UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Lucas de Biaggi Januário

Máquina de elementos finitos para seleção de características de anomalias em redes de computadores

São José do Rio Preto, São Paulo, Brasil

2021

Lucas de Biaggi Januário

Máquina de elementos finitos para seleção de características de anomalias em redes de computadores

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Universidade Estadual Paulista "JÚLIO DE MESQUITA FILHO"

Instituto de Biociências, Letras e Ciências Exatas

Programa de Pós-Graduação em Ciência da Computação

Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

São José do Rio Preto, São Paulo, Brasil

2021

J35m

Januário, Lucas de Biaggi
Máquina de elementos finitos para seleção de características de anomalias em redes de computadores / Lucas de Biaggi Januário. -- Bauru, 2021
96 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru
Orientador: Kelton Augusto Pontara da Costa

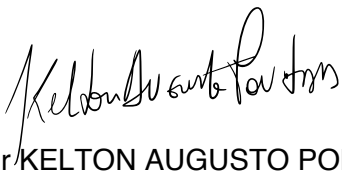
1. Seleção de Característica. 2. Redes de Computadores. 3. Identificação de anomalias. 4. Método de elementos finitos. 5. FEMa. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE LUCAS DE BIAGGI JANUÁRIO, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 03 dias do mês de dezembro do ano de 2021, às 16:00 horas, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de LUCAS DE BIAGGI JANUÁRIO, intitulada **Máquina de elementos finitos para seleção de caractersticas de anomalias em redes de computadores**. A Comissão Examinadora foi constituida pelos seguintes membros: Professor Doutor KELTON AUGUSTO PONTARA DA COSTA (Orientador(a) - Participação Virtual) do(a) Professor Pleno I / Faculdade de Tecnologia de Bauru, Prof. Dr. DANILLO ROBERTO PEREIRA (Participação Virtual) do(a) Faculdade de Informática de Presidente Prudente / Universidade do Oeste Paulista-UNOESTE, Prof. Dr. LEANDRO APARECIDO PASSOS JUNIOR (Participação Virtual) do(a) Pós-doutorando do Depto. de Computação / FC/Bauru - Unesp. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final:_ _ _ _ APROVADO _ _ _ _ _ _ _ _ _ _ _ _ . Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.



Professor Doutor KELTON AUGUSTO PONTARA DA COSTA

Lucas de Biaggi Januário

Máquina de elementos finitos para seleção de características de anomalias em redes de computadores

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

São José do Rio Preto, São Paulo, Brasil, 03 de Dezembro de 2021:

Prof. Dr. Kelton Augusto Pontara da Costa
Orientador

Prof. Dr. Danillo Roberto Pereira
Convidado 1

Prof. Dr. Leandro Passos Junior
Convidado 2

São José do Rio Preto, São Paulo, Brasil
2021

Agradecimentos

Agradeço a todos os professores por me proporcionarem o conhecimento não apenas técnico, por tanto que se dedicaram a minha pessoa, não somente por terem me ensinado, mas por ajudarem a evoluir na forma de aprender. A esta universidade, ao programa de pós-graduação e a sua administração que possibilitaram um novo horizonte. Agradeço ao meu orientador, Prof. Dr. Kelton Augusto Pontara da Costa por acompanhar-me neste projeto.

À minha família, pelo incentivo e apoio incondicional. À Leticia Gaiotte, pessoa com quem amo não apenas partilhar a vida, mas tudo. Graças a você, me sinto capaz de superar qualquer desafio. Meus agradecimentos aos meus queridos amigos Mauricio, Thiago e Tomaz pelas risadas, palavras amigas e conselhos. A todos os meus outros amigos que direta ou indiretamente fizeram parte de minha formação, o meu muito obrigado.

Agradeço aos professores participantes da banca examinadora que dividiram comigo este momento tão importante e esperado.

Resumo

Identificar anomalias tornou-se uma das principais estratégias para procedimentos de segurança e proteção em redes de computadores. No entanto, é uma tarefa desafiadora para os seres humanos, pois requer a avaliação de um grande volume de dados diários para descobrir um comportamento inesperado. Nesse contexto, os métodos baseados em aprendizado de máquina surgem como uma solução elegante para ajudar a identificar esses comportamentos. Além disso, técnicas inteligentes para remover informações irrelevantes de conjuntos de dados, ou seja, selecionar caracaterísticas, podem aumentar a eficiência e reduzir o tempo de processamento. Portanto, esta dissertação propõe uma nova abordagem de seleção de recursos chamada *Finite Element Machine Feature Selection* (FEMa-FS). O método utiliza elementos finitos, como a função inversa de Shepard, para identificar as características mais representativas em conjuntos de dados. Finalmente, o FEMa-FS seleciona as características mais relevantes para identificar anomalias no tráfego da rede, que são posteriormente empregadas para alimentar o classificador Optimum-Path-Forest. O método provou sua eficiência na redução de informações irrelevantes e pode aumentar a precisão da classificação em até 2%.

Palavras-chaves: Aprendizado de máquina. Seleção de Característica. Redes de Computadores. Identificação de anomalias. Método de elementos finitos. FEMa.

Abstract

Identifying anomalies has become one of the primary strategies towards security and protection procedures in computer networks. However, such an approach denotes a challenging task for human beings since it requires assessing a large volume of daily data to uncover unexpected behavior. In this context, machine learning-based methods emerge as an elegant solution to help to identify such unexpected behaviors. Further, intelligent techniques to remove irrelevant information from datasets, namely feature selection, can increase efficiency and reduce the processing time. Therefore, this dissertation proposes a novel feature selection approach called Finite Element Machine Feature Selection (FEMa-FS). The method uses finite elements, such as Shepard's inverse function, to identify the most representative characteristics in datasets. Finally, FEMa-FS selects the most relevant features to identify anomalies in network traffic, which are further employed to feed the Optimum Path Forest classifier. The method has proved its efficiency in reducing irrelevant information and could increase classification accuracy up to 2%.

Keywords: Machine Learning. Feature Selection. Computer Networks. Anomaly Identification. Finite Elements Method. FEMa.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ML - Aprendizado de Máquina

FEMa - Máquina de elementos finitos

OPF - Optimum Path Florest

DoS - Negação de serviço

DDoS - Negação de serviço distribuída

U2R - Acesso ao usuário administrador

R2L - Ataque remote para rede local

NB - Naïve Bayer

SVM - Máquina de vetores de suporte

ANN - Rede de Neurônios Artificiais

PV - Positivo verdadeiro

FN - Falso negativo

NV - Negativo verdadeiro

FP - Falso positivo

MI - Informação mutuá

MSPCA - *Multi-scale principal component analysis*

RF - Floresta aleatória

ARM - Regra de associação de mineração

EM - *Expectation-Maximisation clustering*

GHSOM-pr - *Growing Hierarchical Self-Organizing Maps*

CVM - Core Vector Machine

PSO - Otimização de enxame de partículas

GWO - Otimização do lobo cinzento

FFA - Otimização de libélulas

GA - Algoritmo Genético

Sumário

1 Introdução

O risco de atividades suspeitas ou até mesmo controversas dentro de uma rede de computadores é uma das principais preocupações dos profissionais de segurança da computação, dado que uma de suas responsabilidades é a de identificar tais atividades e reconhecer tentativas maliciosas de acesso não autorizado ou ilegal (GARG; BATRA, 2018). Apesar dos esforços desses profissionais, o problema exige resposta instantânea devido às suas consequências imprevisíveis. Essa necessidade atraiu a atenção de muitos pesquisadores para o desenvolvimento de planos de ação inteligentes e autônomos (MATEL; SISON; MEDINA, 2019). Dentre essas abordagens, os métodos baseados em aprendizado de máquina (ML) obtiveram notória popularidade, visto que pode ser possível aprender o comportamento "normal" a partir dos dados e apontar comportamentos inadequados, fornecendo uma excelente opção para analisar tais fenômenos.

As estratégias de aprendizado de máquina apresentaram uma evolução considerável nos últimos anos devido à demanda por alternativas e ao crescente poder computacional proporcionado pela progressão do equipamento. Entre essas técnicas, pode-se referir-se à Máquina de Elementos Finitos (FEMa) (PEREIRA et al., 2020), que é uma abordagem sem parâmetros baseada em uma análise de método numérico para encontrar soluções aproximadas, chamadas método de elementos finitos (FEM) (ZIENKIEWICZ; TAYLOR; ZHU, 2013), e que praticamente dispensa a etapa de treinamento. Em suma, o FEMa divide um problema em equações mais simples usando funções básicas para construir uma variedade. O processo é realizado interpolando um conjunto de pontos (elementos) e, assim, criando um padrão de classificação.

Apesar dos avanços mencionados acima, as abordagens com aprendizado de máquina ainda enfrentam alguns desafios inerentes à segurança de rede devido à sua dinâmica intrínseca, que se torna cada vez mais complexa à medida que os sistemas computacionais evoluem (FALCÃO et al., 2019). Dentre algumas alternativas projetadas para aliviar a carga resultante de tais complexidades, pode-se referir-se às técnicas de seleção de características, que podem extrair as características mais relevantes dos dados e descartar informações redundantes ou irrelevantes. Normalmente, tais métodos geram um conjunto de dados mais compacto e representativo, que são posteriormente empregados para alimentar algum algoritmo de ML, fornecendo assim uma classificação mais eficiente e assertiva (FARIS et al., 2019).

Nesse contexto, muitos trabalhos abordaram o problema de seleção de características utilizando técnicas de otimização metaheurística. (RODRIGUES et al., 2015),Rodrigues et al. (2015), por exemplo, propôs uma versão binária do Algoritmo de polinização

de flores (YANG, 2012; RODRIGUES et al., 2020) para seleção de características, enquanto Pereira et al. (2019) ofereceu uma abordagem semelhante usando JADE (ZHANG; SANDERSON, 2009). Em ambos os trabalhos, os autores empregaram uma abordagogem baseado em grafos, o classificador Optimum-Path Forest (OPF) (PAPA; FALCÃO; SUZUKI, 2009) para avaliar o desempenho dos métodos propostos. Outros trabalhos também obtiveram resultados satisfatórios para a tarefa usando abordagens distintas, como análise de variância (ANOVA) (AHSAN et al., 2021) e chi^2 (KASONGO; SUN, 2020).

Apesar do sucesso observado nos trabalhos citados, a maioria dos procedimentos ainda apresenta desvantagens relacionadas à sua natureza estocástica e aos desafios de evitar ótimos locais. Portanto, foi proposto uma nova abordagem de seleção de recursos que aproveita os atributos do FEMa para selecionar o melhor conjunto de recursos, denominado FEMa-FS.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver uma técnica baseada em elementos finitos para seleção de característica supervisionada para identificação de anomalias no contexto de redes de computadores que não necessite de uma etapa formal de treinamento para a sua execução.

1.1.2 Objetivos Específicos

Para facilitar a análise do desenvolvimento do trabalho e a validação da proposta, foram definidos os seguintes objetivos específicos, descritos a seguir:

a) Estudar sobre técnicas de aprendizado de máquina supervisionada e não supervisionada para entendimento dos possíveis desafios e aprender sobre as métricas de avaliações;

b) Estudar as técnicas de seleções de características, visando identificar os caminhos para futuras otimizações;

c) Estudar sobre o método de elementos finitos e propor um modelo para seleção de características;

d) Aprimorar o conhecimento sobre as particularidades das bases de dados NSL-KDD, ICSXTor2016 e UNSW-NB15;

e) Propor um modelo de seleção de características que utilize de elementos finitos para a realização da seleção de características de anomalias;

f) Aplicar sempre que possível as ações para a redução do tempo de processamento;

g) Testar e validar o modelo proposto, nas bases de dados escolhidas;

h) Avaliar os resultados obtidos.

1.2 Estrutura da Dissertação

O documento encontra-se organizado da seguinte maneira:

- No capítulo 2 são apresentados os conceitos de anomalias, algoritmos de aprendizado de máquina, seleção de características, sobre os elementos finitos e o algoritmo de classificação FEMa e a proposta de seleção de característica baseada nele.
- Enquanto na seção 2.5 encontra-se uma revisão dos últimos cinco anos de práticas realizadas para seleção de características no contexto de anomalias em redes de computadores, assim como as principais técnicas e algoritmos de ML e seleção de características.
- A metodologia aplicada na experimentação e os critérios de avaliação são apresentadas no capítulo 3;
- A discussão sobre os resultados encontrados na execução dos passos propostos na metodologia estão no capítulo 4.
- A sintetização dos resultados obtidos está no capítulo 5 com propostas de futuros trabalhos.

2 Fundamentação

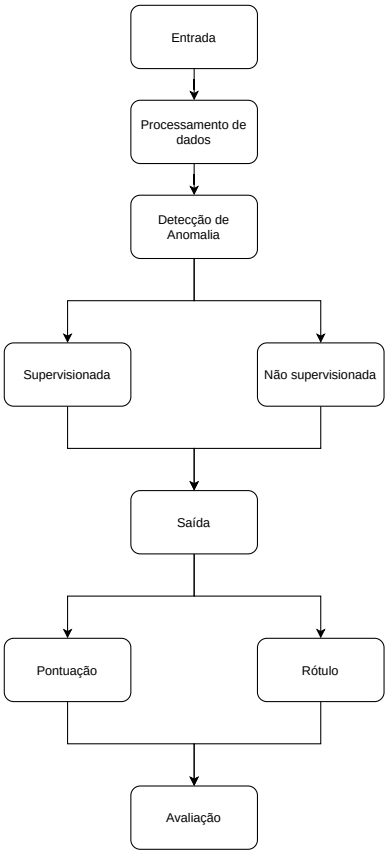
2.1 Identificação de anomalias em redes de computadores

Ahmed, Naser Mahmood e Hu (2016) e Alabi e Yurtkan (2018) definem anomalias como valores encontrados em um conjunto de dados com comportamento diferente do padrão esperado. Elas podem ser categorizadas da seguinte maneira:

- **Pontual**: quando um ponto em particular do conjunto se comporta fora do considerado normal para o *dataset*, por exemplo, se uma pessoa utiliza todo dia cinco litros de combustível, caso um dia qualquer ela utilizar cinquenta litros, tal dado passou a ser uma anomalia.
- **Contextual**: quando o comportamento do dado se comporta de maneira diferente em um particular contexto, por exemplo, uma pessoa que compra no cartão de crédito em uma data festiva, é esperado um gasto alto, então essa ação não poderia ser considerada uma anomalia, entretanto caso ocorra em um período não previsto, ela torna-se uma.
- **Coletiva**: quando um grupo similar do conjunto se comporta diferente do restante.

Os autores Ahmed, Naser Mahmood e Hu (2016) disponibilizaram um modelo genérico proposto para um detector de anomalias utilizando ML para a avaliação, representado na Figura 1, em que o processamento ocorre de acordo com o tipo de entrada, logo o aprendizado pode ser supervisionado ou não supervisionado.

Figura 1 – Modelo de solução para detecção de anomalias.



Fonte: Adaptado de Ahmed, Naser Mahmood e Hu (2016).

Uma ataque cibernético também pode ser considerado uma anomalia em redes de computadores. Serão apresentadas a seguir algumas das principais anomalias no contexto de redes de computadores encontradas na literatura.

a) Negação de Serviço (DoS): o usuário mal-intencionado deixará um serviço indisponível ou com acesso congestionado. Para isso ele tentará utilizar todos os recursos possíveis para realizar a sobrecarga. Caso sejam múltiplos recursos, transforma-se em um ataque distribuído (DDoS); gerando indisponibilidade e afetando a experiência dos outros usuários. (AHMED; NASER MAHMOOD; HU, 2016; ALABI; YURTKAN, 2018; ALKASASSBEH, 2017; BHATIA et al., 2019; GARG et al., 2019; GOTTWALT; CHANG; DILLON, 2019; HAMAMOTO et al., 2018; KHAMMASSI; KRICHEN, 2017; KHAN et al., 2018; MAZINI; SHIRAZI; MAHDAVI, 2019)

b) Acesso ao usuário administrador (U2R): o atacante ilicitamente obtém credenciais válidas com acesso administrativo, possibilitando assim fazer o que permite exercer a função de um administrador, podendo dessa maneira instalar programas para monitoramento e até mesmo roubo de dados. (AHMED; NASER MAHMOOD; HU, 2016; ALABI; YURTKAN, 2018; GARG et al., 2019; KHAMMASSI; KRICHEN, 2017; MAZINI; SHIRAZI; MAHDAVI, 2019)

c) Ataques remotos para rede local (R2L): o atacante pode coletar uma ou mais credenciais válidas para acesso no sistema remoto alvo. (AHMED; NASER MAHMOOD; HU, 2016; ALABI; YURTKAN, 2018; GARG et al., 2019; KHAMMASSI; KRICHEN, 2017; MAZINI; SHIRAZI; MAHDAVI, 2019)

d) Sondagem: utilizada para coletar informações sobre a infraestrutura e arquitetura da rede para decisões futuras e métodos de ataque sem ferramentas administrativas. (AHMED; NASER MAHMOOD; HU, 2016; ALABI; YURTKAN, 2018; GARG et al., 2019; KHAMMASSI; KRICHEN, 2017; MAZINI; SHIRAZI; MAHDAVI, 2019)

e) Enumeração (Infiltração): utiliza da mesma abordagem que a sondagem, recorrendo a ferramentas administrativas. Também pode ser encontrada como infiltração.(ALABI; YURTKAN, 2018; MAZINI; SHIRAZI; MAHDAVI, 2019)

f) Força Bruta: consiste em tentar todas as combinações possíveis de uma interface de um serviço, como por exemplo acesso remoto. Se tal interface não possuir mecanismo de proteção, em questão de tempo o atacante pode obter acesso do administrador ou algum outro usuário válido.(ALABI; YURTKAN, 2018; MAZINI; SHIRAZI; MAHDAVI, 2019).

g) *Botnet*: são redes de computadores infectados por programas maliciosos que executam comandos solicitados através de uma central, tais redes são bastante utilizadas para ataques do tipo de DDoS.(ABRAHAM et al., 2018; ALABI; YURTKAN, 2018; MAZINI; SHIRAZI; MAHDAVI, 2019).

h) *Backdoor*: um programa que permite ignorar a segurança do sistema computacional. Possuem mecanismos para se tornar indetectáveis e podem ser utilizados para para acessar um computador ou coletar informações sobre ele. (GOTTWALT; CHANG; DILLON, 2019)

i) *Exploit*: exploração de uma vulnerabilidade de software conhecida.(MOUSTAFA; SLAY, 2017)

j) Reconhecimento (*Reconnaissance*): possui objetivo de fazer o "reconhecimento" da rede alvejada, (FERNANDES et al., 2019).

2.1.1 Conjunto de *datasets* e anomalias

Como uma anomalia em rede de computadores pode ser utilizada para representar diversas situações fora do padrão, na tabela 1 encontra-se de maneira resumida a abrangência que o termo anomalia reflete em cada trabalho. Ainda na tabela 1 observa-se o grande interesse em atividades que afetam a disponibilidade de um serviço, tais como

DoS, DDoS e *botnets*. Segundo Mazini, Shirazi e Mahdavi (2019) os principais *datasets* utilizados para testes são as KDD-CUP 99 e NSL-KDD.

Os autores Mishra et al. (2019) e Ahmed, Naser Mahmood e Hu (2016) identificaram que os *datasets* públicos disponíveis para teste possuem características distintas e consecutivamente, os resultados podem variar afetando as identificações de anomalias ou até mesmo criar uma forma de viés para o teste. Os autores presentes na tabela 1 utilizam as anomalias para medir a eficiência de detecção de algoritmos de ML.

Tabela 1 – Anomalias utilizadas pelos autores

Autor	*Dataset*	Para o autor anomalia representa
Ahmed, Naser Mahmood e Hu (2016)	NSL-KDD	DoS, U2R, R2L e Sondagem
Chen et al. (2016)	DARPA	DoS
Khammassi e Krichen (2017)	KDD'99, UNSW-NB15	DoS, U2R, R2L e Sondagem
Abraham et al. (2018)	-	*botnets*
Alabi e Yurtkan (2018)	UNB,IDS	DoS, U2R, R2L, Sondagem, Enumeração, Forca Bruta e *Botnets*
Hamamoto et al. (2018)	KDD'99	DoS
Fernández Maimó et al. (2018)	CTU	*Botnets*
Khan et al. (2018)	NSL-KDD	DoS
Garg et al. (2019)	KDD'99	DoS, U2R, R2L e Sondagem

Gottwalt, Chang e Dillon (2019)	UNSW-NB15	DoS e *Backdoor*
Kasongo e Sun (2019)	NSL-KDD	-
k Mazini, Shirazi e Mahdavi (2019)	NSL-KDD	DoS, U2R, R2L, Sodagem, Enumeração, Força Bruta e *Botnets*
Mishra et al. (2019)	-	DoS, U2R, R2L, Sodagem, Enumeração, Força Bruta, *Backdoor* e *Botnets*
Biondi et al. (2019)	Próprio	-
Bhatia et al. (2019)	-	DDoS
Nawir et al. (2019)	UNSW-NB15	-
Palmieri (2019)	ISCX-UNB	*Botnets*
Zhou et al. (2020)	KDD'99, AWID, CIC-IDS2017	-
Liu, Ci e Liu (2020)	UNM_sendmail UNM_live_lpr	-

2.2 Aprendizado de Máquina

Aprendizado de Máquina é uma das linhas de pesquisas em Inteligência Artificial cujo seu propósito é encontrar padrões por de meios matemáticos e estatísticos sobre um conjunto de dados, seja através de agrupamento (*clustering*), regressão ou classificação (SAMMUT; WEBB, 2017). Entre as principais alternativas de algoritmos encontra-se

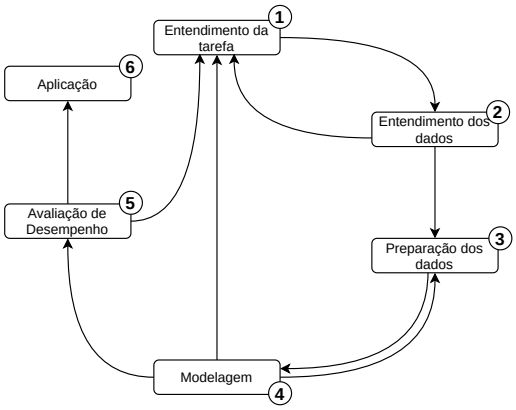
duas grandes classes: supervisionadas e não supervisionadas.

Tanto os algoritmos supervisionados quanto não supervisionados têm objetivos distintos de realizar a predição sobre o valor da entrada. A maior diferença está no fato de que o supervisionado ante de iniciar o treinamento recebe o domínio que irá trabalhar enquanto o não supervisionado não recebe conhecimento sobre esse domínio. (XUE et al., 2019).

Sammut e Webb (2017) e F.Y et al. (2017) definem que um algoritmo de ML quando utilizado para classificação, precisará treinar com uma pequena porção do conjunto e será transformado em treinamento, e o restante será utilizado como teste, permitindo assim observar o desempenho alcançado pelo algoritmo.

O processo do desenvolvimento e utilização de um algoritmo pode ser observado na Figura 2. Pode-se observar a importância que existe no entendimento da tarefa, dos dados e efetividade do modelo proposto.

Figura 2 – Visão geral de ML. Como primeiro passos temos o entendimento da utilização, já em um segundo passo é realizado a análise dos dados para identificar a necessidade de tratativas que serão realizadas, posteriormente é gerado um modelo para definir o processo e por último é avaliado o desempenho e caso positivo, a aplicação do modelo.

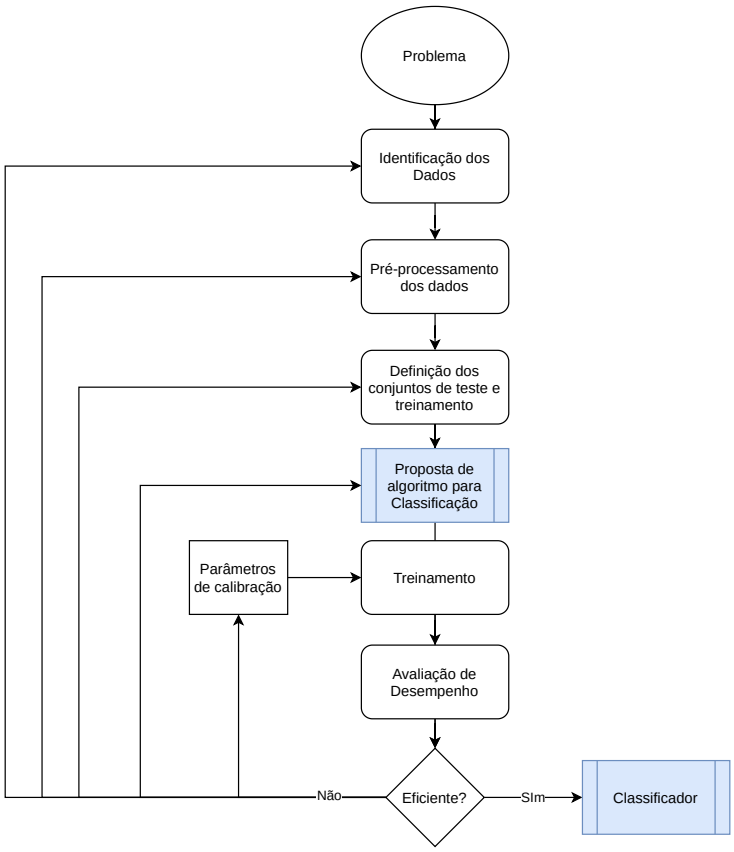


Fonte: Adaptado de Fenner (2019).

A abordagem supervisionada é mais comum para problemas de classificações com rótulos (classes), normalmente o objetivo é fazer o computador ser capaz de operar um sistema de classificação desenvolvido para uma atividade específica (FENNER, 2019). Como exemplo de problema de classificação pode-se citar o reconhecimento de números escritos por seres humanos, nesta abordagem, as entradas indefinidas ou inválidas são descartadas, pois, não será possível realizar uma inferência de um valor nulo ou indefinido.

Os rótulos presentes nas amostras do *dataset* servem para o aprendizado, e os problemas podem variar de uma classificação binária (quando existem apenas duas classes) ou uma classificação multi classe (quando existem mais de duas). Na figura 3 é possível observar o ciclo de um aprendizado supervisionado.

Figura 3 – Aprendizado supervisionado. Partindo da identificação do problema e dos dados, as informações são tratadas (por exemplo: transformar texto em número), para posteriormente ser definido a métrica de divisão do conjunto avaliado. É realizado um treinamento para avaliar a eficiência da proposta, caso positivo pode utilizá-lo como um classificador para o cenário identificado, se não são realizados ajustes.



Fonte: Adaptado de F.Y et al. (2017).

De acordo com F.Y et al. (2017) alguns dos algoritmos mais utilizados de classificação são: classificador linear, regressão logística, classificador Naïve Bayes (NB), máquina de vetores de suporte (SVM), Árvore de decisões.

a) **Classificador Linear**: classifica as amostras em classes combinando linearmente as características da amostra. É capaz de separar a base de dados utilizando retas, o que também permite separar as amostras analisadas. Esse classificador não é recomendado para se utilizar em bases que as informações não sejam linearmente separáveis, mesmo que existam modificações que permitam tais aplicações. (KHAMMASSI; KRICHEN, 2017; SAMMUT; WEBB, 2017). Esta categoria de classificador é eficiente quando existe uma grande dimensão, um cenário que se destaca é a contagem de palavras em documentos.

b) **Regressão logística**: é uma variação de classificador linear, o resultado normalmente se encontra entre 0 e 1 para indicar a relevância para em relação ao resultado da predição. Onde as amostras analisadas são transformadas em uma razão de probabilidades e utilizadas para a predição da representatividade de um

evento. (ABRAHAM et al., 2018; KHAMMASSI; KRICHEN, 2017; PALMIERI, 2019).

c) **Redes Bayesianas**: são modelos baseados no teorema de Bayes que utiliza do conhecimento do incerto e incompleto para criar a ligação de uma probabilidade condicional a outras, permitindo assim então transformar meras casualidades estatísticas em confirmações. São representadas por grafos acíclicos direcionados, onde os nós representam variáveis e as arestas a dependência condicional entre elas. (SAMMUT; WEBB, 2017).

d) **Naïve Bayes**: uma variante das Redes Bayesianas que não considera as dependências das variáveis. Por exemplo, não considera fatores como horário, local, origem do tráfego no momento de predição. (ABRAHAM et al., 2018; AHMED; NASER MAHMOOD; HU, 2016; KHAMMASSI; KRICHEN, 2017; KUNHARE; TIWARI; DHAR, 2020; MISHRA et al., 2019).

Na aprendizagem não supervisionada diferentemente da supervisionada os dados que serão utilizados para treinamentos não possuem rótulos, e a classificação é realizada por meio do agrupamento de características semelhantes. Ela também possui capacidade de reduzir os hiperplanos de um *dataset*, permitindo assim identificar tendências ou concentrar características selecionadas. (SAMMUT; WEBB, 2017).

De acordo com Berry, Mohamed e Yap (2020) entre os algoritmos mais utilizados para aprendizagem não supervisionada estão os *K-means*, Aprendizado hierárquico e Agrupamento (*Clustering*) hierárquico.

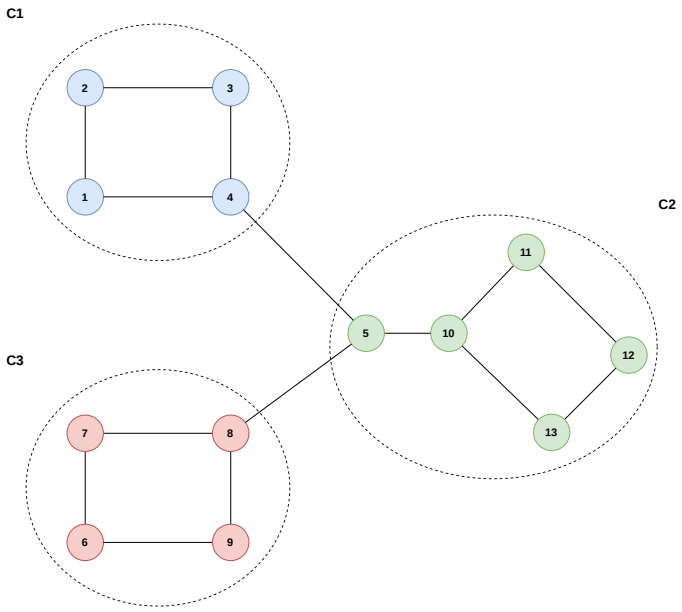
a) ***K-means***: de acordo com Sammut e Webb (2017) um dos algoritmos mais simples não supervisionado para solucionar problemas de agrupamentos. No começo do seu processamento é determinado o número de agrupamentos e é estabelecido aleatoriamente o centro (centroide) de cada agrupamento. Os centroides precisam estar posicionados de maneira inteligente, pois caso estiverem próximos os resultados não serão satisfatórios, a melhor alternativa é posicioná-los o mais longe possível uns dos outros; após isso os dados agrupados como parte do centroide mais próximo, posteriormente é calculado novamente o novo centroide comparando com o resultado anterior, até que se chegue no ponto que não seja mais possível movê-los. (AHMED; NASER MAHMOOD; HU, 2016; KHAMMASSI; KRICHEN, 2017; MISHRA et al., 2019; CAI et al., 2019). Na figura 4 está representado o algoritmo *k-means*.

b) **Aprendizado Profundo**: O aprendizado profundo é inspirado nas redes de neurônios do cérebro, a partir da entrada (camada de entrada), a cada passo (camada oculta) ocorre uma nova avaliação do dado e utilizando os pesos e viés do neurônio

obtém-se o resultado (camada de saída). (FADLULLAH et al., 2017; MISHRA et al., 2019).

c) ***Clustering* Hierárquico**: É um método que prioriza a criação de *clusters* que possuem duas categorias de predominância: de ordem de cima para o baixo (também pode ser denominada *top-down*) ou de baixo para cima (*bottom-up*), sendo a última a mais utilizada. Um dos métodos de agrupamento mais comuns é o de Aninhamento Aglomerativo (predominância *bottom-up*). Este algorítimo pode ser descrito da seguinte maneira: cada dado do conjunto forma um agrupamento (*cluster*) de um único nó, no próximo passo os dois nós mais próximos formam um novo *cluster*, na ação seguinte é criado um *cluster* a partir dos 2 *clusters* mais próximos. (SAMMUT; WEBB, 2017; MISHRA et al., 2019).

Figura 4 – Aprendizado não supervisionado *K-means*. Temos três conjuntos, no $C1$ existe uma associação ao $C2$, partindo do 4, enquanto o $C2$ também interage com o cluster $C3$ partindo do 5 do mesmo valor que interagem com o $C1$, significando que o valor de $C2$ será o centroide central e os pontos em $C1$ e $C3$ serão seus membros.



Fonte: Adaptado de Cai et al. (2019)

Não existem somente as categorias de algoritmos supervisionados e não supervisionados, existem mais métodos descritos na literatura sendo eles: aprendizado semi supervisionado, aprendizado por reforço, transdução e aprendizado para aprender (OLADIPUPO, 2010). Entretanto, eles não serão abordados.

2.2.1 Avaliação de desempenho

Para avaliação de desempenho de um classificador é comumente utilizado a matriz de confusão, a qual descreve de maneira objetiva o resultado obtido por um sistema de

classificação. Na matriz está contida o valor esperado ser identificado pelo classificador e também o resultado determinado (preditado), ela é quadrada de no mínimo de ordem $n = 2$. (DENG, 2016).

Figura 5 – Matriz de confusão binária.

Fonte: Elaborado pelo autor.

Na figura 5 temos uma matriz de confusão binária com os parâmetros utilizados para realizar a avaliação de desempenho sendo eles: positivos verdadeiros (PV), que representam a identificação correta das anomalias, os falsos negativos (FN), que identificam ocorrências de classificações erradas de anomalias como amostras normais, os negativos verdadeiros (NV), classificam os tráfegos que realmente não são normais (ex.: tráfego não esperado naquela rede) e os falsos positivos (FP), que representam as classificações de ações consideradas corretas como anormais. (PALMIERI, 2019). A partir da matriz de confusão somos capazes de calcular facilmente as seguintes métricas:

- **Acurácia** (ACR): é a porção de predições corretas, pode ser obtida por meio da seguinte formula:
$$\frac{PV + NV}{PV + PN + FP + FN}$$
- **Precisão** (PRE): mensura a acurácia de uma determinada classe ser classificada, é visualizada a partir de:
$$\frac{PV}{PV + FP}$$
.
- **Sensibilidade** (SEN): taxa de positivos verdadeiros, contabiliza a quantidade de eitas pelo sistema, podendo ser extraída da matriz com:
$$\frac{PV}{PV + FN}$$
- **Especificidade** (ESP): taxa de negativo verdadeiro, responsável por informar a eficiência das classificações de NV realizadas, podendo ser extraída da matriz com:
$$\frac{NV}{NV + FP}$$

- **Medida F** (F-1): média harmônica entre a sensibilidade e especificidade, bastante utilizada em cenários em que as bases de dados são desbalanceadas como, por exemplo, na detecção de anomalias. Ela é extraída da matriz por meio de:

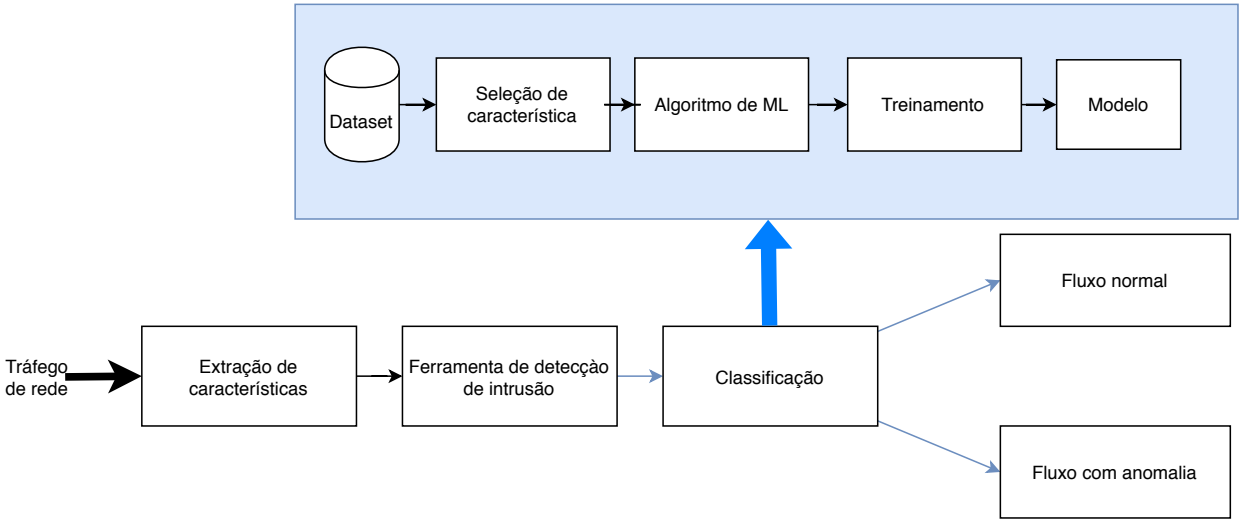
$$2 \cdot \frac{PRE \cdot SEN}{PRE + SEN}$$

2.3 Seleção de característica

Uma característica de um modelo ou amostra pode ser considerada a representação numérica de qualquer dado bruto de um conjunto de dados. A seleção de característica tem como um dos seus objetivos reduzir a quantidade de característica (redução de dimensão) dos dados que será utilizado pelo sistema de classificação. Entretanto, seu principal objetivo não é apenas reduzir o tempo de treinamento, mas sim aumentar a acurácia do resultado de predição do modelo.(ZHENG; CASARI, 2018; KUHN; JOHNSON, 2019).

Na Figura 6 é possível observar o papel que a seleção de característica exerce sobre a redução da dimensão dos dados a serem analisados, seguindo critérios definidos para classificação, por exemplo, ranqueamento das características por peso de importância e por último será classificado em fluxo normal ou com anomalia.

Figura 6 – Proposta de arquitetura para ferramenta de detecção de intrusão.



Fonte: Adaptado de Selvakumar e Karuppiah (2019).

Após isso é realizada a remoção das características que não obtiveram bons resultados de acordo com o critério e por último se a amostra analisada sera classificada como normal ou anomalia.

Quando utilizado de maneira supervisionada, os métodos de seleção de características podem ser divididos em três grandes grupos, sendo eles:

a) ***Wrapper***: São técnicas complexas, pois utilizam de um algoritmo de aprendizado de máquina para verificar a qualidade de um determinado subconjunto de características das amostras presentes no *dataset*. Ela possui um mecanismo de busca para identificar quais subcojuntos de características devem ser avaliados, para assim então identificar o melhor subconjunto para ser utilizado. (ZHENG; CASARI, 2018).

b) **Filtro**: Técnicas de filtro preprocessam as características para identificar de maneira isolada quais são as melhores para o modelo. São técnicas que em sua grande maioria possuem um custo menor que as de *wrapper*, características para remover as que não possuem um valors para o modelo. São técnicas que em sua grande maioria possuem um custo menor que as de *wrapper*, não consideram o classificador que está sendo utilizado, com isso características que possuam um valor de predição significativa podem ser removidas, que consequentemente afetam o desempenho de predição. (ZHENG; CASARI, 2018). De acordo com Kuhn e Johnson (2019) as técnicas de filtro podem considerar cada preditor separadamente, e embora não seja um requerimento, são eficientes em capturar grandes tendências no *dataset*.

c) **Intrínseco**: As técnicas intrínsecas (na literatura podem ser também encontradas como embutidas ou implícitas) fazem parte do modelo de ML. Uma das vantagens que se obtém é a seleção de característica conectada ao objetivo da função, este último é o modelo estatístico que o modelo otimizará. Esta conexão permite visão entre a dispersão das características e o desempenho da predição, permitindo assim uma melhor escolha. (KUHN; JOHNSON, 2019).

Embora tanto a seleção supervisionada quanto a não supervisionada tenham o objetivo de reduzir a quantidade de características que serão utilizadas no modelo no algoritmo de ML, o não supervisionado possui duas grandes vantagens, sendo elas: (1) não são enviesados e são eficientes quando não existe conhecimento prévio disponível, (2) são capazes de reduzir o risco de ajustes excessivos nos dados, em contraste com os métodos supervisionados de seleção de recursos que não podem ser capazes de lidar com uma nova classe de dados. (SOLORIO-FERNÁNDEZ; CARRASCO-OCHOA; MARTÍNEZ-TRINIDAD, 2020).

Dong e Liu (2018) e Alelyani, Tang e Liu (2018) dividiram os métodos não supervisionados nos seguintes grupos:

a) **Filtro**: possuem duas subcategorias, sendo elas, univariada e multivariada. O primeiro também é conhecido como método de ranqueamento pois utiliza alguns critérios para avaliar cada característica, criando dessa maneira uma lista ordenada (ranqueamento) de características, em que o subconjunto final de características é

selecionado de acordo com esse pedido. Ele pode efetivamente identificar e remover características irrelevantes, porém não pode realizar a remoção redundante, pois nele não é considerada a existência de dependências entre as características. Por outro lado, o multivariado avalia a relevância das características em conjunto e não individualmente, e são capazes de lidar com características redundantes e irrelevantes. Assim, em muitos casos, a precisão alcançada pelos algoritmos de aprendizagem usando o subconjunto de características selecionadas por métodos multivariados é melhor que a alcançada pelo uso de métodos univariados. (TABAKHI et al., 2015; THEJAS et al., 2019)

b) ***Wrapper***: podem ser divididos em três grandes categorias: a sequencial, bio inspirada e a iterativa. Na sequencial, as características são adicionadas ou removidas sequencialmente, são de fácil implementação e rápidas. Por outro lado, as bio inspiradas buscam incorporar a aleatoriedade na busca, com o objetivo de escapar dos ótimos locais. Por fim a bordagem iterativa, usa a estimativa dos dados em um *cluster* para evitar pesquisa combinatória. (SOLORIO-FERNÁNDEZ; CARRASCO-OCHOA; MARTÍNEZ-TRINIDAD, 2020; THEJAS et al., 2019).

c) **Híbrido**: União dos dois tipos anteriores, atuando de maneiras distintas em cada etapa do processo. Em poucas palavras inicialmente podem ser um método de filtro e nos subconjuntos das características aplicam estratégias *wrappers* para encontrar uma melhor abordagem de *clustering*. (SOLORIO-FERNÁNDEZ; CARRASCO-OCHOA; MARTÍNEZ-TRINIDAD, 2020; THEJAS et al., 2019)

Um dos objetivos da seleção de características é a redução da dimensionalidade de um modelo de predição, ou seja, remover características que não interfiram no resultado da predição, possibilitando que modelos desenvolvidos para os algoritmos alcancem um desempenho superior. Muitos métodos de seleção de características são baseados em soluções estatísticas. (KUHN; JOHNSON, 2019).

De acordo com Brownlee (2020) um dos métodos estatísticos mais utilizados para avaliar a relação entre as variáveis de entrada e saída (resultados do modelo de predição) é o de correlação. Ao utilizar de tais mecanismos para a seleção, a correlação selecionada deve considerar a categoria do dado que será utilizado pelo modelo.

2.4 Máquina dos elementos finitos

A utilização do método dos elementos finitos (FEM) é amplamente utilizada para diversos fins na ciência e engenharia, entretanto, até recentemente ela não havia sido utilizada para as práticas de aprendizado de máquina. Com a proposta de Pereira et al.

(2020) a utilização de elementos finitos emergiu como uma possibilidade, sendo nomeada como máquina dos elementos finitos (FEMa).

O objetivo da proposta é computar a partir de uma amostra do treinamento a probabilidade de aprendizado usando método dos elementos finitos; dessa forma, permitir que cada amostra do conjunto de treinamento seja tratada como uma função de base discreta, permitindo então uma construção variada por todo o conjunto. Outro fator interessante é que ela não necessita de uma fase de treinamento, tornando bastante interessante para trabalhar com grandes conjuntos de dados como, por exemplo, tráfego de rede computadores.

2.4.1 Método dos elementos finitos

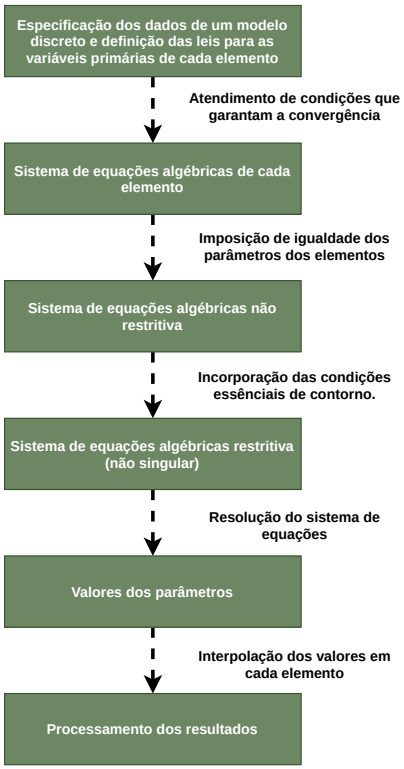
Em modelos contínuos de matemática, a aplicação de métodos analíticos é de alta complexidade e em alguns casos pode ser até mesmo impossível de aplicar. Como alternativa a falta de um método eficiente analítico, utiliza-se de um método que visa reduzir os infinitos graus do modelo contínuo por um número finito de parâmetros ou graus de liberdade de um modelo aproximado. (SORIANO, 2009).

Entre os principais métodos que buscam soluções aproximadas, se encontra o FEM. Ele possui uma fácil capacidade de generalização, programação e utilização; é controlado através de leis simples (em grande maioria polinomiais) para as variáveis primárias em subdomínios conhecidos como elementos finitos, substituindo às leis analíticas para a solução do modelo, permitindo assim uma interface com os elementos. (PINDER, 2018). Para isso os infinitos pontos do modelo contínuo são substituídos por um conjunto finito de elementos, tal processo é conhecido como discretização do modelo matemático contínuo.

De acordo com Soriano (2009) ao aplicar o FEM para resolução de um modelo, as leis locais são aplicadas arbitrariamente sem a imposição de que os pontos precisam coincidir com a solução procurada. O método é aplicado sob condições matemáticas que garantem que a solução aproximada esteja convergente com a solução do modelo original, com isso, garantindo um comportamento similar próximo.

Na figura 7 se encontram as etapas da aplicação generalizada do método, o primeiro passo é a etapa de discretização e os posteriores são passos modulares que possibilitam a sua utilização para as mais diversas aplicações e por último o processamento dos resultados aproximados.

Figura 7 – Etapas de um método dos elementos finitos.



Fonte: Adaptado de Soriano (2009).

2.4.2 Função de aproximação

Uma função de aproximação visa construir um novo conjunto de pontos entre dois valores conhecidos, ou seja, transportar um conjunto $S = \{ a, \ldots, b \}$ de valores para $[c, d]$ sendo esse entre quaisquer pares de números conhecidos. (PINDER, 2018).

Sejam dois conjuntos $\mathcal{D}$ e $\mathcal{V}$ não triviais infinitos, enquanto $\mathcal{F}$ uma função que produz um mapa de infinitas de associações às imagens, ou seja, $\mathcal{F} : \mathcal{D} \to \mathcal{V}$; entretanto $\mathcal{F}$ não será capaz de representar computacionalmente os elementos. Para possibilitar a representação será utilizada uma função de aproximação $\widetilde{\mathcal{F}}$, entre as principais funções de aproximação estão:

1. Função base de aproximação: dada uma base ϕ do espaço $\mathcal{V}$ representar um vetor de funções $\phi = [\phi_1, \phi_2, \ldots, \phi_n]$, em que todos os elementos são linearmente independentes; permitindo obter qualquer elemento $v \in \mathcal{V}$ por meio da seguinte combinação linear:

$$v = \sum_{i=1}^{n} a_i \phi_i, \tag{2.1}$$

   Em que $a = [a_1, a_2, \ldots, a_n]$ para todo $a_i \in \Re$, com isso a função $\widetilde{\mathcal{F}}$ possibilita a representação computacional a partir dos coeficientes reais a quando a base ϕ está presente no espaço finito.

2. Interpolação: no contexto de aproximação de dados discretos, dado um conjunto de ponto $\mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \}$ em que $\mathcal{X} \subset \mathcal{D}$ e os valores associados $\mathcal{Y} = \{ y_1, y_2, \ldots, y_n \}$ e $\mathcal{Y}$ faça parte de $\mathcal{V}$, ou seja, $\mathcal{Y} \subset \mathcal{V}$, permite que a aproximação da função $\widetilde{\mathcal{F}}$, responsável por interpolar os pares $(\mathbf{x}_i, y_i)$, tal que:

$$\widetilde{\mathcal{F}}(\mathbf{x}_i) = y_i, \forall i \in \{ 1, 2, 3, \ldots, n \}, \tag{2.2}$$

sendo possível então, por meio de $\widetilde{\mathcal{F}}$, encontrar combinações de maneira que:

$$\widetilde{\mathcal{F}}(\mathbf{x}_i) = \sum_{j=1}^{n} a_j \phi_j(\mathbf{x}_i) = y_i, \forall i \in \{ 1, 2, 3, \ldots, n \}, \tag{2.3}$$

demonstrando que é possível encontrar por meio da combinação das bases lineares e os seus coeficientes. A formulação desenvolvida na 2.3 é o equivalente ao seguinte sistema linear:

$$\mathcal{Z} a = y \tag{2.4}$$

com $y = [y_1, \ldots, y_n]^T$ e $\mathcal{Z}$ é uma matriz n $\mathbf{x}$ n que representa a significância do elemento da base (ϕ_i) em relação ao ponto x_j, ou seja, $Z_{ij} = \phi_i(\mathbf{x_j})$.

$$\phi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{caso contrário.} \end{cases} \tag{2.5}$$

Com isso podemos dizer que uma função é de interpolação se possuir uma matriz de identidade $\mathcal{A}$, em que $a_i = y_i, \forall i \in \{ 0, 2, \ldots, n \}$. Assim, uma função base que não realizar a interpolação naturalmente, em tais casos é possível obter um novo $\hat{\phi}$, com isso cada elemento é a combinação linear dos elementos de ϕ_i a partir da equação:

$$\hat{\phi}_i(x) = \sum_{j=0}^{n} \mathcal{Z}_{ij}^{-1} \phi_j(x), \tag{2.6}$$

onde $\mathcal{Z}^{-1}$ é a matriz inversa de $\mathcal{Z}$.

A qualidade dela pode ser obtida por meio da norma $||\widetilde{\mathcal{F}} - \mathcal{F}||$, no qual $|| \cdot ||$ pode ser qualquer norma definida sobre o espaço finito. Esta norma também é conhecida como erro de aproximação (PINDER, 2018; PEREIRA et al., 2020).

Uma propriedade muito importante das funções de aproximações apresentadas é que permitem que os subconjuntos se interceptem apenas uma única vez, ou seja, se tornam uma partição de unidade interpoladora (SCHWEITZER, 2003; ODEN; REDDY; ENGINEERING, 2011). Uma base ϕ se torna uma partição se somente se atender os seguintes requisitos:

$$\phi_i(\mathbf{x}) \geq 0, \forall i \; e \; \forall \mathbf{x} \in \mathcal{D}, \tag{2.7}$$

e

$$\sum_{i=1}^{n} \phi_i(\mathbf{x}) = 1, \forall \mathbf{x} \in \mathcal{D}, \tag{2.8}$$

ao atender as condições das equações 2.7 e 2.8, garante-se que possua uma propriedade de suavização:

$$\mathbf{a_l} \geq \sum_{i=1}^{n} \mathbf{a_i}\phi_i(\mathbf{x}) \geq \mathbf{a_h}, \tag{2.9}$$

$\mathbf{a_l}$ e $\mathbf{a_h}$ são mínimo e o máximo coeficiente de $\mathbf{a}$. Em cálculos computacionais de interpolação a utilização de suavização evita a descontinuidade. (PEREIRA et al., 2020).

2.4.3 Algoritmo FEMa

Um *dataset* para o FEMa pode ser dado pela por intermédio da seguinte equação $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{z}$ que pode ser particionado em treinamento e teste, D_1 and D_2 respectivamente.

De maneira sucinta, o FEMa aprende um conjunto de probabilidades $\mathcal{P}(\mathbf{x}) = \{P_1(\mathbf{x}), P_2(\mathbf{x}), \ldots, P_c(\mathbf{x})\}$, de forma em que c define o número total de classes do conjunto, enquanto $P_j(\mathbf{x})$ é a probabilidade da amostra $\mathbf{x} \in \mathcal{D}_1$ estar associada a classe j. Isso garante ao modelo a oportunidade de aprender sobre ele durante a etapa de treinamento, que pode ser desconsiderada ao ser utilizada com funções base considerada nativa de interpolação (PEREIRA et al., 2020).

Com a possibilidade de "pular" a etapa de treinamento, o FEMa possui um grande potencial para lidar com um massivo volume de dados de maneira efetiva, sendo um atrativo para trabalhar com os grandes volumes de dados em redes de computadores. Um exemplo de uma função nativa é a função base de Shepard (DELL'ACCIO; DI TOMMASO; GONNELLI, 2020); em tal base os elementos são descritos como:

$$\phi(\mathbf{x}, \mathbf{x}_i; D_1, k) = \frac{w(\mathbf{x}, \mathbf{x}_i; k)}{\sum\limits_{\mathbf{x}_j \in D_1} w(\mathbf{x}, \mathbf{x}_j; k)}, \tag{2.10}$$

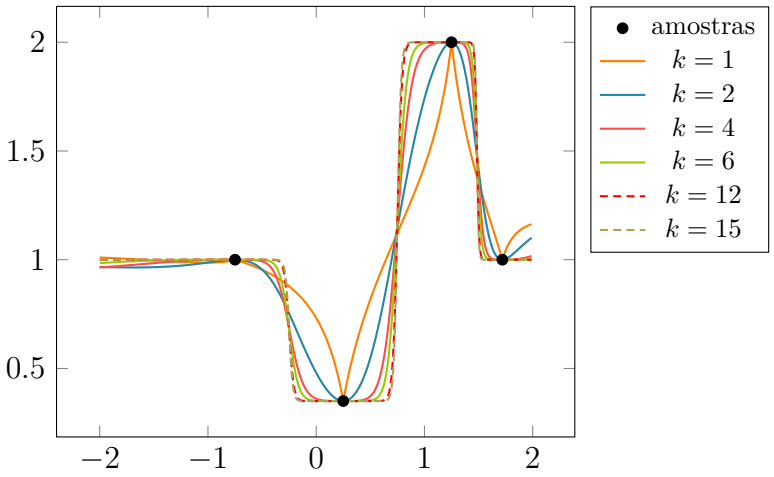
Sendo que $w : \mathcal{D} \times \mathcal{D} \to \Re$ é uma função não negativa, como $w(\mathbf{x}, \mathbf{x}_i) \to \infty$ quando $\mathbf{x} \to \mathbf{x}_i$, ou seja, quanto maior o $\mathbf{x}$ de $\mathbf{x}_i$, maior será a função w. Tais propriedades garantem que a base de Shepard mantenha o controle da interpolação e partição das unidades interpoladoras. De acordo com Rahaman, Ghosh e Thiery (2021) uma função w bastante usada é a função inversa de distância euclidiana, que é descrita como:

$$w(\mathbf{x}, \mathbf{x}_i; k) = \frac{1}{||\mathbf{x}, \mathbf{x}_i||_2^k} \tag{2.11}$$

sendo que $|\mathbf{x}, \mathbf{x}_i|$ representam a distância entre $\mathbf{x}$ and $\mathbf{x}_i$, o parâmetro k possui o controle da interpolação e pode ser manipulada de acordo com a necessidade. Na figura

8 é possível observar o comportamento da base de Shepard utilizando diferentes valores para k, note que quanto maior o seu valor mais rigidas as curvas se tornam.

Figura 8 – Exemplo de interpolação de Shepard.



Fonte: Elaborado pelo autor.

Embora seja possível utilizar o FEMa com funções não interpoladoras como, por exemplo, em base radial, é necessário computar o inverso da matriz que fornece a influência de cada elemento básico, e caso a função não seja capaz de manter a propriedade de partição de unidade interpoladora, será necessário normalizar cada amostra avaliada. Sendo assim, é preferível a utilização de métodos que sejam capazes de realizar interpolação e partição de unidade (PEREIRA et al., 2020).

Quando a função base utilizada segue as recomendações para o contexto, a classificação de uma amostra $\mathbf{x} \in \mathcal{D}_2$, sua probabilidade de pertencer a cada classe j, $j = 1, 2, \ldots, c$ pode ser encontrada com a equação:

$$P_i(\mathbf{x}; D_1; k) = \sum_{j=1}^{m} \rho_j^i \phi(\mathbf{x}, \mathbf{x}_j; D_1; k), \tag{2.12}$$

$\rho_j^i \in [0, 1]$ representa a amostra de treinamento i à qual faz parte da classe j. Uma das principais propriedades do algoritmo consiste na possibilidade de associar a cada elemento de treinamento uma probabilidade de incerteza as amostras e m representa o total de amostras, permitindo assim a tratativa de *Over fitting*. A probabilidade $\rho_j^i \in [0, 1]$ em *dataset* devidamente rotulados pode ser obtida mediante:

$$\rho_j^i = \begin{cases} 1 & \text{se } y_j = i \\ 0 & \text{caso contrário.} \end{cases} \tag{2.13}$$

2.4.4 Seleção de Características com FEMa

Aqui é apresentado a abordagem que adapta o classificador FEMa para a tarefa de seleção de características. Ao invés de associar uma amostra de treinamento $\boldsymbol{x} = [x^1, x^2, \ldots, x^n]$ a um rótulo, o FEMa-FS calculará como cada característica x^j de $\boldsymbol{x}$ contribui para a classificação.

Seja $p \in [\min(\boldsymbol{x}), \max(\boldsymbol{x})]$ um escalar que pode ser calculado da seguinte maneira:

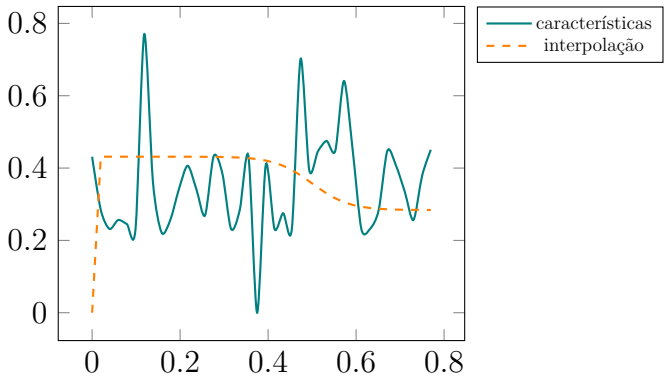
$$p = \min(\boldsymbol{x}) + pos(\boldsymbol{x}) \cdot \frac{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}{m}, \tag{2.14}$$

onde $\min(\boldsymbol{x})$ e $\max(\boldsymbol{x})$ são funções que retornam os valores mínimo e máximo em $\boldsymbol{x}$. Além disso, $pos(\boldsymbol{x}) \in [1, m]$ representa a posição (identificador) da amostra no conjunto de treinamento. A equação 2.14 gera um número que provavelmente diferirá para cada característica. Uma distância d de uma determinada característica x^j tem em relação aos seus valores máximo e mínimo pode ser calculada da seguinte forma:

$$d(x^j; \boldsymbol{x}) = \frac{x^j - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \tag{2.15}$$

Com isso é possível realizar interpolações utilizando a equação 2.10 fornecendo como entrada os resultados das equações 2.14 e 2.15, possibilitando assim a obtenção de uma representação interpolada de uma característica f como na figura 9.

Figura 9 – Exemplo de representação das características interpoladas. Onde a linha continua representa a característica e a linha tracejada a sua interpolação.



Fonte: Elaborado pelo autor.

Para realizar a avaliação da força da representatividade de uma classe c por uma característica f, computa-se a soma do valor da interpolação da característica e então as características com menor variação média na interpolaçào são classificadas como mais expressivas. Para uma melhor compreensão, o algoritmo 1 implementa o pseudo código do FEMa-FS.

No algoritmo 1, o laço principal é executado nas linhas $1-11$, que itera por todas as instâncias do conjunto de treinamento, enquanto o laço apresentado nas linhas $2-10$ sobre

Algoritmo 1: FEMa Feature Selection

Entrada: Conjunto de treinamento D_1, quantidade de amostras m, mapa dos rótulos L, quantidade de características n e total de classes c.
Saída: Mapa de Ranqueamento R.
Auxiliares: Basis function ϕ, posição da amostra p, função de distância d, armazenamento auxiliar $S(m, n, c)$.

```
1  para i = 1, 2, ..., m faça
2      para j = 1, 2, ..., n faça
3          para c = 1, 2, ..., c faça
4              Compute p usando a equação 2.14;
5              Compute d(x_i^j) usando a equação 2.15;
6              se L(x_i) = c então
7                  S(i, j, L(x_i)) ← φ_f(d(x_j^i), p));
8              fim
9          fim
10     fim
11 fim
12 para j = 1, 2, ..., n faça
13     R(j) ← 0;
14     para i = 1, 2, ..., m faça
15         para c = 1, 2, ..., c faça
16             R(j) += S(i, j, c);
17         fim
18     fim
19 fim
20 ordene(F);
21 retorna F;
```

as características de cada amostra. O laço interno é representado nas linhas $3-9$ onde itera-se através de todas as classes disponíveis. As linhas 4 e 5 computam, computam a posição p e sua respectiva distância d para ser utilizada na interpolação. A linha 6 verifica se o rótulo verdadeiro da amostra é igual a c e então armazena a interpolação na linha 7 usando as Equações 2.14 e 2.15 como entrada.

Mais uma vez, as linhas $12-17$ itera-se por todas as características. Nesse contexto, a linha 13 inicializa o mapa de classificações F enquanto o laço interno nas linhas $14-18$ itera por todas as classes. A linha 16 preenche o mapa de classificações. Finalmente, a linha 18 classifica o mapa de classificações, que é retornado como a saída do algoritmo na linha 20.

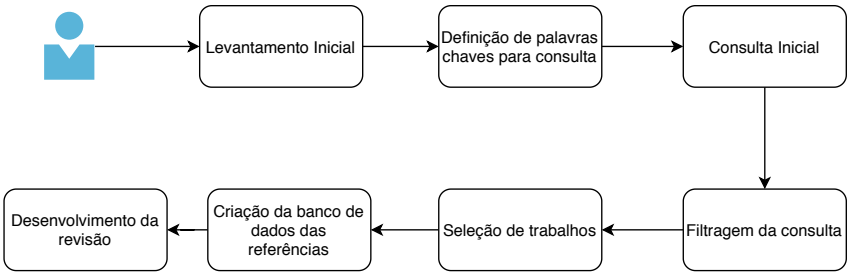
2.5 Trabalhos Relacionados

A seguir, se encontra uma revisão de trabalhos publicados nos últimos cinco anos, são utilizados de métodos de seleção de característica para aumentar a eficiência na identificação de anomalias.

2.5.1 Critérios de trabalhos relacionados

Na figura 10 estão representadas a ordem das atividades realizadas para o desenvolvimento deste trabalho. A primeira atividade realizou o entendimento do que seria revisado e quais seriam as limitações da pesquisa.

Figura 10 – Processo de elaboração desta revisão. Na primeira etapa realizou-se um levantamento inicial, definiu-se as palavras chaves e realizou-se uma consulta inicial bem como as subsequentes tarefas.



Fonte: Elaborado pelo autor.

Após a limitação do escopo foi definido a palavra-chave central para iniciar o levantamento bibliográfico, a palavra-chave definida foi "*feature selection anomaly network traffic detection*". Ela foi utilizada para realizar a consulta inicial no programa *Publish or Perish*[1]. Posteriormente foram realizadas novas consultas com as seguintes palavras chaves:

- "*Finite Element Method Feature Selection*"
- "*Parallel Feature Selection*"
- "*Anomaly network classification*"
- "Feature selection for anomaly detection"

As consultas foram limitadas aos indexadores de trabalhos acadêmicos *Google Scholar* e Elsevier *Scopus*, possibilitando de maneira centralizada e clara obter 418 artigos iniciais. Entre os artigos inicialmente levantados foi dado a preferência para as publicações encontradas nos seguintes repositórios: *IEEE Explorer Digital Library*, *Science Direct*, arXiv.org e *Springer Direct*. O período de busca limitou-se a conteúdos publicados nos últimos cinco anos.

[1] Disponível em: https://harzing.com/resources/publish-or-perish

2.5.2 Detecção de anomalias em redes e extração de características

Aghdam e Kabiri (2016) propuseram um algoritmo de seleção de característica baseando-se na otimização de colônia de formigas e ao reduzir as características em cada categoria de ataque, obteve-se uma precisão similar a utilização de todas as características para predição, nas categorias de ataques remotos para rede local (R2L) e sondagem o resultado foi maior quando comparado a todas as características. Para avaliação da proposta, foram utilizados os *datasets* KDD-CUP 99 e NSL-KDD e a quantidade de características selecionadas para classificação em:

- Tráfego válido: 5
- Negação de Serviço (DoS): 4
- Acesso ao usuário administrador (U2R): 4
- R2L: 3
- Sondagem: 8

No trabalho desenvolvido por Ambusaidi et al. (2016) foi apresentado um algoritmo de filtro para seleção de características baseado em informação mútua (MI) para a extração de características e para classificação foi utilizado o algoritmo de aprendizado de máquina SVM. Para avaliação os autores utilizaram três *datasets*, sendo eles KDD-CUP 99, NSL-KDD e Kyoto 2006+, no local que a quantidade de característica variou, eles selecionaram as seguintes quantidades de características:

- KDD-CUP 99: 19
- NSL-KDD: 18
- Kyoto 2006+: 4

O trabalho alcançou resultados melhores do que quando utilizado todas as características, obtendo uma redução de cerca de 50% nos *datasets* KDD-CUP 99 e NSL-KDD em que ambos possuem do total 41 características, enquanto no de Kyoto 2006+ a redução foi de cerca de 83% do total de 24 características.

Enquanto no trabalho de Chen et al. (2016) a abordagem para seleção de característica foi de filtro utilizando-se do coeficiente máximo de informação. Os autores utilizaram o algoritmo de *Multi-scale principal component analysis* (MSPCA) para identificação de anomalias de DDoS. Para alcançar a mesma eficiência da utilização de todas as características, foi necessário extrair dezesseis destas características do conjunto. A base utilizada para o trabalho foi a KDD-CUP 99.

Na abordagem de Gharaee e Hosseinvand (2016), para detecção de anomalias foi utilizado a seleção de características do tipo *wrapper* baseado em algoritmo genético e como algoritmo de classificação foi utilizado o SVM. Essa combinação alcançou uma taxa de acurácia de mais de 99% nas bases de teste KDD-CUP 99, com uma taxa de falso positivo menor de 1%. Os pesquisadores também realizaram uma avaliação de desempenho com a base de testes UNSW-NB15, e a taxa de acurácia teve uma grande variação, entretanto a taxa de falso positivo não alcançou 0.10%. Na base de teste KDD-CUP 99 foram selecionadas as características por tipo de classe, sendo a quantidade por classe:

- Normal: 7
- DoS: 10
- Sondagem: 9
- R2L: 7
- U2R: 8

Enquanto na outra base de testes a quantidade de características selecionadas por classe foi:

- Normal: 7
- *Fuzzers*: 13
- *Reconnaissance*: 14
- *Shellcode*: 9
- DoS: 12
- *Exploits*: 6
- Ataques Genéricos: 9

O trabalho de Hasan et al. (2016), baseou-se na utilização do algoritmo de aprendizado de máquina Floresta Aleatória, o qual possui uma seleção de característica intrínseca. O processo foi realizado em dois passos. No primeiro passo foi utilizado a permutação de importância de índice para realizar o ranqueamento das características e no segundo passo foi utilizada a floresta aleatória para encontrar o subconjunto de características com a melhor acurácia, conseguindo encontrar assim um subconjunto contendo 25 características utilizando como *dataset* para seleção a KDD-CUP 99. No treinamento o subconjunto selecionado foi capaz de alcançar um percentual um pouco maior do que o teste com todas

as características. Os autores mostraram também que foi possível reduzir o tempo de treinamento em aproximadamente em $\frac{1}{4}$. No entanto, nos testes o subconjunto teve uma taxa de acurácia média de 65.92% contra 82.36% ao utilizar todas as 41 características do conjunto.

Khaokaew e Anusas-amornkul (2016) realizaram um estudo de comparação de diversos métodos para seleção de características baseadas em SVM, e também utilizaram do mesmo para a classificação de anomalias. Foi utilizado o *dataset* KDD-CUP 99 para avaliação de desempenho. A quantidade de características selecionadas para cada cenário foi:

- SVM Padrão - 37
- Envoltória convexa (CH-SVM) - 9
- Seleção de característica baseada em correlação (CFS-SVM) - 11
- *Motif Discovery Using Random Projection* (MDRP-SVM) - 10
- Seleção de características Híbrida (HFS-SVM) - 3

No experimento os autores notaram que as variações HFS-SVM e CH-SVM alcançavam resultados similares quando o conjunto de testes possuía mais de 300.000 registros, com menores números a obtenção de um resultado não fora alcançado. O tempo necessário que o HFS levou para ser treinado foi a metade do tempo necessário para treinar o SVM Padrão.

Osanaiye et al. (2016) apresentaram uma abordagem do algoritmo de multi filtro para seleção de características baseada em *ensemble*. Fazem parte do multi filtro os seguintes algoritmos: *information gain*, *gain ratio*, *chi-squared* e *ReliefF*. A base de teste utilizada foi à NSL-KDD. Neste trabalho os autores, realizaram diversas rodadas foram selecionados 14 características por algoritmo presente no multi filtro, para a seleção final foram selecionadas 13 características mais presentes. Para classificação foi utilizado o algoritmo de árvore de decisão, os resultados foram satisfatórios, houve um pequeno aumento na porcentagem de acurácia e detecção. Também foi identificado um pequeno aumento de falso positivo e o tempo de construção do modelo foi drasticamente reduzido.

Em sua publicação Gadal e Mokhtar (2017) utilizaram dois algoritmos de seleção de características (*ConsistencySebsetEvel* e Busca Genética) para selecionar 22 características do conjunto de teste NSL-KDD para depois realizar a avaliação utilizando o algoritmo não supervisionado *K-means*, permitindo que obtivessem um resultado satisfatório na acurácia e um baixo nível de falso positivo.

Khammassi e Krichen (2017) propuseram uma seleção de características do tipo *wrapper* que utilizou algoritmo genético para seleção das características e regressão linear

para avaliação de relevância dos conjuntos identificados. Para avaliação de desempenho foram utilizados os seguintes três algoritmos de árvores de decisões: o C4.5, floresta aleatória (RF) e árvore NB. Como base de testes foram utilizados os conjuntos KDD-CUP 99 e UNSW-NB15. O total de subconjuntos de características avaliados por base foram 3, sendo distribuídos da seguinte maneira:

- KDD-CUP 99: Amostragem de 1.000 registros - 18
- KDD-CUP 99: Amostragem de 1.500 registros - 16
- KDD-CUP 99: Amostragem de 2.000 registros - 18
- UNSW-NB15: Amostragem de 1.000 registros - 18
- UNSW-NB15: Amostragem de 1.500 registros - 24
- UNSW-NB15: Amostragem de 2.000 registros - 20

Na classificação realizada com a base KDD-CUP 99, o algoritmo de RF em conjunto com as características extraídas da amostragem de 1.000 foi o que obteve melhor resultado, enquanto na outra base o algoritmo C4.5 em conjunto com o subconjunto em união com as características extraídas de 2.000 amostras obteve o melhor resultado, próximo da avaliação com todas as características da base.

No trabalho realizado por Janarthanan e Zargari (2017) foram analisadas as performances de dois dos subconjuntos de características mais recorrentes segundo a literatura em que se basearam, as bases utilizadas foram KDD-CUP 99 e UNSW-NB15. No primeiro encontram-se 8 características, enquanto no segundo 5 características. A performance do segundo grupo foi superior à performance do primeiro. O critério de performance foi avaliado por meio do método estatístico Kappa.

Moustafa e Slay (2017) desenvolveram um modelo de seleção de características híbrido dividido em dois passos, primeiro calcula-se o ponto central de cada característica e a saída é utilizada como entrada. No segundo passo é executado o algoritmo de regra de associação de mineração (ARM),responsável por realizar o ranqueamento das características. Após a execução da seleção de características foram escolhidas 11 características que, segundo os autores equivalem cerca de 25% do total das características dos conjuntos de testes NSL-KDD e UNSW-NB15. Já para classificação foram utilizados três algoritmos:

- *Expectation-Maximization clustering (EM)*
- Regressão Logística
- Naïve Bayes (NB)

O algoritmo de regressão logística foi o que obteve melhor resultado, entretanto ele teve um valor de falso positivo maior que o de EM. Já o NB com UNSW-NB15 teve a pior taxa de falso positivo de 61.4%. Os pesquisadores não disponibilizaram nenhuma análise com todas as características para comparação.

Ullah e Mahmoud (2017) desenvolveram uma abordagem de seleção baseada em filtro utilizando a técnica de ganho de informação para realizar o ranqueamento das características e posteriormente realizar a avaliação. Entretanto os autores não especificaram qual algoritmo de classificação foi utilizado para detecção de anomalia nas bases NSL-KDD e ISCX. Para a primeira base foram selecionadas 6 características enquanto para a outra foram selecionadas 4 características. Foi identificado pelos autores um ganho satisfatório na acurácia em relação à utilização de todas as características da base NSL-KDD e também identificaram uma redução de falsos positivos em aproximadamente $\frac{3}{4}$. Na base ICSX os autores também identificaram um ganho na acurácia quando utilizado o método de seleção de características, entretanto a redução de falso positivo foi de apenas $\frac{1}{6}$.

No trabalho realizado por Zhu et al. (2017) foi apresentado um algoritmo de seleção de características *wrapper* de muitos objetivos baseado em algoritmo genético. Para a avaliação dos pesos das características foi utilizado o método de correlação baseado no indíce jaccard. Os autores utilizaram uma variação do algoritmo *Growing Hierarchical Self-Organizing Maps* (GHSOM-pr) para avaliação do desempenho do método de seleção de características nos *datasets* Guru-KDD e KDD-CUP 99. Em ambas foram selecionadas 20 características pelo seletor. No trabalho foi identificado uma redução de características satisfatória nos treinamentos e testes do modelo proposto de aproximadamente 33% e 22%, respectivamente e o classificador foi capaz de manter um resultado similar ao utilizado com todas as características.

Alabi e Yurtkan (2018) realizaram a comparação de dois métodos de seleção de características, o primeiro baseado em entropia e o segundo em variância, ambos do tipo filtro. Os autores utilizaram para validar a eficiência de ambos métodos o algoritmo de classificação de mínima distância e o dataset escolhido foi o UNB/IDS 2012, que foi subdividido em dois conjuntos de testes menores nos quais foram denominados como D1 e D2. Nos testes ambas soluções selecionaram 3 características, os resultados de acurácia com ambas propostas tiveram um desempenho mais eficaz do que sem a utilização.

Um método proposto por Khan et al. (2018) é o de filtro que utiliza de entropia e computação granular. A base escolhida para seleção foi a NSL-KDD, para consideração da importância das características para seleção foi utilizado a entropia de Shannon em conjunto com uma análise probabilística de peso.

Anwer, Farouk e Abdel-Hamid (2018) realizaram uma avaliação de diferentes métodos para detecção de anomalia, na qual são utilizadas 6 técnicas de filtros e *wrappers* para a identificação de anomalias. Para classificação utilizou-se 2 tipos de classificadores,

o J48 e o Naïve Bayes; o dataset utilizado foi o UNSW-NB15. O Framework consiste na execução das seguintes técnicas de filtro:

- *Information Gain*
- *Gain Ratio*
- *Symmetrical Uncertainty*
- *Relief F*
- *One R*
- *Chi Squared*

Os autores não mencionaram os critérios definidos para os *wrappers*, entretanto na avaliação de desempenho notaram que a combinação do algoritmo de seleção de características de *Gain Ratio* com 18 características selecionadas e o classificador J48 foi a combinação que obteve um desempenho similar a utilização de todas as características.

Divyasree e Sherly (2018) propuseram uma seleção de características de filtro baseada em *chi squared* e uma função de peso para selecionar as 10 características mais relevantes para posteriormente ser alimentada ao algoritmo de classificação de *Core Vector Machine* (CVM). Para avaliação de desempenho foi utilizado a KDD-CUP 99 sendo assim, a proposta foi capaz de alcançar uma taxa de acurácia de 99%, entretanto a taxa de falso positivo foi de 27%.

Aljawarneh, Aldwairi e Yassein (2018) utilizaram como seleção de características o método de *Information Gain* para reduzir o *dataset* NSL-KDD de 41 características para 8 características, em que foram selecionadas apenas as que possuiam um peso de 0.40 ou mais. Para construção do modelo, os autores propuseram uma prática de *ensemble* de 7 algoritmos distintos sendo eles:

- J48
- *Meta Paging*
- Árvore aleatória
- *REPTree*
- *DecisionStump*
- *AdaBoostM1*
- Naïve Bayes

O modelo proposto realiza o treinamento e os algoritmos que possuírem uma acurácia no treinamento quando comparada ao *ensemble* será o modelo selecionado para utilização. Os autores não publicaram a taxa de falso positivo do teste, somente que a taxa de acurácia foi similar ao resultado quando se utiliza todas as características pelos algoritmos J48.

Garg e Batra (2018) elaboraram um modelo híbrido baseado em *ensemble* para seleção de características do tipo intrínseco em árvore de decisão, para classificação da relevância das características foi utilizado entropia e *information gain*. Para identificação de anomalia utilizou-se técnicas de *Cuckoo Search Optimization* para criar os centroides e realizar a identificação com *K-means*. Para avaliação do modelo foi utilizado a base NSL-KDD e a proposta selecionou as características de maneiras distintas de acordo com a classe do ataque sendo para:

- Normal: 12
- DoS: 10
- Sondagem: 13
- U2R: 15
- R2L: 9

Os resultados obtidos por eles demonstraram que em relação ao *K-means* "tradicional"o modelo proposto teve um ganho significativo na acurácia (21,32%) e uma redução de falso positivos de aproximadamente 76%. Não foi divulgado informações sobre tempo de treinamento e construção do modelo.

Selvakumar e Karuppiah (2019) apresentaram um seletor para as características híbrido que utiliza de uma técnica de filtro baseada em informação mútua em conjunto com de *wrapper* bioinspirada em libélulas para seleção. Para classificação foram utilizados os algoritmos árvores de decisões C4.5 e Bayesiana. Para validação do método proposto foi utilizada a base KDD-CUP 99. A proposta reduziu a quantidade de características para 10, e o resultado alcançado pelos algoritmos após a seleção de características foi similar ao desempenho com todas, em alguns casos observou-se uma melhora na identificação dos ataques e a taxa de falso positivo com a árvore Bayesiana atingir o valor de 0.01%.

Gottwalt, Chang e Dillon (2019) utilizaram um método de filtro baseado em correlação das características com a classe respectiva, e utilizaram para classificação um algoritmo bio-inspirado de colônia artificial de abelha (ABC) e cardume artificial de peixe (AFS) para realizar a classificação dos *NSL-KDD* e UNSW-NB15. O seletor selecionou 6 características de cada uma. Na primeira base os autores conseguiram obter uma acurácia

de 98.9% e uma taxa de falso positivo de 0.13%, e na segunda obtiveram uma acurácia de 99% e 0.01%, respectivamente.

Os autores Uysal et al. (2019) trabalharam em um filtro de seleção de características baseado em algoritmo genético, o qual foi implementando usando NSGA-II, utilizado em conjunto com o algoritmo J48 para realizar a classificação da base KDD-CUP 99. O experimento obteve uma taxa de acurácia de aproximadamente 97% com uma seleção de 22 características no treinamento e uma acurácia de 91,1% no teste.

Kasongo e Sun (2019) apresentaram um algoritmo não supervisionado, o qual obteve um ganho de performance na classificação binária e multi classe devido a utilização de um seletor de características baseado no método de *information gain*. Através do método foi possível realizar a remoção de características consideradas ruídos para o algoritmo de aprendizado profundo, permitindo então um aumento significativo na taxa de acurácia deste algoritmo utilizando apenas 21 características do dataset *NSL-KDD*.

Mazini, Shirazi e Mahdavi (2019) apresentaram um algoritmo de seleção de características de filtro baseada em colônia artificial de abelhas e em conjunto com o algoritmo de classificação *AdaBoost* e para validação da proposta foram avaliadas as bases NSL-KDD e ISCXIDS2012. Os autores identificaram que ao permitirem que a seleção reduzisse o total de características para 25 obtiveram o melhor resultado, com um acurácia de 98.9% e uma taxa de falso positivo de 0.017%.

Palmieri (2019) propôs uma metodologia para identificação de anomalias utilizando uma seleção de características baseada em *information gain*. Foram selecionadas as 4 características com maior coeficiente e identificado o fator de correlação entre dados de maneira não linear. Posteriormente, essas informações são enviadas para um modelo de regressão linear realizar a classificação. Diferente de outros modelos, o autor propôs uma revisão dos resultados dúbios para que fossem adicionados posteriormente em uma base de conhecimento para ser utilizada pela regressão no futuro. Para sua validação foi utilizado a base ICSX-UNB, sobre a qual conseguiu obter um resultado de acurácia de 99.846% com um erro absoluto de 0.0015%.

Na proposta elaborada por Isa (2020) foi introduzida uma seleção de características utilizando um filtro de correlação baseado em Pearson em conjunto com um método de otimização, os pesquisadores avaliaram a performance de três classificadores: SVM, floresta de decisão e redes neurais. Os autores apresentaram resultados em que evidenciaram a eficácia da proposta em auxiliar os classificadores a alcançarem bons resultados em três *datasets* (KDD-CUP, NSL-KDD e CICIDS 2017) distintos, desta maneira conseguiram alcançar uma acurácia de 99.9% e uma taxa de falso positivo que variou entre 0.0001% à 0.9%.

Kunhare, Tiwari e Dhar (2020) elaboraram um algoritmo de filtro para seleção de

características baseado em floresta aleatória usando a otimização do enxame de partículas (PSO). Foram selecionadas as 10 características com melhor desempenho para ser utilizada pelo algoritmo responsável pela classificação das características. A base selecionada para avaliar o desempenho do modelo proposto foi a NSL-KDD. Constatou-se que a eficiência obtida pelo modelo obteve uma taxa de acurácia mais eficiente que trabalhos relacionados anteriormente.

No trabalho de Zhou et al. (2020) foi proposto um algoritmo de seleção de característica híbrido da correlação de características com o algoritmo do morcego (CFS-BA) em conjunto com um algoritmo de *ensemble* para a realização de detecção de anomalias. Os *datasets* utilizados por eles e a quantidade de características selecionadas em cada foi:

- NSL-KDD: 10
- AWID: 8
- CIC-IDS2017: 13

Na avaliação de eficiência os pesquisadores notaram que houve um ganho considerável na taxa de acurácia e uma redução considerável nas características dos três conjuntos de testes, e o melhor resultado obtido por eles foi na última (CIC-IDS2017), o modelo proposto obteve uma taxa de acurácia de 99.99% e um percentual de falso positivo de 0.01%.

Almomani (2020) realizou uma extensa avaliação de diversos algoritmos de seleção de características e utilizou para classificação os algoritmos J48 e SVM. A base escolhida pelo autor para a extensa análise foi a UNSW-NB15 e foram avaliados os seguintes algoritmos para seleção de características e combinações entre eles:

- Otimização de enxame de particula (PSO) - 25
- Otimização do lobo cinzento (GWO) - 20
- Otimização de libélulas (FFA) - 21
- Algoritmo Genético (GA) - 23

Em sua análise o autor realizou a comparação de dois classificadores com todas as características do dataset, bem como o performance de cada um dos algoritmos de seleção de características e suas respectivas combinações, incluindo a combinação de todas. A melhor dupla sem combinação de métodos de seleção de características foi alcançada pelo par PSO e SVM. Os autores identificaram uma taxa de acurácia 89.152% e uma taxa de falso positivo de 2.596%, enquanto a combinação de todos os métodos de seleção, que resultaram numa seleção 30 características, obteve um melhor desempenho com o algoritmo

J48. Enquanto com todas as características os algoritmos de classificação obtiveram uma acurácia de 81.158% e 81.29% com uma taxa de falso positivo de 4.809% e 4.57%.

Chkirbene et al. (2020) desenvolveram um *framework* para detecção de anomalias em redes de computadores, para isso eles utilizaram um algoritmo próprio para *clustering* das características e realizaram a classificação utilizando árvore de decisão. Para critérios de avaliação os autores utilizaram dois *datasets*, NSL-KDD e UNSW-NB15. Na etapa de seleção de características em ambas das bases foram selecionadas um grupo contendo apenas 5 características. Os resultados de acurácia foram bastante satisfatórios visto que na primeira foi obtido uma acurácia de 98% enquanto na segunda foi obtido 91%.

Na abordagem feita por Zhang et al. (2020) foi utilizado um filtro baseado em dominância de características utilizando a base NSL-KDD, e como classificadores os algoritmos SVM e KNN. Os autores identificaram que os melhores resultados obtidos em cada um dos classificadores a quantidade de características variou, com o SVM obter uma acurácia de 97.694% foram necessárias 18 características, enquanto o KNN precisou somente de 16 para obter 99.083%.

2.5.3 Compilado - Método de seleção, *dataset* e quantidade de características utilizadas

A seguir temos na tabela 2 o compilado das publicações citadas na seção anterior organizado em ordem crescente, na tabela podemos observar a referência para o trabalho citado, o método de seleção de característica utilizado, as bases de dados utilizadas e a quantidade de características selecionadas no trabalho.

Tabela 2 – Compilado de seleção de características

Autor	Algoritmo de Seleção de Característica	*dataset*	Quantidade de Características
Aghdam e Kabiri (2016)	Otimização de colônia de formigas	KDD-CUP 99	Normal - 5 DoS - 4 U2R - 4 R2L - 3 Sondagem - 8
Ambusaidi et al. (2016)	Informação mutua	KDD-CUP 99, NSL-KDD e Kyoto 2006+	19 18 4

Chen et al. (2016)	Coefiente máximo de informação	KDD-CUP 99	16
Gharaee e Hosseinvand (2016)	Algoritmo Genético	KDD-CUP 99 e UNSW-NB15	KDD Normal - 5 KDD DoS - 4 KDD U2R - 4 KDD R2L - 3 KDD Sondagem - 8 USNW Normal - 7 USNW Fuzzers - 13 USNW Recon - 14 USNW Shellcode - 9 USNW DoS - 12 USNW Exploits - 6 USNW Genéricos - 9
Hasan et al. (2016)	Floresta Aleatória	KDD-CUP 99	25
Khaokaew e Anusas-amornkul (2016)	SVM Padrão CH-SVM CFS-SVM MDRP-SVM HFS-SVM	KDD-CUP 99	37 9 11 10 3
Osanaiye et al. (2016)	*Information Gain* *Gain Ratio* *chi-squared* *ReliefF* *Ensemble*	NSL-KDD	14 14 14 14 13
Gadal e Mokhtar (2017)	*ConsistencySebsetEvel* e Busca Genética	NSL-KDD	22

Khammassi e Krichen (2017)	Algoritmo Genético e Regressão Linear	KDD-CUP 99 e UNSW-NB15	1000 - 18 1500 - 16 2000 - 18 1000 - 18 1500 - 24 2000 - 20
Janarthanan e Zargari (2017)	-	KDD-CUP e UNSW-NB15	8 5
Moustafa e Slay (2017)	Ponto Central e Regra de associação de mineração	NSL-KDD e UNSW-NB15	11 (25%)
Ullah e Mahmoud (2017)	Ganho de Informação	NSL-KDD e ICSX	4 6
Zhu et al. (2017)	Algoritmo Genético	Gure-KDD e KDD-CUP 99	20
Alabi e Yurtkan (2018)	Entropia e Variância	UNB/IDS 2012	3
Khan et al. (2018)	Entropia e Computação Granular	NSL-KDD	6
Anwer, Farouk e Abdel-Hamid (2018)	*Ensemble* (*Gain Ratio Melhor*)	UNSW-NB15	18

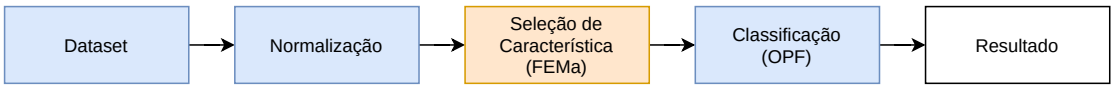
Divyasree e Sherly (2018)	*Chi Square* + função de peso	KDD-CUP 99	10
Aljawarneh, Aldwairi e Yassein (2018)	*Information Gain*	NSL-KDD	8
Garg e Batra (2018)	Entropia +*Information Gain*	NSL-KDD	Normal - 12 DoS - 10 Sondagem - 13 U2R - 15 R2L - 9
Selvakumar e Karuppiah (2019)	Informação Mutua e Informação Mutua de libélulas	KDD-CUP 99	10
Gottwalt, Chang e Dillon (2019)	Correlação de Característica	NSL-KDD e UNSW-NB15	6
Uysal et al. (2019)	Algoritmo genético (variado de NSGA-II)	KDD-CUP 99	22
Kasongo e Sun (2019)	*Information Gain*	NSL-KDD	21
Mazini, Shirazi e Mahdavi (2019)	Colônia artificial de abelhas	NSL-KDD e ICSXIDS2012	25
Palmieri (2019)	*Information Gain*	ISCX-UNB	4

Isa (2020)	Correlação de Pearson	KDD-CUP NSL-KDD CICIDS 2017	-
Kunhare, Tiwari e Dhar (2020)	PSO	NSL-KDD	10
Zhou et al. (2020)	Híbrido	NSL-KDD AWID CIC-IDS2017	10 8 13
Almomani (2020)	PSO GWO FFA GA Todos	UNSW-NB15	25 20 21 23 30
Chkirbene et al. (2020)	*Clustering*	NSL-KDD e UNSW-NB15	5
Zhang et al. (2020)	Filtro de Dominância	NSL-KDD	SVM - 18 KNN - 16

3 Metodologia

Neste capítulo serão abordadas ações elaboradas para a execução do experimento, é apresentado também o processo de execução do FEMa-FS. Na figura 11 é possível observar o macroprocesso que será executado para produção da avaliação dos testes em que são executados para avaliação de desempenho do FEMa-FS na seleção de características.

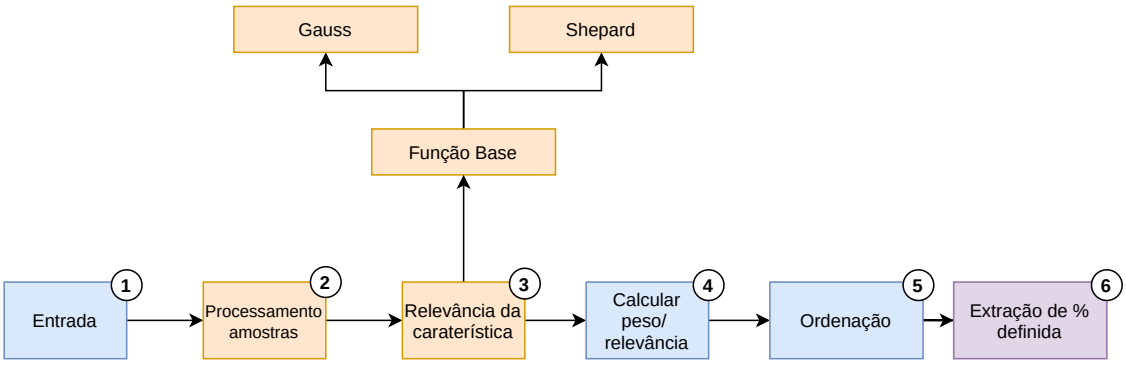
Figura 11 – Macro processo do experimento. Primeiramente é tratado as informações do conjunto do *dataset*, posteriormente se transforma os dados deixando-os entre 0 e 1 e posteriormente é realizado a seleção de característica, classificação e analisado o resultado da etapa de classificação.



Fonte: Elaborado pelo autor.

O processo detalhado das etapas executadas pelo FEMa-FS como um algoritmo de filtro para seleção de característica é explanado na figura 12. Durante o seu desenvolvimento foi encontrado importantes pontos de melhorias na etapa 2 e 3 que possibilitou uma redução significativa em relação ao tempo de processamento inicial, e que são relatadas no decorrer do texto.

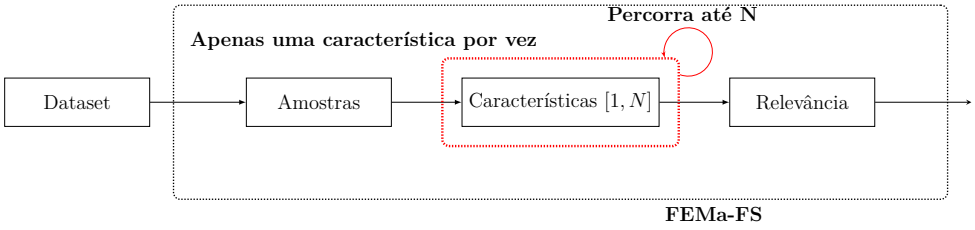
Figura 12 – Representação da arquitetura do FEMa-FS como seletor de característica. No primeiro passo é processado um conjunto de amostras que são reparadas e são enviadas para a função base estabelecida para processamento, posteriormente é calculado a importância de cada característica, ordenado por relevância iniciando da mais importante e com a última sendo menos importante. Finalmente é extraído a quantidade definida.



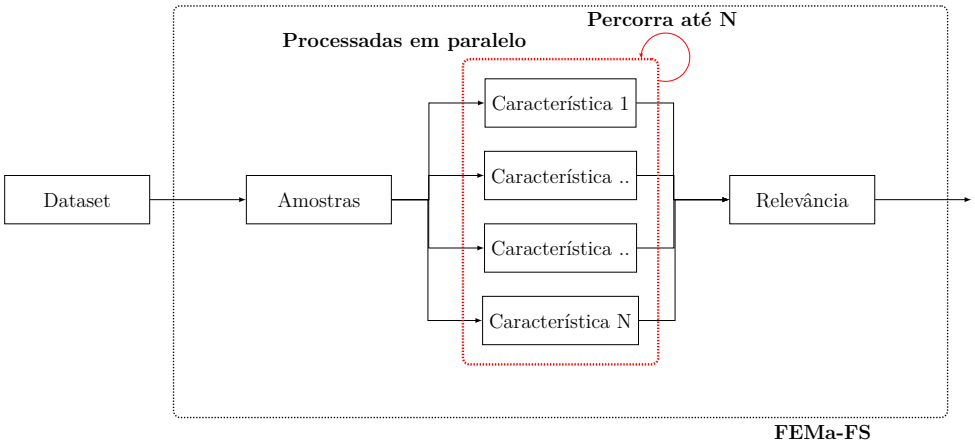
Fonte: Elaborado pelo autor.

A diferença entre a versão inicial e a paralelizada é observada na figura 13. Enquanto na figura 13a as informações da significância da característica eram processadas uma por vez e posteriormente armazenadas em um local sequêncial. Enquanto na figura 13b foi possível isolar a análise de cada caracaterística de maneira que o local de armazenamento do resultado não fosse sobrescrito permitindo assim alcançar uma paralelização.

Figura 13 – Demostração gráfica da parelização. Onde em 13a o passo de análise avaliava apenas uma característica por amostra. Na versão paralelizada 13b são avaliadas uma quantidade arbitrária de característica por vez.



(a) Versão Inicial



(b) Versão Paralelizada

3.1 Algoritmo de classificação

Para avaliar o desempenho do FEMa-FS como seletor de características foi utilizado o classificador *Optimum Path Forest* (OPF). A opção desse classificador é justificada pela sua utilização bem sucedida em trabalhos similares (PEREIRA et al., 2012), em que alcançou uma taxa de 99.8% na detecção de anomalias em redes de computadores usando seleção de características, e também por já ter sido utilizado para classificação em trabalhos anteriores realizados por Rodrigues et al. (2015) e Pereira et al. (2019).

3.2 Pré processamento dos dados

Devido à natureza determinística do OPF e a necessidade de formato numérico, todos os *datasets* foram normalizados para serem distribuídos em um intervalo padronizado. Os valores das características foram normalizados para o intervalo $[0, 1]$ utilizando $\log x + 1$ de maneira que valores que apresentavam métricas distintas não atuem de maneira negativa à análise do algoritmo.

3.3 *Datasets*

Os *datasets* NSL-KDD[2], ISCxTor2016[3] e UNSW-NB15[4] foram escolhidos para a avaliação, o primeiro foi selecionado por ser um *dataset* com algumas das anomalias em redes de computadores consideradas críticas, enquanto o segundo foi selecionado por trazer uma coleção de registros de tráfego normal e camuflado, sendo que o último pode ser considerado uma anomalia e o terceiro por possuir anomalias mais recentes e diversificadas em relação ao primeiro.

3.3.1 NSL-KDD

A base NSL-KDD é divida da seguinte maneira: 125,973 registros para treinamento enquanto para testes são fornecidos 22,544 registros. As anomalias neste conjunto estão divididas em quatro classes: DoS, Sondagem, U2R e R2l. As características se encontram detalhadas na tabela 3.

Tabela 3 – Características NSL-KDD

Característica	Descrição	Tipo
1. duration	Tempo de duração da conexão	Numérico
2. protocol_type	Protocolo usado na conexão	Nominal
3. service	Serviço usado no destino da comunicação	Nominal
4. flag	Estado da conexão	Nominal

[2] Disponível em: http://recogna.tech/files/datasets/nsl-kdd.tar.gz
[3] Disponível em: http://recogna.tech/files/datasets/tor-nontor.tar.gz
[4] Disponível em: https://researchdata.edu.au/unsw-nb15-dataset/1425943

5. src_bytes	Total de bytes transferidos da origem ao destino em uma única conexão	Numérico
6. dst_bytes	Total de bytes transferidos do destino para a origem em uma única conexão	Numérico
7. land	Se a origem e o destino forem o mesmo endereço IP e as portas utilizadas na conexão é a mesma, o valor é 1 se não 0	Binário
8. wrong_fragment	Total de fragmentos classificados como errados na conexão	Numérico
9. urgent	Número de pacotes urgentes na conexão	Numérico
10. hot	Representa o sucesso do ataque como, por exemplo, de entrar no sistema	Numérico
11. num_failed_logins	Contador de falhas de tentativa de acesso	Numérico

12. logged_in	Quando 1 o ataque foi capaz de se autenticar no sistema se não 0	Binário
13. num_compromised	Números de condições de comprometimento	Numérico
14. root_shell	1 se acesso administrativo foi obtido, se não 0	Binário
15. su_attempted	1 se foi tentado a obtenção de acesso administrativo, se não 0	Binário
16. num_root	Número de operações com acesso administrativo realizado na conexão	Numérico
17. num_file_creations	Quantidade de arquivos criados nesta conexão	Numérico
18. num_shells	Quantidade de sessões de usuários criado nesta conexão	Numérico
19. num_access_files	Números de arquivos acessados na conexão	Numérico

20. num_outbound_cmds	Quantidade de comandos de saída em uma sessão FTP.	Numérico
21. is_host_login	1 se o login de acesso faz parte da característica 10 "hot", se não 0	Binário
22. is_guest_login	1 se o login é "visitante"(*guest*), se não 0	Binário
23. count	Quantidade de conexões para o mesmo destino nos dois últimos segundos	Numérico
24. srv_count	Quantidade de conexões para o mesmo serviço nos dois últimos segundos	Numérico
25. serror_rate	Porcentagem de conexões que possuem 4. flags nos valores s0,s1,s2 ou s3, nas conexões agregadas da característica 23. count	Numérico
26. srv_serror_rate	Porcentagem de conexões que possuem 4. flags nos valores s0,s1,s2 ou s3, nas conexões agregadas da característica 24. srv_count	Numérico

27. rerror_rate	Porcentagem de conexões que possuem 4. flags no valor REJ, nas conexões agregadas da característica 23. count	Numérico
28. srv_rerror_rate	Porcentagem de conexões que possuem 4. flags nos valor REJ, nas conexões agregadas da característica 24. srv_count	Numérico
29. same_srv_rate	Porcentagem de conexões que possuem o mesmo serviços nas conexões agregadas da característica 23. count	Numérico
30. diff_srv_rate	Porcentagem de conexões que possuem serviços diferentes nas conexões agregadas da característica 23. count	Numérico
31. srv_diff_host_rate	Porcentagem de conexões para diferentes destinos nas conexões agregadas da característica 24. srv_count	Numérico
32. dst_host_count	Quantidade de conexões para o mesmo destino	Numérico

33. dst_host_srv_count	Quantidade de conexões utilizando a mesma porta de serviço	Numérico
34.dst_host_same_srv_rate	Porcentagem de conexões que utilizam o mesmo serviço nas conexões agregadas da características 32. dst_host_count	Numérico
35. dst_host_diff_srv_rate	Porcentagem de conexões que utilizam diferentes serviços nas conexões agregadas da características 32. dst_host_count	Numérico
36. dst_host_same_src_port_rate	Porcentagem de conexões que utilizam a mesma porta de conexão da origem nas conexões agregadas da características 33. dst_srv_count	Numérico
37. dst_host_srv_diff_host_rate	Porcentagem de conexões que possuem destinos diferentes nas conexões agregadas da características 33. dst_srv_count	Numérico

38. dst_host_serror_rate	Porcentagem de conexões que possuem 4. flags nos valores s0,s1,s2 ou s3, nas conexões agregadas da característica 32. dst_srv_count	Numérico
39. dst_host_srv_serror_rate	Porcentagem de conexões que possuem 4. flags nos valores s0,s1,s2 ou s3, nas conexões agregadas da característica 33. dst_host_srv_count	Numérico
40. dst_host_rerror_rate	Porcentagem de conexões que possuem 4. flags no valor REJ, nas conexões agregadas da característica 32. dst_host_count	Numérico
41. dst_host_srv_rerror_rate	Porcentagem de conexões que possuem 4. flags no valor REJ, nas conexões agregadas da característica 33. dst_host_srv_count	Numérico

3.3.2 ISCxTor2016

Diferente dos outros dois *datasets* o ICSxTor2016 não é um conjunto em que os ataques são identificados como anomalias, mas sim o de tráfego criptografado através de uma rede virtual privada. Embora não esteja relacionado diretamente com alguma ameaça, muitos ataques utilizam de tais práticas para sua execução. O conjunto se encontra divido com 10,745 para testes e 57,653 para treinamento, o descritivo de suas características se encontram na tabela 4.

Tabela 4 – Características ISCxTor2016

Característica	Descrição	Tipo
1. Source IP	Origem	Objeto
2. Source Port	Porta utilizada pela origem	Numérico
3. Destination IP	Destino	Objeto
4. Destination Port	Porta utilizada pelo destino	Numérico
5. Protocol	Denota protocolo utilizado	Numérico
6. Flow Duration	Duração da conexão	Numérico
7. Flow Bytes/s	Bytes por segundos enviado	Numérico
8. Flow Packets/s	Pacotes por segundos enviado, pode conter fragmentos "infinitos"	Objeto

9. Flow IAT Mean	Média do tempo de chegada do fluxo	Numérico
10. Flow IAT Std	Desvio padrão de chegada do fluxo	Numérico
11. Flow IAT Max	Máximo do tempo de chegada do fluxo	Numérico
12. Flow IAT Min	Mínimo do tempo de chegada do fluxo	Numérico
13. Fwd IAT Mean	Média do tempo de encaminhamento do fluxo	Numérico
14. Fwd IAT Std	Desvio padrão do tempo de encaminhamento do fluxo	Numérico
15. Fwd IAT Max	Máximo do tempo de encaminhamento do fluxo	Numérico
16. Fwd IAT Min	Mínimo do tempo de encaminhamento do fluxo	Numérico

17. Bwd IAT Mean	Média do tempo de encaminhamento reverso do fluxo	Numérico
18. Bwd IAT Std	Desvio padrão do tempo de encaminhamento reverso do fluxo	Numérico
19. Bwd IAT Max	Máximo do tempo de encaminhamento reverso do fluxo	Numérico
20. Bwd IAT Min	Mínimo do tempo de encaminhamento reverso do fluxo	Numérico
21. Active Mean	Média do tempo que a conexão se manteve ativa	Numérico
22. Active Std	Desvio padrão do tempo que a conexão se manteve ativa	Numérico
23. Active Max	Máximo do tempo que a conexão se manteve ativa	Numérico

24. Active Min	Mínimo do tempo que a conexão se manteve ativa	Numérico
25. Idle Mean	Média do tempo que a conexão se manteve ociosa	Numérico
26. Idle Std	Desvio padrão do tempo que a conexão se manteve ociosa	Numérico
27. Idle Max	Máximo do tempo que a conexão se manteve ociosa	Numérico
28. Idle Min	Mínimo do tempo que a conexão se manteve ociosa	Numérico

3.3.3 UNSW-NB15

Este *dataset* consiste em dois de amostras, sendo elas: amostras de operações legítimas e anomalias geradas sinteticamente. O conjunto é compost por dez classes, sendo uma exclusivamente para representar todo o tráfego legítimo, enquanto as outras nove representam as seguintes anomalias: Fuzzers, Analizadores (como por exemplo varredura de porta, spam de correio eletrônico e arquivos html), *Backdoor*, DoS, Exploits, Genéricos, *Reconnaissance*, *Shellcode* e Verme. Esta base é divido em duas partes treinamento e teste, na parte de testes existem cerca de $82,332$ registros, enquanto na de treinamento existem $175,341$.A seguir na tabela 5 se encontra o detalhamento das características.

Tabela 5 – Características UNSW-NB15

Característica	Descrição	Tipo
1. dur	Tempo de duração da conexão	Numérico
2. proto	Protocolo de transmissão	Nominal
3. service	Serviço utilizado, em caso desconhecido "-"	Nominal
4. state	Indica o estado do 3. service usando bandeiras TCP (i.e. SYN, ACK...), quando não, usasse "-"	Nominal
5. spkts	Contagem de pacotes a partir da origem	Numérico
6. dpkts	Contagem de pacotes a partir do destino	Numérico
7. sbytes	Contagem de *bytes* da origem ao destino	Numérico

8. dbytes	Contagem de *bytes* do destino à origem	Numérico
9. rate	Taxa média de conexão	Numérico
10. sttl	Tempo de vida da origem ao destino	Numérico
11. dttl	Tempo de vida do destino à origem	Numérico
12. sload	*bits* por segundo da origem	Numérico
13. dload	*bits* por segundo do destino	Numérico
14. sloss	Pacotes da origem retransmitidos	Numérico
15. dloss	Pacotes do destino retransmitidos	Numérico
16. sinpkt	Intervalos entre chegadas de pacotes da origem	Numérico

17. dinpkt	Intervalos entre chegadas de pacotes do destino	Numérico
18. sjit	Jitter da origem	Numérico
19. djit	Jitter do destino	Numérico
20. swin	Valor da janela de propagação de pacote TCP da origem	Numérico
21. stcpb	Número da sequência base do pacote TCP da origem	Numérico
22. dtcpb	Número da sequência base do pacote TCP do destino	Numérico
23. dwin	Valor da janela de propagação de pacote TCP do destino	Numérico
24. tcprtt	Tempo de configuração da conexão TCP (soma do 25. synack e 26. ackdat)	Numérico

25. synack	Tempo da troca de SYN para SYN ACK	Numérico
26. ackdat	Tempo da troca de SYN ACK para SYN	Numérico
27. smean	Média do fluxo dos pacotes TCP transmitidos pela origem	Numérico
28. dmean	Média do fluxo dos pacotes TCP transmitidos pelo destino	Numérico
29. trans_depth	Representa a profundidade do *pipeline* entre a requisição http e a resposta	Numérico
30. response_body_len	O tamanho do conteúdo transmitido pelo serviço http	Numérico
31. ct_srv_src	Quantidade de 3. services que possuam a mesma origem a cada 100 conexões	Numérico

32. ct_state_ttl	Quantidade de estados do 4. state a cada 100 conexões de 10. sttl e 11. dtll	Numérico
33. ct_dst_ltm	Quantidade de conexões que o mesmo destino a cada 100 conexões	Numérico
34. ct_src_dport_ltm	Quantidade de conexões com a mesma origem e porta de destino a cada 100 conexões	Numérico
35. ct_dst_sport_ltm	Quantidade de conexões com o mesmo destino e porta de origem a cada 100 conexões	Numérico
36. ct_dst_src_ltm	Quantidade de conexões com o mesmo destino e origem a cada 100 conexões	Numérico
37. is_ftp_login	Quando existir for um serviço ftp e houve acesso com usuário e senha 1, se não 0	Binário

38. ct_ftp_cmd	Quantidade de fluxos de comandos ftps executados na sessão	Numérico
39. ct_flw_http_mthd	Quantidade de fluxos de operações de *GET* e *POST* na sessão http	Numérico
40. ct_src_ltm	Quantidade de conexões que possuem a mesma origem a cada 100 conexões	Numérico
41. ct_srv_dst	Quantidade de conexões que possuem o mesmo 3. service e o mesmo destino a cada 100 conexões	Numérico
42. is_sm_ip_ports	Se a conexão possuir a mesma origem,destino,endereço e porta de comunicação	Binário
43. attack_cat	Categoria do ataque (no total são 9) + categorização normal	Nominal

3.4 Avaliação

Foram utilizadas duas funções base no experimento: inversão da potência de Shepard (SHEPARD, 1968) e a base de Gauss (YAMAKAWA; HYODO, 2005) em conjunto com a distância euclidiana (RAHAMAN; GHOSH; THIERY, 2021) para calcular a distância entre a localização esperada média e a real da característica. Os valores utilizados para critério de eficiência serão: a medida f e acurácia. Enquanto a quantidade de seleção de características para uma melhor avaliação, serão extraídas entre 10% até 60% pois eles se encontram entre os valores mais utilizadas na literatura como observado na tabela 2. O mesmo processo será executado utilizando CHI^2 e ANOVA no lugar do FEMa-FS-FS para comparar o seu desempenho.

Para cada um dos *dataset* mencionados na seção 3.3 foram realizados dois experimentos distintos. Para o primeiro grupo fora realizado os seguintes procedimentos:

- Extração aleatória de 10% do conjunto de teste e treinamento.
- Realização da classificação com o OPF sem seleção de característica.
- Execução do FEMa-FS utilizando Shepard como função base extraindo as seguintes quantidades de características: 10%, 15%, 20%, 25% 35%, 40%, 45%, 50%, 55%, 60% e classificação do resultado com o OPF.
- Repetido o processo do item anterior alterando somente a seleção de característica para CHI^2 e ANOVA.

Enquanto no segundo foram executadas as mesmas ações exceto que a função base utilizada foi a de Gauss:

- Extração aleatória de 10% do conjunto de teste e treinamento.
- Realização da classificação com o OPF sem seleção de característica.
- Execução do FEMa-FS utilizando Gauss como função base extraindo as seguintes quantidades de características: 10%, 15%, 20%, 25% 35%, 40%, 45%, 50%, 55%, 60% e classificação do resultado com o OPF.
- Repetido o processo do item anterior alterando somente a seleção de característica para CHI^2 e ANOVA.

Sendo assim, as atividades descritas nos procedimentos acima foram executadas 25 vezes e os resultados encontrados foram avaliados utilizando o teste de ranqueamento estatístico pareado de Wilcoxon com significância de 0,05. De acordo com Volpato e Barreto

(2016) o teste de Wilcoxon é um dos testes estatísticos mais eficiente para assegurar que os resultados obtidos nos testes sejam satisfatórios, identificar os casos reais de eficiência e avaliar se os resultados encontrados são distintos.

4 Resultados

Este capítulo descreve os resultados obtidos nos experimentos realizados para a avaliação deste trabalho.

Para execução dos experimentos foi utilizado um computador com sistema operacional Arch Linux (64 *bits*), processador da marca Intel modelo i7-3770K de 3.50 GHz e 24 GB de memória RAM DDR3 com frequência de 1600MHz. As classificações foram realizadas de maneira binária, ou seja, se foi identificado ou não uma anomalia, sem a identificação de algum tipo específico.

O algoritmo foi implementado em C[5], o compilador utilizado foi o Clang 11[6], enquanto a paralelização foi implementado com o conjunto OpenMP[7] na versão 4.5, presente no compilador.

4.1 Grupo 1 — FEMa com Shepard

Para os experimentos executados com Shepard foi identificado por testes de amostragem que os valores do controle da interpolação k seria necessário ser alterado para cada um dos *datasets* para que fosse alcançado o melhor desempenho, sendo assim os valores selecionados foram:

- **NSL-KDD**: $k = 5$;
- **ISCxTor2016**: $k = 6$;
- **UNSW-NB15**: $k = 6$.

Enquanto a paralelização das ações aumentou a eficiência de processamento do algoritmo em cerca de quatro vezes em relação à versão inicial alcançando os valores médios de:

- **NSL-KDD**: o tempo médio de processamento era cerca de 1.278 ± 55 segundos foi reduzido para 272 ± 12 segundos, obtendo uma redução de $78,72\%$;
- **ISCxTor2016**: foi de 176 ± 10 segundos para 40 ± 2 segundos, reduzindo em 77.28%;
- **UNSW-NB15**: reduziu de 2.954 ± 212 segundos para 717 ± 80 segundos, ou seja, foi alcançado uma redução de $75,73\%$.

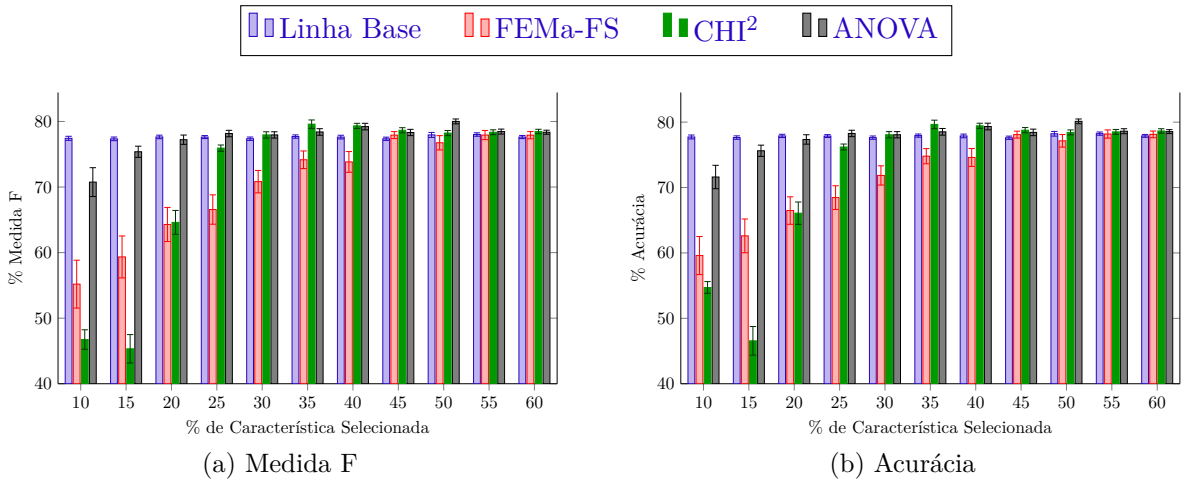
[5] Disponível em: https://github.com/lbiaggi/fema_cmake
[6] Disponível em: https://releases.llvm.org/download.html#11.0.0
[7] Detalhes sobre disponível em: https://www.openmp.org/wp-content/uploads/openmp-4.5.pdf

Com essa função foram identificados melhoras significativas para os *datasets* UNSW-NB15 e ICSxTor2016, enquanto na NSL-KDD manteve-se um resultado próximo ao da utilização de todas as características.

4.1.1 NSL-KDD

Na figura 14 temos os resultados do experimento na base NSL-KDD, estes obtidos por meio da avaliação média das execuções, nos gráficos 14a e 14b é possível notar que ocorreu um aumento quando utilizado o FEMa-FS nas extrações de 45% e 60% de características, houve um ganho de 0,56% e 0,28% respectivamente. Entretanto, as propostas com CHI^2 e ANOVA obtiveram um melhor resultado. O método FEM de Shepard, este não é capaz de trabalhar de maneira efetiva com dados dispersos, criando uma linha generalizada com poucos valores (THACKER et al., 2010).

Figura 14 – Resultados NSL-KDD dos experimentos grupo Shepard. Na figura encontram-se as comparações de eficiências do OPF sem nenhuma seleção de característica (Linha Base), em relação a modelos propostos com o FEMa-FS, CHI^2 e ANOVA. Na figura (a) temos a medida f, enquanto na (b) acurácia.



(a) Medida F (b) Acurácia

Fonte: Elaborado pelo autor.

O resultado do teste estatístico presente na tabela 6, demonstra que os ganhos identificados nos resultados são estaticamente similares aos resultados da Linha base, entretanto podemos observar o comportamento de similaridade ao aumentar a quantidade de caracaterśitica para esse conjunto.

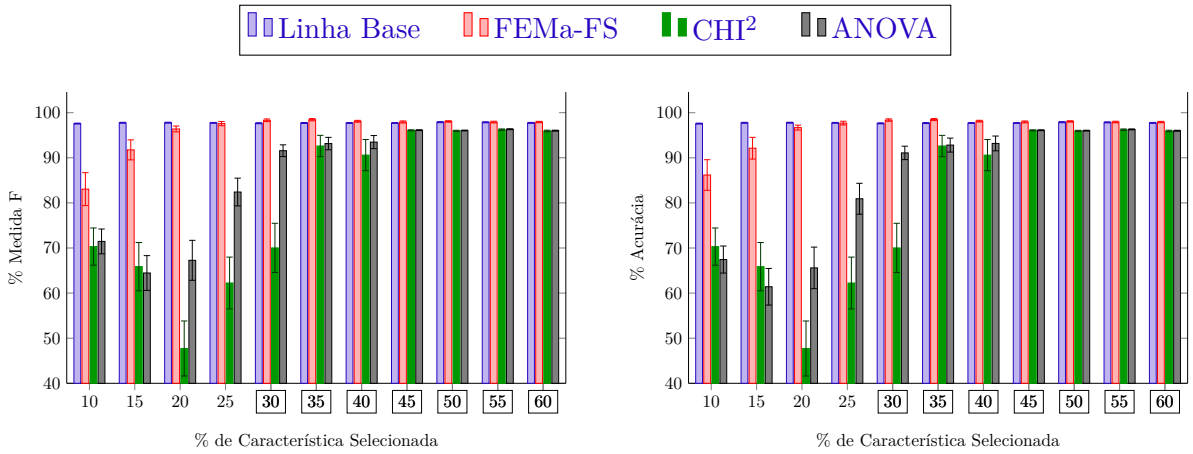
Tabela 6 – Resultados do teste estatístico medida F de significância com 5% de significância na NSL-KDD comparando o FEMa-FS usando Shepard com os outros modelos. Simbolo "-", representa onde o FEMa-FS teve uma efiência superior ao teste referenciado, o "+" representa o oposto do "-" e "=" significa que os resultados obtidos são similares.

Experimento	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	=	-	=	+	+	+	+	=	=	=	=
ANOVA	=	+	+	+	+	+	+	+	+	=	=
Linha Base	+	+	+	+	+	+	=	=	=	=	=

4.1.2 ISCxTor2016

Nos resultados com a base ICSxTor2016 dispostos na figura 15, os resultados médios são mais expressivos e significativos em relação ao anterior. Nos testes com 30%, 35%, 40%, 45%, 50%, 55% e 60% houve um aumento tanto na acurácia e medida f, sendo o maior aumento de 35% no qual obteve uma melhora de 0,73%. O FEMa-FS conseguiu superar os resultados obtidos por CHI^2 e ANOVA em todos os testes.

Figura 15 – Resultados ISCxTor2016 dos experimentos grupo Shepard. Na figura encontram-se as comparações de eficiências do OPF sem nenhuma seleção de característica (Linha Base), em relação a modelos propostos com o FEMa-FS, CHI^2 e ANOVA. Na figura (a) temos a medida f, enquanto na (b) acurácia. Os locais destacados representam os resultados superiores do FEMa-FS aos modelos comparados.



(a) Medida F (b) Acurácia

Fonte: Elaborado pelo autor.

Na avaliação estatística do experimento com a ISCxTor2016 encontrados na tabela 7 é possível observar que as porcentagens de 30% até 40% demonstram ganhos reais para o modelo proposto. Enquanto nos testes entre 45% e 60% não se mostraram significativos para a medida f e acurácia em relação à linha base, o modelo proposto foi capaz de superar CHI^2 e ANOVA.

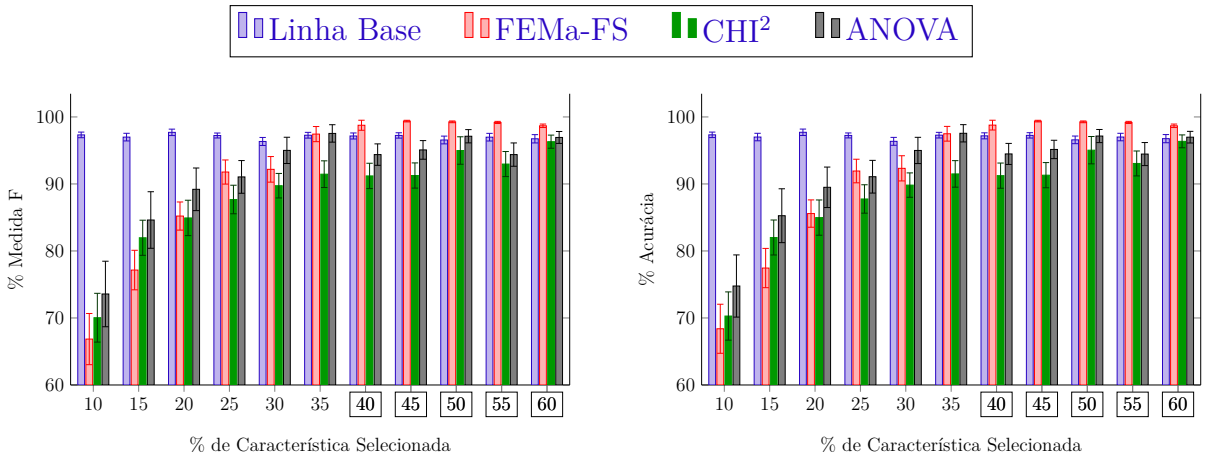
Tabela 7 – Resultados do teste estatístico medida F de significância de 5% no conjunto ICSxTor2016 comparando FEMa-FS com função base shepard com os outros modelos. Simbolo "-", representa onde o FEMa-FS teve uma efiência superior ao teste referenciado, o "+" representa o oposto do "-" e "=" significa que os resultados obtidos são similares.

Experimento	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	-	-	-	-	-	-	-	-	-	-	-
ANOVA	-	-	-	-	-	-	-	-	-	-	-
Linha Base	+	+	=	=	-	-	-	=	=	=	=

4.1.3 UNSW-NB15

Os resultados obtidos no dataset UNSW-NB15 foram positivos, ocorreu um aumento significativo em acurácia e medida f, como podemos observar nos gráficos 16a e 16b. A diferença entre os casos de 45%, 50%, 55% é de cerca de $0,1\%$ e o maior ganho foi de $2,14\%$ nos 45%. A apartir dos resultados de 40% FEMa-FS manteve-se na liderança dos melhores resultados.

Figura 16 – Resultados UNSW-NB15 dos experimentos grupo Shepard. Medida F (a) e Acurácia (b), onde o OPF sem seleção de característica é a Linha Base e os testes realizados com FEMa-FS, CHI^2 e ANOVA. Os resultados destacados com quadrados no eixo x são os locais que o FEMa-FS obteve os melhores resultados.



(a) F1-Score (b) Acurácia

Fonte: Elaborado pelo autor.

Como podemos observar nos testes de significância presentes na tabela 8, todos os testes acima de 35% não demostraram similaridade com a linha base, demonstrando a validação do ganho de eficiência. A partir de 40% com ANOVA, todos os resultados apontaram superioridade em relação ao FEMa-FS, não tendo diferença significativa. O modelo proposto demonstrou um melhor desmepenho nos testes quando comparado ao CHI^2 em 45% e 60%.

Tabela 8 – Resultados do teste estatístico de significância UNSW-NB15 Shepard. Simbolo "-", representa onde o FEMa-FS teve uma efiência superior ao teste referenciado, o "+" representa o oposto do "-" e "=" significa que os resultados obtidos são similares.

Test Case	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	=	=	=	=	=	=	=	-	=	=	-
ANOVA	=	+	=	=	=	+	=	=	=	=	=
Baseline	+	+	+	=	=	-	-	-	-	-	-

4.1.4 Detalhamento numérico dos resultados

A seguir os dados utilizados para a construção das figuras 14, 15 e 16 com o seu respectivo desvio padrão da seção anterior. Na tabela 9 as informações foram organizadas ascendentemente por porcentagem.

Tabela 9 – Resultados dos experimentos do grupo 1.

Dataset	Porcentagem	Método	Medida F	Acurácia
NSL-KDD	10%	FEMa-FS	$55,19\% \pm 3,65\%$	$59,6\% \pm 2,91\%$
		CHI^2	$46,74\% \pm 1,49\%$	$54,72\% \pm 0,9\%$
		ANOVA	$70,75\% \pm 2,20\%$	$71,61\% \pm 1,79\%$
		OPF	$77,43\% \pm 0,31\%$	$77,73\% \pm 0,3\%$
ISCxTor2016	10%	FEMa-FS	$83,05\% \pm 3,65\%$	$86,19\% \pm 3,39\%$
		CHI^2	$70,31\% \pm 4,13\%$	$66,25\% \pm 4,53\%$
		ANOVA	$71,46\% \pm 2,75\%$	$67,44\% \pm 3,0\%$
		OPF	$97,59\% \pm 0,1\%$	$97,59\% \pm 0,1\%$
UNSW-NB15	10%	FEMa-FS	$66,85\% \pm 3,82\%$	$68,39\% \pm 3,66\%$
		CHI^2	$70,04\% \pm 3,64\%$	$70,29\% \pm 3,6\%$
		ANOVA	$73,58\% \pm 4,89\%$	$74,77\% \pm 4,64\%$
		OPF	$97,32\% \pm 0,42\%$	$97,33\% \pm 0,41\%$
NSL-KDD	15%	FEMa-FS	$59,34\% \pm 3,2\%$	$62,61\% \pm 2,58\%$
		CHI^2	$45,33\% \pm 2,17\%$	$46,56\% \pm 2,19\%$
		ANOVA	$75,38\% \pm 0,85\%$	$75,62\% \pm 0,86\%$
		OPF	$77,35\% \pm 0,28\%$	$77,64\% \pm 0,26\%$
ISCxTor2016	15%	FEMa-FS	$91,76\% \pm 2,22\%$	$92,13\% \pm 2,41\%$
		CHI^2	$65,87\% \pm 5,35\%$	$62,7\% \pm 5,59\%$
		ANOVA	$64,46\% \pm 3,85\%$	$61,41\% \pm 4,06\%$
		OPF	$97,77\% \pm 0,1\%$	$97,77\% \pm 0,1\%$
UNSW-NB15	15%	FEMa-FS	$77,15\% \pm 2,95\%$	$77,45\% \pm 2,93\%$

		CHI^2	$81,97\% \pm 2,62\%$	$82,01\% \pm 2,61\%$
		ANOVA	$77,22\% \pm 0,71\%$	$77,35\% \pm 0,72\%$
		OPF	$96,99\% \pm 0,57\%$	$97\% \pm 0,56\%$
NSL-KDD	20%	FEMa-FS	$64,29\% \pm 2,6\%$	$66,48\% \pm 2,1\%$
		CHI^2	$64,61\% \pm 1,82\%$	$66,08\% \pm 1,71\%$
		ANOVA	$77,22\% \pm 0,71\%$	$77,35\% \pm 0,72\%$
		OPF	$77,64\% \pm 0,27\%$	$77,89\% \pm 0,25\%$
ISCxTor2016	20%	FEMa-FS	$96,39\% \pm 0,65\%$	$96,7\% \pm 0,54\%$
		CHI^2	$47,61\% \pm 6,1\%$	$45,28\% \pm 6,26\%$
		ANOVA	$67,22\% \pm 4,43\%$	$65,59\% \pm 4,61\%$
		OPF	$97,79\% \pm 0,1\%$	$97,79\% \pm 0,1\%$
UNSW-NB15	20%	FEMa-FS	$85,21\% \pm 2,1\%$	$85,58\% \pm 2,04\%$
		CHI^2	$84,93\% \pm 2,64\%$	$84,98\% \pm 2,63\%$
		ANOVA	$89,2\% \pm 3,18\%$	$89,5\% \pm 3,02\%$
		OPF	$97,7\% \pm 0,48\%$	$97,71\% \pm 0,48\%$
NSL-KDD	25%	FEMa-FS	$66,57\% \pm 2,24\%$	$68,46\% \pm 1,81\%$
		CHI^2	$75,94\% \pm 0,47\%$	$76,2\% \pm 0,45\%$
		ANOVA	$78,16\% \pm 0,49\%$	$78,27\% \pm 0,49\%$
		OPF	$77,62\% \pm 0,23\%$	$77,87\% \pm 0,22\%$
ISCxTor2016	25%	FEMa-FS	$97,57\% \pm 0,46\%$	$97,7\% \pm 0,41\%$
		CHI^2	$65,24\% \pm 5,74\%$	$59,25\% \pm 6,08\%$
		ANOVA	$82,41\% \pm 3,08\%$	$80,92\% \pm 3,43\%$
		OPF	$97,75\% \pm 0,09\%$	$97,76\% \pm 0,09\%$
UNSW-NB15	25%	FEMa-FS	$91,78\% \pm 1,8\%$	$91,93\% \pm 1,76\%$
		CHI^2	$87,68\% \pm 2,13\%$	$87,76\% \pm 2,12\%$
		ANOVA	$91,05\% \pm 2,13\%$	$87,76\% \pm 2,12\%$
		OPF	$97,23\% \pm 0,37\%$	$97,24\% \pm 0,37\%$
NSL-KDD	30%	FEMa-FS	$70,82\% \pm 1,71\%$	$71,85\% \pm 1,47\%$
		CHI^2	$77,95\% \pm 0,48\%$	$76,20\% \pm 0,48\%$
		ANOVA	$77,95\% \pm 0,48\%$	$78,07\% \pm 0,48\%$
		OPF	$77,37\% \pm 0,26\%$	$77,63\% \pm 0,25\%$
ISCxTor2016	30%	FEMa-FS	$98,32\% \pm 0,29\%$	$98,37\% \pm 0,27\%$
		CHI^2	$70,02\% \pm 5,46\%$	$67,32\% \pm 5,96\%$
		ANOVA	$91,57\% \pm 1,3\%$	$91,09\% \pm 1,49\%$
		OPF	$97,66\% \pm 0,11\%$	$97,64\% \pm 0,11\%$
UNSW-NB15	30%	FEMa-FS	$92,18\% \pm 1,91\%$	$92,33\% \pm 1,88\%$

		CHI^2	$89,74\% \pm 1,83\%$	$89,83\% \pm 1,82\%$
		ANOVA	$95,01\% \pm 1,96\%$	$95\% \pm 1,96\%$
		OPF	$96,34\% \pm 0,6\%$	$96,36\% \pm 0,59\%$
NSL-KDD	35%	FEMa-FS	$74,16\% \pm 1,34\%$	$74,79\% \pm 1,16\%$
		CHI^2	$79,59\% \pm 0,64\%$	$79,65\% \pm 0,65\%$
		ANOVA	$78,4\% \pm 0,5\%$	$78,52\% \pm 0,51\%$
		OPF	$77,7\% \pm 0,25\%$	$77,94\% \pm 0,24\%$
ISCxTor2016	35%	FEMa-FS	$98,45\% \pm 0,25\%$	$98,48\% \pm 0,23\%$
		CHI^2	$92,62\% \pm 2,36\%$	$92,3\% \pm 2,56\%$
		ANOVA	$93,15\% \pm 1,38\%$	$92,82\% \pm 1,55\%$
		OPF	$97,72\% \pm 0,11\%$	$97,71\% \pm 0,11\%$
UNSW-NB15	35%	FEMa-FS	$97,44\% \pm 1,13\%$	$97,49\% \pm 1,1\%$
		CHI^2	$91,46\% \pm 1,99\%$	$91,5\% \pm 1,98\%$
		ANOVA	$97,54\% \pm 1,3\%$	$97,56\% \pm 1,29\%$
		OPF	$97,27\% \pm 0,43\%$	$97,28\% \pm 0,42\%$
NSL-KDD	40%	FEMa-FS	$73,83\% \pm 1,58\%$	$74,6\% \pm 1,37\%$
		CHI^2	$79,33\% \pm 0,39\%$	$79,44\% \pm 0,39\%$
		ANOVA	$79,22\% \pm 0,5\%$	$78,52\% \pm 0,51\%$
		OPF	$77,62\% \pm 0,28\%$	$77,89\% \pm 0,28\%$
ISCxTor2016	40%	FEMa-FS	$98,1\% \pm 0,2\%$	$98,13\% \pm 0,19\%$
		CHI^2	$90,6\% \pm 3,45\%$	$90,24\% \pm 3,64\%$
		ANOVA	$93,48\% \pm 1,45\%$	$93,19\% \pm 1,64\%$
		OPF	$97,72\% \pm 0,09\%$	$97,72\% \pm 0,09\%$
UNSW-NB15	40%	FEMa-FS	$98,76\% \pm 0,75\%$	$98,78\% \pm 0,73\%$
		CHI^2	$91,2\% \pm 1,89\%$	$91,24\% \pm 1,88\%$
		ANOVA	$94,38\% \pm 1,6\%$	$94,48\% \pm 1,57\%$
		OPF	$97,16\% \pm 0,45\%$	$97,18\% \pm 0,45\%$
NSL-KDD	45%	FEMa-FS	$77,92\% \pm 0,53\%$	$78,09\% \pm 0,53\%$
		CHI^2	$78,66\% \pm 0,39\%$	$78,77\% \pm 0,4\%$
		ANOVA	$78,32\% \pm 0,48\%$	$78,43\% \pm 0,5\%$
		OPF	$77,36\% \pm 0,25\%$	$77,61\% \pm 0,25\%$
ISCxTor2016	45%	FEMa-FS	$97,94\% \pm 0,23\%$	$97,96\% \pm 0,21\%$
		CHI^2	$96,09\% \pm 0,11\%$	$96,08\% \pm 0,11\%$
		ANOVA	$96,12\% \pm 0,09\%$	$96,11\% \pm 0,09\%$
		OPF	$97,72\% \pm 0,08\%$	$97,72\% \pm 0,08\%$
UNSW-NB15	45%	FEMa-FS	$99,38\% \pm 0,12\%$	$99,38\% \pm 0,12\%$

		CHI^2	$91,26\% \pm 1,88\%$	$91,31\% \pm 1,88\%$
		ANOVA	$95,08\% \pm 1,39\%$	$95,15\% \pm 1,37\%$
		OPF	$97,24\% \pm 0,4\%$	$97,25\% \pm 0,4\%$
NSL-KDD	50%	FEMa-FS	$76,75\% \pm 1,09\%$	$77,14\% \pm 0,96\%$
		CHI^2	$78,23\% \pm 0,39\%$	$78,43\% \pm 0,38\%$
		ANOVA	$80,01\% \pm 0,38\%$	$80,12\% \pm 0,36\%$
		OPF	$77,95\% \pm 0,38\%$	$78,22\% \pm 0,36\%$
ISCxTor2016	50%	FEMa-FS	$98,06\% \pm 0,16\%$	$98,07\% \pm 0,15\%$
		CHI^2	$95,97\% \pm 0,14\%$	$95,94\% \pm 0,14\%$
		ANOVA	$96,05\% \pm 0,09\%$	$96,03\% \pm 0,09\%$
		OPF	$97,9\% \pm 0,12\%$	$97,9\% \pm 0,12\%$
UNSW-NB15	50%	FEMa-FS	$99,27\% \pm 0,13\%$	$99,27\% \pm 0,13\%$
		CHI^2	$94,99\% \pm 2,05\%$	$95,04\% \pm 2,04\%$
		ANOVA	$97,12\% \pm 0,99\%$	$97,15\% \pm 0,98\%$
		OPF	$96,56\% \pm 0,58\%$	$96,58\% \pm 0,57\%$
NSL-KDD	55%	FEMa-FS	$77,93\% \pm 0,7\%$	$78,18\% \pm 0,63\%$
		CHI^2	$78,36\% \pm 0,37\%$	$78,51\% \pm 0,37\%$
		ANOVA	$78,46\% \pm 0,38\%$	$78,61\% \pm 0,38\%$
		OPF	$78,01\% \pm 0,26\%$	$78,24\% \pm 0,25\%$
ISCxTor2016	55%	FEMa-FS	$97,92\% \pm 0,19\%$	$97,93\% \pm 0,18\%$
		CHI^2	$96,24\% \pm 0,16\%$	$96,21\% \pm 0,15\%$
		ANOVA	$96,33\% \pm 0,1\%$	$96,3\% \pm 0,1\%$
		OPF	$97,87\% \pm 0,11\%$	$97,86\% \pm 0,11\%$
UNSW-NB15	55	FEMa-FS	$99,18\% \pm 0,16\%$	$99,18\% \pm 0,16\%$
		CHI^2	$92,97\% \pm 1,87\%$	$93,06\% \pm 1,86\%$
		ANOVA	$94,38\% \pm 1,74\%$	$94,46\% \pm 1,72\%$
		OPF	$96,98\% \pm 0,57\%$	$97\% \pm 0,56\%$
NSL-KDD	60%	FEMa-FS	$77,91\% \pm 0,55\%$	$78,13\% \pm 0,51\%$
		CHI^2	$78,47\% \pm 0,34\%$	$78,65\% \pm 0,34\%$
		ANOVA	$78,36\% \pm 0,31\%$	$78,55\% \pm 0,3\%$
		OPF	$77,63\% \pm 0,24\%$	$77,9\% \pm 0,22\%$
ISCxTor2016	60%	FEMa-FS	$97,93\% \pm 0,15\%$	$97,93\% \pm 0,15\%$
		CHI^2	$95,95\% \pm 0,16\%$	$96,21\% \pm 0,12\%$
		ANOVA	$96,33\% \pm 0,1\%$	$96,3\% \pm 0,1\%$
		OPF	$97,75\% \pm 0,1\%$	$97,75\% \pm 0,1\%$
UNSW-NB15	60%	FEMa-FS	$98,68\% \pm 0,26\%$	$98,68\% \pm 0,26\%$

CHI^2	$96,31\% \pm 0,98\%$	$96,36\% \pm 0,96\%$
ANOVA	$96,94\% \pm 0,98\%$	$96,98\% \pm 0,87\%$
OPF	$96,74\% \pm 0,62\%$	$96,76\% \pm 0,61\%$

4.2 Grupo 2 — FEMa com Gauss

Para a função Gauss a definição de k também foi estabelecida seguindo a mesma prática do grupo 1 por amostragem e os valores encontrados foram:

- **NSL-KDD**: $k = 6$;
- **ISCxTor2016**: $k = 6$;
- **UNSW-NB15**: $k = 9$.

Enquanto a paralelização contou com um ganho próximo de cinco vezes alcançando resultados como:

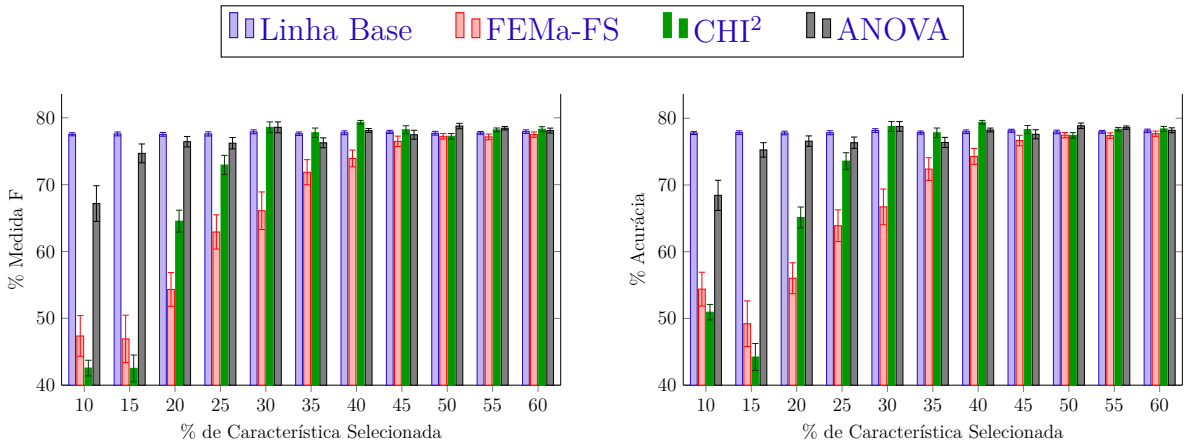
- **NSL-KDD**: de 1.278 ± 55 segundos foi reduzido para 241 ± 9 segundos, obtendo uma redução próxima de 81.15%;
- **ISCxTor2016**: foi de 176 ± 10 segundos para 29 ± 2 segundos, reduzindo em aproximadamente 83.52%;
- **UNSW-NB15**: reduziu de 2.954 ± 212 segundos para 507 ± 38 segundos, ou seja, reduziram-se 82.84%.

A função não alcançou um ganho significativo na medida f e de acurácia em nenhum dos três conjuntos avaliados, mesmo assim o método FEM Gaussiano foi capaz de manter um resultado próximo ao da utilização de todas as características com pequenas quedas na sua eficácia nos testes executados nos conjunto NSL-KDD e ICSxTor2016, entretanto, com a USNW-NB15, a seleção de característica não manteve o nível que nas outras.

4.2.1 NSL-KDD

Os resultados presentes nos gráficos 17a e 17b se comparados com os resultados obtidos utilizando Shepard, notamos um comportamento similar, podendo estar relacionado à distância das características em relação ao ponto identificado para a mesma, fazendo com que características com pouca significância obtenham valores próximos das importantes e consecutivamente sendo selecionados pela proposta.

Figura 17 – Resultados NSL-KDD dos experimentos grupo Gauss. Na figura encontram-se as comparações de eficiências do OPF sem nenhuma seleção de característica (Linha Base), em relação a modelos propostos com o FEMa-FS, CHI^2 e ANOVA. Na figura (a) temos a medida f, enquanto na (b) acurácia.



(a) Medida F (b) Acurácia

Fonte: Elaborado pelo autor.

Com os resultados dos testes estátisticos da tabela 10, observamos que embora o FEMa-FS com base de Gauss também não conseguiu um resultado de otimização direto muito dos testes deram similaridades com CHI^2 e ANOVA nos cenários menores que 40%. No teste de 35% ambos foram melhor que a Linha Base.

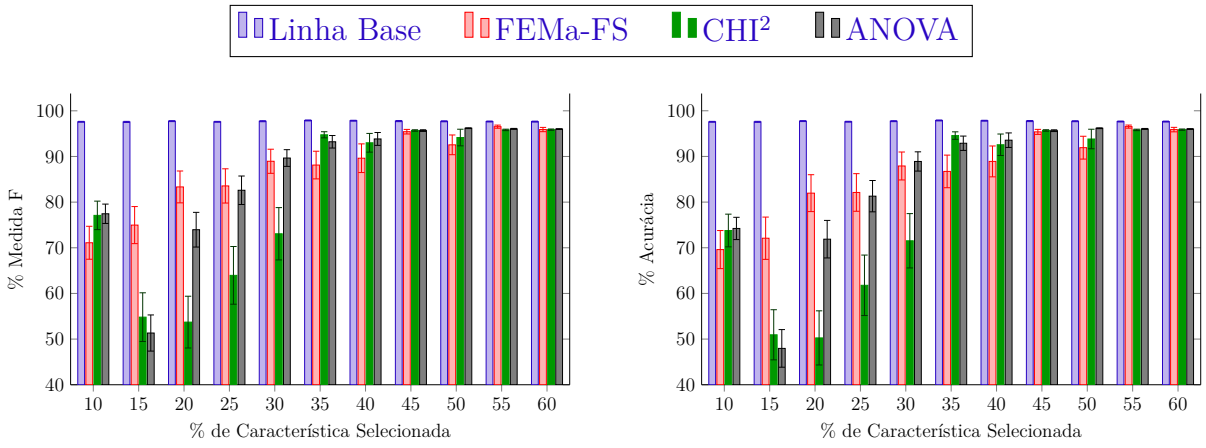
Tabela 10 – Resultados do teste estatístico de significância NSL-KDD com Gauss. Simbolo "-", representa onde o FEMa-FS teve uma efiência superior ao teste referenciado, o "+" representa o oposto do "-" e "=" significa que os resultados obtidos são similares.

Experimento	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	=	=	=	+	+	=	=	=	=	+	+
ANOVA	=	=	+	+	+	=	+	+	+	+	+
Linha Base	+	+	+	+	+	+	+	=	=	=	+

4.2.2 ISCxTor2016

Seguindo as tendências apresentadas nos resultados do conjunto NSL-KDD os resultados presentes nos gráficos 18 apresentam a mesma estabilidade, ou seja, mesmo que não tenha sido capaz de aumentar a eficácia do OPF, o filtro manteve um resultado próximo utilizando menos recursos com pequenas variações que podem ser consideradas aceitáveis.

Figura 18 – Resultados ISCxTor2016 dos experimentos grupo Gauss. Na figura encontram-se as comparações de eficiências do OPF sem nenhuma seleção de característica (Linha Base), em relação a modelos propostos com o FEMa-FS, CHI^2 e ANOVA. Na figura (a) temos a medida f, enquanto na (b) acurácia.



(a) Medida F (b) Acurácia

Fonte: Elaborado pelo autor.

Neste *dataset* o comportamento do que houve nos testes com Shepard, CHI^2 e ANOVA não foram capazes de auxiliar o OPF. Diferente do que houve também com o FEMa-FS usando Shepard, o FEMa-FS com Gauss também não foi capaz de auxiliar o classificador. Os testes estatísticos disponíveis na tabela 11 demonstram que dessa vez não houve nenhuma similaridade entre a Linha Base e o modelo proposto com o FEMa-FS e que houve uma forte similaridade dele com os outros modelos testados.

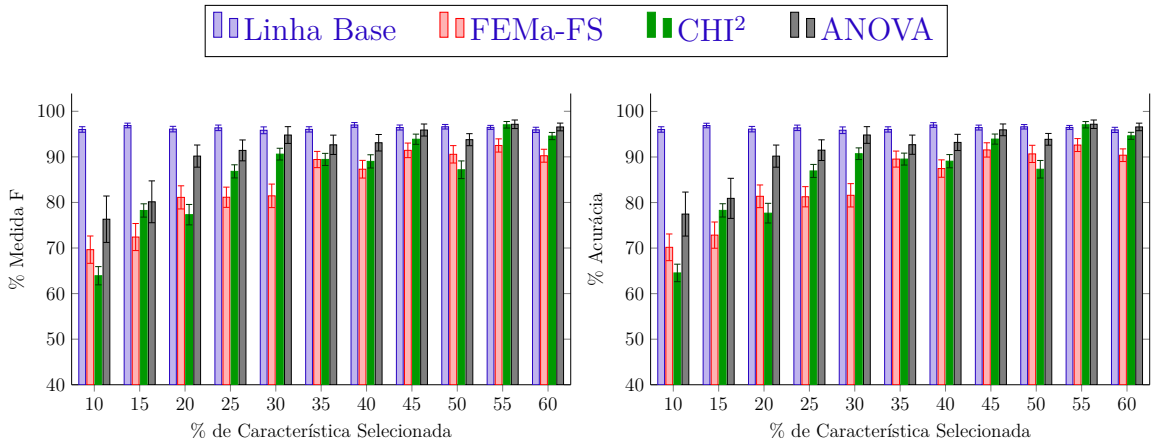
Tabela 11 – Resultados do teste estatístico de significância ISCxTor2016 com Gauss.

Experimento	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	=	-	-	-	-	=	=	=	=	+	=
ANOVA	=	-	=	=	=	=	+	=	=	=	=
Linha Base	+	+	+	+	+	+	+	+	+	+	+

4.2.3 UNSW-NB15

Os testes com Gauss atingiram diferentes resultados quando comparado com Shepard, a seleção de característica acabou comprometendo de maneira negativa o classificador, como pode ser observado na tabela 19 demonstrando que a base de Gauss não foi capaz de ser efetiva.

Figura 19 – Resultados UNSW-NB15 dos experimentos grupo Gauss. Na figura encontram-se as comparações de eficiências do OPF sem nenhuma seleção de característica (Linha Base), em relação a modelos propostos com o FEMa-FS, CHI^2 e ANOVA. Na figura (a) temos a medida f, enquanto na (b) acurácia.



(a) Medida F (b) Acurácia

Fonte: Elaborado pelo autor.

Ao observamos os testes estátisticos presentes na tabela 12 do contexto, podemos observar à tendência de similaridade novamente do modelo proposto com CHI^2 como ocorreu no teste do mesmo conjunto com Shepard, podendo ser interpretado como positivo, pois 50% do teste com CHI^2 foi superior à linha base, já em relação ao ANOVA seu compartamento diferiu, teve uma quantidade bem menor de similaridade, em contrapartida, aos testes com Shepard. A similaridade com a linha base foi a mesma quantidade, porém com maior quantidade de características.

Tabela 12 – Resultados do teste estatístico de significância USNW-NB15 com Gauss. Simbolo "-", representa onde o FEMa-FS teve uma efiência superior ao teste referenciado, o "+" representa o oposto do "-" e "=" significa que os resultados obtidos são similares.

Experimento	10%	15%	20%	25%	30%	35%	40%	45%	50%	55%	60%
CHI^2	=	=	=	+	+	+	=	=	=	+	+
ANOVA	=	=	+	+	+	=	+	+	+	+	+
Linha Base	+	+	+	+	+	+	+	=	+	=	+

4.2.4 Detalhamento numérico dos resultados

Os dados foram utilizados para geração das análises propostas de seleção de características apresentadas nas figuras 17, 18 e 19 com um método de elemento finito Gaussiano apresentados na seção anterior. A tabela 13 segue o mesmo padrão da tabela descrita que sumariza os testes do grupo usando Shepard.

Tabela 13 – Resultados dos experimentos do grupo 2.

Dataset	Porcentagem	Método	Medida F	Acurácia
NSL-KDD	10%	FEMa-FS	$47,36\% \pm 3,04\%$	$54,39\% \pm 2,54\%$
		CHI^2	$42,56\% \pm 1,17\%$	$50,94\% \pm 1,16\%$
		ANOVA	$67,17\% \pm 2,67\%$	$68,45\% \pm 2,26\%$
		OPF	$77,54\% \pm 0,23\%$	$77,79\% \pm 0,22\%$
ISCxTor2016	10%	FEMa-FS	$71,1\% \pm 3,61\%$	$69,6\% \pm 4,17\%$
		CHI^2	$77,1\% \pm 3,12\%$	$73,78\% \pm 3,58\%$
		ANOVA	$77,45\% \pm 2,12\%$	$74,23\% \pm 2,44\%$
		OPF	$97,6\% \pm 0,09\%$	$97,59\% \pm 0,09\%$
UNSW-NB15	10%	FEMa-FS	$69,64\% \pm 3\%$	$70,17\% \pm 2,93\%$
		CHI^2	$63,92\% \pm 2\%$	$64,55\% \pm 1,92\%$
		ANOVA	$76,32\% \pm 5,09\%$	$92,82\% \pm 4,83\%$
		OPF	$96,01\% \pm 0,61\%$	$96,04\% \pm 0,6\%$
NSL-KDD	15%	FEMa-FS	$46,92\% \pm 3,55\%$	$49,21\% \pm 3,42\%$
		CHI^2	$42,52\% \pm 1,99\%$	$44,22\% \pm 2,03\%$
		ANOVA	$74,68\% \pm 1,41\%$	$75,26\% \pm 1,08\%$
		OPF	$77,59\% \pm 0,29\%$	$77,85\% \pm 0,27\%$
ISCxTor2016	15%	FEMa-FS	$74,97\% \pm 4,06\%$	$72,09\% \pm 4,63\%$
		CHI^2	$54,81\% \pm 5,33\%$	$50,94\% \pm 5,49\%$
		ANOVA	$51,33\% \pm 3,96\%$	$47,96\% \pm 4,12\%$
		OPF	$97,57\% \pm 0,12\%$	$97,58\% \pm 0,11\%$
UNSW-NB15	15%	FEMa-FS	$72,41\% \pm 2,97\%$	$72,85\% \pm 2,88\%$
		CHI^2	$78,24\% \pm 1,46\%$	$78,28\% \pm 1,46\%$
		ANOVA	$80,14\% \pm 4,59\%$	$80,92\% \pm 4,4\%$
		OPF	$96,89\% \pm 0,51\%$	$96,91\% \pm 0,5\%$
NSL-KDD	20%	FEMa-FS	$54,32\% \pm 2,52\%$	$56,03\% \pm 2,31\%$
		CHI^2	$64,54\% \pm 1,64\%$	$65,13\% \pm 1,57\%$
		ANOVA	$76,43\% \pm 0,78\%$	$76,58\% \pm 0,78\%$
		OPF	$77,53\% \pm 0,28\%$	$77,79\% \pm 0,27\%$
ISCxTor2016	20%	FEMa-FS	$83,33\% \pm 3,49\%$	$81,96\% \pm 4,04\%$
		CHI^2	$53,72\% \pm 5,67\%$	$50,26\% \pm 5,93\%$
		ANOVA	$73,96\% \pm 3,8\%$	$71,88\% \pm 4,11\%$
		OPF	$97,74\% \pm 0,1\%$	$97,74\% \pm 0,1\%$
UNSW-NB15	20%	FEMa-FS	$81,11\% \pm 2,54\%$	$81,37\% \pm 2,49\%$
		CHI^2	$77,32\% \pm 2,24\%$	$77,67\% \pm 2,16\%$

		ANOVA	$90,17\% \pm 2,43\%$	$90,19\% \pm 2,43\%$
		OPF	$96,09\% \pm 0,61\%$	$96,11\% \pm 0,6\%$
NSL-KDD	25%	FEMa-FS	$62,92\% \pm 2,56\%$	$63,9\% \pm 2,38\%$
		CHI^2	$72,94\% \pm 1,44\%$	$73,59\% \pm 1,25\%$
		ANOVA	$76,2\% \pm 0,85\%$	$76,34\% \pm 0,85\%$
		OPF	$77,59\% \pm 0,31\%$	$77,83\% \pm 0,3\%$
ISCxTor2016	25%	FEMa-FS	$83,55\% \pm 3,75\%$	$82,11\% \pm 4,12\%$
		CHI^2	$63,96\% \pm 6,32\%$	$61,78\% \pm 6,62\%$
		ANOVA	$82,6\% \pm 3,12\%$	$81,3\% \pm 3,43\%$
		OPF	$97,6\% \pm 0,09\%$	$97,61\% \pm 0,09\%$
UNSW-NB15	25%	FEMa-FS	$81,14\% \pm 2,22\%$	$81,27\% \pm 2,22\%$
		CHI^2	$86,82\% \pm 1,44\%$	$86,94\% \pm 1,43\%$
		ANOVA	$91,42\% \pm 2,29\%$	$91,5\% \pm 2,27\%$
		OPF	$96,37\% \pm 0,61\%$	$96,4\% \pm 0,6\%$
NSL-KDD	30%	FEMa-FS	$66,1\% \pm 2,81\%$	$66,72\% \pm 2,68\%$
		CHI^2	$78,58\% \pm 0,81\%$	$78,78\% \pm 0,78\%$
		ANOVA	$78,58\% \pm 0,81\%$	$78,78\% \pm 0,78\%$
		OPF	$77,9\% \pm 0,3\%$	$78,16\% \pm 0,28\%$
ISCxTor2016	30%	FEMa-FS	$88,95\% \pm 2,64\%$	$87,91\% \pm 3,06\%$
		CHI^2	$73,07\% \pm 5,73\%$	$71,53\% \pm 5,95\%$
		ANOVA	$89,66\% \pm 1,83\%$	$88,9\% \pm 2,11\%$
		OPF	$97,72\% \pm 0,11\%$	$97,72\% \pm 0,1\%$
UNSW-NB15	30%	FEMa-FS	$81,45\% \pm 2,58\%$	$81,59\% \pm 2,56\%$
		CHI^2	$90,6\% \pm 1,29\%$	$90,72\% \pm 1,28\%$
		ANOVA	$94,79\% \pm 1,84\%$	$94,83\% \pm 1,83\%$
		OPF	$95,83\% \pm 0,75\%$	$95,87\% \pm 0,73\%$
NSL-KDD	35%	FEMa-FS	$71,86\% \pm 1,88\%$	$72,39\% \pm 1,7\%$
		CHI^2	$77,77\% \pm 0,7\%$	$77,83\% \pm 0,71\%$
		ANOVA	$76,6\% \pm 0,74\%$	$76,38\% \pm 0,76\%$
		OPF	$77,62\% \pm 0,26\%$	$77,85\% \pm 0,24\%$
ISCxTor2016	35%	FEMa-FS	$88,12\% \pm 3,03\%$	$86,72\% \pm 3,57\%$
		CHI^2	$94,75\% \pm 0,67\%$	$94,57\% \pm 0,82\%$
		ANOVA	$93,22\% \pm 1,37\%$	$92,89\% \pm 1,57\%$
		OPF	$97,89\% \pm 0,1\%$	$97,88\% \pm 0,1\%$
UNSW-NB15	35%	FEMa-FS	$89,42\% \pm 1,78\%$	$89,54\% \pm 1,76\%$

		CHI^2	$89,44\% \pm 1,33\%$	$89,54\% \pm 1,31\%$
		ANOVA	$92,65\% \pm 2,12\%$	$92,7\% \pm 2,11\%$
		OPF	$96,03\% \pm 0,58\%$	$96,05\% \pm 0,58\%$
NSL-KDD	40%	FEMa-FS	$73,92\% \pm 1,26\%$	$74,29\% \pm 1,19\%$
		CHI^2	$79,33\% \pm 0,28\%$	$79,4\% \pm 0,28\%$
		ANOVA	$78,11\% \pm 0,29\%$	$78,24\% \pm 0,29\%$
		OPF	$77,76\% \pm 0,29\%$	$77,98\% \pm 0,27\%$
ISCxTor2016	40%	FEMa-FS	$89,62\% \pm 3,14\%$	$88,91\% \pm 3,38\%$
		CHI^2	$93\% \pm 2,05\%$	$92,56\% \pm 2,36\%$
		ANOVA	$93,84\% \pm 1,41\%$	$93,56\% \pm 1,59\%$
		OPF	$97,84\% \pm 0,09\%$	$97,83\% \pm 0,09\%$
UNSW-NB15	40%	FEMa-FS	$87,29\% \pm 1,94\%$	$87,46\% \pm 1,91\%$
		CHI^2	$89,01\% \pm 1,45\%$	$89,08\% \pm 1,45\%$
		ANOVA	$93,1\% \pm 1,81\%$	$93,2\% \pm 1,78\%$
		OPF	$97\% \pm 0,53\%$	$97,02\% \pm 0,52\%$
NSL-KDD	45%	FEMa-FS	$76,48\% \pm 0,78\%$	$76,66\% \pm 0,79\%$
		CHI^2	$78,22\% \pm 0,59\%$	$78,32\% \pm 0,6\%$
		ANOVA	$77,47\% \pm 0,67\%$	$77,6\% \pm 0,68\%$
		OPF	$77,89\% \pm 0,24\%$	$78,13\% \pm 0,23\%$
ISCxTor2016	45%	FEMa-FS	$95,43\% \pm 0,5\%$	$95,38\% \pm 0,55\%$
		CHI^2	$95,69\% \pm 0,16\%$	$95,69\% \pm 0,16\%$
		ANOVA	$95,67\% \pm 0,18\%$	$95,64\% \pm 0,19\%$
		OPF	$97,75\% \pm 0,12\%$	$97,75\% \pm 0,12\%$
UNSW-NB15	45%	FEMa-FS	$91,43\% \pm 1,6\%$	$91,55\% \pm 1,57\%$
		CHI^2	$93,87\% \pm 1,12\%$	$93,93\% \pm 1,11\%$
		ANOVA	$95,89\% \pm 1,32\%$	$95,96\% \pm 1,29\%$
		OPF	$96,44\% \pm 0,55\%$	$96,46\% \pm 0,54\%$
NSL-KDD	50%	FEMa-FS	$77,24\% \pm 0,41\%$	$77,47\% \pm 0,39\%$
		CHI^2	$77,62\% \pm 0,41\%$	$77,44\% \pm 0,41\%$
		ANOVA	$78,77\% \pm 0,41\%$	$78,88\% \pm 0,41\%$
		OPF	$77,7\% \pm 0,28\%$	$77,95\% \pm 0,27\%$
ISCxTor2016	50%	FEMa-FS	$92,55\% \pm 2,16\%$	$91,92\% \pm 2,5\%$
		CHI^2	$94,15\% \pm 1,82\%$	$93,82\% \pm 2,14\%$
		ANOVA	$96,19\% \pm 0,08\%$	$96,17\% \pm 0,08\%$
		OPF	$97,7\% \pm 0,1\%$	$97,71\% \pm 0,1\%$
UNSW-NB15	50%	FEMa-FS	$90,56\% \pm 1,91\%$	$90,68\% \pm 1,89\%$

		CHI^2	$87,17\% \pm 1,94\%$	$87,31\% \pm 1,93\%$
		ANOVA	$93,77\% \pm 1,32\%$	$93,88\% \pm 1,29\%$
		OPF	$96,61\% \pm 0,5\%$	$96,63\% \pm 0,49\%$
NSL-KDD	55%	FEMa-FS	$77,15\% \pm 0,43\%$	$77,41\% \pm 0,41\%$
		CHI^2	$78,2\% \pm 0,28\%$	$78,36\% \pm 0,27\%$
		ANOVA	$78,45\% \pm 0,26\%$	$78,6\% \pm 0,25\%$
		OPF	$77,74\% \pm 0,21\%$	$77,97\% \pm 0,21\%$
ISCxTor2016	55%	FEMa-FS	$96,53\% \pm 0,34\%$	$96,53\% \pm 0,34\%$
		CHI^2	$95,82\% \pm 0,16\%$	$95,81\% \pm 0,16\%$
		ANOVA	$96,05\% \pm 0,1\%$	$96,03\% \pm 0,1\%$
		OPF	$97,66\% \pm 0,1\%$	$97,65\% \pm 0,1\%$
UNSW-NB15	55%	FEMa-FS	$92,5\% \pm 1,45\%$	$92,62\% \pm 1,42\%$
		CHI^2	$97,04\% \pm 0,72\%$	$97,08\% \pm 0,7\%$
		ANOVA	$97,15\% \pm 0,94\%$	$97,19\% \pm 0,93\%$
		OPF	$96,47\% \pm 0,44\%$	$96,49\% \pm 0,44\%$
NSL-KDD	60%	FEMa-FS	$77,49\% \pm 0,4\%$	$77,69\% \pm 0,39\%$
		CHI^2	$78,3\% \pm 0,37\%$	$78,41\% \pm 0,37\%$
		ANOVA	$78,09\% \pm 0,37\%$	$78,2\% \pm 0,37\%$
		OPF	$77,93\% \pm 0,27\%$	$78,12\% \pm 0,26\%$
ISCxTor2016	60%	FEMa-FS	$95,9\% \pm 0,47\%$	$95,88\% \pm 0,49\%$
		CHI^2	$95,88\% \pm 0,14\%$	$95,87\% \pm 0,14\%$
		ANOVA	$96\% \pm 0,08\%$	$96\% \pm 0,08\%$
		OPF	$97,64\% \pm 0,1\%$	$97,64\% \pm 0,1\%$
UNSW-NB15	60%	FEMa-FS	$90,25\% \pm 1,42\%$	$90,39\% \pm 1,39\%$
		CHI^2	$94,57\% \pm 0,79\%$	$94,65\% \pm 0,77\%$
		ANOVA	$96,56\% \pm 0,85\%$	$96,6\% \pm 0,83\%$
		OPF	$95,93\% \pm 0,57\%$	$95,95\% \pm 0,57\%$

5 Considerações finais

O trabalho proposto apresentou um novo modelo de filtro para seleção de característica baseado no recente trabalho de classificação utilizando métodos de elementos finitos FEMa-FS. Ao invés de utilizar a distância euclidiana para classificar amostras como projetado originalmente, foi ajustado para identificar a distância das características.

O FEMa-FS diferente de outros métodos de seleção de características dispensa a necessidade de uma etapa formal de treinamento quando utilizado uma função base que possua partição de unidade interpoladora. Em situações de avaliação de grandes volumes e de rápida decisão como a segurança de um ambiente virtual, tal propriedade pode aumentar a eficiência e durante o ciclo de desenvolvimento foi possível inserir novas ações que possibilitaram reduzir o seu tempo de processamento.

Quando testado com a função de interpolação inversa de Shepard em dois (UNSW-NB15 e ISCxTor2016) dos três *dataset* testados, o método proposto de seleção de característica conseguiu aumentar o desempenho do OPF. Com o NSL-KDD identificou-se que as características encontram-se distantes da linha traçada. Tal situação gerou uma distância genérica com poucos valores distintos, esse comportamento foi identificado em situações adversas pelos pesquisadores Thacker et al. (2010) em outras utilizações da base.

Nos testes utilizando uma função base Gaussiana não foi possível aumentar a eficácia do classificador utilizado, no entanto, a queda de eficiência pode ser um efeito de degradação aceitável, se considerar a diferença do tempo de execução em relação aos testes com Shepard.

Quando comparado com modelos já consolidados como CHI^2 e ANOVA, o FEMa-FS foi capaz de ser superior a eles. Nos testes com Gauss ele alcançou similaridade com os testes que foram superior em relação à Linha Base em sua grande maioria.

Sendo assim, com os cenários executados e os testes de Wilcoxon identificando em sua grande maioria os resultados foram distintos em relação à linha base, conclui-se que o FEMa-FS é uma possível ferramenta para seleção de características para anomalias e possui uma alta capacidade de adaptação. Com os resultados obtidos nos experimentos executados conclui-se que a utilização de funções base que dependam da distribuição normal, não são interessantes para seleção de características.

5.1 Trabalhos Futuros

Como trabalho futuro, planeja-se a implementação da solução proposta por Thacker et al. (2010) para obtenção de uma melhor generalização, além da utilização de es-

trutura de dados espaciais que possuam particionamento como arvores K-D para o armazenamento de informações. A utilização de outras métricas de distância como Geometria do táxi e Hamming ao invés somente da euclidiana.

Referências

ABRAHAM, Brendan; MANDYA, Abhijith; BAPAT, Rohan; ALALI, Fatma; BROWN, Don E.; VEERARAGHAVAN, Malathi. A Comparison of Machine Learning Approaches to Detect Botnet Traffic. In: 2018 International Joint Conference on Neural Networks (IJCNN). Jul. 2018. P. 1–8. DOI: 10.1109/IJCNN.2018.8489096.

AGHDAM, Mehdi Hosseinzadeh; KABIRI, Peyman. Feature Selection for Intrusion Detection System Using Ant Colony Optimization, p. 13, 2016.

AHMED, Mohiuddin; NASER MAHMOOD, Abdun; HU, Jiankun. A Survey of Network Anomaly Detection Techniques. **Journal of Network and Computer Applications**, v. 60, p. 19–31, 1 jan. 2016. ISSN 1084-8045. DOI: 10.1016/j.jnca.2015.11.016. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804515002891>. Acesso em: 29 mai. 2020.

AHSAN, Mostofa; GOMES, Rahul; CHOWDHURY, Md. Minhaz; NYGARD, Kendall E. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. **Journal of Cybersecurity and Privacy**, v. 1, n. 1, p. 199–218, 2021. ISSN 2624-800X. DOI: 10.3390/jcp1010011.

ALABI, Ruth; YURTKAN, Kamil. Entropy-Based Feature Selection for Network Anomaly Detection. In: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). Out. 2018. P. 1–7. DOI: 10.1109/ISMSIT.2018.8566694.

ALELYANI, Salem; TANG, Jiliang; LIU, Huan. Feature Selection for Clustering: A Review. In: AGGARWAL, Charu C.; REDDY, Chandan K. (Ed.). **Data Clustering**. First: Chapman and Hall/CRC, set. 2018. P. 29–60. ISBN 978-1-315-37351-5. DOI: 10.1201/9781315373515-2.

ALJAWARNEH, Shadi; ALDWAIRI, Monther; YASSEIN, Muneer Bani. Anomaly-Based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model. **Journal of Computational Science**, v. 25, p. 152–160, 1 mar. 2018. ISSN 1877-7503. DOI: 10.1016/j.jocs.2017.03.006. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877750316305099>. Acesso em: 10 out. 2020.

ALKASASSBEH, Mouhammd. An Empirical Evaluation for the Intrusion Detection Features Based on Machine Learning and Feature Selection Methods. **arXiv:1712.09623 [cs]**, dez. 2017. arXiv: 1712.09623 [cs].

ALMOMANI, Omar. A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms. **Symmetry**, Multidisciplinary Digital Publishing Institute, v. 12, n. 6, p. 1046, 6 jun. 2020. DOI: 10.3390/sym12061046.

AMBUSAIDI, Mohammed A.; HE, Xiangjian; NANDA, Priyadarsi; TAN, Zhiyuan. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. **IEEE Trans. Comput.**, v. 65, n. 10, p. 2986–2998, out. 2016. ISSN 1557-9956. DOI: 10.1109/TC.2016.2519914.

ANWER, Hebatallah Mostafa; FAROUK, Mohamed; ABDEL-HAMID, Ayman. A Framework for Efficient Network Anomaly Intrusion Detection with Features Selection. In: 2018 9TH International Conference ON Information AND Communication Systems (ICICS). **2018 9th International Conference on Information and Communication Systems (ICICS)**. Abr. 2018. P. 157–162. DOI: 10.1109/IACS.2018.8355459.

BERRY, Michael W.; MOHAMED, Azlinah; YAP, Bee Wah (Ed.). **Supervised and Unsupervised Learning for Data Science**. Cham: Springer International Publishing, 2020. (Unsupervised and Semi-Supervised Learning). ISBN 978-3-030-22474-5 978-3-030-22475-2. DOI: 10.1007/978-3-030-22475-2.

BHATIA, Randeep; BENNO, Steven; ESTEBAN, Jairo; LAKSHMAN, T. V.; GROGAN, John. Unsupervised Machine Learning for Network-Centric Anomaly Detection in IoT. en. In: PROCEEDINGS of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks - Big-DAMA '19. Orlando, FL, USA: ACM Press, 2019. P. 42–48. ISBN 978-1-4503-6999-2. DOI: 10.1145/3359992.3366641.

BIONDI, Fabrizio; ENESCU, Michael A.; GIVEN-WILSON, Thomas; LEGAY, Axel; NOUREDDINE, Lamine; VERMA, Vivek. Effective, Efficient, and Robust Packing Detection and Classification. en. **Computers & Security**, v. 85, p. 436–451, ago. 2019. ISSN 0167-4048. DOI: 10.1016/j.cose.2019.05.007.

BROWNLEE, Jason. **Data Preparation for Machine Learning**. 2020. 380 p. Disponível em: <https://machinelearningmastery.com/>.

CAI, Biao; ZENG, Lina; WANG, Yanpeng; LI, Hongjun; HU, Yanmei. Community Detection Method Based on Node Density, Degree Centrality, and K-Means Clustering in Complex Network. **Entropy**, Multidisciplinary Digital Publishing Institute, v. 21, n. 12, p. 1145, 12 dez. 2019. DOI: 10.3390/e21121145. Disponível em: <https://www.mdpi.com/1099-4300/21/12/1145>. Acesso em: 5 out. 2020.

CHEN, Zhaomin; YEO, Chai Kiat; FRANCIS, Bu Sung Lee; LAU, Chiew Tong. Combining MIC Feature Selection and Feature-Based MSPCA for Network Traffic Anomaly Detection. In: 2016 Third International Conference ON Digital Information

Processing, Data Mining, AND Wireless Communications (DIPDMWC). **2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)**. Jul. 2016. P. 176–181. DOI: 10.1109/DIPDMWC.2016.7529385.

CHKIRBENE, Zina; ERBAD, Aiman; HAMILA, Ridha; MOHAMED, Amr; GUIZANI, Mohsen; HAMDI, Mounir. TIDCS: A Dynamic Intrusion Detection and Classification System Based Feature Selection. **IEEE Access**, v. 8, p. 95864–95877, 2020. ISSN 2169-3536. DOI: 10.1109/ACCESS.2020.2994931.

DELL'ACCIO, Francesco; DI TOMMASO, Filomena; GONNELLI, Domenico. Comparison of Shepard's Like Methods with Different Basis Functions. In_____. **Numerical Computations: Theory and Algorithms**. Cham: Springer International Publishing, 2020. (Lecture Notes in Computer Science), p. 47–55. ISBN 978-3-030-39081-5. DOI: 10.1007/978-3-030-39081-5_6.

DENG, Xinyang. An Improved Method to Construct Basic Probability Assignment Based on the Confusion Matrix for Classification Problem. **Inf. Sci.**, p. 12, 2016.

DIVYASREE, T. H.; SHERLY, K. K. A Network Intrusion Detection System Based On Ensemble CVM Using Efficient Feature Selection Approach. **Procedia Computer Science**, v. 143, p. 442–449, 1 jan. 2018. ISSN 1877-0509. DOI: 10.1016/j.procs.2018.10.416. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050918321136>. Acesso em: 10 out. 2020.

DONG, Guozhu; LIU, Huan (Ed.). **Feature Engineering for Machine Learning and Data Analytics**. Edição: 1: CRC Press, 2018.

F.Y, Osisanwo; J.E.T, Akinsola; O, Awodele; J. O, Hinmikaiye; O, Olakanmi; J, Akinjobi. Supervised Machine Learning Algorithms: Classification and Comparison. en. **IJCTT**, v. 48, n. 3, p. 128–138, jun. 2017. ISSN 22312803. DOI: 10.14445/22312803/IJCTT-V48P126.

FADLULLAH, Zubair Md.; TANG, Fengxiao; MAO, Bomin; KATO, Nei; AKASHI, Osamu; INOUE, Takeru; MIZUTANI, Kimihiro. State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. **IEEE Communications Surveys Tutorials**, v. 19, n. 4, p. 2432–2455, 2017. ISSN 1553-877X. DOI: 10.1109/COMST.2017.2707140.

FALCÃO, Filipe; ZOPPI, Tommaso; SILVA, Caio Barbosa Viera; SANTOS, Anderson; FONSECA, Baldoino; CECCARELLI, Andrea; BONDAVALLI, Andrea. Quantitative Comparison of Unsupervised Anomaly Detection Algorithms for Intrusion Detection. en. In: PROCEEDINGS of the 34th ACM/SIGAPP Symposium on Applied Computing.

Limassol Cyprus: ACM, abr. 2019. P. 318–327. ISBN 978-1-4503-5933-7. DOI: 10.1145/3297280.3297314.

FARIS, Hossam; AL-ZOUBI, Ala' M.; HEIDARI, Ali Asghar; ALJARAH, Ibrahim; MAFARJA, Majdi; HASSONAH, Mohammad A.; FUJITA, Hamido. An Intelligent System for Spam Detection and Identification of the Most Relevant Features Based on Evolutionary Random Weight Networks. en. **Information Fusion**, v. 48, p. 67–83, ago. 2019. ISSN 1566-2535. DOI: 10.1016/j.inffus.2018.08.002.

FENNER, Mark. **Machine Learning with Python for Everyone**. 1 edition. Boston, MA: Addison-Wesley Professional, ago. 2019. ISBN 978-0-13-484562-3.

FERNANDES, Gilberto; RODRIGUES, Joel J. P. C.; CARVALHO, Luiz Fernando; AL-MUHTADI, Jalal F.; PROENÇA, Mario Lemes. A Comprehensive Survey on Network Anomaly Detection. **Telecommun Syst**, v. 70, n. 3, p. 447–489, 1 mar. 2019. ISSN 1572-9451. DOI: 10.1007/s11235-018-0475-8. Disponível em: <https://doi.org/10.1007/s11235-018-0475-8>. Acesso em: 11 out. 2020.

FERNÁNDEZ MAIMÓ, Lorenzo; PERALES GÓMEZ, Ángel Luis; GARCÍA CLEMENTE, Félix J.; GIL PÉREZ, Manuel; MARTÍNEZ PÉREZ, Gregorio. A Self-Adaptive Deep Learning-Based System for Anomaly Detection in 5G Networks. **IEEE Access**, v. 6, p. 7700–7712, 2018. ISSN 2169-3536. DOI: 10.1109/ACCESS.2018.2803446.

GADAL, Saad Mohamed Ali Mohamed; MOKHTAR, Rania A. Anomaly Detection Approach Using Hybrid Algorithm of Data Mining Technique. In: 2017 International Conference ON Communication, Control, Computing AND Electronics Engineering (ICCCCEE). **2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)**. Jan. 2017. P. 1–6. DOI: 10.1109/ICCCCEE.2017.7867661.

GARG, Sahil; BATRA, Shalini. Fuzzified Cuckoo Based Clustering Technique for Network Anomaly Detection. **Computers & Electrical Engineering**, v. 71, p. 798–817, 1 out. 2018. ISSN 0045-7906. DOI: 10.1016/j.compeleceng.2017.07.008.

GARG, Sahil; KAUR, Kuljeet; KUMAR, Neeraj; RODRIGUES, Joel J. P. C. Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective. **IEEE Transactions on Multimedia**, v. 21, n. 3, p. 566–578, mar. 2019. ISSN 1941-0077. DOI: 10.1109/TMM.2019.2893549.

GHARAEE, Hossein; HOSSEINVAND, Hamid. A New Feature Selection IDS Based on Genetic Algorithm and SVM. In: 2016 8TH International Symposium ON Telecommunications (IST). **2016 8th International Symposium on Telecommunications (IST)**. Set. 2016. P. 139–144. DOI: 10.1109/ISTEL.2016.7881798.

GOTTWALT, Florian; CHANG, Elizabeth; DILLON, Tharam. CorrCorr: A Feature Selection Method for Multivariate Correlation Network Anomaly Detection Techniques. en. **Computers & Security**, v. 83, p. 234–245, jun. 2019. ISSN 0167-4048. DOI: 10.1016/j.cose.2019.02.008.

HAMAMOTO, Anderson Hiroshi; CARVALHO, Luiz Fernando; SAMPAIO, Lucas Dias Hiera; ABRÃO, Taufik; PROENÇA, Mario Lemes. Network Anomaly Detection System Using Genetic Algorithm and Fuzzy Logic. en. **Expert Systems with Applications**, v. 92, p. 390–402, fev. 2018. ISSN 0957-4174. DOI: 10.1016/j.eswa.2017.09.013.

HASAN, Md Al Mehedi; NASSER, Mohammed; AHMAD, Shamim; MOLLA, Khademul Islam. Feature Selection for Intrusion Detection Using Random Forest. **J. Inf. Secur.**, Scientific Research Publishing, v. 7, n. 3, p. 129–140, 3 1 abr. 2016. DOI: 10.4236/jis.2016.73009. Disponível em: <http://www.scirp.org/Journal/Paperabs.aspx?paperid=65359>. Acesso em: 8 out. 2020.

ISA, Fuad Mat. Optimizing the Effectiveness of Intrusion Detection System by Using Pearson Correlation and Tune Model Hyper Parameter on Microsoft Azure Platform. **IJATCSE**, v. 9, n. 1.3, p. 132–138, 25 jun. 2020. ISSN 22783091. DOI: 10.30534/ijatcse/2020/1991.32020. Disponível em: <http://www.warse.org/IJATCSE/static/pdf/file/ijatcse19913sl2020.pdf>. Acesso em: 6 out. 2020.

JANARTHANAN, Tharmini; ZARGARI, Shahrzad. Feature Selection in UNSW-NB15 and KDDCUP'99 Datasets. In: 2017 IEEE 26TH International Symposium ON Industrial Electronics (ISIE). **2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)**. Jun. 2017. P. 1881–1886. DOI: 10.1109/ISIE.2017.8001537.

KASONGO, Sydney M; SUN, Yanxia. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. **Journal of Big Data**, Springer, v. 7, n. 1, p. 1–20, 2020.

KASONGO, Sydney Mambwe; SUN, Yanxia. A Deep Learning Method With Filter Based Feature Engineering for Wireless Intrusion Detection System. **IEEE Access**, v. 7, p. 38597–38607, 2019. ISSN 2169-3536. DOI: 10.1109/ACCESS.2019.2905633.

KHAMMASSI, Chaouki; KRICHEN, Saoussen. A GA-LR Wrapper Approach for Feature Selection in Network Intrusion Detection. en. **Computers & Security**, v. 70, p. 255–277, set. 2017. ISSN 0167-4048. DOI: 10.1016/j.cose.2017.06.005.

KHAN, Suleman; GANI, Abdullah; WAHAB, Ainuddin Wahid Abdul; SINGH, Prem Kumar. Feature Selection of Denial-of-Service Attacks Using Entropy and Granular Computing. en. **Arab J Sci Eng**, v. 43, n. 2, p. 499–508, fev. 2018. ISSN 2193-567X, 2191-4281. DOI: 10.1007/s13369-017-2634-8.

KHAOKAEW, Yonchanok; ANUSAS-AMORNKUL, Tanapat. A Performance Comparison of Feature Selection Techniques with SVM for Network Anomaly Detection. In: 2016 4TH International Symposium ON Computational AND Business Intelligence (ISCBI). **2016 4th International Symposium on Computational and Business Intelligence (ISCBI)**. Set. 2016. P. 85–89. DOI: 10.1109/ISCBI.2016.7743263.

KUHN, Max; JOHNSON, Kjell. **Feature Engineering and Selection: A Practical Approach for Predictive Models**. 1ª Edição: Chapman and Hall/CRC, 25 jul. 2019. 310 p.

KUNHARE, Nilesh; TIWARI, Ritu; DHAR, Joydip. Particle Swarm Optimization and Feature Selection for Intrusion Detection System. en. **Sādhanā**, v. 45, n. 1, p. 109, dez. 2020. ISSN 0256-2499, 0973-7677. DOI: 10.1007/s12046-020-1308-5.

LIU, Wei; CI, LinLin; LIU, LiPing. A New Method of Fuzzy Support Vector Machine Algorithm for Intrusion Detection. en. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 3, p. 1065, jan. 2020. DOI: 10.3390/app10031065.

MATEL, Elmer C.; SISON, Ariel M.; MEDINA, Ruji P. Optimization of Network Intrusion Detection System Using Genetic Algorithm with Improved Feature Selection Technique. en. In: 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM). Laoag, Philippines: IEEE, nov. 2019. P. 1–6. ISBN 978-1-72813-044-6. DOI: 10.1109/HNICEM48295.2019.9073439.

MAZINI, Mehrnaz; SHIRAZI, Babak; MAHDAVI, Iraj. Anomaly Network-Based Intrusion Detection System Using a Reliable Hybrid Artificial Bee Colony and AdaBoost Algorithms. en. **Journal of King Saud University - Computer and Information Sciences**, v. 31, n. 4, p. 541–553, out. 2019. ISSN 1319-1578. DOI: 10.1016/j.jksuci.2018.03.011.

MISHRA, Preeti; VARADHARAJAN, Vijay; TUPAKULA, Uday; PILLI, Emmanuel S. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. **IEEE Communications Surveys Tutorials**, v. 21, n. 1, p. 686–728, 2019. ISSN 1553-877X. DOI: 10.1109/COMST.2018.2847722.

MOUSTAFA, Nour; SLAY, Jill. A Hybrid Feature Selection for Network Intrusion Detection Systems: Central Points. **Proc. 16th Aust. Inf. Warf. Conf. Pp 5-13**, held on the 30 November - 2 December, 2017. DOI: 10.4225/75/57a84d4fbefbb. arXiv:

1707.05505. Disponível em: <http://arxiv.org/abs/1707.05505>. Acesso em: 9 out. 2020.

NAWIR, Mukrimah; AMIR, Amiza; YAAKOB, Naimah; LYNN, Ong Bi. Effective and Efficient Network Anomaly Detection System Using Machine Learning Algorithm. en. **Bulletin of Electrical Engineering and Informatics**, v. 8, n. 1, p. 46-51–51, jun. 2019. ISSN 2302-9285. DOI: 10.11591/eei.v8i1.1387.

ODEN, J. T.; REDDY, J. N.; ENGINEERING. **An Introduction to the Mathematical Theory of Finite Elements**. Mineola, N.Y: Dover Publications, 20 abr. 2011. ISBN 978-0-486-46299-8.

OLADIPUPO, Taiwo. Types of Machine Learning Algorithms. In: ZHANG, Yagang (Ed.). **New Advances in Machine Learning**. InTech, fev. 2010. ISBN 978-953-307-034-6. DOI: 10.5772/9385.

OSANAIYE, Opeyemi; CAI, Haibin; CHOO, Kim-Kwang Raymond; DEHGHANTANHA, Ali; XU, Zheng; DLODLO, Mqhele. Ensemble-Based Multi-Filter Feature Selection Method for DDoS Detection in Cloud Computing. **J Wireless Com Network**, v. 2016, n. 1, p. 130, 10 mai. 2016. ISSN 1687-1499. DOI: 10.1186/s13638-016-0623-3. Disponível em: <https://doi.org/10.1186/s13638-016-0623-3>. Acesso em: 8 out. 2020.

PALMIERI, Francesco. Network Anomaly Detection Based on Logistic Regression of Nonlinear Chaotic Invariants. en. **Journal of Network and Computer Applications**, v. 148, p. 102460, dez. 2019. ISSN 10848045. DOI: 10.1016/j.jnca.2019.102460.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. **International Journal of Imaging Systems and Technology**, John Wiley & Sons, Inc., New York, NY, USA, v. 19, n. 2, p. 120–131, 2009.

PEREIRA, Clayton R.; NAKAMURA, Rodrigo Y. M.; COSTA, Kelton A. P.; PAPA, João P. An Optimum-Path Forest Framework for Intrusion Detection in Computer Networks. **Engineering Applications of Artificial Intelligence**, v. 25, n. 6, p. 1226–1234, 1 set. 2012. ISSN 0952-1976. DOI: 10.1016/j.engappai.2012.03.008.

PEREIRA, Clayton Reginaldo; PASSOS, Leandro Aparecido; RODRIGUES, Douglas; SOUZA, André Nunes de; PAPA, João P. JADE-based feature selection for non-technical losses detection. In: SPRINGER. ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing. 2019. P. 141–156.

PEREIRA, Danilo; PITERI, Marco Antonio; SOUZA, André; PAPA, João Paulo; ADELI, Hojjat. FEMa: A Finite Element Machine for Fast Learning. **Neural Comput & Applic**, v. 32, n. 10, p. 6393–6404, mai. 2020. ISSN 0941-0643, 1433-3058. DOI: 10.1007/s00521-019-04146-4.

PINDER, George Francis. **Numerical Methods for Solving Partial Differential Equations: A Comprehensive Introduction for Scientists and Engineers**. Hoboken, NJ, USA: John Wiley and Sons, Inc, 2018. 295 p. ISBN 978-1-119-31611-4.

RAHAMAN, Rahul; GHOSH, Atin; THIERY, Alexandre H. **Pretrained Equivariant Features Improve Unsupervised Landmark Discovery**. 7 abr. 2021. arXiv: 2104.02925 [cs].

RODRIGUES, Douglas; ROSA, Gustavo Henrique de; PASSOS, Leandro Aparecido; PAPA, João Paulo. Adaptive improved flower pollination algorithm for global optimization. In: NATURE-INSPIRED Computation in Data Mining and Machine Learning. Springer, 2020. P. 1–21.

RODRIGUES, Douglas; YANG, Xin-She; DE SOUZA, André Nunes; PAPA, João Paulo. Binary flower pollination algorithm and its application to feature selection. In: RECENT advances in swarm intelligence and evolutionary computation. Springer, 2015. P. 85–100.

SAMMUT, Claude; WEBB, Geoffrey I. **Encyclopedia of Machine Learning and Data Mining**. Second: Springer Publishing Company, Incorporated, 2017. ISBN 978-1-4899-7685-7.

SCHWEITZER, Marc Alexander. **A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations**. Berlin Heidelberg: Springer-Verlag, 2003. (Lecture Notes in Computational Science and Engineering). ISBN 978-3-540-00351-9. DOI: 10.1007/978-3-642-59325-3. Disponível em: <https://www.springer.com/gp/book/9783540003519>. Acesso em: 13 abr. 2021.

SELVAKUMAR; KARUPPIAH, Muneeswaran. Firefly Algorithm Based Feature Selection for Network Intrusion Detection. **Computers & Security**, v. 81, p. 148–155, mar. 2019. ISSN 01674048. DOI: 10.1016/j.cose.2018.11.005. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0167404818303936>. Acesso em: 6 out. 2020.

SHEPARD, Donald. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In: PROCEEDINGS of the 1968 23rd ACM National Conference. New York, NY, USA: Association for Computing Machinery, 1 jan. 1968. (ACM '68), p. 517–524. ISBN 978-1-4503-7486-6. DOI: 10.1145/800186.810616.

SOLORIO-FERNÁNDEZ, Saúl; CARRASCO-OCHOA, J. Ariel; MARTÍNEZ-TRINIDAD, José Fco. A Review of Unsupervised Feature Selection Methods. en. **Artif Intell Rev**, v. 53, n. 2, p. 907–948, fev. 2020. ISSN 1573-7462. DOI: 10.1007/s10462-019-09682-y.

SORIANO, Humberto Lima. **Elementos Finitos**. 1ª edição. Rio de Janeiro: Ciencia Moderna, 2009. ISBN 978-85-7393-880-7.

TABAKHI, Sina; NAJAFI, Ali; RANJBAR, Reza; MORADI, Parham. Gene Selection for Microarray Data Classification Using a Novel Ant Colony Optimization. en. **Neurocomputing**, v. 168, p. 1024–1036, nov. 2015. ISSN 0925-2312. DOI: 10.1016/j.neucom.2015.05.022.

THACKER, William I; ZHANG, Jingwei; WATSON, Layne T; BIRCH, Jeffrey B; IYER, Manjula A. Algorithm XXX: SHEPPACK: Modified Shepard Algorithm for Interpolation of Scattered Multivariate Data. Department of Computer Science, Virginia Polytechnic Institute & State University, p. 21, 2010.

THEJAS, G. S.; JOSHI, Sajal Raj; IYENGAR, S. S.; SUNITHA, N. R.; BADRINATH, Prajwal. Mini-Batch Normalized Mutual Information: A Hybrid Feature Selection Method. **IEEE Access**, v. 7, p. 116875–116885, 2019. ISSN 2169-3536. DOI: 10.1109/ACCESS.2019.2936346.

ULLAH, Imtiaz; MAHMOUD, Qusay H. A Filter-Based Feature Selection Model for Anomaly-Based Intrusion Detection Systems. In: 2017 IEEE International Conference ON Big Data (Big Data). **2017 IEEE International Conference on Big Data (Big Data)**. Dez. 2017. P. 2151–2159. DOI: 10.1109/BigData.2017.8258163.

UYSAL, Elif İpek; DEMIRCIOĞLU, Gülnur; KALE, Gülsade; BOSTANCI, Erkan; GÜZEL, Mehmet Serdar; MOHAMMED, Sarmad N. Network Anomaly Detection System Using Genetic Algorithm, Feature Selection and Classification. In: 2019 3RD International Symposium ON Multidisciplinary Studies AND Innovative Technologies (ISMSIT). **2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)**. Out. 2019. P. 1–5. DOI: 10.1109/ISMSIT.2019.8932750.

VOLPATO, Gilson; BARRETO, Rodrigo. **Estatística sem dor!!!** 2. ed. Brasil: Best Writing, 2016. 160 p. ISBN 978-85-64201-10-1.

XUE, Hongfa; SUN, Shaowen; VENKATARAMANI, Guru; LAN, Tian. Machine Learning-Based Analysis of Program Binaries: A Comprehensive Study. **IEEE Access**, v. 7, p. 65889–65912, 2019. ISSN 2169-3536. DOI: 10.1109/ACCESS.2019.2917668.

YAMAKAWA, Shunsuke; HYODO, Shi-aki. Gaussian Finite-Element Mixed-Basis Method for Electronic Structure Calculations. **Phys. Rev. B**, American Physical Society, v. 71, n. 3, p. 035113, 26 jan. 2005. DOI: 10.1103/PhysRevB.71.035113. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevB.71.035113>. Acesso em: 24 out. 2020.

YANG, Xin-She. Flower pollination algorithm for global optimization. In: SPRINGER. INTERNATIONAL conference on unconventional computing and natural computation. 2012. P. 240–249.

ZHANG, Jingqiao; SANDERSON, Arthur C. JADE: adaptive differential evolution with optional external archive. **IEEE Transactions on evolutionary computation**, IEEE, v. 13, n. 5, p. 945–958, 2009.

ZHANG, Zhixia; WEN, Jie; ZHANG, Jiangjiang; CAI, Xingjuan; XIE, Liping. A Many Objective-Based Feature Selection Model for Anomaly Detection in Cloud Environment. **IEEE Access**, v. 8, p. 60218–60231, 2020. ISSN 2169-3536. DOI: 10.1109/ACCESS.2020.2981373.

ZHENG, Alice; CASARI, Amanda. **Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists**. Edição: 1: O'Reilly Media, 2018.

ZHOU, Yuyang; CHENG, Guang; JIANG, Shanqing; DAI, Mian. Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier. **Computer Networks**, p. 107247, abr. 2020. Comment: To be published in Computer Networks at https://doi.org/10.1016/j.comnet.2020.107247. ISSN 13891286. DOI: 10.1016/j.comnet.2020.107247. arXiv: 1904.01352. Disponível em: <http://arxiv.org/abs/1904.01352>. Acesso em: 13 abr. 2020.

ZHU, Yingying; LIANG, Junwei; CHEN, Jianyong; MING, Zhong. An Improved NSGA-III Algorithm for Feature Selection Used in Intrusion Detection. **Knowledge-Based Systems**, v. 116, p. 74–85, 15 jan. 2017. ISSN 0950-7051. DOI: 10.1016/j.knosys.2016.10.030. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705116304245>.

ZIENKIEWICZ, Olek C.; TAYLOR, Robert L.; ZHU, J. Z. **The Finite Element Method: Its Basis and Fundamentals**. 7ª edition: Butterworth-Heinemann, 31 ago. 2013. 1369 p.