exemplos (também chamada aprendizagem preguiçosa) algo- rithm, o k-NN, é outro método ML popular em que a classificação de um ponto é determinada pelo k neigh- bors mais próximo desse ponto de dados. Um exemplo desta abordagem para o ciberespaço é descrito em [47]. A determinação da categoria do ponto exerce- cise uma votação por maioria, que pode ser um inconveniente se a distribuição de classes for enviesada. Como os dados de alta dimensão afectam negativamente os métodos k-NN (ou seja, a maldição da dimensionalidade), é quase sempre necessária uma redução da característica. A escolha da ordem de vizinhança ou da ordem no conjunto de dados é também considerada como tendo um impacto elevado. Entre os benefícios da k-NNN está a sua simplicidade e ausência de pressupostos paramétricos (excepto o número k).

Na teoria gráfica, o coeficiente de agregação representa a afinidade dos nós que estão próximos uns dos outros [48]. Certos tipos de dados, tais como redes sociais, têm coeficientes de agregação elevados. O coeficiente global de agregação é definido como o rácio de num- ber de trigémeos fechados para o número de trigémeos ligados de vértices. O coeficiente de agregação local de um nó é definido como o rácio entre o número de sub-gráficos com três arestas e três vértices de que o nó faz parte e o número de triplos de que o nó faz parte [49]. O coeficiente está entre 0 e 1, sendo 0 um único nó e 1 cada vizinho ligado ao nó.

1) Detecção de uso indevido: Na literatura, há menos aplicações dos métodos de agrupamento para a detecção de utilizações indevidas do que

para a detecção de anomalias. Contudo, como demonstrado por um estudo [50], a geração de assinaturas em tempo real por um detector de anomalias pode ser uma vantagem importante. Um esquema de agrupamento baseado na densidade chamado Simple Logfile Clustering Tool (SLCT) é utilizado para criar clusters - ters de tráfego de rede normal e malicioso. Para diferenciar o tráfego normal do anómalo, o estudo utilizou um parâmetro M para definir a percentagem de características fixas que o aglomerado contém. Uma característica fixa para o aglomerado corresponde a um valor constante para essa característica. O estudo define como uma região densa os valores da característica que estão pelo menos em N por cento das instâncias. Este é essencialmente o valor de suporte de um único antecedente, uma das métricas que a Association Rule Mining utiliza. Desta forma, quando M é zero, todos os dados são agrupados e quando M tem um valor elevado, idealmente só restam agrupamentos maliciosos. Como exemplo, ao fixar o valor de M a 97%, o estudo detecta 98% dos dados de ataque com um FAR de 15%. O método pode detectar ataques anteriormente não vistos (novos ou de dia zero). Após a fase de agrupamento com os parametros especificados, todos os agrupamentos são considerados como ataques que tratam os centróides de agrupamento como as assinaturas.

O sistema é composto por dois esquemas de agrupamento: o primeiro é utilizado para a detecção normal ou de ataque (como descrito anteriormente), e o segundo é utilizado de uma forma supervisionada para determinar o tráfego normal. A diferença entre eles é a modificação da definição dos parâmetros, tendo praticamente dois esquemas de agrupamento para detectar o tráfego normal e anómalo em paralelo. A saída desta fase vai para um módulo baseado em regras para extrair assinaturas a serem utilizadas quer pelo sistema quer pelos profissionais de segurança cibernética. Uma das novidades deste estudo é que cada centróide de agrupamento anómalo é tratado como uma assinatura a ser filtrada pelo sistema (por exemplo, após actualizações horárias ou diárias da assinatura pelo sistema).

O estudo utilizou o conjunto de dados KDD em várias experiências. Os conjuntos de dados foram preparados com percentagens de ataque de 0%, 1%, 5%, 10%, 25%, 50%, e 80%. Foram utilizadas métricas de desempenho, tais como integridade do cluster, para além da precisão. O estudo relatou o seu desempenho como uma taxa de detecção de 70% a 80% para ataques anteriormente desconhecidos. Os resultados são impressionantes, especialmente dado o facto de o sistema não ter conhecimento prévio de nenhum dos ataques ou padrões de ataque no conjunto de dados KDD.

2) Detecção de Anomalias e Detecção Híbrida: No seu estudo, Blowers e Williams [51] utilizam um método de agrupamento DBSCAN para agrupar pacotes de rede normais versus anómalos. O conjunto de dados KDD é pré-processado para seleccionar características utilizando uma análise de correlação. Embora o FAR real não seja relatado, o estudo utiliza o limiar do método de agrupamento para o controlling do FAR do sistema. Durante o pré-processamento dos dados é definida uma taxa de ataque de 10% para no-attack. O desempenho relatado é de 98% para detecção de ataque ou no-attack. Este é um valor muito elevado para um detector de anomalias baseado no agrupamento, mais elevado do que a maioria dos estudos na literatura. Globalmente, o estudo apresenta um bom exemplo de resumo da aplicação de métodos ML a operações cibernéticas.

Sequeira e Zaki [52] capturaram 500 sessões com um longo fluxo de comandos de nove utilizadores na Universidade Purdue. Foram utilizados dados de nível de comando do utilizador (comandos shell) para detectar se o utilizador é um utilizador regular ou um intruso. O fluxo de comandos de cada utilizador numa sessão foi analisado em fichas e

posteriormente representada como uma sequência de fichas. Sequeira e Zaki tentaram várias abordagens envolvendo sequências, tais como medidas de semelhança de sequências e sequência correspondentes a algo- rithms. Uma das abordagens promissoras utilizadas foi baseada na métrica subsequence mais longa comum.

As capturas de comando de nove utilizadores sugerem que os dados são pequenos, possivelmente devido aos requisitos de processamento. Não é claro quantos comandos uma sessão típica inclui, mas há uma boa probabilidade de ser muito maior do que o comprimento máximo da sequência, que o estudo escolheu como 20. O estudo indicou o seu perfor- mance de 80% de precisão com 15% FAR como um resultado bem sucedido. Sequeira e Zaki [52] referem-se a estudos anteriores que utilizaram o mesmo conjunto de dados e alcançaram 74% com 28% FAR e apontam para uma melhoria. Também poderia ser uma boa ideia utilizar os registos do servidor do conjunto de dados DARPA ou KDD para comparação.

E. Árvores de decisão

Uma árvore de decisão é uma estrutura em forma de árvore que tem folhas, que representam classificações e ramos, que por sua vez representam as conjunções de características que levam a essas classificações. Um exemplar é rotulado (classificado) testando os seus valores de característica (atributo) contra os nós da árvore de decisão. Os métodos mais conhecidos para a construção automática de árvores de decisão são o ID3
[53] e algoritmos C4.5 [54]. Ambos os algoritmos constroem árvores de decisão a partir de um conjunto de dados de formação, utilizando o conceito de entropia da informação. Ao construir a árvore de decisão, em cada nó da árvore, C4.5 escolhe o atributo dos dados que mais eficazmente divide o seu conjunto de exemplos em subconjuntos. O critério de divisão é o ganho de informação normalizada (diferença na entropia). O atributo com o maior ganho de informação normalizada é escolhido para tomar a decisão. O algoritmo C4.5 efectua então a repetição dos subconjuntos mais pequenos até que todos os exemplos de formação tenham sido classificados.

As vantagens das árvores de decisão são a expressão intuitiva do conhecimento, a elevada precisão de classificação, e a simples mentação de implementos. A principal desvantagem é que para dados incluindo variáveis categóricas com um número diferente de níveis, os valores de ganho de informação são tendenciosos em favor de características com mais níveis. A árvore de decisão é construída maximizando o ganho de informação em cada divisão de variáveis, resultando numa classificação natural de variáveis ou selecção de características. As árvores pequenas (como a descrita na Fig. 4) têm uma expressão de conhecimento intuitiva para os especialistas num determinado domínio, porque é fácil extrair regras dessas árvores apenas ao examiná-las. Para árvores mais profundas e largas, é muito mais difícil extrair as regras e, portanto, quanto maior for a árvore, menos intuitiva será a sua expressão de conhecimento. Árvores mais pequenas são obtidas de árvores maiores através da poda. Árvores maiores têm frequentemente alta precisão de classificação mas não têm muito boas capacidades de generalização. Através da poda de árvores maiores, obtêm-se árvores menores que muitas vezes têm melhores capacidades de generalização (evitam o ajuste excessivo). Os algoritmos de construção de árvores de decisão (e.g., C4.5) são relativamente mais simples do que os algoritmos mais complexos como os SVM. Como tal, têm também uma implementação mais simples.

1) Detecção de uso indevido: A maioria dos sistemas de detecção de uso indevido por forma de detecção comparando cada entrada com todas as regras (signa- tures). Snort [16], uma ferramenta de código aberto bem conhecida, segue a

abordagem baseada na assinatura. Em Snort, cada assinatura tem uma descrição da linha sin- gle. O processo de correspondência entre os dados introduzidos e as assinaturas é normalmente lento (especialmente quando o número de assinaturas é grande) e, portanto, ineficiente de utilizar no front end.

Kruegel e Toth [55] substituíram o motor de detecção de uso indevido de Snort por árvores de decisão. Primeiro realizaram um agrupamento de regras utilizadas pelo Snort 2.0 e depois derivaram uma árvore de decisão utilizando uma variante do algoritmo ID3. O agrupamento de regras minimiza o número de comparações necessárias para determinar que regras são desencadeadas por determinados dados de entrada. A árvore de decisão escolhe as características mais discriminatórias do conjunto de regras, permitindo assim uma avaliação em paralelo de cada característica. Isto permite um desempenho muito superior ao do Snort.

A técnica proposta foi aplicada aos ficheiros tcpdump dos 10 dias de dados de teste produzidos pelo MIT/LL para a avaliação da detecção de intrusão DARPA de 1999. Para esse conjunto de dados, a velocidade de funcionamento do Snort e a técnica de árvore de decisão foram comparadas. O ganho real de desempenho varia conside- eravelmente dependendo do tipo de tráfego; a velocidade máxima - acima foi de 105%, a média de 40,3%, e a mínima de 5%. As experiências também foram realizadas aumentando o número de regras de 150 para 1581 (conjunto completo utilizado por Snort 2.0). Com o número crescente de regras, a aceleração do método da árvore de decisão sobre o Snort 2.0 é ainda mais pronunciada.

Este estudo mostrou que os métodos de agrupamento associados a árvores de decisão podem reduzir substancialmente o tempo de processamento de um sistema de detecção de utilização incorrecta, permitindo possivelmente a sua utilização eficiente na parte da frente.

2) Detecção de Anomalias e Detecção Híbrida: EXPOSIÇÃO [56], [57] é um sistema que emprega técnicas de análise em larga escala e passiva do Domain Name Service (DNS) para detectar domínios que estão envolvidos em actividade maliciosa. O sistema é constituído por cinco componentes principais: Colector de Dados, Componente de Atribuição de Característica, Colector de Domínios Maliciosos e Benignos, Módulo de Aprendizagem, e Classificador. O Classificador é construído pelo programa de árvore de decisão Weka J48, que é um implemento do algoritmo C4.5 capaz de gerar árvores de decisão podadas ou não podadas. A experiência mostrou que o erro mínimo foi alcançado quando todas as características foram combinadas, e portanto todas as 15 características foram utilizadas pela árvore de decisão.

Os dados utilizados consistem em dados DNS recolhidos durante um período de 2,5 meses (100 mil milhões de consultas DNS que resultaram em 4,8 milhões de nomes de domínio distintos). O estudo examinou vários milhares de domínios maliciosos e benignos e utilizou-os para estruturar o conjunto de formação. Os domínios maliciosos foram obtidos a partir de www.malwaredomains.com, da Lista de Blocos Zeus, Anubis, etc. A lista inicial de domínios maliciosos é composta por 3500 domínios. Os domínios benignos eram da lista de domínios do Alexa top 1000.

Ao experimentar diferentes valores de duração de períodos, o estudo determinou que o período ideal de treino inicial para o sistema era de 7 dias. Após este treino inicial, o classificador foi requalificado todos os dias. Os resultados variam consideravelmente - em função do conjunto de dados. Globalmente, utilizando uma validação cruzada de dez vezes, a precisão de detecção de domínios maliciosos foi de 98,5% e o FAR foi de 0,9%.

Foram realizadas experiências adicionais para determinar se o método conseguia detectar domínios maliciosos que eram

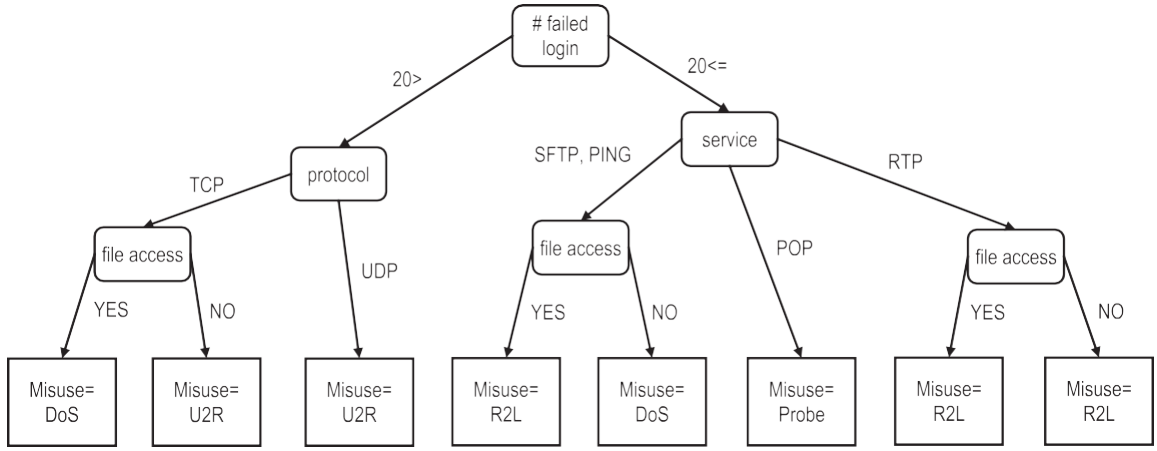


Fig. 4. Um exemplo de árvore de decisão.

não presentes no conjunto de formação. Na primeira experiência, entre os 50 domínios escolhidos aleatoriamente da lista de 17.686 domínios, o classificador detectou três domínios benignos como sendo maliciosos (6% de FP). Na segunda experiência, o estudo cruzou automaticamente os domínios maliciosos e suspeitos identificados pelo classificador utilizando ferramentas de classificação de sites on-line, tais como o McAfee Site Advisor, Google Safe Browsing, e Norton Safe Web. A taxa de PF foi de 7,9% para os domínios maliciosos identificados. Na terceira experiência, EXPOSURE classificou 100 milhões de consultas DNS durante um período de 2 semanas, detectou 3117 novos domínios maliciosos (anteriormente desconhecidos pelo sistema e não utilizados em formação), e não gerou quaisquer PF durante esse período.

Os resultados experimentais mostram que a precisão e o FAR da EXPOSIÇÃO são satisfatórios e que a EXPOSIÇÃO é utilizada - plenamente na identificação automática de uma categoria variada de domínios maliciosos (por exemplo, servidores de comando e controlo de botnet, sites de phishing, hospedeiros de esquemas). A capacidade da EXPOSURE de detectar um elevado número de domínios maliciosos anteriormente desconhecidos a partir do tráfego DNS é uma conquista importante.

F. Aprendizagem em conjunto

Em geral, os algoritmos de aprendizagem supervisionados pesquisam o espaço de hipotética para determinar a hipótese certa que fará boas previsões para um determinado problema. Embora possam existir boas hipóteses, pode ser difícil encontrar uma. Ensemble methods com- bine múltiplas hipóteses, na esperança de formar uma melhor do que a melhor hipótese isoladamente. Frequentemente, os métodos de ensemble utilizam múltiplos aprendentes fracos para construir um aprendente forte [58].

Um aprendiz fraco é aquele que gera consistentemente melhores pré-dicções do que ditados aleatórios. Um dos algoritmos do conjunto utiliza o boosting para treinar vários algoritmos de aprendizagem fracos e combinar (i.e., soma) os seus resultados ponderados. Adaptive Boosting (AdaBoost) [59] é um dos algoritmos mais populares utilizados para reduzir o problema de sobreposição inerente ao ML. O reforço pode ser visto como uma regressão linear em que as características dos dados são a entrada para o aprendiz fraco *h* (por exemplo, uma linha recta que divide os pontos de dados de entrada em duas categorias no espaço), e a saída do reforço é a soma ponderada destas funções *h.* Long e Servedio [60] criticaram o reforço sugerindo que uma fracção não nula de dados mal etiquetados pode causar a falha do reforço

completamente e resultar num ROC de 0,5 (o que é equivalente a um palpite aleatório).

O ensacamento (bootstrap aggregating) é um método para melhorar a generalidade do modelo de previsão para reduzir o excesso de ajuste. Baseia-se numa técnica de modelo-agregador de modelos e é conhecido por melhorar o desempenho de agregação de um vizinho mais próximo.

O classificador Random Forest [61] é um método ML que combina as árvores de decisão e a aprendizagem em conjunto. O for- est é composto por muitas árvores que utilizam características de dados colhidos aleatoriamente (atributos) como sua entrada. O pro- cess de geração florestal constrói uma colecção de árvores com variância controlada. A previsão resultante pode ser decidida por votação por maioria ou votação ponderada.

As Florestas Aleatórias têm várias vantagens: um baixo número de parâmetros de controlo e de modelo; resistência ao excesso de ajuste; sem exigência de selecção de características porque podem utilizar um grande número de atributos potenciais. Uma vantagem importante da Random Forest é que a variação do modelo diminui à medida que o número de árvores na floresta aumenta, enquanto que o viés permanece o mesmo. A Random Forests também tem algumas desvantagens, tais como baixa interpretabilidade do modelo, perda de desempenho devido a variáveis correlacionadas, e dependência do gerador aleatório da implementação.

1) Detecção de uso indevido: Zhang et al. [62] aproximam-se da sonda de detecção de uso indevido, empregando um módulo de detecção de uso indevido na parte da frente do seu sistema. Se a entrada for classificada como tráfego de rede anormal, os dados são ainda classificados como pertencentes a uma das categorias de ataque do conjunto de dados KDD 1999. O estudo fornece uma solução completa do sistema incluindo um detector outlier, um preditor de ataques baseado em assinatura, e uma base de dados pat- tern. Uma base de dados de anomalias é também utilizada para armazenar os padrões que são rotulados como anomalia por um utilizador (man- ually) ou pelo sistema (automaticamente) utilizando dados pré-rotulados. Os parâmetros da Floresta Aleatória são determinados e optimizados tentando diferentes valores no conjunto de treino equilibrado. O estudo criou um conjunto de dados equilibrado ao replicar as instâncias de ataque menos ocorridas, o que pode ser considerado uma abordagem imprópria. Não existe uma fase de validação, o que não é recomendado, porque os parâmetros do sistema são dissuadidos - minados pelos dados do conjunto de treino, o que reduz a generalização do modelo. No entanto, a abordagem global é

som porque utiliza com sucesso as propriedades úteis das Florestas Aleatórias.

O estudo agrupou o conjunto de dados em ataques (DoS e Sonda ou Scan) e ataques de minorias (U2R e R2L). O desempenho relatado na detecção do uso indevido foi de 1,92% de erro no conjunto original e 0,05% no conjunto equilibrado com 0,01% e 0% FAR, respectivamente. O estudo também agrupou os ataques DoS e Sonda ou Scan em quatro níveis de ataque (1%, 2%, 5%, e 10%), que representam a percentagem de dados de ataque no conjunto de testes. A precisão obtida para estes quatro níveis de ataque é de 95%, 93%, 90%, e 87%, respectivamente, com uma FAR constante de 1%. A detecção de ataques minoritários (U2R e R2L) é relatada como 65% com um FAR de 1%. A implementação é suficientemente rápida para ser empregada como uma solução online.

Outro estudo [63] utilizou um Random Forest para detectar o uso indevido dos mesmos dados do KDD 1999, de uma forma semelhante a Zhang et al. [62]. A principal diferença é que uma fase d e normalização tem lugar no módulo de pré-processamento. Este estudo compara o desempenho da Random Forests com a máxima probabilidade Gaussiana, Naïve Bayes, e a árvore de decisão classifica - fiers. A precisão relatada para Random Forest é de 97%, 76%, 5%, e 35% para DoS, Probe ou Scan, R2L, e U2R, respectivamente.

Um conjunto de três ANNs (ver Subsecção A, um SVM (ver Subsecção L), e um Spline de Regressão Adaptativa Multivariada (MARS) [64] é utilizado por Mukkamala et al. [65] para a detecção de intru- sion. A primeira ANN é uma retropropagação resiliente NN, a segunda é uma ANN de gradiente conjugado em escala, e a terceira é uma ANN de uma etapa de algoritmo de secante. A maioria dos votos é utilizada para combinar os resultados dos cinco classificadores.

Os dados utilizados são um subconjunto do conjunto de dados da DARPA 1998. Este subconjunto é composto em 80% dos ataques e em 20% dos dados do nor- mal. Os desempenhos de 99,71%, 99,85%, 99,97%, 76%, e 100% são reportados para as categorias Normal, Sonda ou Scan, DoS, U2R, e R2L, respectivamente. O método do conjunto supera - forma a precisão dos classificadores individuais que são uma entrada para o conjunto. A precisão de quatro classes está no intervalo de 99%; apenas a precisão na classe U2R é muito inferior a 76%.

2) Detecção de Anomalias e Detecção Híbrida: O sistema DISCLOSURE de Bilge et al. [66] realiza a detecção de botnet de Comando e Controlo (C&C) utilizando Florestas Aleatórias. Utiliza dados NetFlow facilmente disponíveis mas inclui apenas metadados agregados relacionados com a duração do fluxo e o número de pacotes transferidos, e não cargas úteis de pacotes completos. São extraídos dos dados (por exemplo, tamanho do fluxo, padrões de acesso do cliente, características temporais). Não há detalhes no seu papel sobre quantas árvores foram utilizadas no classificador Floresta Aleatória ou quantos atributos as árvores utilizaram em média. A DISCLUSÃO foi testada em duas redes do mundo real e alcançou uma Taxa Positiva (detecção) Verdadeira de 65%, e uma Taxa Positiva Falsa de 1%.

A aplicação de Random Forests à detecção de anomalias é descrita por Zhang et al. [62], onde um detector de anomalias (out- lier) foi utilizado para alimentar um segundo classificador de ameaças. Com efeito, o método é híbrido, utilizando dois classificadores Floresta Aleatória - um para a detecção de anomalias e o outro para a detecção de uso indevido. O desempenho da detecção de out- lier foi de 95% de precisão com 1% de FAR. O estudo ilustrou como uma anomalia

pode ser implementado utilizando uma medida de proximidade (ou seja, a distância entre duas instâncias calculada a partir da floresta) da Floresta Aleatória. Utilizava como medida de proximidade a soma dos quadrados das proximidades entre as árvores e as categorias.

G. Computação Evolutiva

O termo *computação evolutiva* engloba Algoritmos Genéticos (AG) [67], Programação Genética (GP) [68], Estratégias de Evolução [69], Optimização de Enxame de Partículas [70], Optimização de Colónias de Formigas [71], e Sistemas Imunitários Artificiais [72]. Esta subsecção centra-se nos dois métodos de computação evolutiva mais amplamente utilizados - GA e GP. Ambos se baseiam nos princípios de sobrevivência do mais apto. Operam sobre uma população de indivíduos (cromossomas) que são evoluídos utilizando certos operadores. Os operadores básicos são a selecção, cruzamento, e mutação. Começam normalmente com uma população gerada de forma aleatória. Para cada indivíduo da população, é calculado um valor de aptidão que revela quão bom um determinado indivíduo é para resolver o problema em questão. O indivíduo - assim como os indivíduos com maior aptidão têm uma maior probabilidade de serem escolhidos para o grupo de acasalamento e assim serem capazes de se reproduzir. Dois indivíduos do grupo de acasalamento podem fazer cruzamento (ou seja, trocar material genético entre eles) e cada um pode também sofrer mutação, o que é uma alteração aleatória do material genético do indivíduo. Os indivíduos de maior aptidão são copiados para a geração seguinte.

A principal diferença entre a AG e o GP é a forma como os indivíduos são representados. Na AG, são representados como cordas de bits e as operações de cruzamento e mutação são muito simples. No GP, os indivíduos representam programas e, por conseguinte, repre-mitem árvores com operadores tais como *mais*, *menos*, *multiplicar*, *dividir*, *ou*, *e*, *não*, ou mesmo blocos de programação tais como *se então*, *loop*, etc. Em GP, os operadores de crossover e mutação são muito mais complexos do que os utilizados em AG.

1) Detecção de uso indevido: Li [73] desenvolveu um método que utiliza a AG para a evolução das regras de detecção de uso indevido. Foi utilizado o conjunto de dados de detecção de intrusão DARPA. O cromossoma GA foi concebido para conter o endereço IP de origem, endereço IP de destino, número da porta de origem, número da porta de destino, duração da ligação, protocolo, número de bytes enviados pelo originador e respondedor, e estado da ligação. Na função de adequação, o acordo sobre certas partes do cromossoma (por exemplo, endereço IP de destino) é mais pesado do que noutras (por exemplo, tipo de protocolo). Os operadores tradicionais de crossover e muta- tion são aplicados aos indivíduos da população. São utilizadas técnicas de nichagem para encontrar múltiplos máximos locais (porque são necessárias muitas regras, não apenas uma). As regras mais evoluídas tornam-se parte da base de regras para a detecção de intrusão. Embora o artigo de Li descreva uma técnica interessante e mostre algumas regras desenvolvidas pelo sistema, o que falta, apesar das 181 citações do artigo, é a exactidão do conjunto de regras nos dados de teste.

Abraham et al. [74] usam o GP para desenvolver programas simples para classificar os ataques. As três técnicas de GP utilizadas nas experiências são Programação Genética Linear (LGP), Programação Multi-Expressão (MEP), e Programação de Expressão Genética (GEP). Os programas envolvidos fizeram uso de

+, -, ⍰, /, pecado, cos, sqrt, ln, lg, log2, min, max, e abs

TABELA V
SENSIBILIDADE E ESPECIFICIDADE DO GP COM CROSSOVER HOMÓLOGO

Type of Attack	Sensitivity	Specificity
Smurf	99.93	99.95
Satan	100.00	99.64
IP Sweep	88.89	100.00
Port Sweep	86.36	100.00
Back	100.00	100.00
Normal	100.00	100.00
Buffer Overflow	100.00	100.00
WarezClient	66.67	99.97
Neptune	100.00	99.56

como o conjunto de funções. Os diferentes subconjuntos de características são utilizados pelas diferentes técnicas de GP. Foi utilizado o conjunto de dados de detecção de intrusão DARPA 1998. O conjunto de dados contém 24 tipos de ataque que podem ser classificados em quatro categorias principais: DoS, acesso não autorizado a partir de uma máquina remota (R2L), acesso não autorizado ao super utilizador local (U2R), e vigilância e outras sondagens (Sonda ou Scan). A FAR varia de 0% a 5,7%, dependendo do método utilizado e do tipo de ataque.

Hansen et al. [75] utilizaram GP com crossover homólogo para programas em evolução que realizavam a detecção de intrusão. O crossover homólogo é um operador de crossover especial concebido para reduzir a tendência dos programas evoluídos para crescerem cada vez mais com o número crescente de gerações. Foi utilizado um subconjunto dos dados do KDD 1999, com 30.000 casos para formação e 10.000 para testes. Os conjuntos de dados de formação e testes foram escolhidos de modo a terem as mesmas proporções de ataques de cada tipo que o conjunto de dados completo. A sua precisão variou de 66,7% a 100% dependendo do tipo de ataque (mais baixo para U2R e mais alto para DoS). A especificidade (Tabela V) variou de 99,56% a 100%, o que significa que a FAR é muito baixa. Os GP com resultados homólogos cruzados são também comparados com os resultados do vencedor da Taça KDD 1999, e são melhores. No entanto, essa comparação pode ser inválida porque os resultados neste trabalho são apenas para um conjunto de teste escolhido de 10.000 assinaturas e os resultados para o vencedor da Taça KDD poderiam ter sido para um conjunto de dados de teste diferente.

2) Detecção de Anomalias e Detecção Híbrida: Khan [76] utiliza a AG para desenvolver regras para a detecção de intrusões. Foram utilizados dois subconjuntos de 10.000 casos cada um do conjunto de dados KDD 1999: um para treino e outro para testes. Para o grande número de atributos desses dados, foram escolhidos oito utilizando a Análise de Componentes Principais [77]. Foi utilizada uma população de 10 indivíduos, e parece que apenas uma regra da população final foi utilizada para a classificação dos dados em duas classes: normal e ataque. É surpreendente que os resultados obtidos no conjunto de dados do teste sejam de facto melhores do que os obtidos no conjunto de dados de formação. A precisão e as FAR são de 93,45% (10,8% FAR) e 94,19% (2,75% FAR) para as classes normal e de ataque, respectivamente.

Lu e Traore [78] apresentaram uma abordagem de evolução de regras baseada em GP para detectar ataques conhecidos e novos à rede. A população inicial de regras foi seleccionada com base no conhecimento do terreno dos ataques conhecidos, e cada regra pode ser representada como uma árvore parse. O GP evolui estas regras iniciais para gerar novas regras utilizando quatro operadores genéticos: operador de reprodução, cruzamento, mutação, e operador de condição de queda. O

O operador que deixa de estar em condição aleatória selecciona uma condição na regra, e depois esta condição já não é considerada na regra, impedindo assim que os programas se tornem cada vez mais complicados com o número de gerações. A função de aptidão utilizada baseou-se no apoio e confiança de uma regra (ver Subsecção 1). O artigo utiliza um subconjunto dos dados de detecção de intrusão DARPA. O conjunto de dados de treino consiste em 1 dia de registos de ligação (ou seja, 10.000 registos de ligação) com oito tipos de ataque. O conjunto de dados de teste consiste em registos de outro dia de con- necção com 10 tipos de ataque (dois tipos de ataque são novos). Na avaliação prática, a base de regras em vez de uma única regra é utilizada para testar o desempenho do IDS. Dez mil execuções de GP são executadas e os resultados médios são comunicados. O valor médio do FAR é 0,41% e o valor médio do P_D é 0,5714. O ROC mostra P_D próximo dos 100% quando o FAR está na faixa entre 1,4% e 1,8%. Contudo, quando o FAR está próximo de 0%, o P_D é apenas cerca de 40%. O P_D cai num amplo intervalo entre 40% e 100% porque o número de regras na base de regras é diferente para cada corrida. Os resultados são relatados em dados que contêm novos ataques, bem como ataques conhecidos. Ao contrário de muitos outros artigos que relatam apenas os melhores resultados, Lu e Traore incluem os resultados médios. Os seus melhores resultados, com P_D perto dos 100% e FAR entre 1,4% e 1,8% são bons.

H. Modelos de Markov escondidos

As correntes Markov e os modelos Markov Escondidos (HMMs) pertencem à categoria dos modelos Markov. Uma cadeia de Markov [79] é um conjunto de estados interligados através de probabilidades de transição que determinam a topologia do modelo. Um HMM [80] é um modelo statistical onde o sistema a ser modelado é assumido como sendo um processo Markov com parâmetros desconhecidos. O desafio principal é determinar os parâmetros ocultos a partir dos parâmetros observáveis. Os estados de um HMM representam condições não observáveis que estão a ser modeladas. Ao ter diferentes distribuições de probabilidades de saída em cada estado e permitindo ao sistema mudar de estado ao longo do tempo, o modelo é capaz de representar sequências não estacionárias.

A Figura 5 [81] mostra um exemplo de um HMM para a detecção de intrusão no hospedeiro. Neste exemplo, cada hospedeiro é modelado por quatro estados: *Bom*, *Probed*, *Attack*, e *Compromissado*. A borda de um nó para outro representa o facto de que, quando um hospedeiro está no estado indicado pelo nó de origem, pode transitar para o estado indicado pelo nó de destino. A matriz de probabilidade de transição de estado P descreve as probabilidades de transições entre os estados do modelo. A matriz de probabilidades de observação Q descreve as probabilidades de receber diferentes observações, dado que o hospedeiro se encontra num determinado estado. π é a distribuição do estado inicial. Um HMM é denotado por (P, Q, π).

1) Detecção de uso indevido: No estudo de Ariu et al. [82], os ataques a aplicações web (tais como XSS e SQL-Injection) são considerados, e os HMMs são utilizados para extrair assinaturas de ataque. De acordo com o estudo, 50% das vulnerabilidades descobertas em 2009 afectaram aplicações web. O estudo descreve um sistema chamado HMMPayl, que examina as cargas úteis HTTP utilizando n-gramas e constrói múltiplos HMMs para serem utilizados num esquema de fusão de clas- sifier. O multi-classificador delineado também pode ser considerado como uma abordagem de aprendizagem em conjunto; no entanto, ao contrário de

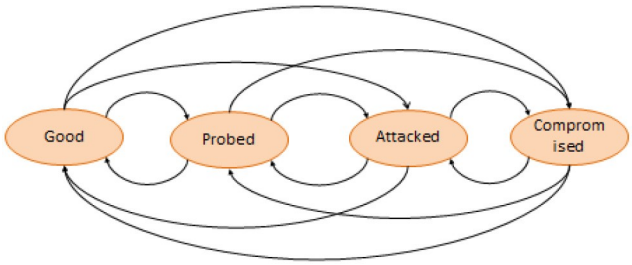


Fig. 5. Um Exemplo de Modelo Markov Escondido.

um método de conjunto, Ariu et al. [82] estabeleceram os classificadores de uma forma competitiva e não complementar. Na secção experi- ment, o estudo de Ariu também utilizou o conjunto de dados DARPA 1999, bem como alguns outros conjuntos de dados HTTP. Na maioria dos experi- mentos, a área média sob a Curva ROC (AUC) de 0,915 a 0,976 é alcançada, para uma taxa de FP que varia de 10E-4 a 10E- [82].

1. Para taxas de PF superiores a 10E-3, HMMPayl atinge taxas de detecção superiores a 0,85, um resultado digno de nota. Para taxas de PF menores, a percentagem de ataques detectados diminui, mas continua a ser superior a 70% para uma taxa de PF de 10E-4.

2) Detecção de Anomalias e Detecção Híbrida: HMMs foram utilizados para detecção de intrusão por Joshi e Phoha [83]. Eles utilizam um HMM com cinco estados e seis símbolos de observação por estado. Os estados do modelo estão interligados de tal forma que qualquer estado pode ser alcançado a partir de qualquer outro estado. O método Baum-Welch [84] é utilizado para estimar os parâmetros HMM. Foi utilizado o conjunto de dados KDD 1999, e 5 das 41 características foram escolhidas para modelação. P_D foi 79%; os restantes 21% são contabilizados como uma taxa de PF (ou seja, classificação da anomalia como nor- mal) e uma taxa de FN (ou seja, classificação do normal como um ataque). Os autores afirmam que poderiam melhorar significativamente a exactidão utilizando mais de cinco características. Embora isto seja possível, está longe de estar provado que ao utilizar um subconjunto maior de características (ou o conjunto completo), a precisão de um HMM aumentará.

I. Aprendizagem Indutiva

Duas técnicas básicas de inferir informação a partir de dados são a dedução e a indução. A dedução infere a informação que é uma consequência lógica da informação presente nos dados e procede de cima para baixo. O raciocínio indutivo move-se de baixo para cima, ou seja, de observações específicas para generalizações e teorias mais amplas. Na aprendizagem indutiva, começa-se com observações e medidas específicas, começa-se a detectar pat- terns e regularidades, formulam-se algumas hipóteses provisórias a explorar, e por último acaba por desenvolver alguns con-clusions ou teorias gerais. Vários algoritmos ML são indutivos (por exemplo, C4.5 para construir árvores de decisão), mas quando os investigadores se referem à aprendizagem indutiva, geralmente significam a poda incremental repetida para produzir redução de erros (RIPPER) [85] e o algoritmo quasi-óptimo (AQ) [86].

A RIPPER constrói regras para problemas de duas classes. Ela induz regras directamente a partir dos dados de formação, utilizando uma abordagem separada e de conquista. Aprende uma regra de cada vez, de tal forma que a regra cobre um conjunto máximo de exemplos no actual conjunto de formação. A regra é podada para maximizar uma medida de desempenho desejada. Todos os exemplos que são correctamente etiquetados pelo

A regra resultante é eliminada do conjunto de formação. O pro- cess é repetido até que o conjunto de treino fique vazio ou até que seja alcançado um critério de paragem pré-determinado.

Um exemplo da regra RIPPER [87] poderia ser:

Adivinhe :- logins falhados **>=** 5

O que significa que se o número de logins falhados for maior ou igual a 5, então esta ligação é um "palpite" (ou seja, um ataque de palpite de senha)

1) Detecção de uso indevido: Lee et al. [87] desenvolveram um trabalho de enquadramento em que foram utilizadas várias técnicas de ML e DM (por exemplo, Aprendizagem Indutiva, Regras de Associação, Exploração Sequencial de Minas de Padrão). Em primeiro lugar, a Sequential Pattern Mining (também chamada Frequent Episodes) foi utilizada para a construção de medidas temporais e estatísticas. É necessária muita experimentação com a janela temporal quando se faz isto. Quando a Sequential Pattern Mining é executada sobre os padrões de intrusão, extrai características que serão utilizadas na detecção do uso indevido. No passo seguinte, essas características são utilizadas pela RIPPER para gerar regras. No entanto, a RIPPER não utiliza todas as características identificadas nas suas regras porque, como acontece com a maioria dos algoritmos de classificação, tem um mecanismo de selecção de características incorporado.

RIPPER gera regras para classificar as ligações telnet a partir do conjunto de dados da DARPA 1998. Sete semanas de dados foram utilizadas para formação, e duas semanas separadas de dados foram utilizadas para o teste. Os dados de teste contêm 38 tipos de ataque, com 14 tipos presentes apenas nos dados de teste. A precisão alcançada para um novo tipo de ataque variou de 5,9 (R2L) a 96,7% (Sonda ou Scan), enquanto que para um tipo de ataque antigo variou de 60 (R2L) a 97% (Sonda ou Scan). O seu método foi capaz de detectar com alta precisão novos ataques que são Sondagem ou Varrimento (96,7%) e U2R (81,8%). A precisão na detecção de novos tipos de ataques de DoS e R2L foi inferior a 25%.

2) Detecção de Anomalias e Detecção Híbrida: Um verdadeiro problema de detecção de anomalias, por definição, não oferece o tipo de dados anormais com antecedência. Por conseguinte, a principal diffi- culdade da detecção da anomalia reside na descoberta dos áries de fronteira entre categorias conhecidas e desconhecidas [88]. No estudo, um gerador artificial de anomalias foi concebido para melhorar o desempenho do detector de anomalias em termos da sua capacidade de generalização. A primeira abordagem foi a geração de anomalias baseadas na distribuição e a segunda foi a geração de anomalias artificiais filtradas. O conjunto de dados da DARPA 1998 foi utilizado na configuração da experiência. Foi alcançada uma impressionante taxa de detecção de anomalias de 94% com um FAR de 2%. O FAR é geralmente desejado ser muito inferior a 1%, especialmente com os tipos de conjuntos de dados DARPA 1998 onde a taxa de pacotes está ao nível do TCP/IP (milhares de pacotes por seg- ond). No entanto, o estudo demonstrou com sucesso como a detecção de anomalias verdadeiras deve ser conduzida e como os dados devem ser utilizados, não empregando um classificador binário para a detecção de anomalias, como muitos outros estudos realizados.

J. Naïve Bayes

Os classificadores naïve Bayes [89] são simples classificadores probabilísticos - aplicando o teorema Bayes. O nome tem origem no facto de as características de entrada serem assumidas como indepen- dentes, enquanto que na prática isto raramente é verdade. O teorema condicional

probabilidades, $p(C \mid f_1, f_2, \ldots, f_m)$, formam o modelo de classificador, e o classificador atribui uma etiqueta de classe como se segue:

$$y(f_1, f_2, \ldots, f_m) = \underset{k \in \{1,\ldots K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{m} p(f_i \mid C_k) \quad (3)$$

onde m é o número de características, K é o número de classes, f_i é a i-ésima característica, C_k é a k-ésima classe, $p(C_k)$ é a probabilidade anterior de C_k, e $P(f_i \mid C_k)$ é a probabilidade condicional de característica f_i dada a classe C_k.

Os classificadores Naive Bayes podem lidar com um número arbitrário de características independentes, sejam elas contínuas ou categóricas. Reduzem uma tarefa de estimativa de alta densidade dimensional a uma estimativa de densidade de núcleo unidimensional, utilizando a hipótese de que as características são independentes.

Embora o classificador Naïve Bayes tenha várias limitações, é um classificador óptimo se as características forem condicionalmente independe- pendente dada a verdadeira classe. Geralmente, é um dos primeiros classificadores que é comparado com os algo- rithms mais sofisticados. Além disso, certos tipos de utilizadores expressaram que compreendem o modelo de classificação de forma mais intuitiva em comparação com outros classificadores complexos (por exemplo, SVM). Uma das maiores vantagens do classificador Naïve Bayes é que é um algoritmo online e a sua formação pode ser completada em tempo linear.

1) Detecção de uso indevido: Panda e Patra [90] utilizaram o classificador Naïve Bayes do pacote Weka [11] e utilizaram o conjunto de dados KDD 1999 para formação e testes. Os dados foram agrupados em quatro tipos de ataque (Sonda ou Scan, DoS, U2R, e R2L) e o classificador atingiu 96%, 99%, 90%, e 90% de precisão de teste, respectivamente, nestas categorias. A FAR acumulada foi reportada como 3%.

O documento comparou os resultados com um classificador NN e declarou que o classificador Naïve Bayes tem uma maior precisão mas um FAR mais elevado, o que é indesejável. Os resultados são relatados como sendo melhores do que a pontuação mais alta alcançada na competição KDD.

2) Detecção de Anomalias e Detecção Híbrida: O trabalho de enquadramento desenvolvido por Amor et al. [91] utiliza uma forma simples de uma rede Bayesiana que pode ser considerada um classificador Naïve Bayes. A construção de uma rede Bayesiana geral é um problema NP- completo, mas a simples rede de um nó de raiz e folhas de atributos formam a estrutura de classificação dos Naïve Bayes.

Este documento também utilizou o conjunto de dados KDD 1999 e agrupou as categorias em três arranjos para reflectir os diferentes tipos de ataque - nários e medições de desempenho. Um único ataque e os dados normais foram contidos no primeiro conjunto. O segundo conjunto con- tingiu os quatro tipos de ataque do conjunto de dados KDD 1999, e o problema a ser resolvido foi uma classificação multi-classe para detecção de uso indevido. O terceiro conjunto continha os dados normais e os quatro tipos de ataque (combinados numa única categoria), e o problema a ser resolvido era um

K. Mineração de Padrões Sequenciais

A mineração de padrões sequenciais emergiu como um dos importantes métodos de DM [92] com o advento das transacções
bases de dados onde cada transacção tem um ID temporal, um ID de utilizador, e um conjunto de itens. Um conjunto de itens é um conjunto de itens distintos adquiridos numa transacção (uma representação estritamente binária na qual um item problema de detecção de anomalias.

O artigo relatou os resultados como 97%, 96%, 9%, 12%, e 88% de exactidão alcançados para as categorias Normal, DoS, R2L, U2R, R2L, e Sonda ou Scan, respectivamente. Nenhum falso alarme é relatado no trabalho, mas como se atinge 97% de normalidade, pode-se assumir que a FAR é inferior a 3%. A experiência de detecção de anomalias é reportada como 98% e 89% de precisão para as categorias Normal e Anormal, respectivamente.

foi ou não comprado). Uma sequência é uma lista ordenada de conjuntos de artigos. A duração da sequência é definida como o num- ber de conjuntos de artigos na sequência. A ordem é determinada pela identificação do tempo. A sequência A (de comprimento n) está contida numa sequência sec- ond, B (de comprimento m), quando todos os conjuntos de itens A, $\{a_i\}$, são subconjuntos de conjuntos de itens B, $\{b_j\}$, com um mapeamento de 1 para 1 entre os índices i a j de modo a que $a_1 \subseteq b_{j1}$, $a_2 \subseteq b$ [illegible], ..., $a_n \subseteq b_{jk}$ e $j_1 \leq j_2 \leq \cdots \leq j_k$. Por outras palavras, cada um dos conjuntos de itens de A é um subconjunto de um conjunto de itens em B. Se um conjunto de itens a_i é um subconjunto de um conjunto de itens b_j, então o próximo conjunto de itens a_{i+1} deve ser um subconjunto de um conjunto de itens b_{j+m}, onde $m > 0$. Na sequência B, são permitidos conjuntos de itens que não são um subconjunto de um conjunto de itens em A (i.e., $n \leq m$).

Num conjunto de sequências, a sequência A é máxima se não for contida em qualquer outra sequência. Considere uma base de dados, *D*, que contém sequências agrupadas por uma certa variável nominal, como o endereço IP, denotado por *p*. Se a sequência A estiver contida em *D(p)* [isto é, uma das sequências de *D(p)* contém A], então A sup- portas *D(p)*. O suporte é a fracção de sequências em *D(.)* que A suporta. Uma grande sequência é definida como a sequência que suporta um limiar mínimo, *Th*. O problema da extracção de sequências é então encontrar todas as sequências máximas que estão contidas em *D* com um suporte mínimo dado pelo utilizador *Th*. As próprias sequências máximas são geradas a partir das sequências em *D,* enumerando todas as sequências possíveis.

1) Detecção de uso indevido: Num campo ligeiramente diferente, a detecção de intrusão em bases de dados [93] utilizou a extracção sequencial de padrões para detectar intrusões em bases de dados através do exame dos padrões sequenciais dos registos das bases de dados. Estes registos incluem vários campos tais como tipo de operação, nome da transacção, ID da transacção, hora de início, hora de fim, etc. A abordagem geral apresentada neste trabalho pode ser aplicada à análise de intrusão semelhante para a segurança cibernética, tais como ataques U2R. O estudo utiliza o algoritmo AprioriAll [92] e gera as sequências máximas com um conjunto de utilizadores de suporte antigo. Os padrões gerados são as assinaturas para as intrusões. O desempenho máximo reportado foi de 91% de taxa de detecção e o pior FAR foi de cerca de 30%. No entanto, o trabalho de mineração de padrões sequenciais aplicados com sucesso, para detectar intrusões em bases de dados através do exame de registos de bases de dados.

2) Detecção de Anomalias e Detecção Híbrida: Li et al. [94] dão um bom exemplo de mineração de padrões sequenciais e a sua aplicação para reduzir a redundância de alertas e minimizar as FARs. Utilizaram o algoritmo AprioriAll para descobrir padrões de ataque em várias fases. As sequências foram os padrões de ataque previamente descobertos ou fornecidos pelo ciberadministrador-trator. Foi também gerada uma ferramenta de visualização de padrões para o utilizador cibernético. Foi utilizado um limiar de apoio de 40%, e as sequências máximas geradas foram utilizadas para verificação de alerta e correlação. O trabalho utilizou os conjuntos de dados DARPA 1999 e DARPA 2000 e foi capaz de detectar 93% dos ataques em 20 segundos. Num segundo conjunto de experiências, o estudo relatou um cenário simulado em tempo real onde foi alcançada uma detecção de 84% dos ataques. O documento afirmava que o limiar de apoio era o

parâmetro principal para controlar a taxa de detecção e o FAR. Os padrões não detectados simplesmente não excederam o limiar de apoio.

L. Máquina Vectorial de Apoio

O SVM é um classificador baseado em encontrar um hiperplano separador no espaço de características entre duas classes de tal forma que a distância entre o hiperplano e os pontos de dados mais próximos de cada classe é maximizada. A abordagem baseia-se num risco de classificação minimizado [95] em vez de uma classificação óptima - ficação. As SVM são bem conhecidas pela sua capacidade de generalização e são particularmente úteis quando o número de características, *m*, é elevado e o número de pontos de dados, *n*, é baixo (*m >> n*).

Quando as duas classes não são separáveis, são adicionadas variáveis frouxas e é atribuído um parâmetro de custo para os pontos de dados sobrepostos. A margem máxima e o lugar do hiperplano é determinado por uma optimização quadrática com um tempo de execução prático de $O(n^2)$, colocando o SVM entre os algoritmos rápidos mesmo quando o número de atributos é elevado.

Vários tipos de superfícies de classificação podem ser realizadas pela aplicação de um núcleo, tais como linear, polinomial, Função de Base Radial Gaussiana (RBF), ou tangente hiperbólica. As SVMs são classificadores binários e a classificação multiclasse é realizada através do desenvolvimento de uma SVM para cada par de classes.

1) Detecção de uso indevido: No trabalho de Li et al. [96], um classificador SVM com um núcleo RBF foi utilizado para classificar o conjunto de dados KDD 1999 em categorias predefinidas (DoS, Probe ou Scan, U2R, R2L, e Normal). Das 41 características, foi seleccionado um subconjunto de atributos, seguindo uma política de remoção de características e uma política de selecção de características únicas. Finalmente, o estudo utilizou 19 características determinadas pelo método de selecção de características.

No mesmo trabalho, um subconjunto do conjunto de formação foi determinado pela Optimização de Colónias de Formigas (ver Subsecção G). Uma possível vantagem desta abordagem é maximizar a classificação geral e minimizar o enviesamento no conjunto KDD, tal como relatado na Secção II. O estudo relatou a sua validação cruzada de dez vezes por- formança como exactidão global de 98% com variância desconhecida. O desempenho mais baixo, de 53%, foi para a categoria U2R. A estratégia de utilizar um subconjunto de um conjunto de formação para ultrapassar as limitações do conjunto de dados KDD vale definitivamente a pena, tal como demonstrado neste artigo.

Amiri et al. [97] utilizaram um SVM menos quadrado para ter um sistema mais rápido para treinar em grandes conjuntos de dados. Para reduzir o número de características no conjunto de dados KDD de 41 para 19 ou menos, utilizaram três algoritmos diferentes de selecção de características. O primeiro foi baseado na escolha da característica que maximiza a classificação perfor- mance, o segundo foi baseado na informação mútua, e o terceiro foi baseado na correlação. Os resultados experimentais mostraram que a abordagem baseada na informação mútua é mais promissora (embora ligeiramente) do que as outras duas.

O conjunto de dados foi amostrado aleatoriamente para ter cerca de 7000 instâncias (de um total de 5 milhões) para cada uma das cinco classes (DoS, Probe ou Scan, U2R, R2L, e Normal). Foi utilizada uma técnica de bootstrapping para fazer uma nova amostragem dos ataques U2R. Para prever o tipo de ataque, foram construídos cinco classificadores para cada categoria. Desta forma, um custo é associado a cada categoria e a

QUADRO VI
COMPARAÇÃO DA PRECISÃO DA SVM MELHORADA COM SNORT E BRO

Method	Test Set	Accuracy (%)	FP Rate (%)	FN Rate (%)
Enhanced SVM	Normal	94.19	5.81	0.00
	Attack #1	66.60	0.00	33.40
	Attack #2	65.76	0.00	34.24
	Real	99.90	0.09	0.00
Snort	Normal	94.77	5.23	–
	Attack #1	80.00	–	20.00
	Attack #2	88.88	–	11.12
	Real	93.62	6.38	–
Bro	Normal	96.56	3.44	–
	Attack #1	70.00	–	30.00
	Attack #2	77.78	–	22.22
	Real	97.29	2.71	–

A classificação final foi determinada. A classificação performance foi reportada como 99% nas classes DoS, Probe ou Scan, R2L, e Normal, e como 93% na classe U2R com 99% de intervalo de confiança.

2) Detecção de Anomalias e Detecção Híbrida: Hu et al. [98] utilizaram a robusta máquina vectorial de suporte (RSVM), uma variação da SVM em que a média do hiperplano discriminador é mais suave e o parâmetro de regularização é automaticamente determinado, como o classificador da anomalia no seu estudo. O Módulo de Segurança Básica [18] partes do conjunto de dados da DARPA 1998 foram utilizadas para pré-processar dados de treino e teste. O estudo mostrou um bom desempenho de classificação na presença de ruído (como alguma má etiquetagem do conjunto de dados de formação) e relatou uma precisão de 75% sem falsos alarmes e uma precisão de 100% com 3% de FAR.

Wagner et al. [99] utilizaram dados NetFlow recolhidos do mundo real e dados de ataques simulados utilizando a ferramenta Flame [100] e outras fontes de fornecedores de serviços de Internet que forneceram dados de ataques do mundo real, tais como varreduras NetBIOS, ataques DoS, spams POP, e varreduras Secure Shell (SSH). O estudo empregou um classificador SVM de uma classe, que é considerado uma abordagem natural para a detecção de anomalias. Foi introduzido um novo núcleo de janela para ajudar a encontrar uma anomalia com base na posição temporal dos dados NetFlow. Mais do que um registo NetFlow entrou neste núcleo, sugerindo que a informação sequencial foi também retida para classificação. Além disso, a janela foi verificada quanto ao endereço IP e volume de tráfego. O kernel da janela funciona como uma medida de semelhança entre os registos NetFlow sequenciais semelhantes. O desempenho foi reportado como sendo 89% a 94% exacto em diferentes tipos de ataques com taxas de FP de 0% a 3%.

A abordagem descrita por Shon e Moon [101] é um trabalho de enquadramento para a detecção de novos ataques no tráfego de rede. A sua abordagem é uma combinação do Mapa de Características de Auto-Organização (SOFM), GA, e SVM. É descrito aqui porque a principal novidade do método é o SVM Melhorado.

As quatro etapas do sistema proposto são as seguintes:

- Um tipo de ANN não supervisionado que executa o SOFM cluster- ing [102] é utilizado para a definição de perfis de pacotes.
- A filtragem de pacotes é realizada utilizando a impressão passiva TCP/IP Fingerprinting.
- A GA selecciona as características.
- Os dados são classificados utilizando o SVM Melhorado, que é derivado de um SVM de uma classe, e o SVM supervisionado

SVM de margem macia. A primeira fornece a capacidade de classificação não rotulada da SVM de uma classe, e a segunda fornece o elevado desempenho de detecção da SVM supervisionada.

O estudo utilizou o conjunto de dados de detecção de intrusão de 1999 da DARPA. Para tornar o conjunto de dados mais realístico, o subconjunto escolhido con- sisted de ataques de 1% a 1,5% e 98,5% a 99% de tráfego normal. Os resultados dos SVM melhorados tiveram 87,74% de precisão, uma taxa de FP de 10,20%, e uma taxa de FN de 27,27%. Esses resultados foram sub-estantilmente melhores do que os das SVM de uma classe, mas não tão bons como as SVM de margem suave. No entanto, a SVM melhorada pode detectar novos padrões de ataque, enquanto que uma SVM de margem macia não pode.

Foi realizada outra experiência com um conjunto de dados recolhidos no instituto onde um dos autores trabalha. O desempenho do sistema melhorado baseado em SVM foi comparado com o por- formance de Snort [16] e Bro [103]. Foram realizados testes sobre dados normais, dois conjuntos de ataques conhecidos, e os chamados dados reais (ou seja, dados recolhidos pelo instituto do autor). No caso de dados nor- mal (ver Quadro VI), Bro efectuou o melhor. O Snort e o SVM melhorado tiveram um desempenho semelhante, sendo o Snort ligeiramente melhor. No caso de ataques conhecidos, o desempenho de Snort e Bro foi melhor do que o do SVM Melhorado. No caso de dados reais, o desempenho do SVM Melhorado foi superior ao do Snort e do Bro.

V. Complexidade Computacional dos Métodos ML e DM

A literatura contém um número limitado de comparações de desempenho para algoritmos ML e DM. O conceito de calibrar a previsão envolve uma espécie de suavização da produção das previsões para as adequar mais adequadamente a uma distribuição. Portanto, uma comparação adequada do desempenho deve incluir a calibração e nenhuma calibração das previsões com uma abordagem adequada, tal como Platt Scaling e Isotonic Regression [104]. De acordo com a comparação empírica em [104], árvores ensacadas, Random Forests, e ANNs dão os melhores resultados. Após a calibração, as árvores e as SVMs têm um melhor desempenho. O estudo também relata que as generalizações não se mantêm, há uma variabilidade significativa entre os problemas e métricas, e os desempenhos dos modelos nem sempre são consistentes. Embora alguns algo- rithms sejam aceites para um melhor desempenho do que outros, o desempenho de um determinado algoritmo ML depende da aplicação e implementação.

O Quadro VII fornece a complexidade computacional (ou seja, a complexidade temporal) dos diferentes algoritmos de ML e DM. Os ele- mentos do Quadro VII foram encontrados através de uma extensa literatura e pesquisa na Internet. Naturalmente, algumas das complexidades de tempo são discutíveis, e baseiam-se na experiência do utilizador e nas capacidades do implementador. A maioria destes algoritmos tem implementações de código aberto bem conservadas. As sugestões feitas no Quadro VII são que os dados consistem em n instâncias, cada uma descrita por m atributos, e n é muito maior do que m.

Como regra geral, os algoritmos O(n) e O(n log n) são considerados como tempo linear e são utilizáveis para abordagens em linha. $O(n^2)$ é considerado como um tempo complexo aceitável para a maioria das práticas. $O(n^3)$ e superiores são considerados algoritmos muito mais lentos e utilizados para abordagens offline.

Uma percentagem mais elevada dos trabalhos abrangidos por este inquérito apresenta as suas abordagens como métodos offline. Os dados pro- cessados estão prontos e são introduzidos no sistema como um todo. Quando a tubulação do sistema é concebida para funcionar como um sistema em linha, ou quando o sistema processa dados em fluxo, várias preocupações têm de ser abordadas, tais como os fluxos de entrada/saída de dados e o buffering, a execução dos métodos em linha, e o des-playing dos resultados com a informação de tempo apropriada. Alguns estudos [35], [42], [52], [62] descreveram os seus sistemas como funcionando em linha e processando os dados de entrada em tempo real em modo suave. Curiosamente, alguns destes estudos utilizam até algoritmos mais lentos, tais como a extracção de sequências [52] para efectuar a detecção de intrusão. Há também questões ao nível do sistema a serem abordadas, tais como o par- titioning dos fluxos de dados de entrada (por exemplo, MapReduce framework [105]), empregando os métodos de aprendizagem, e recolhendo e agregando os resultados em paralelo.

Em geral, quando modelos de previsão de formação, ou características de tráfego de aprendizagem em rede, um método adequado em linha aborda, no mínimo, três factores: complexidade temporal, capacidade de actualização incremental, e capacidade de generalização.

- A complexidade temporal de cada algoritmo é apresentada no Quadro VII. Um método deve estar próximo de aproximadamente O(n log n) para ser considerado um algoritmo de streaming. No entanto, lento
  algoritmos tais como métodos de mineração de sequências ou ANNs são também utilizados dentro de sistemas de streaming, mantendo a janela de dados de entrada e tendo um pequeno n.
- Para a capacidade de actualização incremental, o agrupamento de algo- rithms, métodos estatísticos (por exemplo, HMM, Bayesian net- works), e modelos de conjunto podem ser facilmente actualizados
  de forma crescente [89], [106]. No entanto, actualizações de ANNs, SVMs, ou modelos evolutivos podem causar complicações [89], [106], [106].
- É necessária uma boa capacidade de generalização para que o modelo formado não se desvie drasticamente do modelo inicial quando são vistos novos dados de entrada. A maioria dos
  Os métodos ML e DM de última geração têm uma muito boa capacidade de generalização.

A fase de teste dos métodos é geralmente rápida, na sua maioria por ordem de tempo linear em relação ao tamanho dos dados de entrada. Por conseguinte, uma vez treinados, a maioria dos métodos pode ser utilizada em linha.

VI. Observações e Recomendações

A extensão dos trabalhos encontrados sobre ML e DM para detecção de intrusão cibernética mostra que estes métodos são uma área de investigação prevalecente e em crescimento para a segurança cibernética. A questão é: Quais destes métodos são os mais eficazes para aplicações cibernéticas? Infelizmente, isto ainda não está estabelecido.

A. Observações relacionadas com os conjuntos de dados

O Quadro VIII lista os documentos representativos dos métodos ML e DM aplicados ao domínio cibernético que foram revistos (e descritos na Secção IV), incluindo o número de vezes que foram citados, o problema cibernético que estão a resolver, e os dados utilizados. É interessante que dos 39 documentos listados no Quadro VIII, 28 utilizaram o DARPA 1998, DARPA 1999, DARPA 2000, ou KDD 1999
conjuntos de dados. Apenas dois dados NetFlow usados, dois dados tcpdump usados,

QUADRO VII
COMPLEXIDADE DOS ALGORITMOS ML E DM DURANTE A FORMAÇÃO

Algorithm	Typical Time Complexity	Streaming Capable	Comments
ANN	O(emnk)	low	Jain et al. [107] e: number of epochs k: number of neurons
Association Rules	>> $O(n^3)$	low	Agrawal et al. [108]
Bayesian Network	>> O(mn)	high	Jensen [41]
Clustering, k-means	O(kmni)	high	Jain and Dubes [46] i: number of iterations until threshold is reached k: number of clusters
Clustering, hierarchical	$O(n^3)$	low	Jain and Dubes [46]
Clustering, DBSCAN	O(n log n)	high	Ester et al. [109]
Decision Trees	$O(mn^2)$	medium	Quinlan [54]
GA	O(gkmn)	medium	Oliveto et al. [110] g: number of generations k: population size
Naïve Bayes	O(mn)	high	Witten and Frank [89]
Nearest Neighbor k-NN	O(n log k)	high	Witten and Frank [89] k: number of neighbors
HMM	$O(nc^2)$	medium	Forney [111] c: number of states (categories)
Random Forest	O(Mmn log n)	medium	Witten and Frank [89] M: number of trees
Sequence Mining	>> $O(n^3)$	low	Agrawal and Srikant [92]
SVMs	$O(n^2)$	medium	Burges [112]

um usou dados DNS, um usou comandos SSH, e quatro usaram algum outro tipo de dados. Ao identificar os artigos representativos, este estudo considerou principalmente o método ML ou DM que os autores utilizaram e o facto de os artigos representarem uma abordagem de utilização incorrecta, anomalia ou híbrida. Outro factor importante foi o facto de os artigos terem sido altamente referenciados, o que foi considerado como sendo uma indicação da sua qualidade. No entanto, alguns métodos emergentes promissores foram também incluídos, mesmo que ainda não tenham tido a oportunidade de serem altamente referenciados. Embora o estudo visasse trabalhos escritos em 2000 ou mais tarde, dois trabalhos anteriores foram bem escritos e altamente citados e, por conseguinte, mereceram ser incluídos neste estudo.

O facto de tantos documentos utilizarem os conjuntos de dados DARPA e KDD está relacionado com o quão difícil e demorado é a obtenção de um conjunto de dados representativo. Uma vez que tal conjunto de dados esteja disponível, os investigadores tendem a reutilizá-lo. Além disso, a reutilização do mesmo conjunto de dados deve permitir uma comparação fácil da precisão dos diferentes métodos. Tal como discutido anteriormente, no caso dos conjuntos de dados DARPA e KDD, isto não foi inteiramente verdade porque esses conjuntos de dados são tão grandes que os investigadores optaram por trabalhar em subconjuntos diferentes. Os dois artigos que discutiram a utilização do NetFlow também discutiram o desempenho da detecção de anomalias. Isto é óbvio porque o NetFlow não tem um conjunto tão rico de características como o tcpdump ou os dados DARPA ou KDD, nem as características necessárias para detectar assinaturas particulares na detecção de uso indevido. (As suas características limitam-se à informação de fluxo gerada por routers de gama superior).

B. Factores relacionados com o desempenho do IDS

Um dos factores mais importantes relacionados com o desempenho dos IDSs é o tipo e o nível dos dados introduzidos. Tal como anteriormente afirmado, vários estudos utilizaram conjuntos de dados DARPA ou KDD porque são fáceis de obter e contêm dados ao nível da rede (tcpdump ou NetFlow), bem como dados ao nível do SO (por exemplo, rede

registos, registos de segurança, chamadas de sistema de kernel). Em primeiro lugar, os dados de ataque que chegavam à pilha da rede e o efeito destes pacotes sobre o nível do SO transportavam informações importantes. Como resultado, é vantajoso que um IDS seja capaz de alcançar dados a nível da rede e do kernel. Se apenas estiverem disponíveis dados NetFlow (muito mais fáceis de obter e processar) para o IDS, estes dados devem ser aumentados por dados ao nível da rede, tais como sensores de rede que geram características adicionais de pacotes ou fluxos. Se possível, os dados de trabalho da rede devem ser aumentados por dados ao nível do kernel do SO. Como foi descoberto, vários estudos abordaram o problema da detecção de intrusão examinando comandos a nível de SO (isto é, IDS baseado em host), e não pacotes de rede.

O segundo factor relacionado com o desempenho dos IDSs é o tipo de algoritmos ML e DM utilizados e a concepção do sistema em geral. O inquérito bibliográfico revelou que muitos estudos utilizaram conjuntos de dados DARPA e KDD e aplicaram diferentes tipos de métodos de ML. Estes estudos não construíram realmente um IDS, mas examinaram o desempenho dos métodos ML e DM em alguns dados de segurança cibernética. No entanto, a categorização dos studies em relação às afiliações dos autores revela estudos que construíram IDSs reais e empregaram dados do mundo real capturados de redes de campus ou backbones da Internet. Todos estes estudos parecem ter utilizado sistemas integrados com mais do que um método ML e vários módulos relacionados com a captura de assinaturas de ataque, base de dados de assinaturas, etc.

C. Critérios de comparação

Existem vários critérios através dos quais os métodos ML/DM para cibernéticos poderiam ser comparados:

- Precisão
- Tempo para formar um modelo
- Tempo para classificar uma instância desconhecida com um modelo treinado
- Compreensibilidade da solução final (classificação)

QUADRO VIII
MÉTODOS E DADOS ML E DM QUE UTILIZAM

ML/DM Method	Paper	No. of Times Cited (as of 7/28/2015)	Cyber Approach	Data Used
ANN	Cannady [24]	463	misuse	Network packet-level data
ANN	Lippmann & Cunningham [27]	235	anomaly	DARPA 1998
ANN	Bivens et al. [28]	135	anomaly	DARPA 1999
Association rules	Brahmi et al. [32]	3	misuse	DARPA 1998
Association rules	Zhengbing et al. [33]	33	misuse	Attack signatures (10 to 0 MB)
Association rules	Apiletti et al. [35]	14	anomaly	NetFlow
Association rules – Fuzzy	Luo and Bridges [39]	192	anomaly	tcpdump
Association rules – Fuzzy	Tajbakhsh et al. [38]	124	hybrid	KDD 1999 (corrected)
Bayesian network	Livadas et al. [42]	208	misuse	tcpdump – botnet traffic
Bayesian network	Jemili et al. [43]	31	misuse	KDD 1999
Bayesian network	Kruegel et al. [44]	260	anomaly	DARPA 1999
Clustering – density based	Hendry and Yang [50]	6	misuse	KDD 1999
Clustering – density based	Blowers and Williams [51]	2	anomaly	KDD 1999
Clustering – sequence	Sequeira and Zaki [52]	214	anomaly	shell commands
Decision tree	Kruegel and Toth [55]	155	misuse	DARPA 1999
Decision tree	Bilge et al. [56]	187	anomaly	DNS data
Ensemble learning	Mukkamala et al. [65]	255	misuse	DARPA 1998
Ensemble – Random Forest	Gharibian and Ghorbani [63]	19	misuse	KDD 1999
Ensemble – Random Forest	Bilge et al.[66]	49	anomaly	NetFlow
Ensemble – Random Forest	Zhang et al. [62]	92	hybrid	KDD 1999
Evolutionary Comp - GA	Li [73]	235	misuse	DARPA 2000
Evolutionary Comp - GP	Abraham et al. [74]	83	misuse	DARPA 1998
Evolutionary Comp - GP	Hansen et al. [75]	52	misuse	KDD 1999 – subset
Evolutionary Comp - GA	Khan [76]	15	anomaly	KDD 1999
Evolutionary Comp - GP	Lu and Traore [78]	124	anomaly	DARPA 1999
HMM	Ariu et al. [82]	25	misuse	HTTP payload
HMM	Joshi and Phoha [83]	61	anomaly	KDD 1999
Inductive learning	Lee et al. [87]	1358	misuse	DARPA 1998
Inductive learning	Fan et al. [88]	195	anomaly	DARPA 1998
Naïve Bayes	Benferhat et al. [45]	17	anomaly	DARPA 2000
Naïve Bayes	Panda and Patra [90]	90	misuse	KDD 1999
Naïve Bayes	Amor et al. [91]	277	anomaly	KDD 1999
Sequence mining	Hu and Panda [93]	151	misuse	Database logs
Sequence mining	Li et al. [94]	18	anomaly	DARPA 2000
SVM	Li et al. [96]	56	misuse	KDD 1999
SVM	Amiri et al. [97]	84	misuse	KDD 1999
SVM – Robust	Hu et al. [98]	114	anomaly	DARPA 1998
SVM	Wagner et al. [99]	1	anomaly	NetFlow
One-class SVM and GA	Shon and Moon [101]	166	hybrid	DARPA 1999

Se se comparar a precisão de vários métodos ML/DM, esses métodos devem ser treinados exactamente com os mesmos dados de treino e testados exactamente com os mesmos dados de teste. Infelizmente, mesmo nos estudos que utilizaram o mesmo conjunto de dados (por exemplo, KDD 1999), quando compararam os seus resultados com os melhores métodos da Taça KDD (e geralmente afirmaram que os seus resultados eram melhores), fizeram-no de forma imperfeita - utilizaram um subconjunto do conjunto de dados KDD, mas não necessariamente o mesmo subconjunto que o outro método utilizado. Por conseguinte, a exactidão destes resultados não é comparável.

O tempo para treinar um modelo é um factor importante devido à constante mudança dos tipos e características dos ciberataques. Mesmo os detectores de anomalias precisam de ser treinados frequentemente, talvez incrementalmente, com novas actualizações de assinaturas de malware.

O tempo para classificar uma nova instância é um factor importante que reflecte o tempo de reacção e o poder de processamento de pacotes do sistema de detecção de intrusão.

A compreensibilidade ou legibilidade do modelo de classificação é um meio de ajudar os administradores a examinar facilmente o modelo de fea- tures a fim de remendar os seus sistemas mais rapidamente. Este é um meio para ajudar os administradores a examinar facilmente o modelo de classificação.

informação (tal como tipo de pacote, número de porta, ou alguma outra característica de pacote de rede de alto nível que reflicta o caminho de pé de ciber-ataque) estará disponível através dos vectores de característica que são marcados pelo classificador como uma categoria de intrusão.

D. Peculiaridades do ML e DM para Cyber

O ML e o DM têm sido extremamente úteis em muitas aplicações. O domínio cibernético tem algumas peculiaridades que tornam esses métodos mais difíceis de utilizar. Essas peculiaridades estão especialmente relacionadas com a frequência com que o modelo precisa de ser requalificado e a disponibilidade de dados etiquetados.

Na maioria das aplicações ML e DM, um modelo (por exemplo, classificador) é treinado e depois utilizado durante muito tempo, sem quaisquer alterações. Nessas aplicações, assume-se que os processos são quase estacionários e, portanto, a reciclagem do modelo não acontece com frequência. A situação na detecção de intrusão cibernética é dif- ferente. Os modelos são treinados diariamente [56], sempre que o analista requer [43], ou cada vez que uma nova intrusão é identificada e o seu padrão se torna conhecido [75]. Especialmente quando os modelos são

supostamente treinados diariamente, o seu tempo de formação torna-se impor- tante. (Deve definitivamente demorar menos de 1 dia inteiro para a reciclagem do modelo.) Tradicionalmente, os métodos ML e DM iniciam a formação a partir do zero. No entanto, se um modelo precisa de ser requalificado frequentemente (por exemplo, diariamente) devido a apenas algumas mudanças nos dados, faz mais sentido começar pelo modelo treinado e continuar a treiná-lo ou então usar modelos auto-adaptativos. Uma área fértil de investigação seria investigar os métodos de aprendizagem incremental rápida que poderiam ser utilizados para a actualização diária de modelos para a detecção de anomalias e de uso indevido.

Há muitos domínios em que é fácil obter dados de formação, e nesses domínios os métodos ML e DM geralmente prosperam (por exemplo, recomendações que a Amazon faz para os seus clientes). Em outras áreas onde os dados são difíceis de obter (por exemplo, dados de monitorização sanitária para máquinas ou aeronaves), as aplicações de ML e DM podem ser abundantes. No domínio cibernético, muitos dados poderiam ser colhidos colocando sensores em redes (por exemplo, para obter NetFlow ou TCP), o que, embora não seja uma tarefa fácil, vale definitivamente a pena.

No entanto, há um problema com o enorme volume desses dados - há demasiados dados para armazenar (terabytes por dia). Outro problema é que alguns dos dados precisam de ser rotulados para serem úteis, o que pode ser uma tarefa trabalhosa. Os dados para o treino precisam definitivamente de ser rotulados, e isto é mesmo verdade para os métodos de detecção de anomalias puras. Estes métodos têm de utilizar dados que são normais; não podem desenvolver um modelo com os dados de ataque misturados. Além disso, porque têm de ser testados com novos ataques, também são necessários alguns dados novos de ataque. A maior lacuna observada é a disponibilidade dos dados etiquetados, e definitivamente um investimento que valeria a pena seria a recolha de dados e a etiquetagem de alguns deles. Utilizando este novo conjunto de dados, poderiam ser feitos avanços significativos aos métodos ML e DM em matéria de segurança cibernética e poderiam ser possíveis avanços. Caso contrário, o melhor conjunto de dados disponível neste momento é o conjunto de dados corrigido do KDD 1999. (No entanto, tendo 15 anos, este conjunto de dados não tem exemplos de todos os novos ataques que ocorreram nos últimos 15 anos).

E. Recomendações ML e DM para Detecção de Mau uso e Anomalias

Os IDS são geralmente híbridos e têm módulos de detecção de anomalias e de detecção de uso indevido. O módulo de detecção de anomalias classifica o tráfego anormal na rede. O mod- ule de detecção de uso indevido classifica padrões de ataque com assinaturas conhecidas ou extrai novas assinaturas a partir dos dados marcados de ataque provenientes do módulo de anomalias.

Muitas vezes, um detector de anomalias é baseado num método de agrupamento. Entre os algoritmos de clustering, os métodos baseados na densidade (por exemplo, DBSCAN) são os mais versáteis, fáceis de implementar, menos dependentes de parâmetros ou distribuição, e têm velocidades de processamento elevadas. Nos detectores de anomalias, os SVM de uma classe também funcionam bem e muito pode ser aprendido extraindo regras de associação ou padrões sequenciais dos dados de tráfego normal disponíveis.

Entre os detectores de mau uso, porque as assinaturas precisam de ser capturadas, é importante que o classificador seja capaz de gerar assinaturas legíveis, tais como características de ramos numa árvore de decisão, genes num algoritmo genético, regras na Association Rule Mining,

ou sequências em Sequence Mining. Por conseguinte, os classificadores de caixa negra como ANNs e SVMs não são bem adequados para a detecção de uso indevido.

Vários algoritmos de última geração ML e DM são adequados para a detecção de uso indevido. Alguns destes métodos são estatísticos, tais como redes Bayesianas e HMMs; alguns são baseados na entropia, tais como árvores de decisão; alguns são evolutivos, tais como algoritmos genéticos; alguns são métodos de conjunto, tais como Florestas Aleatórias; e alguns são baseados em regras de associação. A concepção do sistema deve investigar se os dados de formação são de qualidade suficiente e têm propriedades estatísticas que podem ser exploradas (por exemplo, a distribuição Gaussiana). É também importante saber se o sistema necessário funcionará online ou offline. As respostas a tais perguntas determinarão a abordagem ML mais adequada. Na opinião dos autores deste artigo, os dados da rede não podem ser devidamente modelados utilizando uma simples distribuição (por exemplo, Gaussiano) devido ao facto de, na prática, um único pacote de rede poder conter uma carga útil que pode ser associada a dezenas de protocolos de rede e comportamentos de utilizadores [113]. A variabilidade na carga útil é caracterizada por somas de distribuições múltiplas de probabilidade ou distribuições conjuntas de probabilidade, que não são directamente separáveis. Portanto, métodos como as redes Bayesianas ou HMMs podem não ser a abordagem mais forte porque os dados não têm as propriedades que lhes são mais apropriadas. Os métodos evolutivos de cálculo podem demorar muito tempo a funcionar e, por conseguinte, podem não ser adequados para os sistemas que treinam em linha. Se os dados de formação forem escassos, a Random Forests pode ter uma vantagem. Se o ture de assinatura de ataque for importante, árvores de decisão, cálculo evolutivo, e regras de associação podem ser úteis.

VII. Conclusões

O artigo descreve a revisão bibliográfica dos métodos ML e DM utilizados para o ciberespaço. Foi dada especial ênfase à procura de exemplos de trabalhos que descrevem a utilização de diferentes técnicas de ML e DM no domínio cibernético, tanto para uso indevido como para detecção de anomalias. Infelizmente, os métodos que são os mais eficazes para aplicações cibernéticas não foram estabelecidos; e dada a riqueza e complexidade dos métodos, é impossível fazer uma recomendação para cada método, com base no tipo de ataque que o sistema deve detectar. Ao determinar a eficácia dos métodos, não há um critério, mas vários critérios que precisam de ser tidos em conta. Estes incluem (como descrito na Secção VI, Subsecção C) precisão, complexidade, tempo para classificar uma instância desconhecida com um modelo treinado, e compreensibilidade da solução final (classificação) de cada método ML ou DM. Dependendo do IDS específico, alguns podem ser mais importantes do que outros.

Outro aspecto crucial do ML e DM para a detecção de intrusão cibernética é a importância dos conjuntos de dados para a formação e teste dos sistemas. Os métodos ML e DM não podem funcionar sem dados representativos, e é difícil e demorado obter tais conjuntos de dados. Para ser capaz de realizar a detecção de anomalias e detecção de uso indevido, é vantajoso para um IDS ser capaz de alcançar dados ao nível da rede e do núcleo. Se apenas estiverem disponíveis dados NetFlow, estes dados devem ser aumentados por dados ao nível da rede, tais como sensores de rede que geram características adicionais

de pacotes ou fluxos. Se possível, os dados da rede devem ser aumentados por dados ao nível do kernel do SO.

A maior lacuna observada é a disponibilidade de dados rotulados, e definitivamente um investimento que valeria a pena seria a recolha de dados e a rotulagem de alguns deles. Utilizando este novo conjunto de dados, vários métodos promissores de ML e DM poderiam ser utilizados para desenvolver modelos e comparar, estreitando a lista de ML e DM eficazes para aplicações cibernéticas. Poderiam ser feitos avanços significativos aos métodos de ML e DM na segurança cibernética utilizando este conjunto de dados e poderiam ser possíveis avanços significativos.

Existem algumas peculiaridades do problema cibernético que tornam os métodos ML e DM mais difíceis de utilizar (como descrito na Secção VI, Subsecção D). Elas estão especialmente relacionadas com a frequência com que o modelo precisa de ser requalificado. Uma área fértil de investigação seria investigar os métodos de aprendizagem incremental rápida que poderiam ser utilizados para actualizações diárias de modelos para a detecção de utilizações indevidas e anomalias.

REFERÊNCIAS

[1] A. Mukkamala, A. Sung, e A. Abraham, "Cyber security challenges": Designing efficient intrusion detection systems and antivirus tools," in *Enhancing Computer Security with Smart Technology*, V. R. Vemuri, Ed. Nova Iorque, NY, EUA: Auerbach, 2005, pp. 125-163.

[2] M. Bhuyan, D. Bhattacharyya, e J. Kalita, "Network anomaly detection": Métodos, sistemas e ferramentas", *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 303-336, First Quart. 2014.

[3] T. T. T. Nguyen e G. Armitage, "A survey of techniques for inter net traffic classification using machine learning", *IEEE Commun. Surv. T. Tuts.*, vol. 10, no. 4, pp. 56-76, Fourth Quart. 2008.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, e E. Vázquez, "Detecção de intrusão de rede com base em anomalias: Técnicas, sistemas e desafios," *Informática. Secur.*, vol. 28, no. 1, pp. 18-28, 2009.

[5] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, e B. Stiller, "An overview of IP flow-based intrusion detection," *IEEE Commun. Surv. Tuts.*, vol. 12, no. 3, pp. 343-356, Third Quart. 2010.

[6] S. X. Wu e W. Banzhaf, "O uso da inteligência computacional em sistemas de detecção de intrusão": A review", *Appl. Soft Comput.*, vol. 10, no. 1, pp. 1-35, 2010.

[7] Y. Zhang, L. Wenke, e Y.-A. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Netw.*, vol. 9, no. 5, pp. 545-556, 2003.

[8] U. Fayyad, G. Piatetsky-Shapiro, e P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, no. 11, pp. 27-34, 1996.

[9] C. Shearer, "O modelo CRISP-DM": The new blueprint for data mining", *J. Data Warehouse*, vol. 5, pp. 13-22, 2000.

[10] A. Guazzelli, M. Zeller, W. Chen, e G. Williams, "PMML um padrão aberto para partilhar modelos", *R J.*, vol. 1, no. 1, pp. 60-65, Maio de 2009.

[11] M. Hall, E. Frank, J. Holmes, B. Pfahringer, P. Reutemann, e I. Witten, "O software de mineração de dados WEKA": Uma actualização", "*ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10-18, 2009.

[12] R Definição da língua. (2000). *R Core Team* [Online]. Disponível: ftp://155.232.191.133/cran/doc/manuals/r-devel/R-lang.pdf, acedido em Novembro de 2015.

[13] M. Graczyk, T. Lasota, e B. Trawinski, "Análise comparativa de modelos de avaliação de instalações usando KEEL, RapidMiner, e WEKA," *Inteligência Colectiva Computacional. Web Semântica, Redes Sociais e Sistemas Multiagentes*. Nova Iorque, NY, EUA: Springer, 2009, pp. 800–812.

[14] V. Jacobson, C. Leres, e S. McCanne, *The Tcpdump Manual Page*. Berkeley, CA, EUA: Lawrence Berkeley Laboratory, 1989.

[15] G. Pentes. *Wireshark* [Online]. Disponível: http://www.wireshark.org, acedido em Jun. 2014.

[16] Snort 2.0. *Sourcefire* [Online]. Disponível: http://www.sourcefire.com/technology/whitepapers.htm, acedido em Jun. 2014.

[17] G. F. Lyon, *Nmap Network Scanning: O Guia Oficial do Projecto Nmap para a Descoberta de Redes e Scanning de Segurança*. EUA: Insecure, 2009.

[18] R. Lippmann, J. Haines, D. Fried, J. Korba, e K. Das, "The 1999 DARPA offline intrusion detection evaluation," *Computador. Netw.*, vol. 34, pp. 579–595, 2000.
[19] R. Lippmann *et al.*, "Evaluating intrusion detection systems: The 1998 DARPA offline intrusion detection evaluation," in *Proc. IEEE DARPA Inf. Sobreviver. Conf. Expo.*, 2000, pp. 12-26.
[20] S. J. Stolfo, *KDD Cup 1999 Data Set*, University of California Irvine, KDD repositório [Online]. Disponível: http://kdd.ics.uci.edu, acedido em Jun. 2014.
[21] M. Tavallaee, E. Bagheri, W. Lu, e A. Ghorbani, "A detailed analysis of the KDD Cup 1999 data set," in *Proc. 2nd IEEE Symp. Informática. Intell. Secur. Defense Appl.*, 2009, pp. 1-6.
[22] K. Hornik, M. Stinchcombe, e H. White, "Multilayer feedforward networks are universal approximators", *Neural Netw.*, vol. 2, pp. 359-366, 1989.
[23] F. Rosenblatt, "O perceptron: Um modelo probabilístico para o armazenamento e organização da informação no cérebro", *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.
[24] J. Cannady, "Artificial neural networks for misuse detection," in *Proc. 1998 Nat. Inf. Syst. Secur. Conf.*, Arlington, VA, USA, 1998, pp. 443-456.
[25] Scanner de Segurança na Internet (ISS). *IBM* [Online]. Disponível: http://www.iss.net, acedido em Fev. 2015.
[26] B. Morel, "A inteligência artificial e o futuro da ciber-segurança", em *Proc. 4th ACM Workshop Secur. Artif. Intell.*, 2011. pp. 93-98.
[27] R. P. Lippmann e R. K. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Computador. Netw.*, vol. 34, pp. 597-603, 2000.
[28] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, e M. Embrechts, "Network-based intrusion detection using neural networks," *Intell. Eng. Syst. Artif. Neural Netw.*, vol. 12, no. 1, pp. 579-584, 2002.
[29] R. Agrawal, T. Imielinski, e A. Swami, "Regras de associação mineira entre conjuntos de itens em grandes bases de dados", em *Proc. Int. Conf. Gerir. Data Assoc. Informática. Mach. (ACM)*, 1993, pp. 207-216.
[30] C. M. Kuok, A. Fu, e M. H. Wong, "Mining fuzzy association rules in databases", *ACM SIGMOD Rec.*, vol. 27, no. 1, pp. 41-46, 1998.
[31] L. Zadeh, "Fuzzy sets", *Inf. Control*, vol. 8, no. 3, pp. 338-35, 1965.
[32] H. Brahmi, B. Imen, e B. Sadok, "OMC-IDS": At the cross-roads of OLAP mining and intrusion detection", em *Advances in Knowledge Discovery and Data Mining*. Nova Iorque, NY, EUA: Springer, 2012, pp. 13–24.
[33] H. Zhengbing, L. Zhitang, e W. Junqi, "A novel network intrusion detection system (NIDS) based on signatures search of data mining," in *Proc. 1st Int. Conf. Forensic Appl. Techn. Telecommun. Inf. Workshop Multimédia (e-Forensics '08)*, 2008, pp. 10-16.
[34] H. Han, X. Lu, e L. Ren, "Using data mining to discover signatures in network-based intrusion detection", em *Proc. IEEE Comput. Gráfico. Appl.*, 2002, pp. 212-217.
[35] D. Apiletti, E. Baralis, T. Cerquitelli, e V. D'Elia, "Characterizing network traffic by means of the NetMine framework", *Comput. Netw.*, vol. 53, no. 6, pp. 774-789, Abr. 2009.
[36] NetGroup, *Politecnico di Torino, Analyzer 3.0* [Online]. Disponível: http://analyzer.polito.it, acedido em Jun. 2014.
[37] E. Baralis, T. Cerquitelli, e V. D'Elia. (2008). *Generalized Itemset Discovery by Means of Opportunistic Aggregation (Descoberta de Itens Generalizados por Meios de Agregação Oportunista)*. Técnica. Rep., Politecnico di Torino [Online] https://dbdmg.polito.it/twiki/bin/view/Public/NetworkTrafficAnalysis, acedido em Jun. 2014.
[38] A. Tajbakhsh, M. Rahmati, e A. Mirzaei, "Intrusion detection using fuzzy association rules", *Appl. Soft Comput.*, vol. 9, pp. 462-469, 2009.
[39] J. Luo e S. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection," *Int. J. Intell. Syst.*, vol. 15, no. 8, pp. 687-703, 2000.
[40] D. Heckerman, *A Tutorial on Learning with Bayesian Networks*. Nova Iorque, NY, EUA: Springer, 1998.
[41] F. V. Jensen, *Bayesian Networks and Decision Graphs*. Nova Iorque, NY, EUA: Springer, 2001.
[42] C. Livadas, R. Walsh, D. Lapsley, e W. Strayer, "Using machine learning- ing techniques to identify botnet traffic," in *Proc 31st IEEE Conf. Local Comput. Netw.*, 2006, pp. 967-974.
[43] F. Jemili, M. Zaghdoud, e A. Ben, "A framework for an adaptive intrusion detection system using Bayesian network," in *Proc. IEEE Intell. Secur. Informat.*, 2007, pp. 66-70.
[44] C. Kruegel, D. Mutz, W. Robertson, e F. Valeur, "Bayesian event classification for intrusion detection," in *Proc. IEEE 19th Annu. Informática. Secur. Appl. Conf.*, 2003, pp. 14-23.

[45] S. Benferhat, T. Kenaza, e A. Mokhtari, "A Naïve Bayes approach for detecting coordinated attacks," in *Proc. 32nd Annu. IEEE Int. Comput. Software Appl. Conf.*, 2008, pp. 704-709.
[46] K. Jain e R. C. Dubes, *Algoritmos para Agrupamento de Dados*. Englewood Cliffs, NJ, EUA: Prentice-Hall, 1988.
[47] K. Leung e C. Leckie, "Unsupervised anomaly detection in net- work intrusion detection using clusters," in *Proc. 28th Australas. Conf. Comput. Sci.*, vol. 38, 2005, pp. 333-342.
[48] P. W. Holland e S. Leinhardt, "Transitivity in structural models of small groups", *Comp. Grupo Stud.*, vol. 2, pp. 107-124, 1971.
[49] J. Watts e S. Strogatz, "Collective dynamics of 'small-world' net- works", *Nature*, vol. 393, pp. 440-442, Jun. 1998.
[50] R. Hendry e S. J. Yang, "Intrusion signature creation via cluster- ing anomalies", em *Proc. SPIE Defense Secur. Symp. Int. Soc. Opt. Photonics*, 2008, pp. 69730C-69730C.
[51] M. Blowers e J. Williams, "Machine learning applied to cyber oper- ations", em *Network Science and Cybersecurity*. Nova Iorque, NY, EUA: Springer, 2014, pp. 55-175.
[52] K. Sequeira e M. Zaki, "ADMIT: Anomaly-based data mining for intrusions," in *Proc 8 ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2002, pp. 386-395.
[53] R. Quinlan, "Indução de árvores de decisão", *Mach. Learn.*, vol. 1, no. 1,
pp. 81–106, 1986.
[54] R. Quinlan, *C4.5: Programas para a aprendizagem mecânica*. San Mateo, CA, EUA: Morgan Kaufmann, 1993.
[55] C. Kruegel e T. Toth, "Using decision trees to improve signature-based intrusion detection," in *Proc. 6th Int. Workshop Recent Adv. Intrusion Detect.*, West Lafayette, IN, USA, 2003, pp. 173-191.
[56] L. Bilge, E. Kirda, C. Kruegel, e M. Balduzzi, "EXPOSIÇÃO": Finding malicious domains using passive DNS analysis", apresentado no 18º Annu. Netw. Distrib. Syst. Secur. Conf., 2011.
[57] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, e C. Kruegel, "2014 Exposure: Um serviço passivo de análise DNS para detectar e denunciar domínios maliciosos", *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, Abril. 2014.
[58] R. Polikar, "Ensemble based systems in decision making", *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21-45, Third Quart. 2006.
[59] Y. Freund e R. Schapire, "Experiments with a new boosting algo- rithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, vol. 96, pp. 148-156.
[60] P. Long e R. Servedio, "Impulsionar a área sob a curva ROC", *Adv. Neural Inf. Neural. Syst.*, pp. 945-952, 2007.
[61] L. Breiman, "Random forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, 2001.
[62] J. Zhang, M. Zulkernine, e A. Haque, "Random-forests-based net- work intrusion detection systems", *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.*, vol. 38, no. 5, pp. 649-659, Sep. 2008.
[63] F. Gharibian e A. Ghorbani, "Estudo comparativo de técnicas de aprendizagem supervisionada de máquinas para detecção de intrusão", em *Proc. 5th Annu. Conf. Commun. Netw. Serv. Res.*, 2007, pp. 350-358.
[64] J. H. Friedman, "Multivariate adaptative regression splines," *Anal. Statist.*, vol. 19, pp. 1-141, 1991.
[65] S. Mukkamala, A. Sunga, e A. Abraham, "Detecção de intrusão usando um conjunto de paradigmas inteligentes", *J. Netw. Comput. Appl.*, vol. 28, no. 2, pp. 167-182, 2004.
[66] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, e C. Kruegel, "Divulgação: Detecção de servidores de comando e controlo de botnet através de análise de fluxo de rede em grande escala", em *Proc. 28th Annu. Informática. Secur. Conf. Aplic. (ACSAC'12)*, Orlando, FL, EUA, 3-7 de Dezembro de 2012,
pp. 129–138.
[67] D. E. Goldberg e J. H. Holland, "Genetic algorithms and machine learning", *Mach. Learn.*, vol. 3, no. 2, pp. 95-99, 1988.
[68] J. R. Koza, *Genetic Programming: Sobre a Programação de Computadores por Meios de Selecção Natural*. Cambridge, MA, EUA: MIT Press, 1992.
[69] H. G. Beyer e H. P. Schwefel, "Estratégias de evolução": Uma introdução abrangente", *J. Nat. Comput.*, vol. 1, no. 1, pp. 3-52, 2002.
[70] J. Kennedy e R. Eberhart, "Particle swarm optimization", em *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. IV, pp. 1942-1948.
[71] M. Dorigo e L. M. Gambardella, "Sistema de colónia de formigas": Uma abordagem cooperativa de aprendizagem ao problema do caixeiro-viajante", *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53-66, Abr. 1997.
[72] J. Farmer, N. Packard, e A. Perelson, "The immune system, adap- tation and machine learning", *Phys. D: Nonlinear Phenom.*, vol. 2,
pp. 187–204, 1986.
[73] W. Li, "Utilização de algoritmos genéticos para detecção de intrusão na rede," em
Proc. U.S. Dept. de Energia Cyber Secur. Grupo 2004 Train. Conf., 2004, pp. 1–8.

[74] A. Abraham, C. Grosan, e C. Martin-Vide, "Evolutionary design of intrusion detection programs", *Int. J. Netw. Secur.*, vol. 4, no. 3, pp. 328- 339, 2007.
[75] J. Hansen, P. Lowry, D. Meservy, e D. McDonald, "Genetic program- ming for prevention of cyberterrorism through dynamic and evolving intrusion detection," *Decis. Support Syst.*, vol. 43, no. 4, pp. 1362-1374, Ago. 2007.
[76] S. Khan, "Detecção de intrusão na rede baseada em regras usando algo-ritmo genético," *Int. J. Informática. Appl.*, vol. 18, no. 8, pp. 26-29, Mar. 2011.
[77] T. Jolliffe, *Principal Component Analysis*, 2ª ed. New York, NY, USA: Springer, 2002.
[78] W. Lu e I. Traore, "Detecção de novas formas de intrusão na rede utilizando programação genética", *Computador. Intell.*, vol. 20, pp. 470-489, 2004.
[79] A. Markov, "Extension of the limit teorems of probability theory to a sum of variables connected in a chain", *Dynamic Probabilistic Systems*, vol. 1, R. Howard. Hoboken, NJ, EUA: Wiley, 1971 (Reproduzido no Apêndice B).
[80] L. E. Baum e J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", *Bull. Amer. Math. Soc.*, vol. 73, no. 3, p. 360, 1967.
[81] A. Arnes, F. Valeur, G. Vigna, e R. A. Kemmerer, "Usando modelos de markov escondidos para avaliar os riscos de intrusões: System architec- ture e validação de modelos", *Lect. Notas Informáticas. Sci.*, pp. 145-164, 2006.
[82] D. Ariu, R. Tronci, e G. Giacinto, "HMMPayl: Um sistema de detecção de intrusão baseado em modelos Markov escondidos", *Computador. Secur.*, vol. 30, no. 4, pp. 221–241, 2011.
[83] S. S. Joshi e V. V. Phoha, "Investigando modelos Markov ocultos capa- bilidades em detecção de anomalias", em *Proc. ACM 43rd Annu. Southeast Reg. Conf.*, 2005, vol. 1, pp. 98-103.
[84] P. Dempster, N. M. Laird, e D. B. Robin, "Maximum likelihood from incomplete data via the EM algoritmo", *J. Roy. Statist. Soc.*, Série B (metodológica), pp. 1-38, 1977.
[85] W. W. Cohen, "Rápida indução eficaz de regras", em *Proc. 12th Int. Conf. Mach. Learn.*, Lake Tahoe, CA, EUA, 1995, pp. 115-123.
[86] R. Michalski, "Uma teoria e metodologia de aprendizagem indutiva", *Mach. Learn.*, vol. 1, pp. 83-134, 1983.
[87] W. Lee, S. Stolfo, e K. Mok, "A data mining framework for building intrusion detection models", em *Proc. IEEE Symp. Secur. Privacy*, 1999, pp. 120–132.
[88] W. Fan, M. Miller, S. Stolfo, W. Lee, e P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowl. Inf. Syst.*, vol. 6, no. 5, pp. 507-527, 2004.
[89] I. H. Witten e E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 3ª ed. San Mateo, CA, EUA: Morgan Kaufmann, 2011.
[90] M. Panda e M. R. Patra, "Network intrusion detection using Naive Bayes," *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 258-263, 2007.
[91] N. B. Amor, S. Benferhat, e Z. Elouedi, "Naïve Bayes vs. árvores de decisão em sistemas de detecção de intrusão", em *Proc ACM Symp. Appl. Comput.*, 2004, pp. 420-424.
[92] R. Agrawal e R. Srikant, "Mining sequential patterns," in *Proc. IEEE 11th Int. Conf. Data Eng.*, 1995, pp. 3-14.
[93] Y. Hu e B. Panda, "A data mining approach for database intrusion detection", em *Proc. ACM Symp. Appl. Comput.*, 2004, pp. 711-716.
[94] Z. Li, A. Zhang, J. Lei, e L. Wang, "Real-time correlation of network security alerts," in *Proc. IEEE Int. Conf. e-Business Eng.*, 2007, pp. 73- 80.
[95] V. Vapnik, *A Natureza da Teoria da Aprendizagem Estatística*. Nova Iorque, NY, EUA: Springer, 2010.
[96] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, e K. Dai, "An efficient intrusion detection system based on support vector machines and gradual fea- ture removal method", *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424-430, 2012.
[97] F. Amiri, M. Mahdi, R. Yousefi, C. Lucas, A. Shakery, e N. Yazdani, "Mutual information-based feature selection for IDSs", *J. Netw. Informática. Appl.*, vol. 34, no. 4, pp. 1184-1199, 2011.
[98] W. J. Hu, Y. H. Liao, e V. R. Vemuri, "Máquinas robustas de suporte vectorial para detecção de anomalias em segurança informática", em *Proc. 20 Int. Conf. Mach. Learn.*, 2003, pp. 282-289.
[99] C. Wagner, F. Jérôme, e E. Thomas, "Machine learning approach for IP-flow record anomaly detection", em *Networking 2011*. Nova Iorque, NY, EUA: Springer, 2011, pp. 28-39.
[100] D. Brauckhoff, A. Wagner, e M. May, "Flame: Um motor de modelagem de anomalias de baixo nível", em *Proc. Conf. Cyber Secur. Exp. Teste*, 2008.

[101] T. Shon e J. Moon, "A hybrid machine learning approach to net- work anomaly detection", *Inf. Sci.*, vol. 177, no. 18, pp. 3799-3821, Sep. 2007.
[102] T. Kohonen, *Mapa Auto-Organizador*. Nova Iorque, NY, EUA: Springer, 1995.
[103] V. Paxson. (2004). *Ir. 0.9* [Online]. Disponível: http://bro-ids.org, acedido em Jun. 2014.
[104] R. Caruana e A. Niculescu-Mizil, "An empirical comparison of super- vised learning algorithms", em *Proc. ACM 23rd Int. Conf. Mach. Learn.*, 2006, pp. 161-168.
[105] J. Dean e S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107-113, 2008.
[106] H. Guang-Bin, D. H. Wang, e Y. Lan, "Extreme learning machines": Um inquérito", *Int. J. Mach. Aprender. Cybern.*, vol. 2, no. 2, pp. 107-122, 2011.
[107] K. Jain, J. Mao, e K. M. Mohiuddin, "Artificial neural networks" (Redes neurais artificiais): A tutorial", *Computador*, vol. 29, no. 3, pp. 31-44, 1996.
[108] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, e A. I. Verkamo, "Fast discovery of association rules", *Adv. Knowl. Descobrir. Data Min.*, vol. 12, no. 1, pp. 307-328, 1996.
[109] M. Ester, H. P. Kriegel, J. Sander, e X. Xu, "A density-based algo- rithm for discovering clusters in large spatial databases with noise", *Knowl. Discov. Data Min.*, vol. 96, pp. 226-231, 1996.
[110] P. S. Oliveto, J. He, e X. Yao, "Complexidade temporal dos algoritmos evolutivos para optimização combinatória": Uma década de resultados", *Int. J. Autom. Comput.*, vol. 4, no. 3, pp. 281-293, 2007.
[111] G. D. Forney, "The Viterbi algorithm", *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.
[112] J. C. Burges, "A tutorial on support vector machines for pattern recog- nition", *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121-167, 1998.
[113] K. Ahsan e D. Kundur, "Dados práticos escondidos no TCP/IP", em *Proc. ACM Multimedia Secur. Workshop*, 2002, vol. 2, no. 7.

Anna L. Buczak fotografia e biografia não disponíveis no momento da Publicação.

Erhan Guvenil fotografia e biografia não disponíveis no momento da Publicação.