APLICABILIDADE DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE INTRUSÃO E ANÁLISE DE ANOMALIAS DE REDE

ELIAS AMADEU DE SOUZA GOMES

APLICABILIDADE DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE INTRUSÃO E ANÁLISE DE ANOMALIAS DE REDE

Dissertação apresentada ao Programa de Pós-Graduação em Informática: Área de Conc: Gestão de Tecnologia da Informação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Especialista em Informática: Área de Conc: Gestão de Tecnologia da Informação.

Orientador: Prof. Dr. Ítalo Cunha

Brasília, DF

Abril de 2019

de Souza Gomes, Elias Amadeu

G633a Aplicabilidade de Algoritmos de Aprendizado de Máquina para Detecção de Intrusão e Análise de Anomalias de Rede / Elias Amadeu de Souza Gomes. — Brasília, DF, 2019
xiii, 45 f. : il. ; 29cm

Dissertação (especialização) — Universidade Federal de Minas Gerais
Orientador: Prof. Dr. Ítalo Cunha

1. Cibersegurança. 2. Aprendizado de Máquina. 3. Detecção de Intrusão. 4. Detecção de Assinatura. 5. Detecção de Anomalias. I. Título.

CDU 519.6*

UNIVERSIDADE FEDERAL DE MINAS GERAIS

INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
ESPECIALIZAÇÃO EM INFORMÁTICA: ÁREA DE CONCENTRAÇÃO GESTÃO EM TECNOLOGIA DA INFORMAÇÃO

Aplicabilidade de Algoritmos de Aprendizado de Máquina para Detecção de Intrusão e Análise de Anomalias de Rede

ELIAS AMADEU DE SOUZA GOMES

Monografia apresentada aos Senhores:

Prof. Ítalo Fernando Scotá Cunha
Orientador
DCC - ICEx – UFMG

Prof. José Nagib Cotrim Árabe
DCC - ICEx - UFMG

Prof. José Marcos Silva Nogueira
DCC - ICEx - UFMG

Belo Horizonte, 14 de março de 2019

Dedico este trabalho a Deus porque sem Ele eu nada seria. Dedico também a minha família, a todos aqueles que atuam na comunidade acadêmica e a todos que atuam como agentes da tecnologia no mundo!

Excelsior!

Agradecimentos

Agradeço, primeiramente, a Deus pelas oportunidades de crescimento pessoal e profissional, pelo cuidado e amor diários. Agradeço também a minha família e a minha esposa pelo apoio incondicional e por acreditarem em mim. Estendo meus agradecimentos a todos que tornaram esse trabalho possível, ao corpo docente da UFMG em parceria com a ENAP e ao meu país.
Obrigado!

"It takes 20 years to build a reputation
and few minutes of cyber-incident to ruin it."
(Stephane Nappo)

Resumo

Este trabalho propôs uma análise da aplicabilidade de algoritmos de aprendizado de máquina para aprimoramento das técnicas de detecção de intrusão em um sistema de cibersegurança. Este tema tem ganhado ampla relevância no cenário atual devido ao fato de que estruturas governamentais e entidades privadas de relevância econômica, política e social ao redor do mundo têm tido como meta prioritária a disponibilização digital de seus serviços e ativos. Dessa forma, vários estudos têm sido propostos, nos últimos 15 anos, pela comunidade acadêmica, acerca do tema supramencionado com o fim de aprimorar os sistemas que fazem a defesa dos serviços e ativos digitais. Com base nisso, o escopo desse trabalho foi definido e contemplou o levantamento e a análise de alguns desses estudos com o fim de determinar qual o atual estado dos sistemas de detecção de intrusão no que tange à adaptação da sua capacidade de detecção utilizando modelos e algoritmos de aprendizado de máquina. A conclusão atestou a evolução dos sistemas de detecção de intrusão, ressaltando as características de cada estudo, o contexto e as possibilidades de expansão e aplicação.

Palavras-chave: Cibersegurança, Aprendizado de Máquina, Detecção de Intrusão, Artefato Malicioso, Análise Estática.

Abstract

This work proposed an analysis of the applicability of machine learning algorithms for the improvement of intrusion detection techniques in a cybersecurity system. This theme has attained wide relevance in the current scenario due to the fact that government structures and private entities of economic, political and social relevance around the world have had as a priority goal the digital availability of their services and assets. In this way, several studies have been proposed in the last 15 years by the academic community about the aforementioned topic in order to improve the systems that defend digital services and assets. Based on this, the scope of this work was defined and contemplated the survey and the analysis of some of these studies in order to determine the current state of intrusion detection systems on the subject of the adaptability of its own capacity for intrusion detection using machine learning models and algorithms. The conclusion attested to the evolution of intrusion detection systems, highlighting the characteristics of each study, the context and possibilities of expansion and application.

Keywords: Cybersecurity, Machine Learning, Intrusion Detection, Malicious Artifact, Static Analysis.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

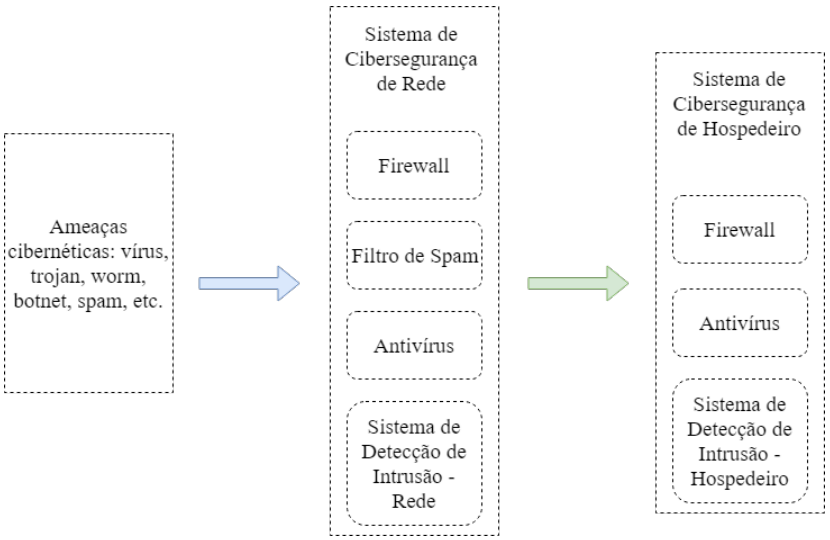
Atualmente, dois campos de estudo têm se destacado na área de segurança de ativos computacionais, a saber, cibersegurança e aprendizado de máquina.

Cibersegurança é o campo de estudo e aplicação de tecnologias e processos cujo objetivo é proteger computadores, redes, programas e dados contra ataques internos e externos e acessos não-autorizados. Um ataque pode ser efetuado utilizando-se de softwares mal intencionados, apelidados comumente de *malwares*, cujo propósito pode ser expor vulnerabilidades no sistema-alvo e possibilitar a exploração deste ou destruir, corromper e impossibilitar o acesso à recursos do sistema-alvo. Entretanto, apesar de ser o método mais comum, nem todo ataque a um sistema ou rede se dá por meio da utilização de *malwares*; existem, por exemplo, ataques que utilizam técnicas de engenharia social para obter acesso, via meios legítimos, a redes e hospedeiros, com o propósito de realizar atividades ilegítimas.

Um sistema de cibersegurança, portanto, deve, dentre outras capacidades, ser capaz de detectar assinaturas conhecidas de ataques (rastros expostos nos pacotes de *malwares* trafegando na rede) e de analisar o ambiente no qual está inserido para descobrir e registrar atividades que fujam do padrão de normalidade do ambiente, ou seja, detectar anomalias no seu ambiente de atuação.

Sistemas de cibersegurança, são compostos por pelo menos um *firewall*, mecanismo que atua na rede para evitar acessos não autorizados ao mesmo tempo que permite o contato com os domínios externso, um software de combate e detecção de programas maliciosos, mecanismo que atua no hospedeiro/serviço comumente chamados de antívirus, e um sistema de detecção de intrusão (IDS) que pode atuar tanto na rede como no hospedeiro fazendo monitoramento de tráfego e analisando o comportamento dos elementos do domínio.

Figura 1.1: Sistema Convencional de Cibersegurança



Fonte: Própria (Gomes, Elias)

Os sistemas de detecção de intrusão (IDS), foco deste trabalho, são os responsáveis pelas atividades supracitadas de detecção de assinaturas e anomalias no ambiente.

Aprendizado de máquina (ML) é o campo de estudo cujo objetivo é permitir aos computadores a habilidade de aprender e formar conhecimento sem explicitamente ser programado para isso. O aprendizado de máquina tem foco na utilização de algoritmos que possibilitem a classificação de amostras, o reconhecimento de padrões em conjuntos de dados, o agrupamento de conjuntos de dados por características semelhantes e a predição de valores fundamentados em um prévio conhecimento adquirido.

Com a expansão do domínio digital no mundo moderno, a crescente banalização, no sentido econômico da palavra, dos preços dos recursos computacionais, aliados com o aumento do poder computacional destes recursos e a explosão dos dados e serviços digitais como substituição permanente dos antigos serviços analógicos, não é de se espantar que a fronteira digital é hoje o cenário de enfoque financeiro, econômico e político mundial. É em decorrência disto que ataques e atividades cibernéticas ilegitimas têm sido alvo de crescente expansão nos países que compõe essa malha digital. Não somente o número de ataques têm crescido, mas também esses ataques têm se modernizado, a ponto de demandar que os mecanismos de defesa também evoluam para contrapor essa onda. É nesse cenário que algoritmos de aprendizado de máquina têm se destacado. Com a utilização destes, é possível permitir aos sistemas de detecção de intrusão, por exemplo, aprenderem novos padrões de ataques e se aprimorarem para detectar novos casos de anomalias não vistos antes (ameaças *zero-day*), sem a necessidade de serem explicitamente reprogramados por um desenvolvedor.

Logicamente, para isso é necessário saber quando utilizar algoritmos de aprendizado de máquina e de que forma utilizá-los, sabendo que nem sempre a utilização de um algoritmo destes significará um aumento de desempenho. Pela natureza destes algoritmos, onde o desempenho está diretamente relacionado à qualidade do conjunto de dados utilizado, é necessário que, para cada cenário de utilização (detecção de ataques novos (*novel attacks*), detecção de assinaturas de pacotes maliciosos, etc.), haja um estudo do impacto de cada modificação paramétrica e de como esses parâmetros influenciam o comportamento dos algoritmos diante de cenários reais, para diferentes tipos de ataques na rede. Por exemplo, um aumento na acurácia de detecção, advindo da modificação dos parâmetros de determinado algoritmo, pode vir acompanhado de um aumento significativo de falsos positivos, o que pode levar à inviabilidade da utilização do algoritmo em casos concretos do mundo real.

Isto posto, no presente trabalho faremos uma análise dos estudos propostos por diversos autores da área de cibersegurança acerca da utilização de algoritmos de aprendizado de máquina para detecção de intrusão. Focaremos nos resultados alcançados, nos conjuntos de dados utilizados e nos algoritmos implementados para determinar a viabilidade e os possíveis desdobramentos futuros para área de prevenção e detecção de ataques nas redes virtuais.

Capítulo 2

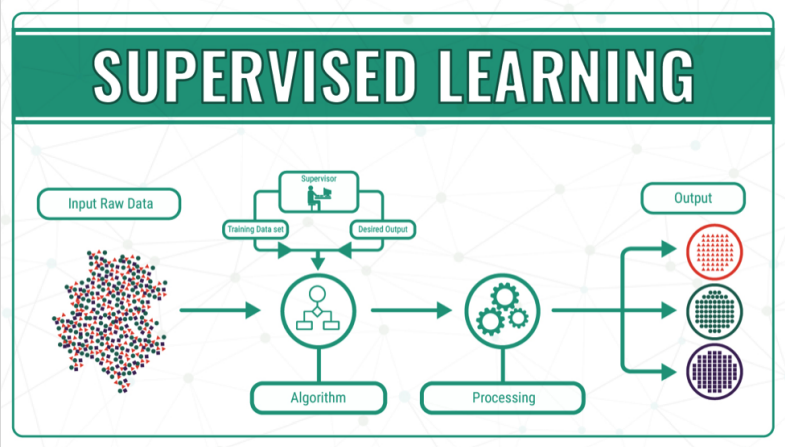
Aprendizado de Máquina

O processo computacional de aprendizado de máquina busca inferir automaticamente e, portanto, generalizar, um modelo de aprendizado com base em um conjunto de dados finito e amostral. Os modelos de aprendizado utilizam-se de funções estatísticas para descrever as dependências entre os dados e a correlação entre as entradas e saídas. De forma simples, este processo permite ao computador aprender, sem ser explicitamente programado, e otimizar sua análise e desempenho com base nos resultados de cada iteração do modelo de aprendizado.

As técnicas de aprendizado podem ser classificadas em técnicas de aprendizado **supervisionado**, técnicas de aprendizado **não-supervisionado**, técnicas de aprendizado **semi-supervisionado** e técnicas de aprendizado **por reforço**.

No aprendizado **supervisionado** (vide figura 2.1), o algoritmo aprenderá com um conjunto de dados rotulados, o que significa dizer que o conjunto de dados utilizado para o treinamento do algoritmo conterá todo o domínio de respostas possíveis para a classificação da amostra avaliada. Sendo assim, a classificação final da amostra (não rotulada) feita pelo algoritmo não poderá extrapolar o domínio de valores possíveis do conjunto de dados de treinamento. A acurácia do algoritmo demonstrará com qual frequência a classificação da amostra está correta. As duas áreas principais de atuação de algoritmos supervisionados são: problemas de predição/classificação e problemas de regressão.

Figura 2.1: Aprendizado Supervisionado

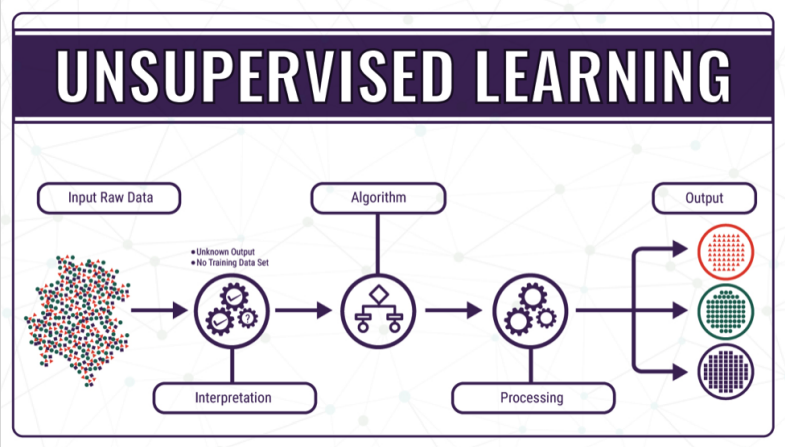


Fonte: Loon [22]

Problemas de predição exigem que o algoritmo classifique uma amostra em um domínio discreto, identificando-a em uma classe dentre um conjunto finito de classes possíveis. Já os problemas de regressão exigem que o algoritmo classifique uma amostra em um domínio contínuo, determinando a relação de dependência entre a amostra e um espaço de variáveis independentes fixas. Geralmente, a relação de dependência entre a amostra e o espaço de variáveis independentes é expresso por meio de uma equação matemática.

Em contraste com o aprendizado supervisionado, no aprendizado **não-supervisionado** (vide figura 2.2), o algoritmo não possui um conjunto de dados de treinamento, e, portanto, o domínio de respostas possíveis não está delimitado, cabendo ao algoritmo extrair as características das amostras e analisar as estruturas dos conjuntos de dados avaliados. Nessa situação, a análise da acurácia do algoritmo está vinculada a sua capacidade de determinar a proximidade das amostras com características semelhantes. As principais áreas de atuação de algoritmos não-supervisionados é na detecção de padrões/reconhecimento de características e na detecção de *outliers*, que são anomalias nas amostras de dados.

Figura 2.2: Aprendizado Não-Supervisionado

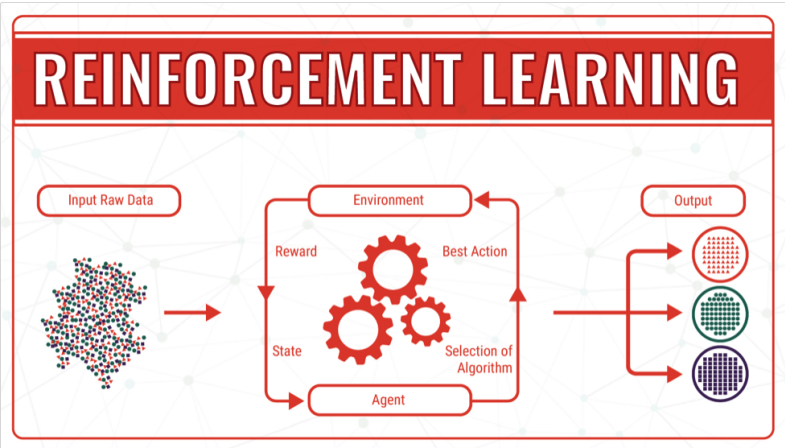


Fonte: Loon [22]

O aprendizado **semi-supervisionado** é exatamente o que o próprio nome denota, o algoritmo de aprendizado possui conjuntos de dados rotulados e não rotulados a sua disposição. Geralmente é utilizado em problemas com domínios bem específicos de classificação e cujos dados não permitem a extração fácil de características, como no caso, por exemplo, da análise de imagens de ressonância magnética.

Por fim, temos as técnicas de aprendizado **por reforço** (vide figura 2.3). Nesse tipo de aprendizado, o objetivo é encontrar a forma ótima de executar uma tarefa específica. Para isso, o algoritmo ajusta sua análise com base em recompensas recebidas a cada iteração de acordo com a resposta que escolhe. Cada iteração é analisada como um sucesso ou falha e isso compõe um feedback que é passado ao algoritmo. Esse aprendizado é mais utilizado no campo da Teoria dos Jogos e no campo da Robótica.

Figura 2.3: Aprendizado Por Reforço



Fonte: Loon [22]

A seguir detalharemos os algoritmos clássicos utilizados em aprendizado de máquina.

Posteriormente, também abordaremos alguns dos novos algoritmos de classificação que têm direcionado o campo de estudo do aprendizado de máquina e consequentemente impulsionado a evolução das ferramentas que se utilizam de aprendizado de máquina para detecção de anomalias e classificação de assinaturas maliciosas na rede.

2.1 Algoritmos Clássicos de Aprendizado de Máquina

Pelo foco do presente trabalho, nesta seção detalharemos os algoritmos mais utilizados em aprendizado de máquina **supervisionado** e **não-supervisionado**.

2.1.1 Aprendizado Supervisionado

A seguir examinaremos os seguintes algoritmos:

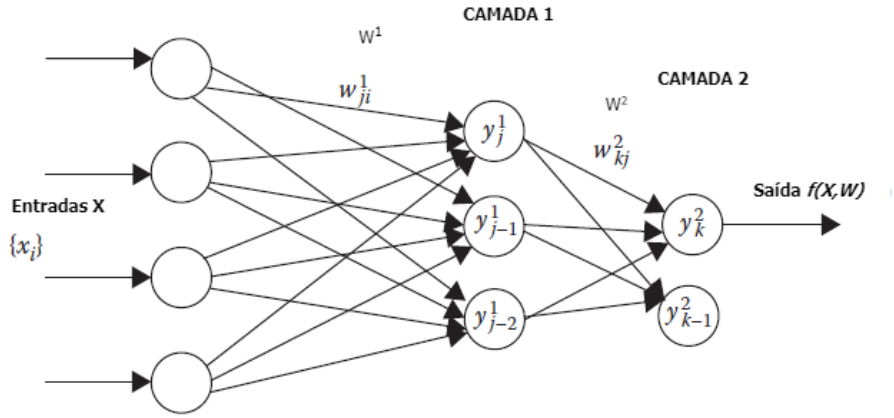
- Redes Neurais Artificiais (ANN)
- *Support Vector Machine* (SVM)
- Árvores de Decisão
- Redes Bayesianas (BNs)
- Modelo Oculto de Markov (HMM)
- *K-nearest Neighbor* (KNN)

2.1.1.1 Redes Neurais Artificiais (ANN)

Uma rede neural artificial é um modelo de aprendizado de máquina que utiliza um processamento não-linear de entradas para compor o resultado final, ou seja, a função de saída é uma correlação não-proporcional das entradas. Isso significa dizer que a saída é definida a partir de um conjunto de etapas de processamento das entradas, onde cada etapa têm um determinado impacto na saída. Essas etapas de processamento são apelidadas de grupos de neurônios pela similaridade metafórica com a atividade cerebral. Esses grupos de neurônios formam as chamadas camadas ocultas da rede neural e, para cada camada, a saída da camada anterior servirá como entrada para a atual. Sendo assim, a última camada será a responsável por gerar a classificação final.

Dessa forma, é necessário que a rede seja capaz de mensurar e priorizar caminhos internos que complementem positivamente a acurácia do resultado final. A forma que a rede encontra para fazer isso é através de um sistema de pesos por camada, onde para cada camada é calculado um peso para os neurônios conforme o desempenho e impacto daqueles neurônios no resultado final de saída da rede. A abstração desse sistema interno da rede neural pode ser vista na figura 2.4, onde a função que define o valor do resultado final (saída) é uma relação entre as entradas(X) e o coeficiente de peso da rede(W), composta pelos pesos internos de cada camada da rede.

Figura 2.4: Rede Neural Clássica

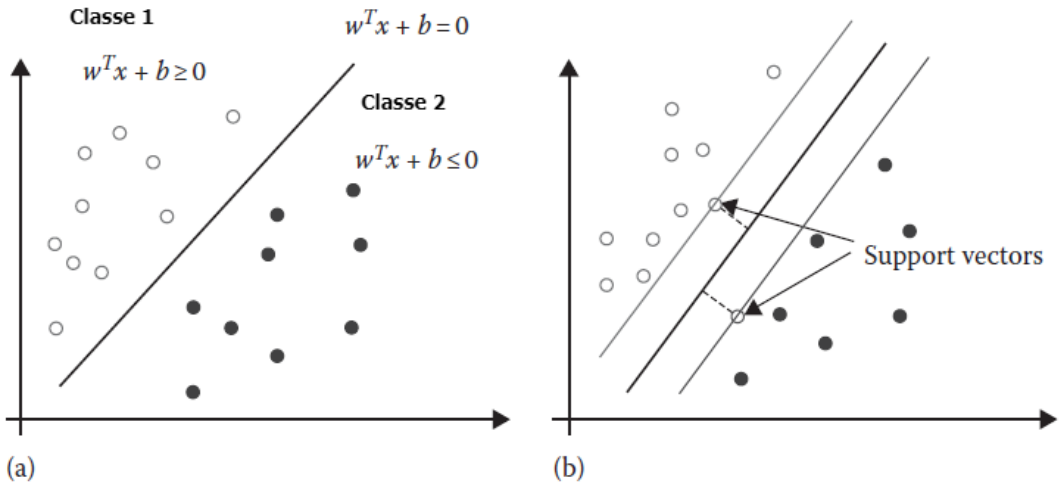


Fonte: Dua & Du [12]

2.1.1.2 *Support Vector Machines* (SVM)

Em um espaço dimensional com X pontos de dados representando n características, uma SVM fará a separação desses pontos de dados com um hiperplano dimensional. O hiperplano terá sua dimensão definida pela dimensão dos dados. Em um espaço de dimensão dois, como é o caso da figura 2.5, o hiperplano será uma linha, por exemplo. O objetivo de uma SVM é classificar os pontos de dados de forma que a distância do hiperplano e os pontos de dados mais próximos seja maximizada para cada classe de atributos, conforme mostrado na figura 2.5. SVMs têm uma ótima capacidade de generalização e são particularmente eficazes quando o número de características é alto e o número de pontos de dados é baixo. A margem de distância máxima e a localização do hiperplano são determinados por um algoritmo de otimização quadrática ($O(n^2)$) o que torna esse algoritmo bem eficiente quando o número de atributos é alto. SVMs são, por natureza, classificadores binários, entretanto, podem ser utilizados em conjunto para realizar classificações multi-classes.

Figura 2.5: SVM (a) Hiperplano (b) *Support Vector*

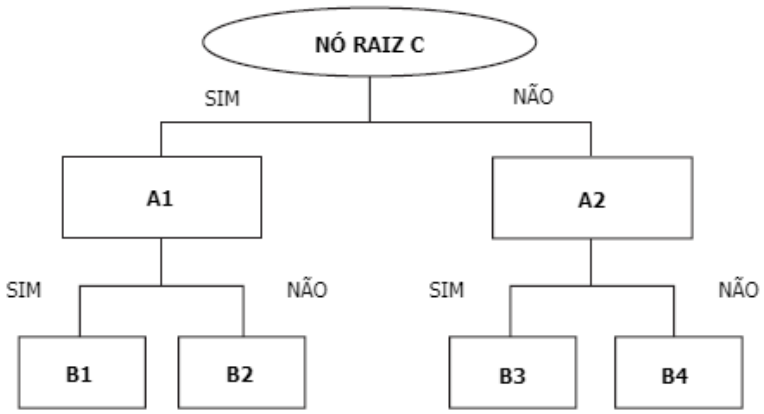


Fonte: Dua & Du [12]

2.1.1.3 Árvores de Decisão (DT)

É um algoritmo cuja estrutura é semelhante a uma árvore onde as folhas representam as classificações e o caminho dos galhos representa a união das características que formou determinada classificação. Essa característica, de subdivisão dos galhos de acordo com as características díspares dos dados, pode levar ao enviesamento da classificação se o número de nós por classe de características for muito destoante. Uma árvore de decisão é construída maximizando o ganho informacional a cada nó, resultando em uma seleção natural de características dentro de um contexto. Os algoritmos mais conhecidos para construção automática de árvores de decisão são chamados ID3 e C4.5 [3]. O tamanho da árvore impacta na classificação final e uma árvore muito grande geralmente possui grande acurácia mas pequena generalização, se tornando muito particular para um subconjunto de problemas. Esse problema é conhecido como *overfitting*, onde o domínio de resolução do algoritmo de árvore se torna restrito a uma subclasse de problemas. Por isso, existem algoritmos de poda (*pruning*) para diminuir o tamanho da árvore e, consequentemente, aprimorar a capacidade de generalização do algoritmo.

Figura 2.6: Árvore de Decisão



Fonte: Dua & Du [12]

***Random Forest* (RF)** A floresta aleatória de decisão (*Random Forest*) é um algoritmo que agrupa várias árvores de decisão e busca combater o problema de enviesamento que pode ocorrer nas árvores de decisão. Este algoritmo busca diluir o viés de classificação das árvores de decisão através de uma técnica de votação e distribuição de pesos para as árvores que o compõe. Outro problema que este algoritmo busca combater é o *overfitting*. Através da adição de um componente de aleatoriedade na divisão e na seleção dos subconjuntos de características, o algoritmo RF consegue ajustar melhor a divisão das árvores de decisão, evitando o crescimento exagerado das árvores e, consequentemente, o *overfitting*.

2.1.1.4 Redes Bayesianas (BN)

A rede bayesiana é um modelo gráfico que representa variáveis e suas relações. Esse modelo se utiliza da distribuição de probabilidade conjunta para tomada de decisões sobre variáveis incertas. O classificador da rede Bayesiana se baseia na regra de Bayes que diz que dada uma hipótese H de classes e um determinado dado x, temos:

$$P(H|x) = \frac{P(x|H)P(H)}{P(x)} \tag{2.1}$$

Onde,

$P(H)$ denota a probabilidade *a priori* de cada classe de forma independente, sem informação sobre a variável x;

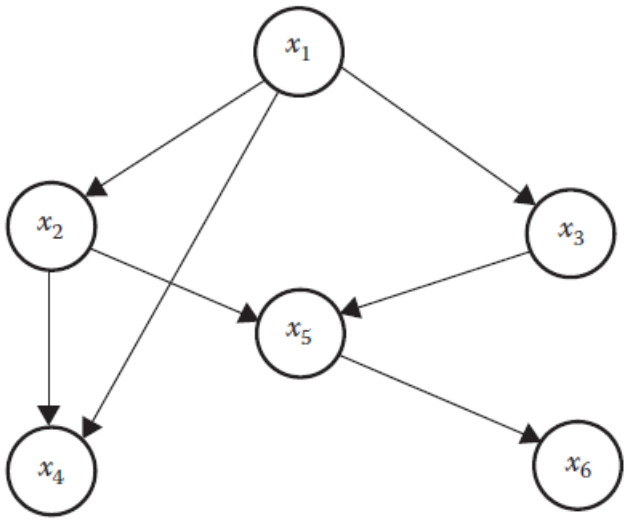
$P(x|H)$ denota a probabilidade *a posteriori* da variável x ocorrer condicionada às possíveis classes de H;

$P(H|x)$ denota a probabilidade *a posteriori* das classes H condicionadas ao dado x;

$P(x)$ denota a probabilidade da variável x de forma independente;

Redes Bayesianas são construídas com nós e arcos, formando um grafo acíclico direcionado, conforme figura 2.7. Os nós do grafo representam as variáveis aleatórias e os arcos representam as dependências probabilísticas entre elas. A existência de nós desconexos representa a existência de variáveis independentes. Cada nó filho, conexo, está associado a uma função de probabilidade relacionada às variáveis do nó pai.

Figura 2.7: Variáveis aleatórias em uma Rede Bayesiana



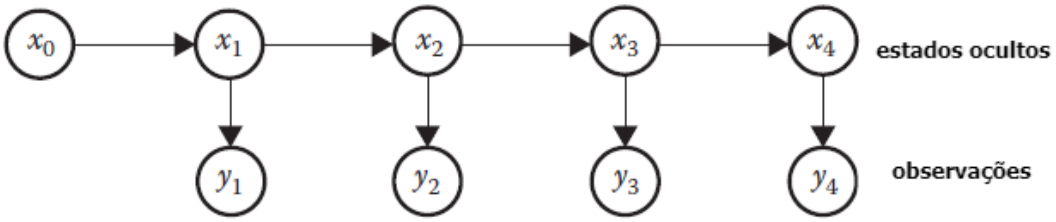
Fonte: Dua & Du [12]

2.1.1.5 Modelo Oculto de Markov (HMM)

Um modelo oculto de Markov é um modelo probabilístico onde assume-se que o sistema modelado é um processo de Markov com parâmetros desconhecidos. O desafio principal é determinar os parâmetros ocultos através dos parâmetros observáveis. Os estados de um modelo oculto de Markov representam condições não observadas sendo modeladas. Sendo assim, é possível ter diferentes distribuições de probabilidade como saídas para

cada estado e isso permite que o modelo seja capaz de representar sequencias não-estacionárias, através da mudança de estados no decorrer do tempo.

Figura 2.8: Modelo Oculto de Markov

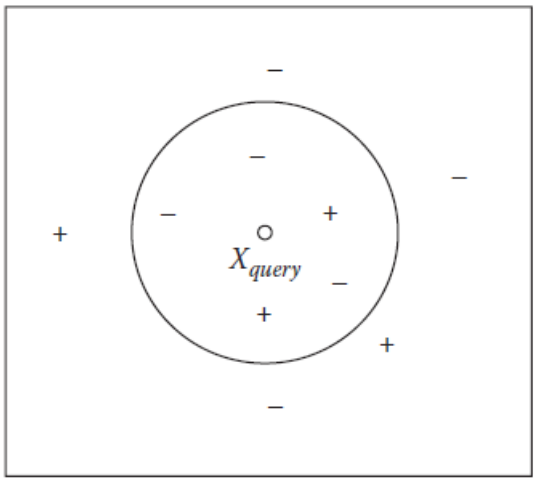


Fonte: Dua & Du [12]

2.1.1.6 *K-nearest Neighbor* (KNN)

Este algoritmo se baseia nas k amostras com menor distância no conjunto de dados de treinamento, e, a partir da avaliação dessas amostras, define um referencial de comparação para os dados do conjunto de teste do algoritmo. Cada ponto de dado de teste será rotulado conforme a distância das amostras do referencial definido pelo conjunto de dados utilizados no treinamento do algoritmo. O número dos vizinhos mais próximos (*nearest neighbors*), representado por k, e a medida de distância são componentes essenciais e impactam diretamente no desempenho desse algoritmo. Como exemplo, temos, na figura 2.9, um ponto de teste X_{query} a ser classificado. Nessa situação hipotético, para $k = 5$, vemos que a maioria dos pontos denota sinal negativo, logo X_{query} será classificado como negativo.

Figura 2.9: Classificação KNN ($k = 5$)



Fonte: Dua & Du [12]

2.1.2 Aprendizado Não-Supervisionado

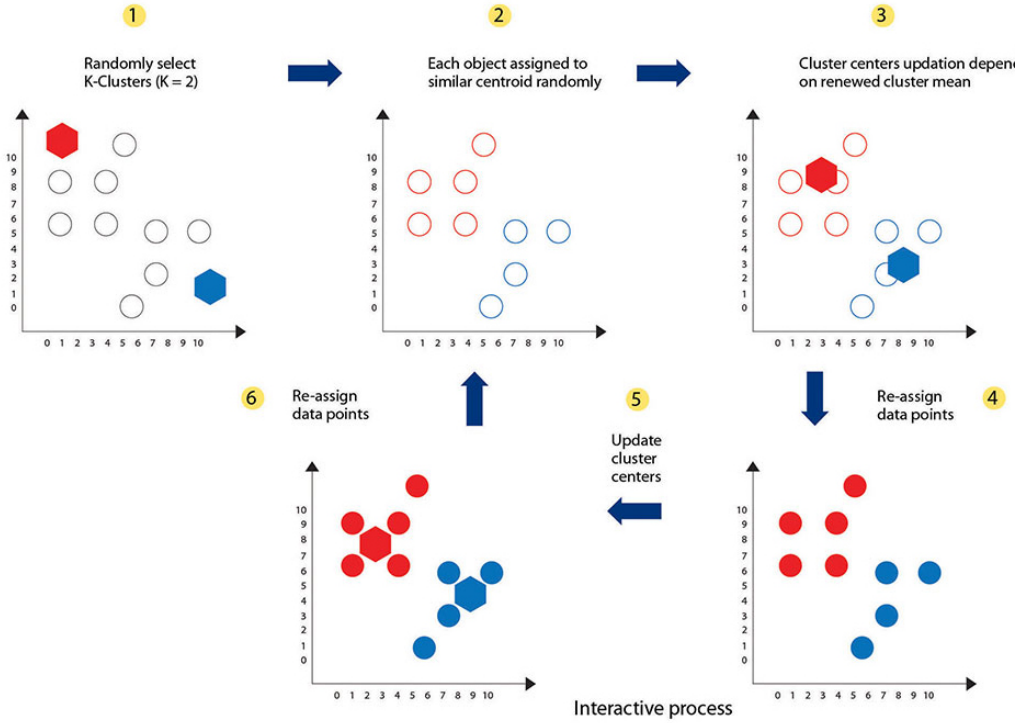
A seguir examinaremos os seguintes algoritmos:

- Clusterização (K-means)
- Regras de Associação Fuzzy

2.1.2.1 K-means

Este algoritmo tem como entrada um número k que indica quantos clusters de pontos de dados teremos como resultado da execução do algoritmo,. A partir dessa entrada, o algoritmo estabelece centroides em diversos pontos aleatórios do espaço dimensional. Para cada ponto de dado existente na amostra de dados, o algoritmo calcula a menor distância do ponto de dado até o centroide mais próximo. Descoberto o centroide mais próximo, aquele ponto de dado será identificado àquele cluster, representado pelo centroide. Depois de computados todos os pontos pertencentes a um cluster, o algoritmo ajustará a posição do centroide do cluster com base na média dos pontos que o compõe. Caso haja um novo ponto pertencente a esse cluster, por conta do ajuste do centroide, este será adicionado ao cluster, conforme mostrado na figura 2.10. O algoritmo irá parar quando não houverem mais ajustes de centroides e nem novas associações entre amostras de dados e clusters.

Figura 2.10: Agrupamento KMeans ($k = 2$)



Fonte: Ltd. [23]

2.1.2.2 Regras de Associação Fuzzy

O objetivo deste algoritmo é descobrir regras de associações desconhecidas previamente nos pontos de dados, como, por exemplo, *SE(A e B) ENTÃO C*. Uma regra de associação descreve um relação entre diferentes atributos. Uma regra de associação relaciona a frequência de ocorrência dos pontos de dados e estabelece métricas para descobrir a correlação entre os pontos de dados. Uma limitação do algoritmo Regra de Associação é que somente funciona para dados binários, entretanto, é possível utilizar lógica fuzzy para expandir o algoritmo e permitir a análise de variáveis categóricas e numéricas, conforme mostrado por Mangalampalli & Pudi.

2.2 Redes Neurais Profundas (*Deep Learning*)

Aprendizado profundo é o nome dado aos algoritmos de aprendizado de máquina que possuem múltiplas camadas de processamento não-linear para extração de características e transformação de modelos de aprendizado. A maior diferença com relação aos algoritmos clássicos é que os algoritmos de aprendizado profundo possuem uma com-

plexidade maior de execução e por isso são capazes de abstrair modelos de aprendizado mais complexos e representativos.

As redes neurais profundas são redes neurais com multíplas camadas ocultas de entrada e saída (*hidden layers*). Geralmente são construídas para abstrair modelos mais complexos e suas camadas possuem relacionamentos não-lineares resultantes da abstração das complexidades pelo modelo representativo.

A maior diferença entre a rede neural clássica e a rede neural profunda está no número de camadas intermediárias e na maneira que tais camadas são ajustadas durante o processo de aprendizado da rede neural. A rede neural clássica se utiliza de pesos e contra-pesos para ajustar os nós de processamento do seu modelo de aprendizado. Já as redes neurais profundas, por serem mais complexas, necessitam de formas mais robustas de aprimoramento de suas camadas intermediárias, utilizando algoritmos de ajuste entre a camada anterior $(n-1)$ e a camada subsequente (n).

2.2.1 *Recurrent Deep Neural Networks* (RNN)

RNN é uma classe de rede neural artificial profunda onde as conexões dos neurônios internos formam grafos direcionados, como loops internos, possibilitando a persistência das informações na rede. Esse loop interno permite que uma informação seja passada de uma camada a outra dentro da rede em várias instâncias da rede. Sendo assim, é possível para a rede operar sobre sequencias de vetores dados como entrada e gerar sequencias de vetores como saída, aumentando a eficiência da rede e gerando modelos de representação bem mais complexos, sem perda de acurácia, se comparado às redes neurais artificiais tradicionais.

2.2.2 *Convolutional Feedfoward Deep Neural Networks* (CNN)

CNN é uma classe de rede neural artificial profunda onde cada neurônio da camada n, que compõe a malha do modelo, recebe sua entrada de apenas um subconjunto dos neurônios da camada $n-1$ anterior. Essa característica faz com que as CNN sejam bastante efetivas em análises de dados espaciais. Por conta dessa característica, essas redes também possuem um menor custo computacional, em geral, do que os outros modelos de redes neurais profundas mencionados. Entretanto, cabe ressaltar que para a maioria das análises envolvendo dados não-espaciais, apesar do custo computacional, os outros modelos de redes neurais profundas tendem a ter uma melhor acurácia.

2.2.3 *Fully-Connected Feedfoward Deep Neural Networks* (FNN)

FNN é uma classe de rede neural artificial profunda onde cada neurônio da camada n, que compõe a malha do modelo, está conectado a todos os outros neurônios da camada $n-1$ anterior. Por conta disso, as *FNN* não utilizam algoritmos de correlação dos dados de entrada, ou seja, não fazem suposições acerca dos dados de entrada. Essa característica faz com que as *FNN* sejam soluções robustas para serem utilizadas em classificações de propósito geral, onde predominam os dados não-espaciais. Essa robustez está aliada ao alto custo computacional decorrente da conexão total da malha de neurônios da camada n e $n-1$.

Capítulo 3

Detecção de Intrusão

Um sistema de detecção de intrusão é um software que monitora uma rede de computadores com o objetivo de detectar atividades maliciosas que visam o extravio ou a censura de informação, a corrupção de protocolos de rede ou a indisponibilização de determinado serviço. No panorama contemporâneo, de um fluxo exorbitante de informações e conexões virtuais, os sistemas de detecção de intrusão não conseguem acompanhar a complexidade e o dinamismo dos ataques cibernéticos. A cada dia os ataques se tornam cada vez mais sofisticados e novas estratégias de invasão são criadas. Para obter adaptabilidade em alto nível escalar, métodos eficientes de generalização e classificação podem ser utilizados para aprimorar os sistemas de detecção de intrusão.

3.1 Detecção de Assinaturas

Detecção de assinatura é uma técnica utilizada para reconhecer, especificamente, padrões únicos de comportamento não-autorizado na rede, ou no hospedeiro, e com isso predizer e detectar qualquer tentativa similar subsequente. Esses padrões são conhecidos como assinaturas e, geralmente, incluem registros específicos de arquivos de log ou cabeçalhos e informações internas de pacotes, trafegando na rede, identificados como ameaça. A assinatura nada mais é que o padrão de bits que está diretamente ligado ao comportamento ameaçador e malicioso. Caso o sistema encontre uma assinatura possivelmente maliciosa, este irá alertar o administrador do sistema sobre o ataque detectado.

A técnica de detecção por assinatura, tipicamente, vasculhará todo o tráfego da rede em busca de padrões similares ou iguais aos padrões previamente conhecidos e estabelecidos. A efetividade dessa técnica está diretamente ligada ao conhecimento armazenado acerca das assinaturas de ataques conhecidos. Sendo assim, é imprescindível

a utilização de uma base de assinaturas maliciosas de qualidade e que sempre esteja atualizada com as assinaturas mais recentes. Por conta dessa característica, esta técnica não é eficiente na descoberta de ataques maliciosos *zero-day*, ou seja, àqueles onde não há precedência de conhecimento ou atividade, ataques com padrões não contemplados anteriormente. Da mesma forma, uma base de conhecimento sem qualidade pode gerar um alto número de falsos positivos e falsos negativos na detecção de atividades maliciosas.

3.1.1 Aprendizado de Máquina em Detecção de Assinaturas

Conforme citado anteriormente, para que a técnica de detecção de assinaturas maliciosas possa ser utilizada com eficiência, é necessário que exista uma base de assinaturas atualizada e de qualidade. Um sistema de detecção de assinaturas consiste, basicamente, de duas grandes etapas, a saber: a coleta/processamento de informação e a identificação da ameaça. A etapa de coleta e processamento de informações prepara os dados de entrada para identificação e classificação de padrões de assinaturas através da normalização e da limpeza dos dados, selecionando e extraindo características impactantes dos conjuntos de dados e eliminando ruídos e redundâncias. Com a base de dados formada, o sistema pode prosseguir para a definição de modelos de aprendizado para a classificação dos pacotes de dados trafegando na rede ou no sistema. Algoritmos de aprendizado de máquina podem ser utilizados tanto na etapa de seleção de características e processamento de dados em informação quanto na etapa de classificação ou agrupamento dos ataques/assinaturas na rede ou no sistema.

3.2 Detecção de Anomalias

Detecção de anomalias tem por alvo qualquer evento cujo comportamento se distancie do conjunto predefinido de comportamentos normais. Sistemas de detecção de anomalias pressupõem que um evento intrusivo pertence a um conjunto maior de atividades anômalas e maliciosas. Diferentemente da detecção de assinaturas, a detecção de anomalias primeiro define um perfil para comportamentos considerados normais no ambiente sistêmico. Esses comportamentos refletem o estado normal da rede. Qualquer comportamento que fuja ao conjunto definido, será considerado como uma anomalia dentro da rede.

Por conta dessa característica, ataques *zero-day*, ou seja, aqueles sem precedência anterior, podem ser detectados na rede. Da mesma forma, como no caso da detecção de assinaturas, o sistema de detecção de anomalias precisa ter um conjunto robusto

e bem definido dos perfis comportamentais da rede em que atuará. Não obstante, também devem ser capaz de adaptar perfis de comportamento conforme a dinamicidade e volatilidade da rede. O maior desafio desse sistema é definir de forma eficaz e eficiente o limiar comportamental de cada domínio de atuação da rede.

3.2.1 Aprendizado de Máquina em Detecção de Anomalias

Assim como no caso da detecção de assinaturas, para anomalias também é necessario haver um processamento de dados e informações, onde será feita uma seleção minuciosa de características impactantes, além da redução de ruídos e redundâncias nos conjuntos de dados. Da mesma forma, atividades e eventos na rede também precisarão ser classificados, com uma extensão onde, por conta do dinamismo e da volatilidade da rede, também existe a necessidade de executar agrupamentos de novos padrões de comportamentos normais para compor a base de informações. Por conta da sensibilidade na determinação dos limiares comportamentais da rede, é necessário estar sempre verificando a taxa de falsos positivos e falsos negativos na rede, isso; portanto, deve ser uma característica de impacto na escolha do algoritmo de aprendizado de máquina.

Capítulo 4

Resultados e Análise - Utilizando ML para Detecção de Intrusão

Nesse capítulo, analisaremos os estudos recentes sobre a utilização de aprendizado de máquina para detecção de intrusão. Faremos um levantamento e uma enumeração dos principais estudos e quais os resultados obtidos por eles. Em seguida, analisaremos qual foi o impacto do algoritmo de ML utilizado e qual a métrica foi utilizada para mensurar esse impacto.

4.1 Conjuntos de Dados Utilizados nas Pesquisas

A seguir serão examinados os principais conjuntos de dados utilizados nas pesquisas de cibersegurança. O foco será naqueles utilizados, principalmente, para as pesquisas de análise de algoritmos de aprendizado de máquina e inteligência artificial em cibersegurança [42], mais especificamente em detecção de intrusão.

4.1.1 KDD Cup 1999 (DARPA1998)

A Agência Avançada de Defesa e Projetos de Pesquisa (DARPA) juntamente com o Laborátorio de Pesquisas da Força Aérea dos Estados Unidos patrocinaram a criação de um conjunto de dados de ataques, DARPA1998, para avaliar os sistemas de detecção de intrusão (IDS). A partir dos dados capturados nessa avaliação, os laboratórios de pesquisa do MIT (*MIT Lincoln Labs*) agregaram dados de tráfego de rede capturados pela ferramenta *tcpdump* e executaram processamentos e normalizações no conjunto de dados resultante. O resultado final foi a criação do conjunto de dados, utilizado na Terceira Competição Internacional de Ferramentas de mineração de Dados e Descoberta

de Conhecimento (*The Third International Knowledge Discovery and Data Mining Tools Competition*), conhecido como KDD Cup 1999 [37].

Esse conjunto de dados foi criado com o propósito de auditar e averiguar o desempenho de um sistema de detecção de intrusão e sua composição consiste de uma variada gama de dados simulados de invasão de redes militares. Por esse motivo, esse conjunto de dados é considerado *benchmark* para avaliação de pesquisas e experimentos envolvendo detecção de intrusão. Os dados deste conjunto estão subdivididos em quatro categorias principais de ataques, a saber:

- Negação de Serviços (DoS)
- *user-to-root* (U2R)
- Ataque Remoto-Local (R2L)
- Ataque de Sondagem

Além disso, esse conjunto de dados também agrega informações sobre as principais características das conexões TCP e do tráfego interno das redes que o utilizam.

4.1.1.1 NSL-KDD

Em uma análise do conjunto de dados KDD 1999 feita por Tavallaee et al. [24] foram identificados alguns limitadores no conjunto de dados que afetam o desempenho dos sistemas que o utilizam. Entre os limitadores encontrados, o que teve o maior impacto no desempenho dos sistemas foi a existência de um número considerado de redundâncias de registros que causava o enviesamento do conjunto de dados, a depender de como este fosse utilizado. Por isso, após uma série de análises estatísticas, Tavallaee et al. [24] propuseram um novo conjunto de dados, chamado de NSL-KDD, que consiste de um conjunto de registros selecionados do conjunto de dados KDD 1999 que não possuem as limitações outrora mencionadas.

4.1.2 ECML-PKDD 2007

Este conjunto de dados foi criado para a Conferência Européia de Aprendizado de Máquina e Descoberta de Conhecimento (*European Conference on Machine Learning and Knowledge Discovery*), em 2007. Este conjunto de dados está descrito em XML e tem enfoque nos ataques de aplicações web, sendo que seus dados foram divididos nas seguintes categorias de ataques:

- *Cross Site Scripting* (XSS)
- *SQL Injection*
- *LDAP Injection*
- *XPATH Injection*
- *Command Injection*
- Ataque de Caminho Dirigido

4.1.3 ISOT

ISOT é o acrônimo estrangeiro de *Information Security and Object Technology* e é um conjunto de dados que contém dados, representando pacotes maliciosos e não-maliciosos, do fluxo de tráfego de redes de *bots*, chamados de *botnets*. Os dados maliciosos foram obtidos do *Honeypot Project*, que consiste de uma rede focada em analisar e coletar dados de ataques maliciosos na web, principalmente os envolvendo *botnets*, como *phishing*, por exemplo. Os dados não-maliciosos, por sua vez, foram obtidos de um laboratório de pesquisa de tráfego de rede da Ericson, na Hungria.

Por fim, esses dados foram combinados, culminando em um conjunto de dados rico para análise de ataques automatizados de coleta de informações, invasões de redes e negação de serviços.

4.1.4 HTTP CSIC 2010

Esse conjunto de dados foi criado para testar os sistemas de proteção contra ataques na web. Basicamente, esse conjunto de dados consiste de 6.000 requisições web normais e mais de 25.000 requisições anômalas. O conjunto de anomalias engloba, em sua maioria, ataques na camada de aplicação, e, pela coleção e adjacência de suas características, esse conjunto de dados têm fornecido subsídios consistentes para um bom desempenho de sistemas de proteção contra anomalias em aplicações web, conforme mostrado por Gimenez *et al.* [4, e.g.].

4.2 Métricas de Performance

Para analisar o desempenho de um algoritmo de aprendizado de máquina, utilizam-se métricas derivadas de uma matriz de confusão, para o caso dos algoritmos super-

visionados, ou de uma matriz de correspondência, para o caso dos algoritmos não-supervisionados.

Essas matrizes nada mais são do que a combinação das possíveis situações finais em que determinada amostra, quando verificada, pode se encontrar ao final da fase de classificação/agrupamento. Dessas possíveis combinações, as métricas são produzidas associando pesos e delegando importância a determinados cenários para cada verficação.

A seguir examinaremos quais as principais métricas foram utilizadas nas pesquisas analisadas neste trabalho. Para isso, para cada métrica será fornecida uma descrição semantica e, em seguida, será detalhada a fórmula matemática que a compõe. Manteremos os nomes das métricas em inglês para padronização e conformidade.

4.2.1 Métricas para Algoritmos Supervisionados

Para a descrição das métricas a seguir, favor considerar o seguinte:

- tp = *true positive*, são os valores verdadeiros positivos.
- tn = *true negative*, são os valores verdadeiros negativos.
- fp = *false positive*, são os valores falso positivos.
- fn = *false negative*, são os valores falso negativos.

4.2.1.1 *Accuracy*

É a proporção das classificações, dado todo o conjunto N de amostras, que estão corretas.

$$acc = \frac{tp + tn}{tp + fp + tn + fn} \tag{4.1}$$

4.2.1.2 *Positive Predictive Value or Precision*

É a proporção de classificações positivas verdadeiras, dado o conjunto de todas as classificações positivas.

$$p = \frac{tp}{tp + fp} \tag{4.2}$$

4.2.1.3 *Negative Predictive Value*

É a proporção de classificações negativas verdadeiras, dado o conjunto de todas as classificações negativas.

$$npv = \frac{tn}{tn + fn} \tag{4.3}$$

4.2.1.4 *TP Rate or Recall or Sensitivity*

É a proporção de classificações positivas verdadeiras, dado o conjunto de todas as classificações verdadeiras (positivas e negativas).

$$r = \frac{tp}{tp + tn} \tag{4.4}$$

4.2.1.5 *TN Rate or Specificity*

É a proporção de classificações negativas verdadeiras, dado o conjunto de todos os elementos que são negativos verdadeiros em sua essência.

$$s = \frac{tn}{tn + fp} \tag{4.5}$$

4.2.1.6 FAR or FP Rate

É a proporção de classificações falso positivas, dado o conjunto de todos os elementos que são negativos verdadeiros em sua essência.

$$far = \frac{fp}{tn + fp} \tag{4.6}$$

4.2.1.7 Miss Rate or FN Rate

É a proporção de classificações falso negativas, dado o conjunto de todos os elementos que são positivos verdadeiros em sua essência.

$$m = \frac{fn}{fn + tp} \tag{4.7}$$

4.2.2 Métricas para Algoritmos Não-Supervisionados

Para os algoritmos não-supersionados, as métricas são baseadas em distâncias e portanto não serão matematicamente formuladas conforme feito anteriormente para as métricas de algoritmos supervisionados.

4.2.2.1 *Internal Metrics*

Essas métricas são usadas em conjuntos de dados que foram clusterizados/agrupados. Geralmente mensuram a distância inter-cluster, ou seja, àquela entre dois clusters diferentes a partir do seu centróide, ou mensuram a distância intra-cluster, ou seja, àquela entre os elementos de um mesmo cluster.

Além disso, podem mensurar também o índice de Dunn (*Dunn index*) que fornece um indicativo da qualidade da clusterização com base nas medidas intra e inter clusters.

4.2.2.2 *External Metrics*

Métricas externas atuam sobre conjuntos de dados que possuem rotulação de classes. Por conta disso, se assemelham às métricas de algoritmos supervisionados, com a única diferença que a interposição dos clusters pode ser um fator atuante na divisão dos dados. Geralmente, assim como com algoritmos supervisionados, fazem uso de métricas como o *FAR* e o *Recall*, citados na seção anterior.

4.3 Estudo 1 - Redes Neurais Artificiais (ANN)

Em sua análise, Buczak *et al.*[3] descrevem um estudo [5] focado na utilização de redes neurais artificiais para auxiliar na detecção de assinaturas através do processamentos de dados de pacotes capturados de uma rede para reconhecimento de padrões de assinaturas de pacotes maliciosos. Os dados, conforme informações do estudo, representam informações dos pacotes que trafegam na rede, tais como, identificador de protocolo (ID), porta de origem, porta de destino, endereço de origem, endereço de destino, código ICMP e o conteúdo do pacote. Os ataques na rede foram simulados com auxílio dos programas Internet Scanner e SATAN (*Security Administrator Tool for Analyzing Networks*). A métrica utilizada no estudo foi a acurácia e este reportou um valor representativo de 93%, sendo que cada pacote na rede foi categorizado como normal ou como pertencente a um grupo de ataque.

No mesmo artigo, Buczak *et al.* [3] também descrevem um estudo [21] focado na utilização das *ANN* para auxiliar na detecção de anomalias através de um sistema de seleção de palavras-chave. A seleção das palavras-chave foi executada utilizando o processamento de uma *ANN* em transcrições de sessões *telnet* e, a partir destas, foram então computadas estatísticas de uso e ocorrência para cada palavra nas sessões. Uma segunda *ANN*, de estrutura similar à primeira, operou sobre as instâncias definidas como ataques e buscou classificar cada ataque utilizando uma nomenclatura pré-definida. Ambas as *ANN* foram compostas de *perceptrons* multi-camadas. O sis-

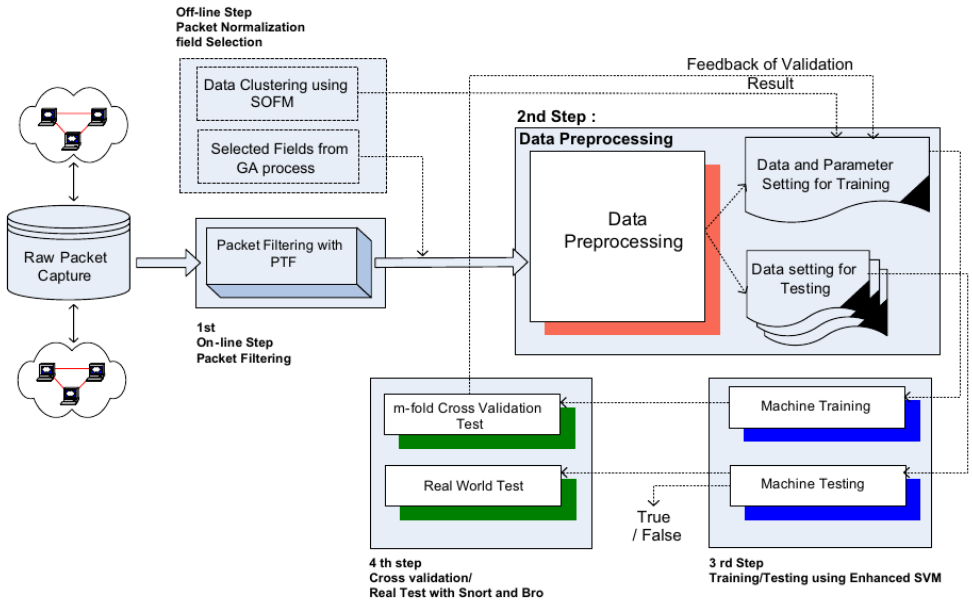
tema supramencionado alcançou uma acurácia de 80% para detecção de ataques com, aproximadamente, 1 resultado falso por dia. Essa taxa de resultados falsos, conforme o estudo, representa uma otimização de desempenho do sistema de detecção de ordem dois na escala de magnitude do sistema padrão, sem nenhuma perda na acurácia.

4.4 Estudo 2 - SVM

Em seus estudos, Shon *et al.* [35] e Parati *et al.* [28] descrevem, em artigos separados, a utilização conjunta de SVM e Algoritmo Genético para detectar e classificar ataques e anomalias de rede.

Shon *et al.* propõe o que eles chamam de *Enhanced SVM*, que é um arcabouço de propósito geral para detecção e classificação de ataques na rede. O arcabouço consiste de um núcleo composto pelo algoritmo SVM auxiliado por mecanismos de agrupamento e processamento/filtro de características utilizando ferramentas de filtragem e algoritmos genéticos (vide figura 4.1) .

Figura 4.1: Arcabouço de Aprendizado de Máquina proposto por Shon *et al.*



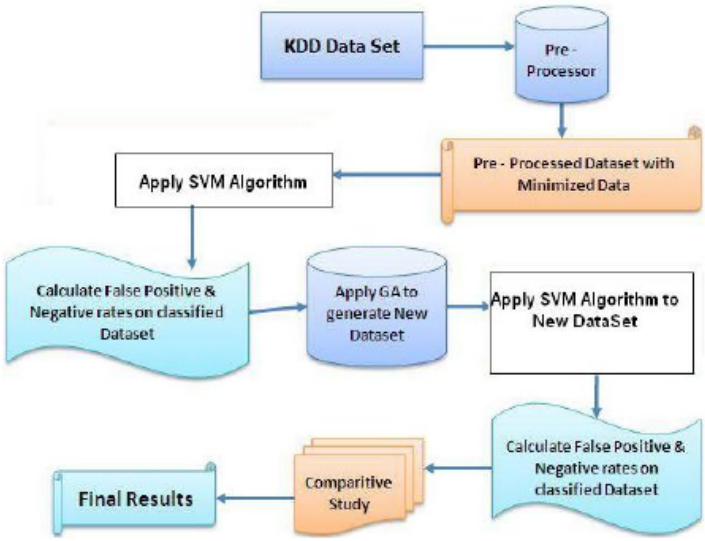
Fonte: Shon & Moon [35]

Os resultados do estudo de Shon [35] mostram que o sistema proposto em torno do algoritmo SVM possui uma acurácia alta, em média 97%, para casos de teste reais, com um baixo índice de resultados falsos (falsos positivos e negativos).

Parati *et al.* [28] também propõe um sistema de detecção de intrusão em torno do algoritmo SVM com auxílio de algoritmos genéticos para instigar a variabilidade no

conjunto de dados analisados com o propóstio de facilitar a detecção de ataques novos, comumente chamado de *novel attacks* (vide figura 4.2).

Figura 4.2: IDS proposto por Parati *et al.*



Fonte: Namita Parati [28]

Os resultados do estudo de Parati *et al.* [28] mostram que o sistema proposto alcançou uma alta acurácia, em torno de 99%, para os casos de teste do conjunto de dados KDD Cup 99 e suas variações criadas utilizando algoritmos genéticos.

4.5 Estudo 3 - Árvores de Decisão

Buczak *et al.* [3] apresentam dois estudos [17] [19] sobre a utilização de árvores de decisão em detecção de intrusão, um sobre a utilização para detecção de assinaturas maliciosas e outro para detecção de anomalias de rede.

No primeiro estudo [17] apresentado por Buczak *et al.* [3], as árvores de decisão são utilizadas para substituir o núcleo de detecção de assinaturas utilizado no IDS *Snort*. Primeiramente foi feito um agrupamento das regras utilizadas no *Snort 2.0* e a partir daí uma árvore de decisão foi derivada utilizando uma variante do algoritmo ID3. O método proposto foi aplicado a um conjunto de dados de testes extraídos utilizando a ferramenta *tcpdump* sobre o conjunto DARPA 1998.

Os resultados foram comparados com o desepenho do IDS *Snort* e o ganho de desempenho (tempo de processamento e detecção) variou entre 5% e 105%, com média de 40.3%. O estudo mostrou que a acurácia de detecção se manteve e que o tempo de processamento foi reduzido significantemente.

No segundo estudo [19] [20] apresentados por Buczak *et al.* [3], foi analisado um sistema chamado *EXPOSURE*, criado para detecção de anomalias envolvendo ataques de DNS cujo classificador é uma árvore de decisão derivada da implementação do algoritmo C4.5. O conjunto de dados analisado consiste de mais de 100 bilhões de consultas de DNS que resultaram em 4.8 milhões de nomes distintos de domínios. O estudo mostrou que, após um treinamento extensivo inicial, o algoritmo obteve uma acurácia em torno de 98.5% com um erro de 0.9% para resultados falsos. Apesar da alta acurácia, o estudo conclui que os resultados de acurácia podem variar consideravelmente dependendo do conjunto de dados testados, demonstrando uma certa instabilidade deste algoritmo para detecção deste tipo de anomalias.

Ugochukwu *et al.* [8] propuseram um IDS de estrutura simples utilizando somente árvores de decisão e comparou com o mesmo IDS utilizando o algoritmo de rede Bayesiana. O conjunto de dados utilizado no estudo é composto por amostras de ataques, que incluem ataques de negação de serviço (DoS - *Denial of service*), *probe*, *U2R* e *R2L*. A partir da classificação desse conjunto, Ugochukwu *et al.* avaliaram o desempenho dos classificadores utilizando as métricas de precisão e *recall*. Os resultados mostram que os algoritmos de árvore de decisão e *random forest* (conjunto de árvores de decisão) obtiveram desempenhos mais consistentes para o conjunto de dados de teste, atingindo uma média de 97% de precisão e 88% de *recall*. O algoritmo de rede Bayesiana obteve um resultado de *recall* melhor que os algoritmos de árvore, atingindo o valor médio de 90%; entretanto, levando em consideração o tempo de processamento e o resultado das outras métricas supramencionadas, os algoritmos com árvores de decisão foram mais eficientes.

Sindhu *et al.* [36], por sua vez, propôs integrar algoritmos de aprendizado de máquina e construir um framework para detecção de intrusão cujo núcleo classificador é uma árvore de decisão. A este framework denominou *neurotree*. O framework proposto utiliza algoritmos genéticos e redes neurais para fazer extração de características e para estruturar o conjunto de dados para classificação. Após o processamento pelos algoritmos supracitados, a árvore de decisão fará a classificação das amostras. A árvore de decisão foi derivada utilizando o algoritmo C4.5 aprimorado (*Enhanced C4.5*) sobre o conjunto de dados resultante. O conjunto de dados utilizado para extração de características e transformação foi o conjunto KDD 1999. Os resultados obtidos foram comparados a outros IDS construídos com classificadores utilizando algoritmos clássicos de ML.

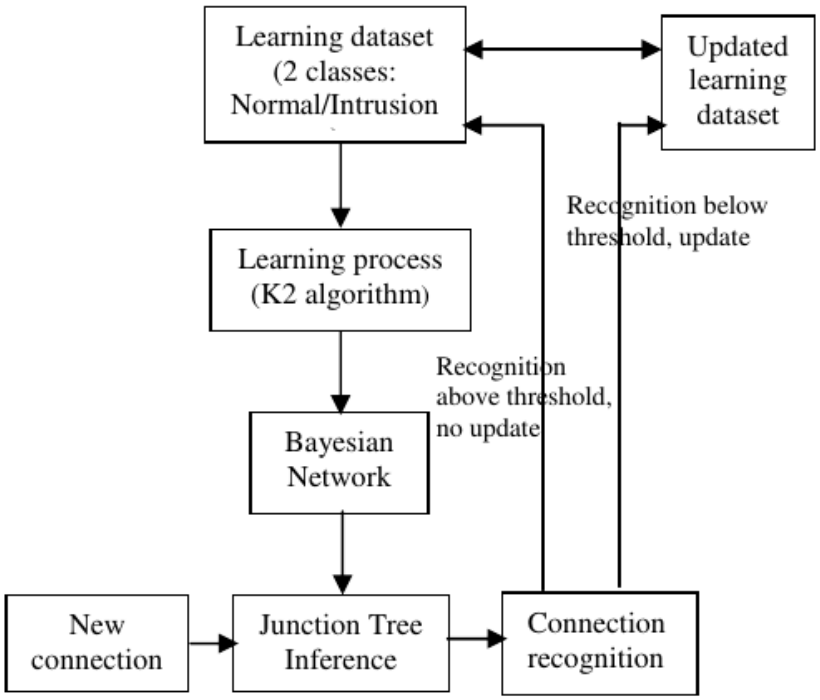
Os resultados obtidos demonstraram um desempenho superior em todos os quesitos avaliados. O sistema proposto por Sindhu *et al.* [36] alcançou uma porcentagem de detecção de 98,38% com erro médio de 1,62%. O estudo chegou à conclusão que o

sistema é capaz de detectar instâncias novas e instâncias específicas de ataques.

4.6 Estudo 4 - Rede Bayesiana (BN)

Jemili *et al.* [14] propuseram o estudo da utilização de Rede Bayesiana para detecção de intrusão através da criação de um arcabouço para um sistema de detecção de intrusão cujo núcleo de classificação de ataques seja uma rede Bayesiana (vide figura 4.3).

Figura 4.3: Framework para o sistema de detecção proposto por Jemili *et al.*



Fonte: Farah Jemili [14]

O estudo utilizou o conjunto de dados KDD 1999, focando em nove das principais características do conjunto, a saber:

- ***Protocol Type***
- ***Service***
- ***Land***
- ***Wrong Fragment***
- ***Num_failed_logins***
- ***Logged_in***

- ***Root_ shell***
- ***Is_ guest_ login***

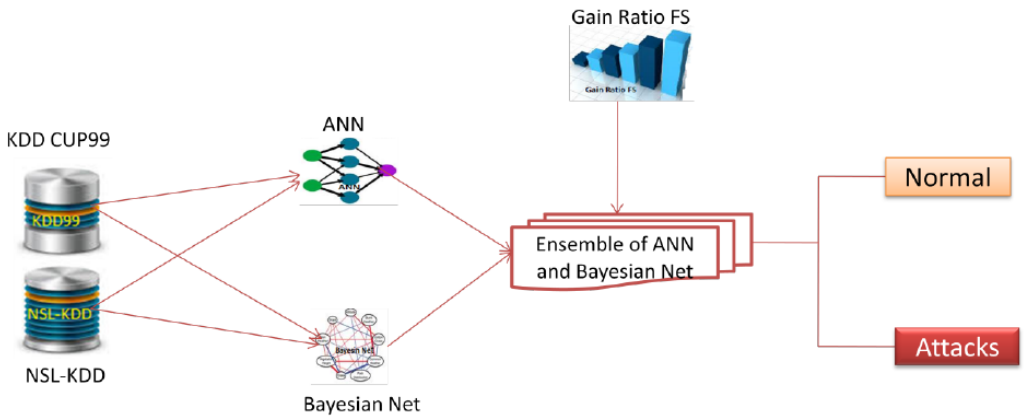
Os seguintes tipos de ataques, pertencentes ao conjunto de dados supramencionado, foram avaliados:

- *DoS*
- *Probe or Scan*
- *R2L*
- *U2R*
- Outros

Os resultados obtidos demonstraram uma atuação razoável do algoritmo de rede Bayesiana, sendo que a média de detecção de ataques alcançada foi aproximadamente 88%. Uma exploração mais aprofundada por Jemili *et al.* [14] buscou analisar a acurácia para cada tipo de ataque e os resultados obtidos foram os seguintes: 89%, 99%, 21%, 7% e 66%, para os ataques de *DoS*, *Probe*, *R2L*, *U2R* e outros, respectivamente. O estudo sugere que o desempenho ruim para os tipos de ataque *R2L* e *U2R* foi em decorrência do baixo número de instâncias de treinamento, o que demonstra uma dependência crítica do algoritmo com o quantitativo de dados para treinamento.

Shrivas *et al.* [1] propuseram um modelo mais robusto de detecção de intrusão através da utilização de uma rede neural em conjunto com uma rede Bayesiana e um extrator de características baseado em ganho (vide figura 4.4).

Figura 4.4: Modelo de IDS proposto por Shrivas *et al.*



Fonte: Akhilesh Kumar Shrivas [1]

O modelo propõe utilizar os algoritmos de *ANN* e *BN* em conjunto para a classificação das amostras e o extrator de características com base em ganho (*GR Feature Selection*) para aprimorar a acurácia e reduzir o número de características necessárias para a classificação, eliminando aquelas consideradas irrelevantes.

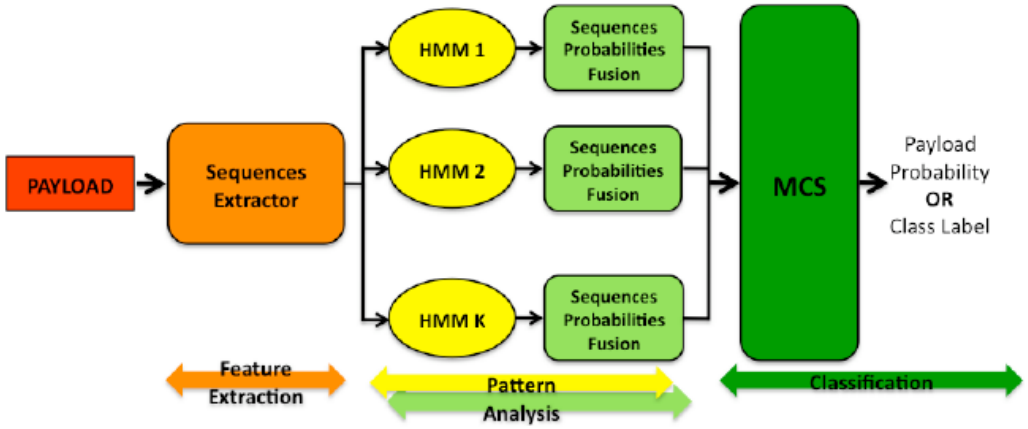
Para mensuração do desempenho, Shrivas *et al.* [1] utilizaram dois conjuntos de dados, o KDD 1999 e sua variante conhecida como NSL-KDD. Para cada conjunto de dados, foram feitas três partições de treinamento/teste com as seguintes porcentagens: 70/30, 80/20, 90/10. Para cada uma dessas partições, o modelo foi executado e a acurácia foi mensurada.

Os resultados alcançados foram muito promissores, com uma acurácia média do modelo, para todos os tipos de ataque, de 99% e 98% para os conjuntos de dados KDD99 e NSL-KDD, respectivamente. O estudo não esclarece se foram feitas as mensurações de resultados falsos (falsos positivos e falsos negativos), portanto, não é possível afirmar se o desempenho alcançado indica a viabilidade de utilização do modelo em cenários reais.

4.7 Estudo 5 - Modelos Ocultos de Markov (HMM)

Ariu *et al.* [10] [3] propuseram, em seus estudos, verificar a utilização de *HMM* para extrair e classificar assinaturas de ataques em aplicações web, como, por exemplo, ataques dos tipos *XSS* e *SQL Injection*. O estudo mencionado descreve um sistema apelidado de HMMPayl, cujo nome deriva-se da atividade de examinar o *payload* de pacotes HTTP na rede utilizando o algoritmo Modelos Ocultos de Markov (*HMM*). O sistema executa o processamento do *payload* dos pacotes HTTP em três etapas, conforme pode ser visto na figura 4.5 a seguir.

Figura 4.5: Sistema HMMPayl proposto por Ariu *et al.*



Fonte: Davide Ariu [10]

Na primeira etapa, Ariu *et al.* [10] propuseram a utilização de um extrator de características para aprimorar a produção de um modelo estatístico robusto e sensível aos detalhes mais importantes dos conjuntos de dados. Na segunda etapa, utilizou-se o algoritmo *HMM* para analisar os dados de *payload* do conjunto de dados de treinamento e então construir modelos internos representativos das classes de ataque conforme padrões de assinatura. Por fim, na terceira etapa é feita a classificação, das amostras de teste, utilizando um classificador múltiplo que tem como entrada o conhecimento formado pelos modelos descritos do algoritmo *HMM*.

O estudo comparou os resultados obtidos com os resultados de dois *IDS* de rede, a saber *McPAD* e *Spectrogram* [32] [41]. O sistema foi testado em diferentes conjuntos de dados, dentre eles o conjunto DARPA, e alcançou resultados de custo computacionais baixos, com um tempo de processamento de 0.035 *ms* por *payload* utilizando uma máquina com recursos limitados. A acurácia foi mensurada baseando-se em uma taxa fixa para falsos positivos e os resultados foram os seguintes:

Falso Positivo	**Acurácia Média**
1%	88,5% +
0,1%	83,5% +
0,01%	50,7% +

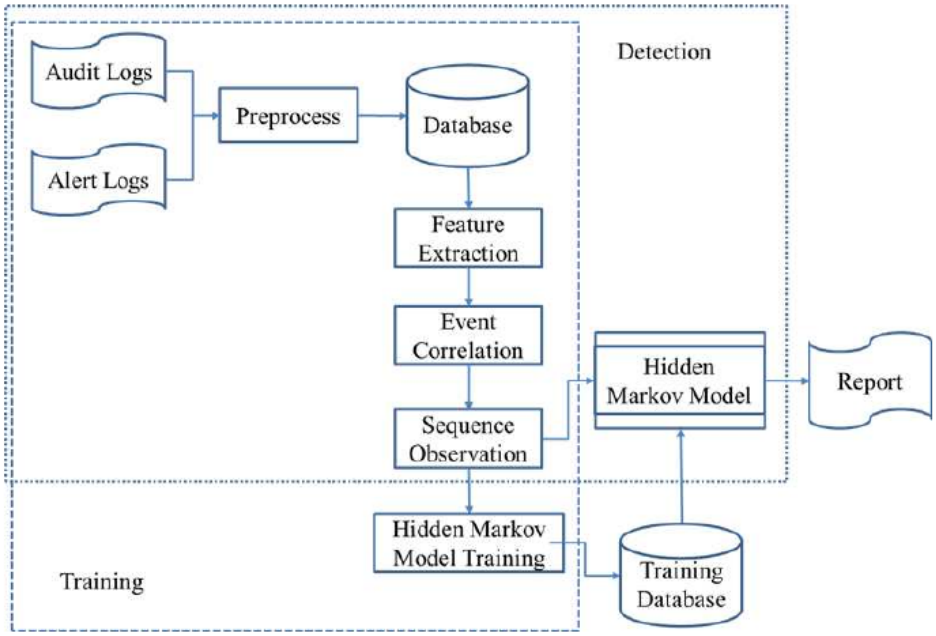
Tabela 4.1: *HMMPayl* - Acurácia com Taxa Fixa de Falsos Positivos

É importante mencionar que os resultados obtidos no estudo supraexposto foram superiores aos alcançados pelos *IDS* utilizados na comparação de desempenho.

Chen *et al.* [7] propuseram um sistema de detecção de anomalias utilizando como classificador um *HMM*. O sistema é composto por um extrator de características,

para aprimorar e extrair as características mais impactantes do conjunto de dados de treinamento, e por dois componentes responsáveis por observar e correlacionar as sequencias de eventos de ataques com as características extraídas, conforme pode ser visto na figura 4.6 a seguir.

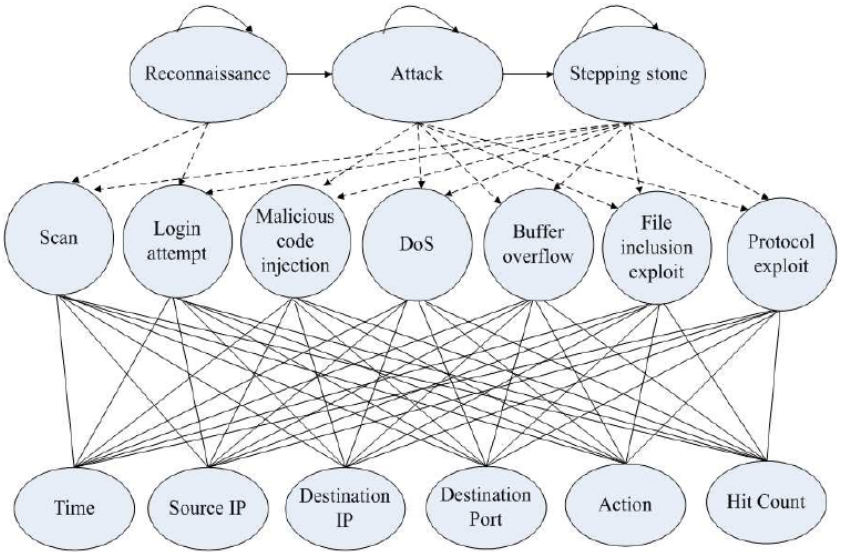
Figura 4.6: Arquitetura do sistema proposto por Chen *et al.*



Fonte: Chia-Mei Chen & Ou [7]

O sistema atua classificando os ataques através de uma sequência de estados pré-definidos, conforme mostrado na figura 4.7.

Figura 4.7: Classificador HMM proposto por Chen *et al.*



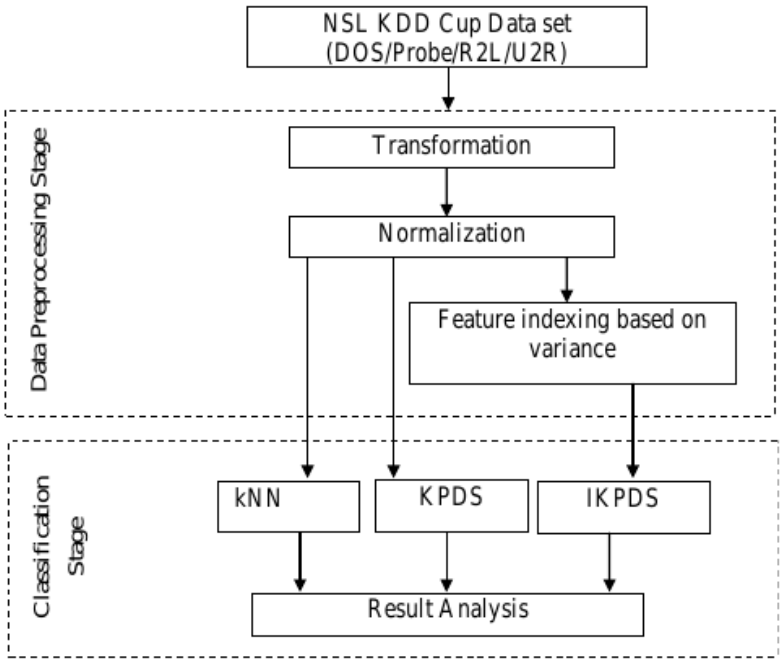
Fonte: Chia-Mei Chen & Ou [7]

O sistema foi testado em um ambiente de rede controlado e em um ambiente de rede real e alcançou uma média de acurácia de detecção de 93,2% nos ambientes de rede real.

4.8 Estudo 6 - KNN

Basaveswara *et al.* [2] propuseram um modelo de classificador utilizando três variações do algoritmo *KNN* em conjunto. Esse classificador integra um sistema maior de detecção de intrusão, vide figura 4.8 abaixo, composto por um transformador e um normalizador de dados responsáveis por transformar os dados categoricos da base em dados númericos e, em seguida, normalizá-los de modo que as características possam ser analisadas e classificadas pelos algoritmos *KNN*. O conjunto de dados utilizado para esta pesquisa foi o NSL-KDD.

Figura 4.8: Arquitetura do sistema proposto por Basaveswara *et al.*



Fonte: Basaveswara & Swathi [2]

Os resultados de acurácia foram extraídos para três valores de k, conforme mostrado na tabela 4.2 a seguir:

	Acurácia		
k	**3**	**5**	**10**
U2R	0.9996	0.9995	0.9994
R2L	0.9988	0.9983	0.9976
Probe	0.9981	0.9976	0.9966
DOS	0.994	0.9991	0.9985

Tabela 4.2: *Acurácia KNN - Basaveswara et al. [2]*

O propóstio desse estudo era encontrar o algoritmo *KNN* mais eficiente e a conclusão do estudo mostra que o modelo proposto é bem eficiente em termos de tempo computacional de processamento e que o algoritmo que mais se destacou dentre as três variações do *KNN* foi o *IKPDS*.

4.9 Estudo 7 - Aprendizado Profundo (Deep Learning)

Apruzzese *et al.* [16] propuseram, em seus estudos, a comparação de desempenho entre algoritmos de aprendizado superficial (*Shallow Learning*) e algoritmos de aprendizado profundo (*Deep Learning*). O estudo propõe testar e comparar os dois tipos de aprendizado para vários domínios de aplicação em cibersegurança, a saber, detecção de intrusão, análise de malware e detecção de spam.

Para detecção de intrusão, Apruzzese *et al.* [16] propôs comparar a utilização do algoritmo *Random Forest*, para aprendizado superficial, e do algoritmo *FNN*, para aprendizado profundo. O conjunto de dados utilizado é um conjunto personalizado para o estudo e contém uma coleção de fluxos benígnos e malígnos de rede da proporção de 100:1, respectivamente.

Os resultados alcançados podem ser visualizados na tabela 4.3 abaixo:

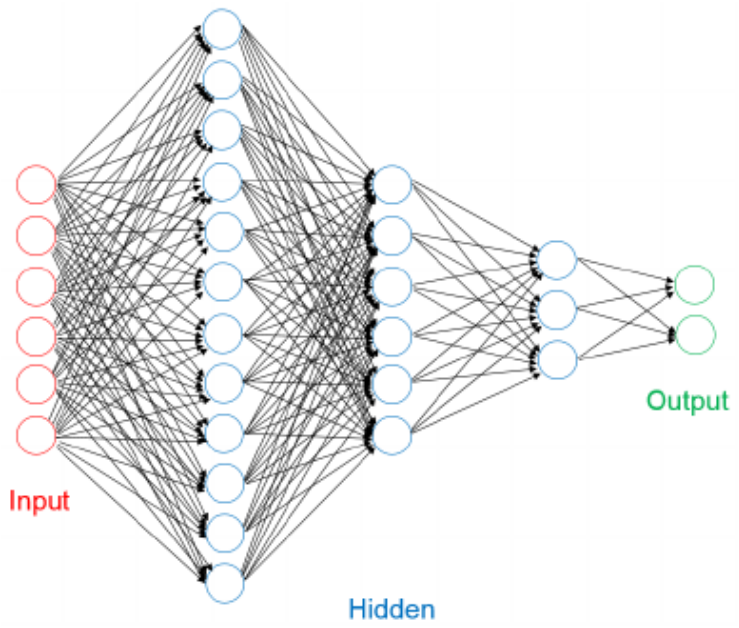
Classificador	**F1-score**	**Precision**	**Recall**
Random Forest (SL)	0.7985	0.8727	0.736
FNN (DL)	0.6085	0.7708	0.5027

Tabela 4.3: Comparação entre classificadores SL e DL

Conforme demonstrado, o algoritmo de *RF* obteve melhor desempenho do que o algoritmo *FNN*. O estudo ressalta, entretanto, que isso não é uma regra. Pela maior complexidade dos algoritmos de *DL*, é de se esperar que os resultados alcançados por estes sejam mais consistentes, o que levaria a melhores resultados, teoricamente; porém, o desempenho de ambos os tipos de aprendizado dependem muito da qualidade da base de dados e da complexidade do domínio avaliado. Conforme conclusão do estudo, não podemos afirmar que um tipo de algoritmo sempre será mais eficiente que outro, é preciso avaliar o conjunto de dados que será utilizado na proposta de solução e a complexidade do domínio onde será aplicado.

Tang *et al.* [38] propuseram a criação de uma rede neural profunda (*DNN*) simples para comparar o desempenho, no domínio da detecção de intrusão, com algoritmos de aprendizado superficial e avaliar a viabilidade do aprendizado profundo para este domínio. A *DNN* está dividida em uma camada de entrada, três camadas ocultas e uma camada de saída. A camada de entrada tem dimensão seis, a camada de saída tem dimensão dois e as camadas ocultas têm dimensões doze, seis e três, respectivamente (vide figura 4.9).

Figura 4.9: *DNN* proposto por Tang *et al.*



Fonte: Tuan A Tang & Ghogho [38]

Para o estudo, foram escolhidas seis características do conjunto de dados NSL-KDD e foram avaliados o desempenho dos algoritmos frente a quatro categorias de tipos de ataques, a saber, *DoS*, *R2L*, *U2R* e *probe*.

Os algoritmos de aprendizado superficial selecionados foram os seguintes:

- *Decision Tree (J48)*
- *Naive Bayes (NB)*
- *NB Tree*
- *Random Forest*
- *Random Tree*
- *Multilayer Perceptron*
- *SVM*

Os resultados de acurácia obtidos, para o cenário proposto pelo estudo, estão expressos na tabela 4.4 a seguir:

Algoritmo	Acurácia (%)
Decision Tree (J48)	81.05
Naive Bayes (NB)	76.56
NB Tree	82.02
Random Forest	80.67
Random Tree	81.59
Multilayer Perceptron	77.41
SVM	69.52
Tang DNN	**75.75**

Tabela 4.4: Comparação de acurácia para diversos algoritmos

O algoritmo de aprendizado profundo obteve resultados de acurácia melhores que alguns algoritmos de aprendizado superficial, porém, piores que outros. Isso demonstra que a escolha do algoritmo pode variar conforme o domínio, sendo importante a análise dos cenários e das variáveis que o compõe antes da escolha do algoritmo de atuação final.

Devaraju *et al.* [33] propuseram comparar e analisar cinco tipos de rede neural profunda para detecção de assinaturas maliciosas na rede. O estudo em questão busca analisar o desempenho das seguintes redes neurais profundas:

- *FNN - Feed Forward Neural Network*
- *ENN - Elman Neural Network*
- *GRNN - Generalized Regression Neural Network*
- *PNN - Probabilistic Neural Network*
- *RBNN - Radial Basis Neural Network*

Para analisar o desempenho, Devaraju *et al.* [33] utilizaram o conjunto de dados KDD 1999 de duas formas, primeiro foi feita a análise de acurácia dos classificadores de rede neural profunda com o conjunto completo, contendo 41 características; posteriormente, utilizou-se um extrator de características para reduzir a dimensionalidade do conjunto de dados às características consideradas mais importantes pelo estudo, gerando um novo conjunto contendo 13 características.

Os resultados alcançados, para as duas etapas supracitadas, estão expostos nas tabelas 4.5 e 4.6, a seguir:

Algoritmo	Acurácia (%)
FNN	79.49
ENN	78.1
GRNN	58.74
PNN	85.56
RBNN	83.51

Tabela 4.5: Acurácia dos Algoritmos de Aprendizado Profundo (41 Características)

Algoritmo	Acurácia (%)
FNN	80.64
ENN	81.38
GRNN	95.32
PNN	96.23
RBNN	72.68

Tabela 4.6: Acurácia dos Algoritmos de Aprendizado Profundo (13 Características)

4.10 Estudo 8 - KMeans

Em seu trabalho, Ranjan *et al.*[31], propuseram a criação de uma variante do algoritmo K-means para detecção de anomalias em um ambiente de rede. O propósito para a criação dessa variante foi devido ao fato de que o algoritmo k-means possui certas desvantagens que prejudicam a acurácia de detecção quando usado em cenários voláteis como o trafego de uma rede.

Essas desvantagens são a dependência inicial muito forte dos centroides dos dados, a dependência da definição ótima do número de *clusters* e a degeneração do algoritmo, que significa dizer que o resultado final pode ter um número de *clusters* menor do que o esperado para a solução.

A variante proposta por Ranjan *et al.*[31] integra uma técnica de particionamento de *clusters* utilizando o algoritmo *apriori*. O conjunto de dados utilizado para o treinamento e teste do algoritmo proposto foi o KDD1999.

Os resultados foram comparados com outros algoritmos variantes do k-means e com o próprio algorimo k-means e podem ser vistos na tabela 4.7 abaixo:

Métricas	K-Means	FCM	Y-Means	Algoritmo Proposto
Precisão (%)	82.3	84.6	86.3	91.2
Acurácia (%)	77.25	82.13	87.15	96.38
Taxa de Resultados Falsos (%)	5.2	4.2	3.9	3.2

Tabela 4.7: Resultado do Algoritmo de Clusterização de Ranjan *et al.*

Capítulo 5

Conclusão

De acordo com os estudos analisados, chega-se à conclusão que utilizando algoritmos de aprendizado de máquina é possível aumentar as taxas de detecção do sistema de detecção de intrusão e manter em nível controlado a incidência de alarmes falsos sem comprometer o custo computacional.

Entretanto, pelos estudos, é notável que não existe um algoritmo que seja melhor para a tarefa de detecção de intrusão, sendo que o ganho aliado à utilização desses algoritmos está diretamente relacionado à qualidade dos conjuntos de dados utilizados para o treinamento do algoritmo e à complexidade do domínio.

As seguintes conclusões também foram extraídas dos estudos:

- A utilização de um único algoritmo de aprendizado de máquina para efetuar a classificação na solução se mostrou menos eficiente, alcançando piores valores de acurácia, do que a combinação de vários algoritmos de aprendizado de máquina em todo o processo de análise da solução.

- As variantes dos algoritmos de aprendizado de máquina, em estado fundamental, obtiveram melhores resultados mostrando que a adequação do algoritmo ao domínio de utilização tende a gerar melhores resultados.

- Algoritmos de aprendizado profundo não são sinônimos de melhores resultados sempre, como muitos acreditam. Os estudos mostraram que a maior complexidade dos algoritmos de aprendizado profundo nem sempre se traduz em melhores análises, e que seu desempenho também depende muito do domínio e da qualidade dos conjuntos de dados.

Conclui-se, portanto, que não só existe a viabilidade da utilização de algoritmos de aprendizado de máquina no processo de detecção de intrusão, mas que há de se tornar

uma necessidade das soluções de detecção de intrusão tendo em vista que os ciber ataques já estão em processo de utilização de aprendizado de máquina para criação de variantes de artefatos maliciosos, gerando alta volatilidade no processo de ataque, e para o treinamento de redes de computadores (*botnets*) no processo de infecção de redes e hosts (utilizando *phishing* e outras técnicas de intrusão).

Referências Bibliográficas

[1] Akhilesh Kumar Shrivas, Amit Kumar Dewangan. 2014. An ensemble model for classification of attacks with feature selection based on KDD-99 & NSL-KDD data set. *International Journal of Computer Applications.*

[2] Basaveswara, B., & Swathi, K. 2017. Fast kNN Classifiers for Network Intrusion Detection System. *Indian Journal of Science and Technology.*

[3] Buczak, Anna L., & Guven, Erhan. 2016. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.*

[4] C. Torrano-Gimenez, A. Perez-Villegas. 2009. A Self-Learning Anomaly-Based Web Application Firewall. *2nd International Workshop in Computational Intelligence in Security for Information Systems (CISIS 09).*

[5] Cannady, J. 1998. Artificial neural networks for misuse detection. *Nat. Inf. Syst. Secur. Conf.*

[6] Chan, Philip K., Mahoney, Matthew V., & Arshad, Muhammad H. 2003. *A Machine Learning Approach to Anomaly Detection.* Tech. rept. Florida Institute of Technology.

[7] Chia-Mei Chen, Dah-Jyh Guan, Yu-Zhi Huang, & Ou, Ya-Hui. 2016. ANOMALY NETWORK INTRUSION DETECTION USING HIDDEN MARKOV MODEL. *International Journal of Innovative Computing, Information and Control.*

[8] Chibuzor john Ugochukwu, E. O Bennet. 2018. An Intrusion Detection System Using Machine Learning Algorithm. *International Journal of Computer Science and Mathematical Theory.*

[9] Damshenas, Mohsen, Dehghantanha, Ali, & Mahmoud, Ramlan. 2013. A SURVEY ON MALWARE PROPAGATION, ANALYSIS, AND DETECTION. *International Journal of Cyber-Security and Digital Forensics.*

[10] Davide Ariu, Roberto Tronci, Giorgio Giacinto. 2011. HMMPayl: an Intrusion Detection System based on Hidden Markov Models. *Elsevier Computers and Security.*

[11] Dong, Bo, & Wang, Xue. 2016. Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection. *8th IEEE International Conference on Communication Software and Networks.*

[12] Dua, Sumeet, & Du, Xian. 2011. *Data Mining and Machine Learning in Cybersecurity.* CRC Press.

[13] Elike Hodo, Xavier Bellekens, Andrew Hamilton Christos Tachtatzis, & Atkinson, Robert. 2017. *Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey.* Tech. rept. University of Strathclyde.

[14] Farah Jemili, Montassar Zaghdoud, Mohamed Ben Ahmed. 2007. A Framework for an Adaptive Intrusion Detection System using Bayesian Network. *Intelligence and Security Informatics IEEE.*

[15] Gilmore, Colin, & Haydaman, Jason. 2016. Anomaly Detection and Machine Learning Methods for Network Intrusion Detection: an Industrially Focused Literature Review. *Int'l Conf. Security and Management.*

[16] Giovanni Apruzzese, Luca Ferretti, Michele Colajanni Alessandro Guido, & Marchetti, Mirco. 2018. On the Effectiveness of Machine and Deep Learning for Cyber Security. *10th International Conference on Cyber Conflict.*

[17] Kruegel, C., & Toth, T. 2013. Using decision trees to improve signature-based intrusion detection. *Int. Workshop Recent Adv. Intrusion Detect.*

[18] Kumar, Dhruba, & Kumar, Jugal. 2014. *Network Anomaly Detection a Machine Learning Perspective.* CRC Press.

[19] L. Bilge, E. Kirda, C.Kruegel, & M.Balduzzi. 2011. EXPOSURE: Finding malicious domains using passive DNS analysis. *Annu. Newt. Distrib. Syst. Secur. Conf.*

[20] L. Bilge, S. Sen, D. Balzarotti E.Kirda C. Kruegel. 2014. EXPOSURE: A passive DNS Analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.*

[21] Lippmann, R. P., & Cunningham, R. K. 2000. Improving intrusion detection performance using keyword selection and neural networks. *Comput. Netw.*

[22] Loon, Ronald Van. 2018. *Understanding supervised, unsupervised, and reinforcement learning.*

[23] Ltd., BrandIdea. 2018. *K-Means and X-Means Clustering.*

[24] M. Tavallaee, E. Bagheri, W. Lu, & Ghorbani, A. 2009. A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications.*

[25] Mangalampalli, Ashish, & Pudi, Vikram. 2009. Fuzzy association rule mining algorithm for fast and efficient performance on very large datasets. *IEEE International Conference on Fuzzy Systems.*

[26] Mazyar Mohammadi Lisehroodi, Zaiton Muda, & Yassin, Warusia. 2013. A HYBRID FRAMEWORK BASED ON NEURAL NETWORK MLP AND K-MEANS CLUSTERING FOR INTRUSION DETECTION SYSTEM. *4th International Conference on Computing and Informatics.*

[27] Monowar Bhuyan, D. K. Bhattacharyya, & Kalita, J. K. 2014. Network Anomaly Detection: Methods, Systems and Tools. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, VOL. 16, NO. 1.*

[28] Namita Parati, Sumalatha Potteti. 2015. Intelligent Intrusion Detection System using SVM and Genetic Algorithm (SVM-GA). *International Journal of Science and Applied Information Technology (IJSAIT).*

[29] Nutan Farah, Abdur Rahman Md. Avishek Khan, Musharrat Rafnim Faisal Muhammad, & Farid, Dewan Md. 2015. Application of Machine Learning Approaches in Intrusion Detection System: A Survey. *International Journal of Advanced Research in Artificial Intelligence, Vol. 4, No.3.*

[30] Ramos, Alex L., & dos Santos, CÃcero N. 2015. *Combinando Algoritmos de ClassificaÃ§Ã£o para DetecÃ§Ã£o de IntrusÃ£o em Redes de Computadores.* M.Phil. thesis, Universidade de Fortaleza.

[31] Ranjan, Ravi, & Sahoo, G. 2014. A NEW CLUSTERING APPROACH FOR ANOMALY INTRUSION DETECTION. *International Journal of Data Mining and Knowledge Management Process.*

[32] R.Perdisci, D. Ariu, P. Fogla G. Giacinto. 2009. McPAD: a multiple classifier system for accurate payload-based anomaly detection. *Computer Networks.*

[33] S. Devaraju, S. Ramakrishnan. 2013. DETECTION OF ACCURACY FOR INTRUSION DETECTION SYSTEM USING NEURAL NETWORK CLASSIFIER. *International Journal of Emerging Technology and Advanced Engineering.*

[34] Shin, Seongjun, Lee, Seungmin, Kim, Hyunwoo, & Kim, Sehun. 2013. Advanced Probabilistic Approach for Network Intrusion Forecasting and Detection. *Expert Syst. Appl.*

[35] Shon, Taeshik, & Moon, Jongsub. 2007. A hybrid machine learning approach to network anomaly detection. *Information Sciences.*

[36] Siva S. Sivatha Sindhu, S. Geetha, A. Kannan. 2012. Decision tree based light weight intrusion detection using a wrapper approach. *ELSEVIER - Expert Systems with Applications 39 129141.*

[37] Stolfo, S. J. 1999. *KDD Cup 1999 Data Set.*

[38] Tuan A Tang, Lotfi Mhamdi, Des McLernon Syed Ali Raza Zaidi, & Ghogho, Mounir. 2016. Deep Learning Approach for Network Intrusion Detection in Software Defined Networking. *International Conference on Wireless Networks and Mobile Communications (WINCOM).*

[39] Volden, Hnerik Hivand. 2016. *Anomaly detection using machine learning techniques.* M.Phil. thesis, University of Oslo.

[40] W. Meng, W. Li, L. Kwok. 2015. Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection. *SECURITY AND COMMUNICATION NETWORKS.*

[41] Y. Song, A. D. Keromytis, S. J. Stolfo. 2009. Spectogram: A mixture-of-markov-chains model for anomaly detection in web traffic. *The Internet Society.*

[42] Yavanoglu, Ozlem, & Aydos, Murat. 2017. A Review on Cyber Security Datasets for Machine Learning Algorithms. *IEEE International Conference on Big Data, Symposium on Data Analytics for Advanced Manufacturing.*

[43] Zamani, Mahdi. 2013. *Machine Learning Techniques for Intrusion Detection.* Tech. rept. University of New Mexico.