UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

YAN ANTONINO COSTA DOS SANTOS

VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS ATRAVÉS DE ANÁLISE DE REDES COMPLEXAS

Recife
2021

YAN ANTONINO COSTA DOS SANTOS

VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS ATRAVÉS DE ANÁLISE DE REDES COMPLEXAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

Área de concentração: Pesquisa Operacional.

Orientador: Leandro Chaves Rêgo, PhD.

Coorientador: Raydonal Ospina Martínez.

Recife

2021

Catalogação na fonte
Bibliotecário Gabriel Luz, CRB-4 / 2222

S237v Santos, Yan Antonino Costa dos
Verificação de assinaturas manuscritas através de análise de redes complexas / Yan Antonino Costa dos Santos – Recife, 2021.
78 f.: figs., tabs.

Orientador: Prof. Dr. Leandro Chaves Rêgo.
Coorientador: Prof. Dr. Raydonal Ospina Martínez.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia de Produção, 2021.
Inclui referências.

1. Engenharia de Produção. 2. Verificação de assinaturas. 3. Análise de redes complexas. 4. Aprendizado de máquina. I. Rêgo, Leandro Chaves (Orientador). II. Martinez, Raydonal Ospina (Coorientador). III. Título.

UFPE

658.5 CDD (22. ed.) BCTG / 2021 - 135

YAN ANTONINO COSTA DOS SANTOS

VERIFICAÇÃO DE ASSINATURAS MANUSCRITAS ATRAVÉS DE ANÁLISE DE REDES COMPLEXAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia de Produção.

Aprovada em: 23 / 02 / 2021

BANCA EXAMINADORA

_____
Prof. Dr. Leandro Chaves Rêgo, PhD (Orientador)
Universidade Federal de Pernambuco

_____
Prof. Dr. Raydonal Ospina Martínez (Coorientador)
Universidade Federal de Pernambuco

_____
Prof. Dr. Márcio José das Chagas Moura (Examinador Interno)
Universidade Federal de Pernambuco

_____
Prof. Dr. Heitor Soares Ramos Filho (Examinador Externo)
Universidade Federal de Minas Gerais

AGRADECIMENTOS

Agradeço aos meus familiares, em especial meu pai, mãe e avós que tanto me ajudaram durante minha vida educacional e me proporcionaram chegar até aqui, principalmente neste ano em que o mundo mudou e as incertezas tomaram conta de todos.

Aos professores Leandro Chaves e Raydonal Ospina, pela dedicação, orientação e compreensão no desenvolvimento deste trabalho.

Ao PPGEP-UFPE, CNPq e Capes, pela oportunidade de cursar o Mestrado e pelo suporte financeiro à pesquisa. Muito obrigado!

RESUMO

Nesta dissertação, um modelo para verificação de assinaturas manuscritas online através de análise de redes complexas é proposto, em que métricas de centralidade são utilizadas como características preditoras. Seis métodos de aprendizado supervisionado são usados como classificador da veracidade da assinatura: Naive Bayes, Árvore de decisão, floresta aleatória, *Extreme Gradient Boosting*, Máquina de vetores de suporte e Regressão logística. No estudo é utilizado o banco de dados de assinaturas do MCYT-100, que é utilizado em problemas de verificação. A base de dados possui informações da dinâmica de assinatura de 100 indivíduos, sendo que cada assinatura é replicada (25 assinaturas genuínas e 25 falsas) a fim de tornar viável a avaliação da variabilidade do indivíduo. No framework, as assinaturas foram consideradas como os nós de uma rede complexa em que a relação de conexão existente entre duas assinaturas é mensurada através das correlações entre as séries temporais das coordenadas da assinatura que excedem um limiar de comparação. Este estudo consiste na implementação e avaliação de 5 etapas: 1) coleta de dados; 2) pré-processamento dos dados: transformar as coordenadas em séries temporais e usá-las para a formação das redes complexas; 3) extração dos dados: cálculo das métricas das redes complexas; 4) aprendizado de máquina: uso dos métodos de machine learning para a classificação das assinaturas; 5) avaliação do modelo de aprendizado. Os atributos para o modelo foram as medidas topológicas de centralidade de grau e de autovetor, coeficiente de clusterização e porcentagem de vizinhos verdadeiros. A entropia de permutação como quantificador da informação das séries temporais também é usada como um preditor. Para o ajuste do modelo foram selecionadas aleatoriamente 20 assinaturas verdadeiras e 20 falsas por indivíduo. Métricas de erro preditivo foram calculadas usando as 10 assinaturas restantes por indivíduo. Esse processo foi repetido em um esquema Monte Carlo com 100 repetições para obter estimativas médias de ajuste. Algumas métricas para medir os erros de classificação são utilizadas, entre elas o False Rejection Rate (FRR) que identifica a porcentagem de assinaturas verdadeiras que são rejeitadas pelo modelo, o False Acceptance Rate (FAR) como sendo a porcentagem de assinaturas falsas que são aceitas pelo mecanismo de aprendizado e a Average Error Rate (AER) como o erro médio. Neste framework obteve-se uma taxa de falsos positivos de 6,19%, uma taxa de falso negativo de 6,39% e um erro médio de aproximadamente 6,28%. Em geral as métricas de rede de assinaturas verdadeiras diferem bastante de assinaturas falsas.

Palavras-chave: Verificação de assinaturas. Análise de redes complexas. Aprendizado de máquina.

ABSTRACT

In this dissertation, a model for online handwritten signatures verification via complex network analysis is proposed, in which centrality metrics are used as predictor features. Six supervised learning methods are used as a classifier of signature veracity: Naive Bayes, Decision Tree, Random forest, Extreme Gradient Boosting, Support Vector Machine, and Logistic Regression. In the study is used the signatures database of the MCYT-100, which is used in verification problems. The database has information on the signature dynamics of 100 individuals, and each signature is replicated (25 genuine signatures and 25 false signatures) in order to make the assessment of the individual's variability feasible. In the framework, signatures were considered as the nodes of a complex network in which the existing connection relationship between two signatures is measured through the correlations between time series of signature coordinates that exceed a comparison threshold. This study consists of the implementation and evaluation of 5 stages: 1) data collection; 2) data pre-processing: transform the coordinates into time series and use them for the formation of complex networks; 3) data extraction: calculation of the metrics of complex networks; 4) machine learning: use of machine learning methods for the classification of signatures; 5) evaluation of the models. The atributes for the model were the topological measures of grade and autovector centrality, cluster coeficiente and percentage of genuine neighbors. Permutation entropy as a quantifier of time series information is also used as a predictor. For the adjustment of the model, 20 genuine signatures and 20 false signatures per individual were randomly selected. Predictive error metrics were calculated using the remaining 10 signatures per individual. This process was repeated in a Monte Carlo scheme with 100 repetitions to obtain average adjustment estimates. Some metrics for measuring classification errors are used, including the False Rejection Rate (FRR), which identifies the percentage of genuine signatures that are rejected by the model, the False Acceptance Rate (FAR) as the percentage of false signatures that are accepted by the learning engine, and the Average Error Rate (AER). In this framework, a false acceptance rate of 6,19% was obtained, a false rejection rate of 6,39% and an average error of approximately 6,28%. In general, network metrics for genuine signatures differ greatly from false signature. Compared to other models in the literature, network metrics are shown to be very promising attributes for signature verification.

Keywords: Signature verification. Complex network analysis. Machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A palavra biometria, segundo Ortega-Garcia (2004), é associada a traços ou comportamentos humanos que podem ser medidos e usados para identificação individual. O reconhecimento biométrico pode ser utilizado em situações onde os usuários precisam ser seguramente identificados. Existem dois tipos de biometria de acordo com os traços pessoais: a) físico/fisiológico que leva em consideração as características biológicas, como íris, impressão digital, face. b) comportamental que considera traços dinâmicos, como assinatura e voz. Com sistemas biométricos os usuários não precisam lembrar senhas ou carregar chaves, e é mais difícil de imitar ou gerar dados biométricos.

A assinatura manuscrita é um dos dados biométricos mais utilizados para verificação e identificação, pois é simples, razoavelmente seguro, barato, não intrusivo e aceitável na sociedade, porém tem uma baixa taxa de identificação se comparado com outras medidas biométricas. Ela pode ser definida como uma marca legal de um indivíduo, executada a mão com o propósito de autenticação da escrita de forma permanente. O processo de assinatura pode ser usado com dois propósitos diferentes: a) identificação (reconhecimento) e b) verificação (autenticação). Na identificação de assinaturas, a entrada é uma assinatura desconhecida e o sistema deve ser capaz de identificar seu dono. Já o objetivo da verificação de assinatura é saber se uma dada assinatura de um indivíduo é verdadeira ou falsa (SIGARI, 2011).

1.1 DESCRIÇÃO DO PROBLEMA

A verificação de assinaturas pode ser abordada de basicamente dois modos: a) a abordagem *offline* é baseada apenas na imagem bidimensional da assinatura. Essas imagens são referidas como assinaturas estáticas. Sistemas *offline* são de interesses em cenários onde estão disponíveis apenas cópias impressas, como em locais em que um grande número de documentos precisa ser autenticado. b) a abordagem *online* requer a coleta de informação temporal, e pode apresentar características como pressão e velocidade. Aqui os dados armazenados são referidos como uma assinatura dinâmica. Sistemas *online* são de maior interesse em aplicações de segurança (COETZER, 2004). Há também um meio termo entre essas duas abordagens que explora a informação temporal presente nas coordenadas das assinaturas, e pode ser chamada de quasi-offline. Com essa informação pode-se originar uma série temporal para cada assinatura com relação as suas coordenadas, tanto para o eixo x quanto para o y.

A maior parte das pesquisas na área de verificação de assinaturas considera a informação de entrada descrita como um processo aleatório (EL-HENAWY, 2013). Assim a informação dinâmica de entrada obtida através da amostra deve ser considerada como uma sequência temporal aleatória discreta.

Estudos recentes têm sido propostos para a análise de séries temporais como, por exemplo, transformar um conjunto de séries temporais em redes complexas (GAO, 2012). Com essa transformação é possível obter dados, a partir da topologia da rede, que não são perceptíveis nas séries temporais, como a formação de componentes conexas, em que são agrupados elementos com características similares, e algumas estatísticas como as distâncias entre diferentes assinaturas que são representadas como vértices de um grafo, e seus respectivos graus de distribuição.

1.2 JUSTIFICATIVA E RELEVÂNCIA

Pessoas assinam todos os dias para atestar sua identidade, entretanto, essa modalidade de identificação é a que mais sofre ataques de falsificação. A verificação de assinatura é um problema no qual a entrada é classificada como verdadeira ou falsa. Embora as assinaturas sejam destinadas a servir como verificação de identidade, uma mesma pessoa possui variações em sua assinatura devido a alguns fatores. Assinaturas possuem três atributos principais: forma, movimento e variação (HILTON, 1992).

A autenticação de assinatura é um processo muito importante, e encontra sua aplicação em vários setores como o bancário, imobiliário, corporações, rede pública, entre outros. Devido a sua importância, muitos tentam falsificar a assinatura de terceiros. Para prevenir este problema é importante desenvolver um sistema para o reconhecimento automático das assinaturas.

Quando é exigida a assinatura de um indivíduo para prosseguimento em certos processos, sistemas de verificação de assinaturas eficientes agilizam a liberação de etapas subsequentes, o que permite maior produtividade do setor que está sendo empregado. Questões logísticas como confirmação de recebimento pelo cliente podem ser monitorados remotamente, podendo evitar extravios e prejuízos. Além da sustentabilidade e economia nos custos de material impresso.

O uso de redes complexas traz novos olhares para o problema de verificação de assinaturas, através do uso de informações mais globais sobre o comportamento das assinaturas do banco de dados, como elas se relacionam umas com as outras e seu papel na vizinhança. Trata-se de uma visão mais holística das assinaturas no banco de dados.

Os resultados preliminares desta pesquisa foram publicados em Santos et al. (2020a) e Santos et al. (2020b).

1.3 OBJETIVO GERAL

Com base no contexto apresentado acima o objetivo do trabalho é explorar o uso de redes complexas para fornecer atributos para o problema de verificação de assinaturas, estabelecendo assim a proposta de uma nova metodologia para abordar o problema, usando para isso o banco de dados de assinaturas do MCYT (Ministerio de Ciencia Y Tecnologia), que é disponível gratuitamente e amplamente utilizado como benchmark para sistemas de verificação. (ORTEGA-GARCIA, 2003).

1.4 OBJETIVOS ESPECÍFICOS

Como objetivos específicos desta dissertação tem-se o seguinte:

- Caracterizar os tipos de assinaturas de acordo com as métricas de análise de redes;
- Desenvolver um framework para o modelo proposto;
- Aplicação do framework em um banco de dados para a validação do modelo proposto.

2 REFERENCIAL TEÓRICO E REVISÃO DA LITERATURA

Esta seção apresenta uma pequena revisão de literatura e em seguida conceitos importantes para o desenvolvimento do estudo proposto, como séries temporais, coeficiente de correlação, teoria dos grafos, redes complexas e aprendizado de máquina.

2.1 REVISÃO DA LITERATURA

Martinez-Diaz et al. (2007) desenvolveram uma abordagem com uma pontuação normalizada para a verificação de assinatura usando uma adaptação do Universal Background model bayesian (UBM) para gerar o modelo do cliente. O UBM é treinado uma vez e pode ser usado por todos os usuários que reivindicarem. O UBM é escolhido para ser o Gaussian Mixture Model (GMM) treinado com o vetor de características de uma lista de usuários. No processo de verificação, a assinatura do cliente é comparada com o template reivindicado e a pontuação resultante é normalizada por sua similaridade com o UBM. A similaridade entre as assinaturas é computada usando técnicas de classificação estatística.

Jonas Richiardi et al. (2009) propuseram um meio termo entre classificadores discriminativos estáticos e modelos geradores dinâmicos, usando um modelo gerador estático junto com passos específicos de extração de características. Eles usaram uma topologia específica de rede bayesiana estática para realizar classificação de séries temporais multivariadas. Os resultados mostraram que usando características apropriadas extraídas das séries temporais, a estrutura da rede bayesiana estática oferece uma abordagem eficiente para classificação das séries temporais multivariadas que representam as assinaturas.

Teymourzadeh et al. (2013) elaboraram um sistema de verificação de assinaturas online baseado em *Discrete Cosine Transformation* (DCT) e *Discrete Wavelet Transformation* (DWT), que cria banco de dados de atributos com espaço d-dimensional para cada assinatura, e depois uma redução de dimensão é aplicada com *Principal Component Analysis* (PCA). O modelo é treinado usando o método de Máquina de Vetores de Suporte. Mostra-se como uma boa opção quando se tem muitos atributos nos dados de entrada e apresenta resultados satisfatórios.

Geralmente são necessárias algumas assinaturas genuínas de referência para se treinar um sistema. Com a proposta de utilizar apenas uma assinatura de referência, Diaz et al. (2018) propuseram um modelo que consiste em duplicar a amostra de referência um certo número de

vezes e treinar o sistema com cada uma das assinaturas resultantes. O esquema de duplicação é baseado em decomposição sigma lognormal da assinatura original. Dois métodos são aplicados, o primeiro varia os parâmetros strokes lognormal (*strokes-wise*), enquanto o segundo modifica seus pontos alvos virtuais (*target-wise*). Seus resultados sugerem que o sistema proposto, com uma única assinatura de referência, alcança performances similares a outros sistemas que usam cinco ou mais assinaturas. Quando se tem poucas assinaturas para treino se apresenta como uma boa escolha, porém apresenta resultados aquém de outros modelos com maior quantidade de dados.

Baseado na ideia de seleção das melhores características das assinaturas, Sharif et al. (2018) desenvolveram um sistema de verificação de assinaturas que consiste em 4 etapas: pré-processamento, extração das características, seleção das características e verificação. As características globais compreendem a área da assinatura, largura, altura, proporção, altura e largura normalizadas. As características locais consistem no centróide da assinatura, inclinação, ângulo e distância. Na seleção das características, um algoritmo genético é utilizado para encontrar o conjunto de características apropriado que depois são usadas em uma máquina de vetores de suporte para a verificação.

Doroz et al. (2018) propuseram um método para verificação de assinaturas online em que o estágio principal da abordagem é a determinação da estabilidade da assinatura de referência. A medida de estabilidade proposta é baseada em conjuntos *fuzzy*. As partes de uma assinatura de referência que diferem das partes correspondentes das assinaturas de referência remanescentes do mesmo indivíduo são tratadas como instáveis e não serão levadas em consideração quando comparadas as amostras de referência e a assinatura sendo verificada. O estudo aplica diferentes métodos de aprendizado de máquinas, como Árvore de decisão, Naive Bayes e Máquina de Vetores de Suportes, atingindo uma taxa de erro em torno de 3,2%. Este método apresenta um dos melhores resultados da literatura.

He et al. (2019) apresentam um método de verificação de assinatura que usa atributos de curvatura e torção. As assinaturas são consideradas como a curva espacial. Como o ambiente de escrita muda, há variação no tamanho da assinatura, localização, rotação, o que pode causar distorções. Então propuseram uma nova técnica para criar um conjunto de atributos através de informação tridimensional, e este processo fornece um vetor de atributos com 8 dimensões para cada assinatura. Baseado na medida de similaridade chamado de distância de Hausdorff foi

verificada a performance entre as assinaturas de teste e as referências. O método proposto tem boa robustez e não é afetado por translação, rotação ou transformação de escala da assinatura.

Okawa (2020) propôs um modelo usando *dinamic time warping* (DTW) múltiplo e ponderado. Para obter um modelo eficaz de cada atributo refletindo a variabilidade entre as amostras de referência, é adotado um método de *time series averaging,* o *euclidian barycenter-based DTW barycenter averaging* (EB-DBA). Com o conjunto de referência são calculadas múltiplas distâncias DTW a partir de séries temporais multivariadas. E para impulsionar o poder discriminativo, é aplicado um esquema de ponderação utilizando um classificador de *gradient boosting*, para combinar as múltiplas distâncias.

O trabalho aqui proposto para verificação de assinaturas procura analisar as assinaturas do banco de dados de uma maneira não apenas local, como a maioria dos modelos propostos, que usam, em geral, informações locais da própria assinatura como, centroide, velocidade média e pressão média, mas também global, verificando o comportamento e o papel de cada um dos elementos da rede na sua vizinhança. Estabelecendo assim novas informações para análise. Vale ressaltar que o estudo tem uma proposta exploratória, e procura analisar se o modelo proposto tem resultados satisfatórios quando comparado com outros modelos da literatura.

2.2 SÉRIES TEMPORAIS

Uma série temporal é um conjunto de observações ordenadas no tempo. Os dados de séries temporais geralmente não são independentes, especialmente se os intervalos da amostra são curtos, as observações próximas costumam ser mais parecidas que as mais distantes, i. e. a autocorrelação diminui com a distância entre as observações. Enquanto em modelos de regressão por exemplo a ordem das observações é irrelevante para a análise, em séries temporais a ordem dos dados é crucial. Também é mais difícil de lidar com observações faltantes e dados discrepantes devido à natureza sequencial (BOX & JENKINS, 1994).

Uma série temporal é dita ser contínua quando as observações são realizadas continuamente no tempo. Definindo o conjunto $T = \{t: t_1 < t < t_2\}$ a série temporal será denotada por $\{X(t): t \in T\}$, em que $X(t)$ representa o valor de uma observação no instante t. Uma série temporal é chamada discreta quando as observações são feitas em instantes discretos no tempo. Definindo o conjunto $T = \{t_1, t_2, \dots, t_n\}$ a série temporal será denotada por $\{X_t: t \in T\}$. Uma série temporal também pode ser multivariada. Se k variáveis são observadas a cada

instante denota-se por $\{X_{1t}, X_{2t}, \dots, X_{kt}, t \in T\}$. Neste caso várias séries correlacionadas devem ser analisadas conjuntamente, ou seja, em cada instante tem-se um vetor de observações (HAMILTON, 1994).

Os principais objetivos em se estudar séries temporais podem ser os seguintes:

- Descrição: Descrever propriedades da série, como o padrão de tendência, existência de variação sazonal ou cíclica, alterações estruturais, outliers, etc.
- Controle: Os valores da série temporal medem a "qualidade" de um processo de manufatura e o objetivo é o controle do processo (controle estatístico da qualidade).
- Explicação: Usar a variação em uma série para explicar a variação em outra série.
- Predição: Predizer valores futuros com base em valores passados. Aqui assume-se que o futuro envolve incerteza, ou seja, as previsões não são perfeitas.

Em algumas situações o objetivo pode ser fazer previsões de valores futuros enquanto em outras a estrutura da série ou sua relação com outras séries pode ser o interesse principal. No caso do estudo aqui presente se está interessado na relação entre séries temporais, que podem representar coordenadas de assinaturas de diferentes indivíduos ou de várias assinaturas, verdadeiras ou falsas, de um mesmo indivíduo.

2.3 COEFICIENTE DE CORRELAÇÃO

Os coeficientes de correlação são métodos estatísticos para se medir as relações de associação entre variáveis e o que elas representam. A correlação busca entender como uma variável se comporta em um cenário em que outra está variando, visando identificar se existe alguma correlação entre a variabilidade de ambas. Apesar de não implicar em causalidade, o coeficiente de correlação expressa em números essa relação, ou seja, quantifica a relação entre variáveis. Não há apenas uma maneira de se calcular o coeficiente de correlação. Dependendo de como se comportam as variáveis, um coeficiente de correlação pode ser mais adequado que outro.

2.3.1 Coeficiente de correlação de Pearson

O coeficiente de correlação de Pearson ou coeficiente de correlação linear ou r de Pearson, mede a direção e o grau de correlação entre duas variáveis quantitativas (MOORE, 2007). É

um índice com valores situados entre -1 e 1, que reflete a intensidade de uma relação linear entre dois conjuntos de dados.

O coeficiente de correlação de Pearson calcula-se da seguinte forma:

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}.\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{cov(X,Y)}{\sqrt{var(X).var(Y)}} \quad (1)$$

Em que x_i e y_i são os valores medidos de ambas as variáveis, e $\bar{x}$ e $\bar{y}$ são as médias aritméticas das variáveis x e y respectivamente.

Quando o coeficiente de correlação se aproxima de 1 significa que há uma relação linear positiva entre as duas variáveis, ou seja, quando uma aumenta a outra também aumenta. Quando o coeficiente se aproxima de -1, também há uma relação linear entre as variáveis, porém nesse caso ao se aumentar o valor de uma variável o da outra diminui, isso é chamado de correlação negativa. Um coeficiente de correlação próximo de zero indica que não relação entre as duas variáveis, e quanto mais se aproximam de 1 ou -1, mais forte é a relação. Para Cohen (1988), valores entre 0,10 e 0,29 podem ser considerados pequenos; escores entre 0,30 e 0,49 podem ser considerados como médios; e valores entre 0,50 e 1 podem ser interpretados como grandes.

2.3.2 Coeficiente de correlação de Kendall

Segundo Bonett e Wright (2000), o coeficiente de Kendall é uma estatística da semelhança entre as ordens dos dados quando classificados por cada uma das quantidades, essa correlação será elevada se as observações tiveram uma classificação semelhante, sendo classificação a descrição das posições relativas das observações no interior de cada variável. A correlação será baixa quando as observações tiverem uma classificação diferente. O coeficiente de Kendall varia entre -1 e 1, em que 1 é a correlação idêntica e -1 é a correlação inversa.

O coeficiente de correlação de Kendall calcula-se da seguinte forma:

$$\tau = \frac{(número\ de\ pares\ concordantes) - (número\ de\ pares\ discordantes)}{n(n-1)/2} \quad (2)$$

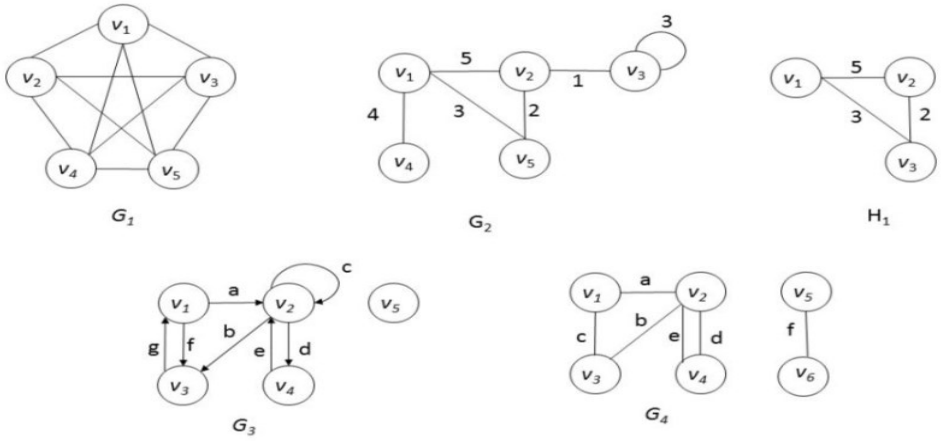
Onde, dado um conjunto de observações $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ de variáveis aleatórias conjuntas X e Y, um par é concordante se $x_i > x_j$ e $y_i > y_j$ ou se $x_i < x_j$ e $y_i < y_j$. E será discordante se $x_i > x_j$ e $y_i < y_j$ ou se $x_i < x_j$ e $y_i > y_j$. O denominador é o número total de combinação de pares.

2.4 GRAFOS

A teoria dos grafos é um ramo da matemática que se utiliza de uma representação baseada em grafos para estudar a relação entre objetos de um conjunto. Um grafo é um conjunto finito e não vazio $V = \{v_1, v_2, \dots, v_n\}$ de objetos chamados vértices, juntamente com um conjunto E de pares $\{v_i, v_j\}$, ordenados ou não, de vértices, os elementos de E são chamados de arestas. Pode-se representar o grafo por $G(V, E)$, que contém o conjunto V de vértices relacionados ao conjunto E de arestas. Em outras palavras, um grafo é uma estrutura que representa um conjunto de elementos denominados vértices e suas relações de interdependência ou arestas.

A figura 1 será usada para demonstrar alguns conceitos sobre grafos, seguindo Furtado (1973) e Feofiloff et al.(2011).

Figura 1 - Exemplos de grafos



Fonte: Adaptado de Amorim (2014)

O número de vértices $V(G)$ de um grafo é chamado de **ordem**, enquanto a quantidade de pares ordenados $E(G)$ é dito ser o **tamanho** do grafo. Na figura 1, o grafo G_1 tem uma ordem igual a 5, enquanto o grafo G_4 tem ordem 6. O tamanho do grafo G_1 é 10, já G_4 tem tamanho 6. Existem três modos diferentes de definir as relações de E, e cada um produz um tipo especial de grafo. As três possibilidades de E são:

- **Aresta**: se a relação em E é simétrica, ou seja, se $\{v_i, v_j\} \in E$ implica que $\{v_j, v_i\} \in E$, diz-se que a relação é uma aresta e o grafo $G(V, E)$ é chamado de grafo **não dirigido**.
- **Arco**: quando a relação é não simétrica diz-se que a relação é um arco, e o grafo é chamado de grafo **dirigido**.

- **Loop**: se a relação é reflexiva, isto é, $\forall\, v \in V$, tem-se $\{v, v\} \in E$, a relação é um loop.

O grafo G_1 é não dirigido, enquanto que G_3 é um grafo dirigido.

Os grafos podem ainda ser classificados em **binários** ou **ponderados**. A diferença entre os dois é que no caso ponderado, se existe uma relação entre dois vértices $\{v_i, v_j\} \in E$ um número w_i é associado como uma medida de força ou proximidade da relação. Enquanto, em um grafo binário, não importa a proximidade da relação, apenas se a relação existe ou não. Na figura 1, o grafo G_1 é binário, já G_2 é um grafo ponderado.

Um grafo pode ser representado de diferentes formas, neste trabalho será utilizada a matriz de adjacências $M(G)$. Dado um grafo G com n nós, a **matriz de adjacência** $M(G)$, é uma matriz $n \times n$. Se a matriz adjacência representar um grafo binário, cada elemento da matriz será definido por:

$$a_{ij}(G) = \begin{cases} 1 \quad se\ \{v_i, v_j\} \in E \\ 0\ caso\ contrario \end{cases} \tag{3}$$

Para um grafo ponderado, cada elemento da matriz será definido por:

$$a_{ij}(G) = \begin{cases} w_{ij} \quad se\ \{v_i, v_j\} \in E \\ 0\ caso\ contrario \end{cases} \tag{4}$$

Caminho é uma sucessão de vértices e arestas, onde cada aresta liga o vértice que a precede ao vértice que a segue, não repetindo vértices e arestas. O comprimento de um caminho que contenha dois determinados vértices será dado pelo número de arestas presentes no caminho.

Outro conceito importante é o de **caminho geodésico**, ou caminho mais curto, que é o caminho mais curto entre dois vértices. O comprimento do caminho geodésico, d_{ij}, também chamado de distância geodésica, ou distância mais curta, é portanto, a mais curta distância no grafo entre dois vértices. Se $(i_0, i_1, i_2, \dots, i_k)$ é um caminho entre os vértices i e j, então:

$$d_{ij} = \min(M_{i_0,i_1} + M_{i_1,i_2} + \cdots + M_{i_{k-1},i_k}) \tag{5}$$

Onde M_{ij} é o elemento ij da matriz adjacente.

Se o grafo for ponderado, a distância geodésica é dada por:

$$d_{ij}^{w} = \min\left(\frac{1}{w_{i_0,i_1}} + \frac{1}{w_{i_1,i_2}} + \cdots + \frac{1}{w_{i_{k-1},i_k}}\right) \tag{6}$$

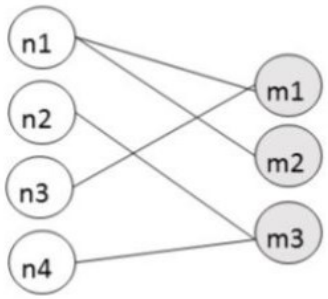
Em que w é o peso da matriz adjacente.

No grafo G_2, $v_1 \rightarrow v_5 \rightarrow v_2 \rightarrow v_3$ é um caminho que leva do vértice v_1 ao v_3, e tem tamanho igual a 3. Porém o tamanho do caminho geodésico entre v_1 e v_3 é 2, pois $v_1 \rightarrow v_2 \rightarrow v_3$ é o menor caminho, entre esses dois vértices.

Um grafo é dito ser **completo** se para todo par de vértices $(v_i, v_j) \in V(G)$ existe uma relação $\{v_i, v_j\} \in E(G)$, ou seja, quando quaisquer dois vértices distintos são adjacentes. O **grau** do vértice é o número de arestas adjacentes a um vértice. Em grafos direcionados, o grau dos vértices é a soma do número de arestas que entram e que saem do vértice. Um grafo onde todos os vértices têm o mesmo grau é chamado de **regular**. G_1 é um grafo completo e regular.

Um grafo é **bipartido** se os vértices são particionados em dois subconjuntos V_1 e V_2 de modo que $V_1 \cap V_2 = \emptyset$ e $V_1 \cup V_2 = V$ e que nenhuma aresta seja incidente em dois vértices do mesmo subconjunto, ou seja, dois vértices do mesmo subconjunto não são adjacentes. A figura 2 mostra um grafo bipartido, onde $V_1 = \{n_1, n_2, n_3, n_4\}$ e $V_2 = \{m_1, m_2, m_3\}$.

Figura 2 - Grafo bipartido



Fonte: Adaptado de Amorim (2014)

Grafo **conexo** é aquele que entre qualquer par de vértices existe sempre um caminho que os une, caso contrário, diz-se que o grafo é desconexo. Os grafo G_1 e G_2 são conexos, enquanto G_3 e G_4 são desconexos.

Um grafo H é um **subgrafo** do grafo G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Se a ordem de H é igual a ordem de G, é dito que H é um subgrafo gerador do grafo G. Um subgrafo H de G é **próprio** se $V(H) \neq V(G)$ ou $E(H) \neq E(G)$. Um subgrafo completo é chamado de **clique**. H_1 é um subgrafo completo de G_2.

Um subgrafo conexo H de um grafo G é **maximal** se H não é subgrafo próprio de algum subgrafo conexo de G. Dado um grafo desconexo, a menor sequência de arestas conectando dois vértices, uma componente conexa é um subgrafo conexo e maximal com respeito a inclusão, ou seja, não existe caminho entre um vértice pertencente ao subgrafo e outro vértice não pertencente ao subgrafo. H_1 é uma componente conexa de G_2.

Um grafo $G(V, E)$ é dito ser um **multigrafo** quando existem múltiplas arestas entre pares de vértices de G. Formalmente, um multigrafo é um par ordenado $G(V, E)$, sendo V um conjunto de vértices, e E um multiconjunto de pares não-ordenados de vértices, ou seja um conjunto de conjuntos. Na figura 1, o grafo G_4 é multigrafo, pois há duas arestas entre os vértices v_2 e v_4. Cada uma dessas arestas pode representar relações diferentes, por exemplo em uma malha viária, uma aresta poderia ser uma rodovia enquanto a outra seria uma ferrovia. No caso deste estudo pode-se considerar uma aresta como sendo a correlação entre as coordenadas x e a outra a correlação entre as coordenadas y de cada assinatura.

2.5 REDES COMPLEXAS

Pode-se agora definir propriamente uma rede. Uma rede N é um grafo que precisa satisfazer as seguintes condições:

- O conjunto de vértices é composto por membros de um sistema, e esses membros representam uma entidade definida;
- A relação entre os vértices é associada com os atributos ou propriedades comuns que os elementos do sistema possuem;
- O conjunto de arestas é criado a partir de uma quantidade ou condição que diz se a relação entre dois elementos existe ou não e se a relação é forte ou fraca.

Para exemplificar, suponha uma festa, os elementos do sistema são as pessoas que estão na festa. O atributo entre os elementos do sistema é a amizade. Logo, se existe uma relação entre as pessoas é porque elas se conhecem. Portanto teremos uma rede cujo conjunto de vértice são as pessoas e o conjunto de arestas é composto pela amizade entre elas.

Redes estão por todos os lados e em todos os domínios do conhecimento. O estudo empírico destas redes reais vem se tornando cada vez mais abrangente não apenas em relação ao que elas representam, mas também com relação a seus tamanhos. Figueiredo (2012) apresenta algumas importantes redes de diferentes domínios entre elas:

- **Redes sociais**: São redes formadas por pessoas ou grupos de pessoas e por algum tipo de relacionamento. Uma rede social bastante estudada são as redes de colaboração, cujo relacionamento é algum tipo de colaboração entre as pessoas. Na rede de colaboração científica, os vértices são pesquisadores e as arestas indicam algum tipo de colaboração científica, como por exemplo, publicação de artigos em conjunto.

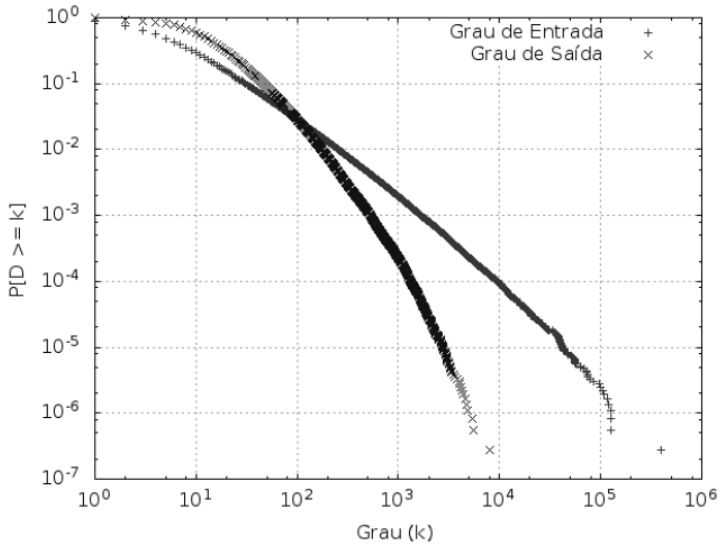
- **Redes de informação**: Em uma rede de informação os vértices representam alguma informação e as arestas representam algum tipo de relacionamento. Uma das maiores e mais populares redes de informação é a Web. Na rede da Web os vértices são páginas da Web identificadas por uma URL e as arestas representam os hiperlinks entre as páginas (entre URLs). Neste caso o relacionamento é assimétrico, pois os hiperlinks da Web são direcionados: um hiperlink da página *i* para a página *j* não implica em um hiperlink no sentido contrário.
- **Redes tecnológicas**: Redes tecnológicas são redes construídas pelo homem em geral para transportar algo. Desta forma, redes tecnológicas são geralmente redes físicas, onde seus objetos e relacionamentos são concretos. Uma importante rede tecnológica é a Internet, formada por roteadores e canais de comunicação, conhecidos como enlaces. Um roteador está ligado a diversos enlaces, que podem ser fios de cobre, fio coaxial, fibra ótica, ou até mesmo canais de rádio e satélite. Na outra ponta de cada enlace há um outro roteador. Um roteador pode ter desde apenas dois canais de comunicação até milhares. Na abstração da Internet cada vértice corresponde a um roteador e uma aresta indica que existe um enlace entre os dois roteadores. Existem muitas redes tecnológicas construídas pelo homem para transportar o próprio homem. Está se falando da rede rodoviária de uma cidade ou um país, da rede ferroviária, e da malha aérea de um país ou do mundo.
- **Redes biológicas**: São geralmente formadas pela natureza e aparecem em muitos contextos. Algumas redes biológicas são construções físicas e outras representações mais abstratas que dependem do relacionamento. Uma das mais intrigantes redes biológicas é a rede neuronal humana, ou seja, a rede formada pelos neurônios de nosso cérebro. Nesta rede, vértices são neurônios e arestas representam ligações físicas entre os neurônios, conhecidas como sinapses. As ligações entre neurônios são direcionadas, com os dendritos recebendo sinais e axônios enviando sinais.

Todas essas redes apresentadas acima possuem características comuns às redes complexas, são elas (FIGUEIREDO, 2012):

- **Distribuição de grau com cauda pesada**: Muitas redes complexas de diferentes domínios possuem uma distribuição de grau com cauda pesada, exibindo um espectro de grau que são ordens de grandeza maiores do que a média. Ou seja, os

graus dos vértices não são nada parecidos uns com os outros, pois podemos ter vértices com graus muito maiores do que a média com probabilidade não desprezível, como mostra a figura 3.

Figura 3 - Distribuição (CCDF) do grau de entrada e de saída de uma rede



Fonte: adaptado de Figueiredo (2012)

- **Distância baixa**: Muitas redes complexas possuem distância relativamente pequena entre seus vértices, inclusive entre os vértices mais distantes da rede, sendo ordens de grandeza menor do que o número de vértices. Em alguns casos a distância média da rede é bem representada por $\log n$, em que n é o número de vértices, sendo exponencialmente menor do que seu tamanho. Essa característica é conhecida como “mundo pequeno”.
- **Esparsas e conectadas**: Muitas redes reais são extremamente esparsas, i. e. exibem uma densidade muito baixa, ordens de grandeza menor do que 1. Apesar disto, estas mesmas redes estão quase completamente conectadas, com quase todos os vértices pertencendo a mesma componente conexa.

Existem algumas métricas importantes no estudo de redes complexas. As métricas podem ser divididas em **globais**, que descrevem as características sobre todo o grafo e **individuais,** que estão relacionadas com a análise das propriedades individuais de elementos da rede.

A densidade quantifica o número de conexões entre os vértices de uma rede. Uma rede "densa" é aquela que o número de arestas na rede se aproxima do número máximo de arestas que a rede poderia ter (ARIF, 2015). Uma rede "completa" é aquela em que todos os vértices são adjacentes um ao outro, ou seja, cada vértice é ligado diretamente a todos os outros vértices,

e sua densidade será de 1. Usando o número de vértices e arestas de uma rede, pode-se definir a densidade da rede, ρ, que representa a fração de arestas que a rede possui. Considere uma rede com n vértices e m arestas. ρ é calculado da seguinte forma:

$$\rho = \frac{m}{n(n-1)/2} \tag{7}$$

Diâmetro refere-se ao tamanho da maior distância geodésica entre qualquer par de vértices. O diâmetro de um grafo pode variar de um mínimo de 1, se o grafo for completo, a um máximo de $n-1$, onde n é a ordem do grafo (JACKSON, 2008). Seja $l(v_i, v_j)$ o comprimento do menor caminho que liga o vértice v_i ao vértice v_j, então o diâmetro do grafo conexo será dado por:

$$L = \max_{\{v_i, v_j \in V\}} l(v_i, v_j) \tag{8}$$

No caso de grafos desconexos, o diâmetro da rede será dado pelo maior valor encontrado entre os diâmetros das componentes conexas.

A **excentricidade**, $e(v_i)$, de um vértice é a distância máxima a partir dele para qualquer outro vértice da rede. Quanto menor a excentricidade de um vértice, melhor é o relacionamento com os outros vértices (WEST, 2000). A excentricidade de v_i é dada por:

$$e(v_i) = \max_{\{v_j \in V\}} l(v_i, v_j) \tag{9}$$

A excentricidade máxima é chamada de diâmetro, enquanto a excentricidade mínima é chamada de raio. No caso de redes desconexas, a excentricidade de cada vértice é calculada levando em consideração o componente ao qual pertence.

Como o grau de um vértice é calculado em torno do número de vértices adjacentes, o grau pode ser considerado como uma medida de centralidade local (ARIF, 2015). Portanto, a **centralidade de grau** de um vértice i em um grafo não ponderado é:

$$C_d = \sum_{j=1}^{n} M_{ij} \tag{10}$$

Caso a rede seja ponderada, a medida é definida por:

$$C_d^w = \sum_{j=1}^{n} w_{ij} \tag{11}$$

Freeman (1979) propôs uma medida de centralidade que normaliza o número de ligações existentes na rede pelo número máximo de ligações que se poderia ter. Assim a **centralidade relativa de grau**, em uma rede ponderada, de um vértice i é dada por:

$$C'_d(i) = \frac{C_d^w(i)}{n-1} \tag{12}$$

Apesar de ser uma medida simples, o grau mede a influência do vértice. Em uma rede sem loops, a centralidade de grau poderá variar de 0, no caso de nós isolados, até $n-1$, quando o vértice possui ligação com todos os demais vértices da rede, em que n é o número total de elementos da rede.

A **centralidade de proximidade** (Closeness) é o inverso da distância média de um determinado nó inicial para todos os demais nós da rede. Seja d_{ij} o número de ligações em um caminho mais curto do vértice i para o vértice j, define-se a centralidade de proximidade do vértice i como (ZHANG; LUO, 2017):

$$C_c(i) = \frac{1}{\sum_{j=1}^{n} d_{i,j}} \tag{13}$$

A medida pode ser normalizada utilizando a maior distância possível entre quaisquer dois vértices de uma rede de n vértices, ou seja, $n-1$. Sendo assim a centralidade relativa de proximidade do vértice i é dada por:

$$C'_c(i) = \frac{C_c^w(i)}{n-1} \tag{14}$$

Segundo Freeman (1979) a centralidade de proximidade é uma medida inversa, ou seja, os vértices mais centrais da rede de acordo com essa medida são aqueles que possuem uma distância menor dos outros vértices. Um vértice que está, em média, numa posição mais próxima dos outros vértices pode obter informações de maneira mais eficiente, ou seja, a medida de proximidade está relacionada a independência e eficiência na comunicação com outros vértices.

A **centralidade de intermediação** (Betweenness) é o número de caminhos mais curtos que passam através de um determinado vértice. Considerando $j \neq k \neq i$, seja g_{jk} o número de caminhos mais curtos do vértice j para o vértice k e g_{jik} é o número de caminhos mais curtos do vértice j para o vértice k passando por i. A centralidade de intermediação do vértice i é determinada por (BRANDES, 2001):

$$C_b(i) = \sum_{j,k} \frac{g_{jik}}{g_{jk}} \tag{15}$$

Do mesmo jeito que a centralidade de grau, existe a necessidade de normalizar essa medida. O betweenness do vértice i é normalizada pelo número máximo possível de caminhos mais curtos, excluindo o vértice i. Dado uma rede não direcionada, o máximo é: $\frac{[(n-1)(n-2)]}{2} = (n^2 - 3n + 2)/2$. Assim a centralidade de intermediação relativa é dada por:

$$C_b'(i) = \frac{2 \times C_b(i)}{(n^2 - 3n + 2)} \tag{16}$$

A intermediação é um indicador do potencial de um vértice de desempenhar um papel de intermediador, podendo "controlar" com maior frequência o fluxo de informação.

Também há medidas de centralidade espectrais que buscam estabelecer propriedades estruturais dos vértices a partir das propriedades de autovalores e autovetores das matrizes associadas a estes grafos. A **centralidade de autovetor** atribui alta importância para um vértice, em função da sua relação com os seus vizinhos, sendo assim, mesmo que um vértice v_k esteja ligado a somente outro v_i (portanto, com uma baixa centralidade de grau) os vizinhos de v_i podem ser importantes, consequentemente o vértice v_k também será importante, obtendo uma elevada centralidade de autovetor (FREITAS, 2010). Seja λ, uma constante, n, o número de vértices do grafo e $M_{i,j}$ a matriz adjacência associada ao $i - ésimo$ vértice da rede, a centralidade de autovetor do vértice i, $C_e(i)$, será dada por:

$$C_e(i) = \frac{1}{\lambda} \sum_{j=1}^{n} M_{i,j}\, C_e(i) \tag{17}$$

Usando a notação vetorial, seja $C = (C_e(1), C_e(2), \dots, C_e(N))$ o vetor de centralidade, pode-se reescrever a equação acima como, $\lambda C = MC$, em que o vetor de centralidade é o autovetor da matriz M associado ao maior autovalor desta matriz.

O **coeficiente de clusterização** junto com o valor médio de caminho mais curto, pode identificar o efeito de "mundo pequeno". Iremos definir o coeficiente de clusterização de um vértice i como sendo a fração de arestas que os vizinhos de i possuem entre si e o máximo de arestas que eles poderiam possuir entre si. Dado que o grau do vértice i é d_i, o maior número de arestas entre seus vizinhos é dado por $\binom{d_i}{2}$, ou seja, todos os pares de vizinhos de i possuem arestas entre si. Sendo E_i o número efetivo de arestas entre os vizinhos do vértice i. Pode-se definir o coeficiente de clusterização do vértice i como (FIGUEIREDO, 2012):

$$c_i = \frac{E_i}{\binom{d_i}{2}} \tag{18}$$

Utilizando o coeficiente de clusterização dos vértices, pode-se agora definir o coeficiente de clusterização da rede como sendo a média aritmética destes.

2.6 ENTROPIA

A entropia é um quantificador de informação quando se olha para a distribuição de padrões da série temporal, que é basicamente associada a desordem e a falta de informação, dada uma sequência de observações cuja evolução pode ser observada ao longo do tempo (BRISSAUD, 2005).

Dada uma função de distribuição de probabilidade contínua (FDP), $f(x)$ com $x \in \Omega \subset \mathbb{R}, \int_{\Omega} f(x)dx = 1$, a entropia de Shannon (S) depende da $f(x)$ e é denotada por $S[f]$ (SHANNON, 1948):

$$S[f] = -\int_{\Omega} f(x)\ln[f(x)]dx \tag{19}$$

$S[f]$ é considerado medida global, pois a distribuição não é muito sensível as mudanças fortes que ocorrem em uma região pequena em Ω, o espaço de f.

Sendo $P = \{p_i; i = 1, \dots, N\}$ com $\sum_{i=1}^{N} p_i = 1$ uma distribuição de probabilidade discreta, com N sendo o número de cenários possíveis no estudo, a média de informação logarítmica de Shannon é dada por:

$$S[P] = -\sum_{i=1}^{N} p_i \ln[p_i] \tag{20}$$

Essa quantidade mede o grau de incerteza relativa à distribuição P e, desse modo, $p_1, \dots, p_n$ é chamado entropia de distribuição P (SHANNON, 1948). Nessa perspectiva, se $S[P] = S_{min} = 0$ sabe-se quais dos possíveis resultados i, sendo a probabilidade associada p_i, irá acontecer e, portanto, o conhecimento sobre o processo descrito por uma distribuição de probabilidade é máximo. No entanto, quando se trata de uma distribuição uniforme nosso conhecimento é mínimo, visto que cada resultado apresenta a mesma probabilidade de ocorrência, $P = \{p_i = 1/N; i = 1, \dots, N\}$, sendo assim, $S[P_e] = S_{max} = \ln N$ e a incerteza é máxima (ROSSO et al.,2016). Para o caso discreto, é definido a entropia de Shannon normalizada, como segue:

$$H[P] = \frac{S[P]}{S_{max}} \tag{21}$$

em que $0 < H < 1$.

Se o sistema estudado estiver em um estado muito ordenado, ou seja a maior parte dos valores de p_i estiverem próximos à zero, então a entropia normalizada de Shannon estará próxima à zero ($H \approx 0$). Por outro lado, se o sistema se encontra em estado muito desordenado, que ocorre quando os valores de p_i estiverem em torno de um mesmo valor, tem-se que $H \approx 1$ (ROSSO et al., 2010).

Para o uso da entropia é necessário fornecer a distribuição de probabilidade mais adequada associada à série temporal (ROSSO et al., 2013). Uma metodologia para determinar a distribuição de probabilidade P a partir das séries temporais é a abordagem de Bandt e Pompe (2002), sendo uma metodologia simbólica simples e robusta que leva em consideração a causalidade da série temporal. Os dados simbólicos são utilizados para classificar os valores da série e definir a reordenação dos dados inseridos em ordem crescente, equivalente a uma reconstrução do espaço de fase com um comprimento padrão D e $lag\ \tau$. Desse modo, é possível quantificar a diversidade dos padrões oriundos de uma série temporal (ROSSO et al., 2013).

2.7 APRENDIZADO DE MÁQUINA

Machine learning, termo original em inglês, ou aprendizado de máquina é um ramo da ciência da computação que evoluiu do estudo de reconhecimentos de padrões em inteligência artificial. Usa-se da teoria estatística para construir modelos matemáticos, pois a principal tarefa é fazer inferências a partir de amostras (ALPAYDIN, 2020). Como o sucesso do aprendizado depende dos dados utilizados, machine learning é inerentemente relacionada à análise de dados e estatística. De forma geral, as técnicas de aprendizagem são métodos baseados em dados combinando conceitos fundamentais da ciência da computação com ideias da estatística, probabilidade e otimização (MOHRI et al.,2018).

Segundo Russel e Norvig (2020) o aprendizado é basicamente classificado em três categorias, de acordo com a natureza dos dados disponíveis.

- No **aprendizado supervisionado** um conjunto de dados rotulados previamente definidos são usados como base para encontrar uma função que seja capaz de predizer rótulos desconhecidos. Nesse tipo de aprendizagem pode-se estimar

números reais ou valores dentro de um conjunto finito. É principalmente usado para classificação de objetos.

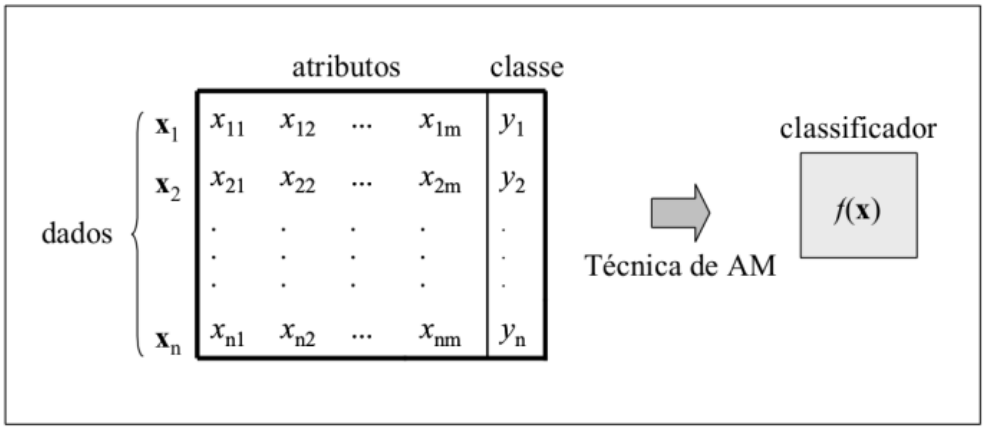
- No **aprendizado não supervisionado** o conjunto de dados não possui rótulos. O objetivo é encontrar similaridades entre os objetos analisados, criando assim agrupamentos a partir de suas características.
- Por fim tem-se a **aprendizagem por reforço**, onde a máquina tenta aprender qual a melhor resposta de acordo com as circunstâncias apresentadas. É um treinamento de modelo para tomar uma sequência de decisões, em que são fornecidos feedbacks quanto a recompensas e punições, na medida em que é navegado o espaço do problema. É mais utilizado na robótica.

O foco deste trabalho será nos métodos supervisionados, para classificação dos objetos, neste caso as assinaturas, que se quer categorizar como falsas ou verdadeiras.

Alguns conceitos são importantes para o aprendizado supervisionado. Dado conjunto de objetos na forma (X_i, y_i), em que X_i representa um objeto, ou instância, e y_i denota o seu rótulo, que também pode ser denotado como classe C_i, deve-se produzir um classificador, também denominado modelo, capaz de predizer o rótulo de novos dados. Esse processo de indução de um classificador a partir de uma amostra de dados é chamado de treinamento. Os rótulos ou classes são o fenômeno de interesse sobre o qual se deseja fazer previsões. Se os rótulos assumem valores discretos, tem-se um problema de classificação. Caso os rótulos possuam valores contínuos, tem-se uma regressão. Cada objeto é representado por um vetor de características. Cada característica, também chamada de atributo, expressa um determinado aspecto do objeto (LORENA; CARVALHO, 2007).

Esses conceitos são representados de forma simplificada na figura 4. Tem-se um conjunto com n objetos. Cada objeto X_i possui m atributos, ou seja, $\mathrm{X}_i = (x_1, \dots, x_m)$. As variáveis y_i representam as classes.

Figura 4 - Conceitos de aprendizagem supervisionada



Fonte: Adaptado de Lorena & Carvalho (2007)

Como mencionado anteriormente a classificação é um método de aprendizagem supervisionada para atribuir rótulos a uma amostra com base nos atributos. São vários os métodos de classificação, entre eles tem-se o Naive Bayes, Árvore de decisões, Floresta aleatória, Máquina de vetores de suporte e regressão logística.

2.7.1 Naive Bayes

O algoritmo de Naive Bayes é um classificador probabilístico bastante usado no *machine learning*, fortemente baseado no teorema de Bayes. A principal característica do algoritmo é que ele desconsidera completamente a correlação entre variáveis, ou seja, cada característica é tratada de forma independente. Assim o método trata sobre probabilidade condicional, isto é, qual a probabilidade do evento A ocorrer, dado o evento B.

De acordo com Narashima e Susheela (2011) dada uma instância a ser classificada, representada por um vetor $\boldsymbol{X} = (\boldsymbol{x_1}, \dots, \boldsymbol{x_n})$ representando n características (variáveis independentes), são atribuídas as probabilidades para essa instância,

$$p(C_k|x_1, \dots, x_n)$$

para cada K possíveis saídas ou classe C_k. Usando o teorema de Bayes, a probabilidade condicional pode ser decomposta como,

$$p(C_k|\boldsymbol{X}) = \frac{p(C_k)p(\boldsymbol{X}|C_k)}{p(\boldsymbol{X})} \tag{22}$$

como esse algoritmo considera as características independentes tem-se que a probabilidade $p(x_i|C)$ é a mesma probabilidade de $p(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, C)$, pode-se então dizer que

$p(x_1, \dots, x_n|C) = p(x_1|C) * p(x_2|C) * \dots * p(x_n|C)$, também temos que a probabilidade de $p(x_1, \dots, x_n)$ é constante, logo:

$$p(C_k|\boldsymbol{X}) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{23}$$

onde Z é uma constante a depender dos valores das características.

O classificador Naive Bayes combina o modelo de probabilidade de Bayes com uma regra de decisão, no caso a regra é selecionar a hipótese com maior probabilidade, isto é conhecido como o *maximum posteriori probability*, sendo assim o classificador é definido pela seguinte função:

$$\hat{C} = argmax\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{24}$$

2.7.2 Árvore de decisão

A árvore de decisão está entre os principais algoritmos para classificação e previsão de dados. Estes modelos utilizam a estratégia de dividir para conquistar, um problema complexo é decomposto em subproblemas mais simples e recursivamente esta técnica é aplicada a cada subproblema (GAMA, 2004). A partir dos dados é possível extrair regras do tipo “se-então” que são facilmente testadas.

Segundo Bramer (2007) uma árvore de decisão é criada através de um processo chamado *splitting on attributes* (partição de atributos), isto é, testando os valores dos atributos e então criando um ramo para cada um dos seus possíveis valores. No caso de atributos contínuos, normalmente, o teste é se o valor é ‘menor que’ ou ‘maior que’ um determinado limite. O processo de particionamento continua até que cada ramo possa ser rotulado com uma única classificação. O critério utilizado para fazer as partições é o da utilidade do atributo para a classificação, procura-se associar a cada nó de decisão o atributo ‘mais informativo’ entre aqueles ainda não utilizados no caminho desde a raiz da árvore. Um dos critérios de partição mais conhecido é o índice Gini.

O índice Gini mede o grau de heterogeneidade dos dados. Logo pode ser usado para medir a impureza dos dados. O índice Gini é calculado para cada atributo. Se existem k classes, com a probabilidade da i-ésima classe sendo p_i, o índice Gini é definido como (BRAMER, 2007):

$$índice\ Gini = 1 - \sum_{i=1}^{k} p_i^2 \quad (25)$$

p_i é a frequência relativa de cada classe em cada nó. Deve-se então calcular a diferença entre o índice Gini antes e após a divisão. O valor do índice Gini do j-ésimo subgrupo resultante da partição por um atributo específico é G_j, definido da seguinte forma:

$$G_j = 1 - \sum_{i=1}^{k} (f_{ij}/N_j)^2 \quad (26)$$

Onde N_j é o número de instâncias no subgrupo j, e f_{ij} é o número de instâncias classificadas como i dentro do subgrupo j. O atributo escolhido é aquele com maior redução no valor do índice Gini.

O processo de criação de uma árvore de decisão, chamado de indução, pode levar a uma alta demanda computacional. De maneira exaustiva, o número de árvores de decisão possíveis cresce fatorialmente à medida que o número de atributos aumenta. Por isso existem heurísticas para a indução de árvore de decisão. O Top-Down Induction Decision Tree (TDIDT) é um algoritmo bastante utilizado para indução de árvores de decisão.

De acordo com Bramer (2007) o algoritmo TDIDT é baseado em três possibilidades sobre um conjunto de treinamento T contendo as classes $C_1, C_2, \dots, C_k$:

- T contém um ou mais objetos, sendo todos da classe C_j. Então a árvore de decisão para T é um nó folha que identifica a classe C_j.
- T não contém objetos. A árvore de decisão também é um nó folha, mas a classe associada deve ser determinada por informação externa. Por exemplo, pode-se utilizar o conhecimento do domínio do problema.
- T contém objetos pertencentes a mais de uma classe. Neste caso, a ideia é dividir em T subconjuntos que são coleção de objetos com classes únicas. Para isso, é escolhido um atributo A, que possui um ou mais possíveis resultados $O_1, O_2, \dots, O_n$. T é particionado em subconjuntos $T_1, T_2, \dots, T_n$, onde T_i contém todos os objetos de T que tem resultado O_i para o atributo A, e uma aresta para cada possível resultado, ou seja, n arestas. No lugar de um único atributo A, pode também ser considerado um subconjunto de atributos.

O algoritmo é então aplicado recursivamente para cada subconjunto de objetos de T_i, com i variando de 1 a n. Basicamente é um algoritmo guloso que procura, sobre um conjunto de

atributos, aqueles que melhor dividem o conjunto de objetos em subconjuntos. No início todos os objetos são colocados em um único nó, o nó raiz. Depois um atributo é escolhido para representar o teste desse nó e, assim, dividir os objetos em subconjuntos. O processo então se repete até que todos os objetos estejam classificados ou até que todos os atributos tenham sido utilizados. O critério de partição define qual atributo é utilizado em cada nó da árvore de decisão (BRAMER, 2007).

2.7.3 Floresta aleatória

A floresta aleatória, do inglês *Random Forest*, é um método de classificação baseado em árvores de decisão, em que múltiplas árvores são geradas e ao final seleciona a classe mais popular, de modo a obter uma classificação com maior acurácia (BREIMAN, 2001). Além de ser uma técnica muito usada em classificação, também apresenta bom desempenho quando usada em regressão e estudo de importância (VERIKAS et al., 2011).

Nesse método tem-se um conjunto de árvores aleatórias não correlacionadas. A construção de uma floresta aleatória utiliza uma técnica de bootstrap para criar o subconjunto de dados utilizados para a construção do crescimento das árvores, reamostrando de forma aleatória e com reposição o vetor das classes Y e a matriz de variáveis explicativas X (BREIMAN, 2001). A floresta aleatória também apresenta bom desempenho com conjuntos de dados de alta dimensão e com problemas de multicolinearidade (BELGIU; DRAGUT, 2016). O resultado da classificação é atingido por um sistema de votação da classe mais popular entre as árvores que foram criadas.

2.7.4 Extreme Gradient Boosting

O *Extreme Gradient Boosting* (XGBoost) utiliza o conceito de árvore de decisão associado a estrutura do *Gradient Boosting*. Sendo assim, esse método se baseia na técnica de *boosting*, que consiste em re-amostrar classificadores, com reposição, várias vezes, entretanto, os dados re-amostrados são construídos de modo que obtenham aprendizado com a classificação realizada na amostragem anterior. Para obter o resultado final, depois de todas as re-amostragens, usa-se um método de combinação ponderado pelo desempenho de classificação em cada modelo (GROVER, 2017).

No XGBoost, o objetivo de cada arvore de decisão é minimizar a função perda, ou seja, minimizar o gradiente da função objetivo do modelo. Desse modo, esse método otimiza a

função objetivo de forma robusta, utilizando um algoritmo mais sensível à dispersão quando se ramifica as árvores (CHEN; GUESTRIN, 2016).

Dado um conjunto de dados de treinamento, com n observações e m variáveis, sendo $D = \{X_i, y_i\}$, em que X são os atributos e y são as classes. O *Extreme Gradient Boosting* utiliza K árvores para fazer a classificação, pois como se trata de um modelo iterativo, tende a melhorar a árvore de decisão anterior a cada rodada, da seguinte forma:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(X_i), \qquad f_k \in \mathbb{F} \tag{27}$$

onde $\hat{y}_i$ é a classe predita pelo modelo e f é uma função no espaço $\mathbb{F}$ de todas as possíveis árvores. A função objetivo é formada por dois elementos, sendo eles, a função perda e um termo de regularização:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{28}$$

em que, l é a função perda e $\Omega(\cdot)$ é o termo de regularização que serve para penalizar a complexidade do modelo, buscando evitar o sobre ajuste (*overfitting*) do modelo.

A cada iteração a função objetivo é melhorada levando em consideração a árvore anterior, da seguinte forma:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i{}^{(t-1)} + f_t(X_i)) + \Omega(f_t) \tag{29}$$

onde $\hat{y}_i{}^{(t-1)}$ é a classe predita do i-ésimo elemento na t-ésima iteração e f_t são as funções de aprendizado que possuem estrutura da árvore (CHEN; GUESTRIN, 2016).

2.7.5 Máquina de vetores de suporte

A máquina de vetores de suporte (SVM, do inglês *Support Vector Machine*) é uma técnica de aprendizado de máquina fundamentada no princípio indutivo da minimização do risco estrutural. Segundo Vapnik (2000) este princípio está presente na teoria do aprendizado estatístico, a qual estabelece condições matemáticas que auxiliam na escolha de um classificador a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e a sua complexidade, com o objetivo de obter um bom desempenho também para novos dados do mesmo domínio.

O objetivo do SVM é a obtenção de hiperplanos que dividam as amostras de tal maneira que sejam otimizados os limites de generalização. De acordo com Lorena e Carvalho (2007) o desempenho desejado de um classificador f é que o mesmo obtenha o menor erro durante o treinamento, sendo o erro mensurado pelo número de predições incorretas de f.

As máquinas de vetores de suporte podem ser empregadas na obtenção de fronteiras lineares para a separação de dados pertencentes a duas classes. Segundo Smola e Schölkopf *apud* Lorena (2002) as SVMs lineares definem fronteiras lineares a partir de dados linearmente separáveis. Seja T um conjunto de treinamento com n dados $x_i \in$ X e seus respectivos rótulos $y_i \in$ Y, em que X constitui o espaço dos dados e Y $= \{-1, +1\}$. T é linearmente separável se é possível separar os dados das classes $+1$ e -1 por um hiperplano.

A equação de um hiperplano é apresentada na equação 30, onde $\mathbf{w} \cdot \mathbf{x}$ é o produto escalar entre os vetores $\mathbf{w}$ e $\mathbf{x}$, $\mathbf{w} \in$ X é o vetor normal ao hiperplano descrito e $\frac{b}{\|\mathbf{w}\|}$ corresponde a distância do hiperplano em relação a origem.

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{30}$$

Essa equação divide o espaço dos dados em duas regiões: $\mathbf{w} \cdot \mathbf{x} + b > 0$ e $\mathbf{w} \cdot \mathbf{x} + b < 0$. Uma função sinal $sgn(f(\mathbf{x}))$ pode ser empregada na obtenção das classificações, conforme a equação a 28 (SMOLA et al. *apud* LORENA, 1999):

$$sgn(f(\mathbf{x})) = \begin{cases} +1 \; se \; \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 \; se \; \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \tag{31}$$

Com isso um número infinito de hiperplanos equivalentes são possíveis, porém deve-se selecionar $\mathbf{w}$ e b de forma que os objetos mais próximos ao hiperplano satisfaçam:

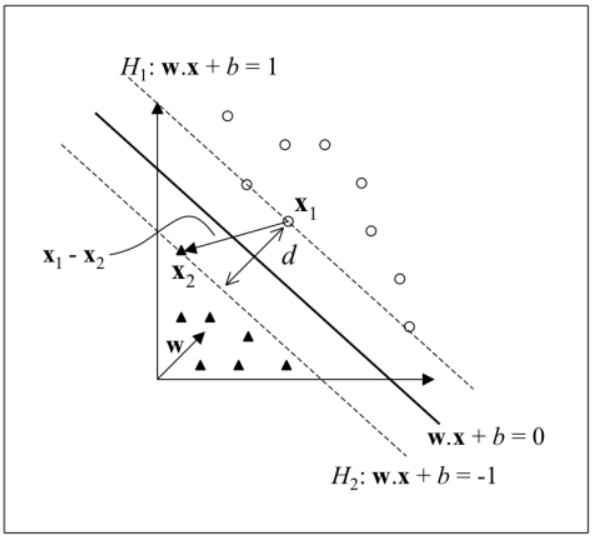
$$|\mathbf{w} \cdot \mathbf{x} + b| = 1 \tag{32}$$

Assim tem-se que

$$\begin{cases} \mathbf{w} \cdot \mathbf{x} + b \geq +1 \; se \; y = +1 \\ \mathbf{w} \cdot \mathbf{x} + b \leq -1 \; se \; y = -1 \end{cases} \tag{33}$$

Seja x_1 um ponto no hiperplano $H_1: \mathbf{w} \cdot \mathbf{x} + b = +1$ e Seja x_2 um ponto no hiperplano $H_2: \mathbf{w} \cdot \mathbf{x} + b = -1$, conforme mostrado na figura. Projetando $x_1 - x_2$ na direção de $\mathbf{w}$, perpendicular ao plano separador $\mathbf{w} \cdot \mathbf{x} + b = 0$, é possível obter a distância entre os hiperplanos.

Figura 5 - Distância d entre os hiperplanos H1 e H2



Fonte: Adaptado de Lorena & Carvalho (2007)

Tem-se que $\mathbf{w} \cdot \mathbf{x_1} + b = +1$ e $\mathbf{w} \cdot \mathbf{x_2} + b = -1$. A diferença entre essas equações fornece $\mathbf{w} \cdot (\mathbf{x_1} - \mathbf{x_2}) = 2$. É chamado de margem o comprimento do vetor diferença projetado na direção de $\mathbf{w}$. Tem-se então que a margem é dada por:

$$\|\mathbf{x_1} - \mathbf{x_2}\| = \frac{2}{\|\mathbf{w}\|} \tag{34}$$

Como $\mathbf{w}$ e b foram escalados de forma a não haver exemplos entre H_1 e H_2, $\frac{1}{\|\mathbf{w}\|}$ é a distância mínima entre o hiperplano separador e os dados de treinamento (CAMPBELL *apud* LORENA, 2000). Logo a maximização da margem de separação de dados envolve a minimização de $\|\mathbf{w}\|$. Assim, tem-se o seguinte problema de otimização:

$$\begin{gathered} \underset{w,b}{minimizar}\ \frac{1}{2}\|\mathbf{w}\|^2 \\ sujeito\ a\colon\ y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \geq 0, \forall i = 1, \ldots, n \end{gathered} \tag{35}$$

As restrições garantem que não haja dados de treinamento entre as margens de separação das classes. Por conta disso, esse tipo de máquina de vetores de suporte é chamado de SVM com margens rígidas.

Trata-se de um problema de otimização quadrático, com restrições lineares. A função objetivo a ser minimizada é convexa, logo há somente um mínimo, que é global (PASSERINI *apud* LORENA, 2004). Problemas desse tipo podem ser solucionados com a introdução de uma função lagrangeana, que engloba as restrições à função objetivo, associada a parâmetros denominados multiplicadores de lagrange α_i.

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i (\, y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \tag{36}$$

Deve-se minimizar a função lagrangeana, o que implica a maximização das variáveis α_i e minimização de $\mathbf{w}$ e b. Tem-se assim um ponto de sela, onde:

$$\frac{\partial L}{\partial b} = 0 \quad \text{e} \quad \frac{\partial L}{\partial \mathbf{w}} = 0 \tag{37}$$

A resolução dessas equações leva, respectivamente a:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{38}$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{39}$$

O problema de otimização passa então a ser:

$$\underset{\alpha}{maximizar} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{40}$$

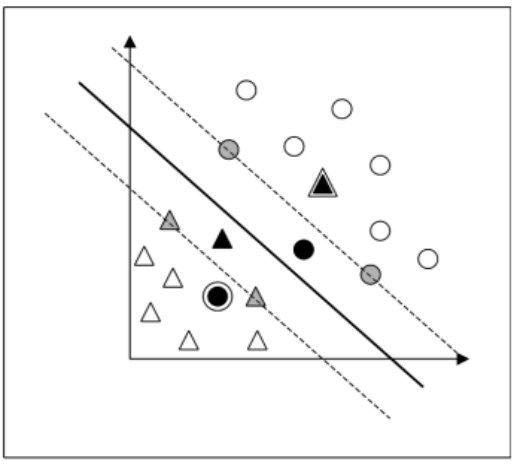
$$sujeito\ a\text{:} \begin{cases} \alpha_i \geq 0, \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{41}$$

Porém há situações que os dados não são linearmente separáveis. Nesse contexto as SVMs podem ser estendidas para lidar com conjuntos de treinamentos mais gerais. Para isso permite-se que alguns dados possam violar a restrição do problema original. Utiliza-se então variáveis de folga ξ_i (SMOLA et al. *apud* LORENA, 1999).

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0, \;\; \forall i = 1, \ldots, n \tag{42}$$

De acordo com Lorena e Carvalho (2007) a aplicação desse procedimento suaviza as margens do classificador linear, permitindo que alguns permaneçam entre os hiperplanos H_1 e H_2 e também a ocorrência de alguns erros de classificação. Por isso, essas SVMs são referenciadas como SVMs de margens suaves. A figura 6 representa um conjunto que não é separável linearmente.

Figura 6 - Conjunto não separável linearmente



Fonte: Adaptado de Lorena & Carvalho (2007)

Se o valor de ξ_i for 0, o objeto está fora da região entre os hiperplanos e é classificado corretamente. Se for positivo entre 0 e 1, mede a distância do objeto em relação aos hiperplanos. Quando o dado é classificado erroneamente, a variável de folga ξ_i, assume valor maior do que 1. A nova função objetivo para o problema passa a ser:

$$\underset{\mathbf{w}, b, \xi}{minimizar} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n} \xi_i\right) \tag{43}$$

Onde C é uma constante que impõe um peso à minimização dos erros no conjunto de treinamento em relação a minimização da complexidade do modelo (PASSERINI, 2004).

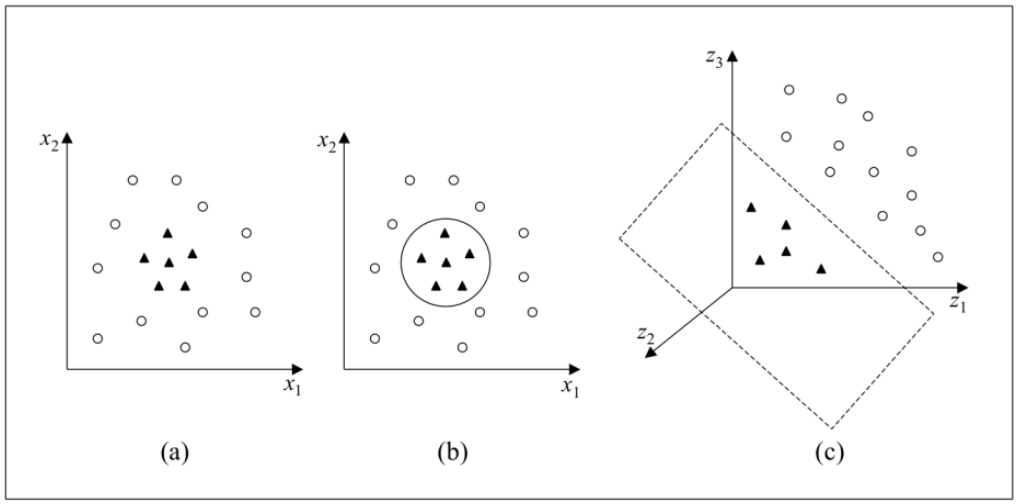
A solução envolve procedimento semelhante ao apresentado anteriormente, com introdução de uma função Lagrangiana e tornando suas derivadas parciais nulas. Tem-se como resultado o seguinte problema:

$$\underset{\alpha}{maximizar} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \tag{44}$$

$$sujeito\ a: \begin{cases} 0 \geq \alpha_i \geq C, \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{45}$$

Há casos que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano. A figura 7 mostra um exemplo em que o uso de uma fronteira curva seria mais adequado na separação das classes.

Figura 7 - (a) Conjunto de dados não linear; (b) Fronteira não linear no espaço de entradas; (c) Fronteira linear no espaço de características



Fonte: Adaptado de Lorena & Carvalho (2007)

Segundo Müller et al. *apud* Lorena (2001) deve-se então mapear o conjunto de treinamento do seu espaço original para um novo espaço de maior dimensão, denominado espaço de características (*feature space*), que é linear. Para isso é preciso encontrar uma transformação não linear. Por exemplo considerando o conjunto da figura a, e transformando os dados de $\Re^2$ para $\Re^3$ com o mapeamento da equação a seguir, o conjunto não linear passa a ser separável linearmente. A função apresentada embora linear em $\Re^3$ (figura c), corresponde a uma fronteira não linear em $\Re^2$ (figura b).

$$\Phi(\mathbf{x}) = \Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \tag{46}$$

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0 \tag{47}$$

Nosso problema de otimização passa a ser:

$$\underset{\alpha}{maximizar} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) \tag{48}$$

Sob as restrições da equação 44.

É preciso realizar o cálculo de produtos escalares entre os dados no espaço de características, isso é obtido com o uso de funções denominadas Kernels. De acordo com Herbrich (2001), um kernel K recebe dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de entrada e computa o produto escalar desses dados no espaço de características. Tem-se então:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{49}$$

Alguns dos Kernels mais utilizados na pratica são os polinomiais, os gaussianos e os sigmoidais, como na tabela 1:

Tabela 1 - Tipos de função kernel

Tipo de Kernel	**Função** $K(\mathbf{x}_i, \mathbf{x}_j)$	**Parâmetros**
Linear	$\mathbf{x}_i \cdot \mathbf{x}_j$	-
Polinomial	$(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + k)^d$	δ, k e d
Gaussiano	$\exp\left(-\sigma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$	σ
Sigmoidal	$\tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + k)$	δ e k

2.7.6 Regressão logística

Um outro método de aprendizagem de máquina para classificação é a regressão logística. Os métodos de regressão tem como objetivo descrever as relações entre a variável resposta (Y) e a variável explicativa (X). No modelo logístico, no entanto, a variável resposta é binária, atribuindo-se o valor 1 para o acontecimento de interesse (sucesso) e o valor 0 para o acontecimento complementar (fracasso), com probabilidades $\pi_i = P(Y = 1|X = x_i)$ e $1 - \pi_i = P(Y = 0|X = x_i)$ respectivamente (HOSMER & LEMESHOW, 2000). Ou seja, a regressão logística é uma técnica que permite estimar a probabilidade associada a ocorrência de determinado evento em face de um conjunto de variáveis explanatórias.

Diferente de um modelo de regressão linear, em um modelo de regressão logística os erros não são normalmente distribuídos, pois o valor da variável resposta dado x_i, é expresso por $Y_i = \pi_i - \varepsilon_i$, como a quantidade Y_i, só pode assumir dois valores, então $\varepsilon_i = 1 - \pi_i$ para $Y_i = 1$ ou $\varepsilon_i = -\pi_i$, para $Y_i = 0$, segue que ε_i tem distribuição com média zero e variância $\pi_i(1 - \pi_i)$.

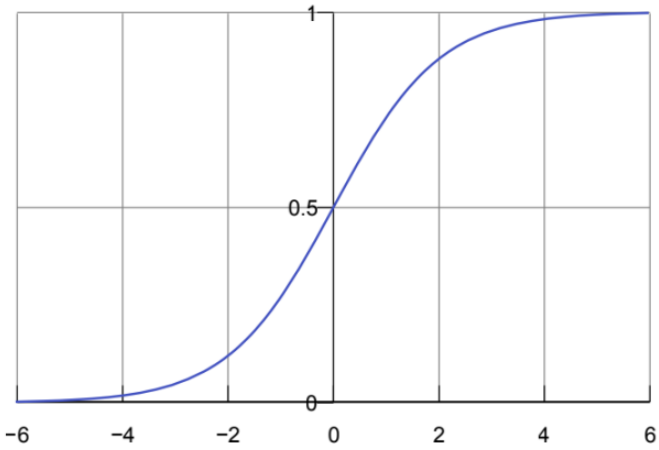
Também não há homogeneidade da variância, pois tem-se que $var(Y_i) = \pi_i(1 - \pi_i) = (\beta_0 + \beta_1 x_i)(1 - \beta_0 + \beta_1 x_i)$ então a variância de Y_i depende de x_i, e consequentemente, não é constante.

Além disso em um modelo de regressão linear, a resposta média $E(Y_i)$, está no intervalo $-\infty < E(Y|X = x_i) < +\infty$, porém em um modelo de regressão logística, devido à natureza de sua variável dependente, existe uma restrição para a resposta média, então tem-se que $0 < E(Y|X = x_i) < 1$, (HOSMER & LEMESHOW, 2000).

Segundo Oliveira (2016) A regressão logística utiliza uma função sigmoide para calcular a probabilidade condicional de um objeto pertencer a uma classe, que recebe como entrada real t e gera como saída um valor entre 0 e 1. Esta delimitação garante que o valor estimado pelo modelo permaneça no intervalo, permitindo a interpretação do valor como figura probabilística.

$$P(Y = 1) = \frac{1}{1 + e^{-t}} = \frac{1}{1 + e^{-\beta X_i}} \tag{44}$$

Figura 8 - Função Sigmoide



Para estimar os coeficientes, utiliza-se a estimativa por máxima verossimilhança

$$P(\pi_i|X, \beta) = \prod_{i=1}^{n} \left[\frac{1}{1 + e^{-\beta X_i}}\right]^{y_i} \left[1 - \frac{1}{1 + e^{-\beta X_i}}\right]^{1-y_i} \tag{45}$$

Devido à grande quantidade de expressões exponenciais, aplica-se o logaritmo negativo, desta forma, o produtório se torna um somatório e algumas expressões exponenciais são substituídas por logaritmos. A função log-verossimilhança a ser minimizada é (GROSSI, 2013):

$$L(\beta) = -\frac{1}{n}\sum_{i=1}^{n} [y_i \log(\pi_i) + (1 - y_i)\log(1 - \pi_i)] \tag{46}$$

3 METODOLOGIA

Nesta seção será abordada a metodologia usada para o estudo proposto nesta dissertação, que consiste em 5 etapas. 1) coleta de dados; 2) pré-processamento dos dados: transformar as coordenadas em séries temporais e usa-las para a formação das redes complexas; 3) extração dos dados: cálculo das métricas das redes complexas; 4) aprendizado de máquina: uso dos métodos de machine learning para a classificação das assinaturas; 5) avaliação do modelo de aprendizado.

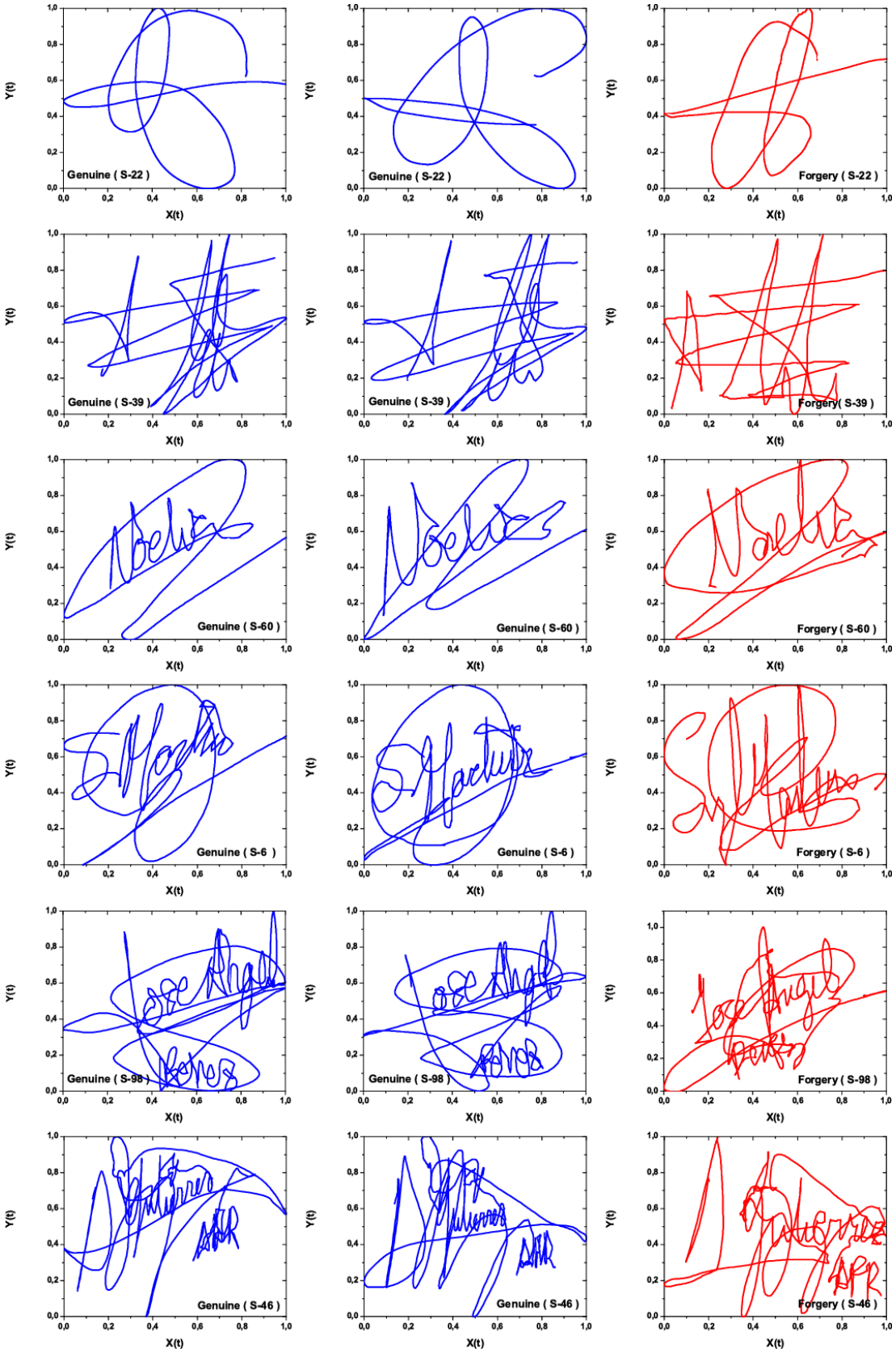
3.1 COLETA DE DADOS

Neste estudo foi utilizado o banco de dados de assinaturas do MCYT (Ministerio de Ciencia Y Tecnologia), que é disponível gratuitamente e amplamente utilizado (ORTEGA-GARCIA, 2003). Mais especificamente o subconjunto MCYT-100 da base de dados, que inclui 100 indivíduos e para cada um, 25 assinaturas genuínas e 25 falsas, contabilizando um total de 5000 assinaturas. As assinaturas falsas são produzidas por 5 outros indivíduos observando as imagens estáticas das assinaturas para imitá-las. O indivíduo n assina um conjunto de 5 amostras de sua assinatura genuína, e então 5 assinaturas falsas do indivíduo $n-1$. Depois novamente um conjunto de 5 assinaturas genuínas, e então 5 assinaturas falsas do indivíduo $n-2$. O procedimento é iterado pelo indivíduo n, fazendo assinaturas genuínas e imitando os indivíduos prévios $n-3$, $n-4$ e $n-5$. Resumindo, o indivíduo n produz 25 amostras de sua assinatura (em conjuntos de 5 amostras) e 25 assinaturas falsas (5 amostras para cada indivíduo, de $n-1$ a $n-5$). Desse modo, para o indivíduo n, 25 assinaturas falsas são produzidas pelos $n+1$ a $n+5$.

Cada assinatura era capturada digitalmente em um tablet que fornecia as seguintes sequências dinâmicas discretas no tempo: 1) a posição no eixo x, x_t; 2) a posição no eixo y, y_t; 3) a pressão aplicada pela caneta, p_t; 4) o ângulo azimutal da caneta em relação ao tablet, γ_t; e 5) a inclinação da caneta em relação ao tablet, φ_t.

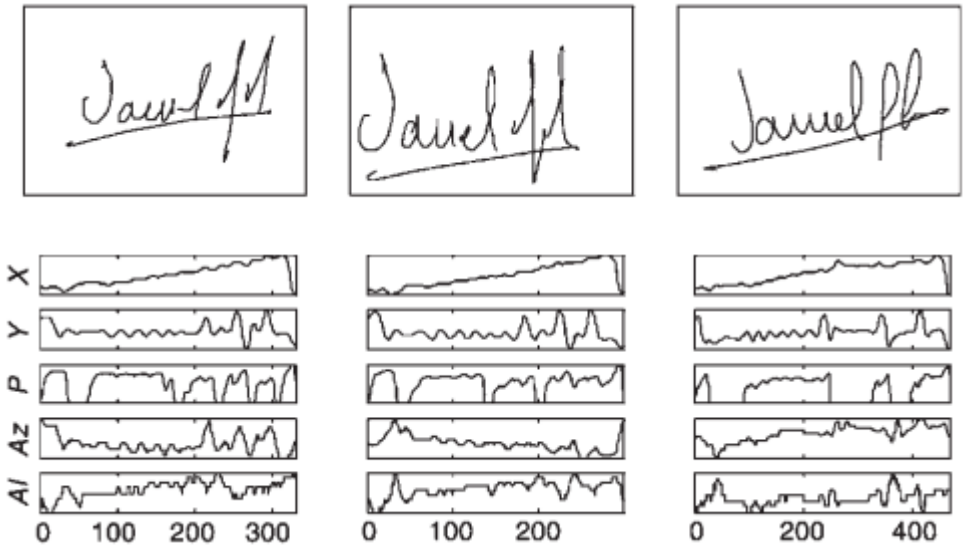
A figura 9 apresenta exemplos de seis indivíduos, sendo as duas primeiras colunas formadas por assinaturas genuínas e a terceira coluna por assinaturas falsas. A figura 10 apresenta gráficos a respeito de cada uma das informações contidas no banco de dados. Novamente as duas primeiras são baseadas em assinaturas verdadeiras e a última em falsas.

Figura 9 - Assinaturas de seis indivíduos do banco de dados MCYT



Fonte: Adaptado de Ortega-Garcia (2003)

Figura 10 - Gráficos das informações contidas no banco de dados: X (posição no eixo x), Y (posição no eixo y), P (pressão aplicada na caneta), Az (ângulo azimutal) e AI (ângulo de inclinação)



Fonte: Adaptado de Ortega-Garcia (2003)

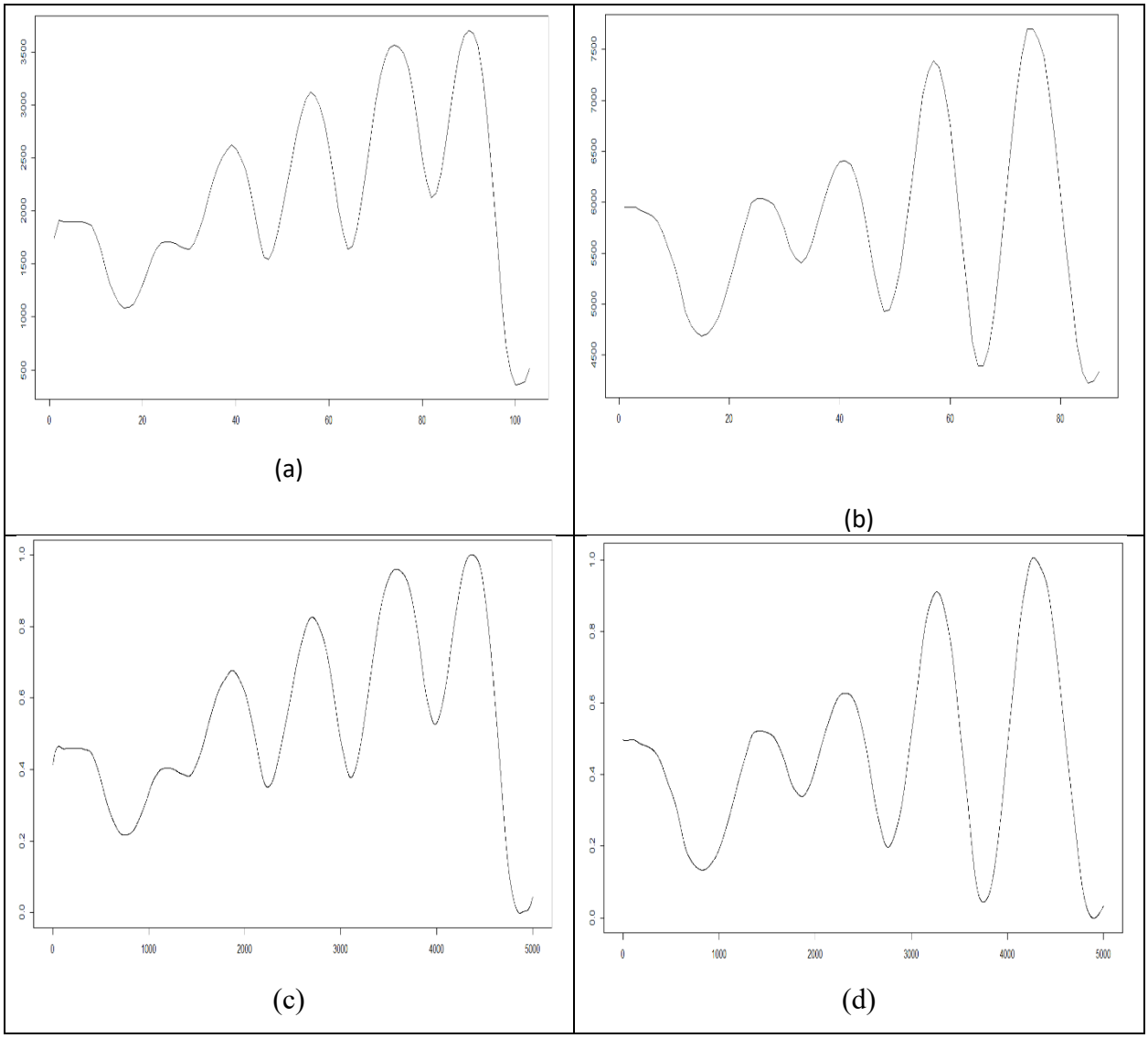
Apesar do banco de dados conter outros dados referentes às assinaturas como pressão, ângulo azimutal e inclinação, nesta dissertação foram utilizados apenas as posições das coordenadas x e y, pois essas quantidades são oferecidas pelos aparelhos de capturas de assinaturas dos mais simples aos mais sofisticados. Além disso a inclusão dos outros dados não apresenta mudanças significativas nos resultados do experimento. Este mesmo fato foi observado em Rosso et al.(2016).

3.2 PRÉ-PROCESSAMENTO DOS DADOS

É importante notar que o comprimento das séries temporais no banco de dados é variável. Sendo assim um pré-processamento deve ser realizado. Primeiro as coordenadas são re-escaladas em um quadrado unitário $[0,1] \times [0,1]$. Depois o número total de pontos original dos dados de cada série temporal é expandido para $M = 5000$ pontos usando um interpolador Hermite cúbico, essa técnica consiste em dividir o intervalo de interesse em vários subintervalos e interpolar, da forma mais suave possível, nestes subintervalos com polinômios pequenos, com grau menor ou igual a três (FRANCO, 2006). Sendo assim para cada indivíduo k $(k = 1, \dots, 100)$ e assinaturas associadas j $(j = 1, \dots, 25)$ serão analisadas duas séries temporais, denotadas por $X_j^{(k,\alpha)} = \{0 \leq \tilde{x}_{j;i}^{(k,\alpha)} \leq 1, i = 1, \dots, M\}$ e $Y_j^{(k,\alpha)} = \{0 \leq \tilde{y}_{j;i}^{(k,\alpha)} \leq 1, i = 1, \dots, M\}$, onde o índice $\alpha = G, F$ denota assinaturas genuínas e falsas respectivamente, e $\tilde{x}$ e $\tilde{y}$ são os valores interpolados.

A figura 11 mostra exemplos das transformações das séries temporais de assinaturas do indivíduo 1. A figura 11(a) apresenta a série temporal das coordenadas x da assinatura verdadeira de número 1, ela varia entre 360 e 3706 e possui 103 observações. Já a figura 11(b) mostra a série temporal das coordenadas x da assinatura verdadeira de número 24, que varia entre 4425 e 7703 e tem 87 observações. As figuras 11(c) e 11(d) apresentam as séries temporais suavizadas das assinaturas 1 e 24 respectivamente.

Figura 11 - séries temporais transformadas: (a) série temporal original das coordenadas x da assinatura verdadeira 1 do indivíduo 1; (b) série temporal original das coordenadas x da assinatura verdadeira 24 do indivíduo 1; (c) série temporal original suavizada da assinatura verdadeira 1; (d) série temporal suavizada da assinatura verdadeira 24.



3.3 EXTRAÇÃO DOS DADOS

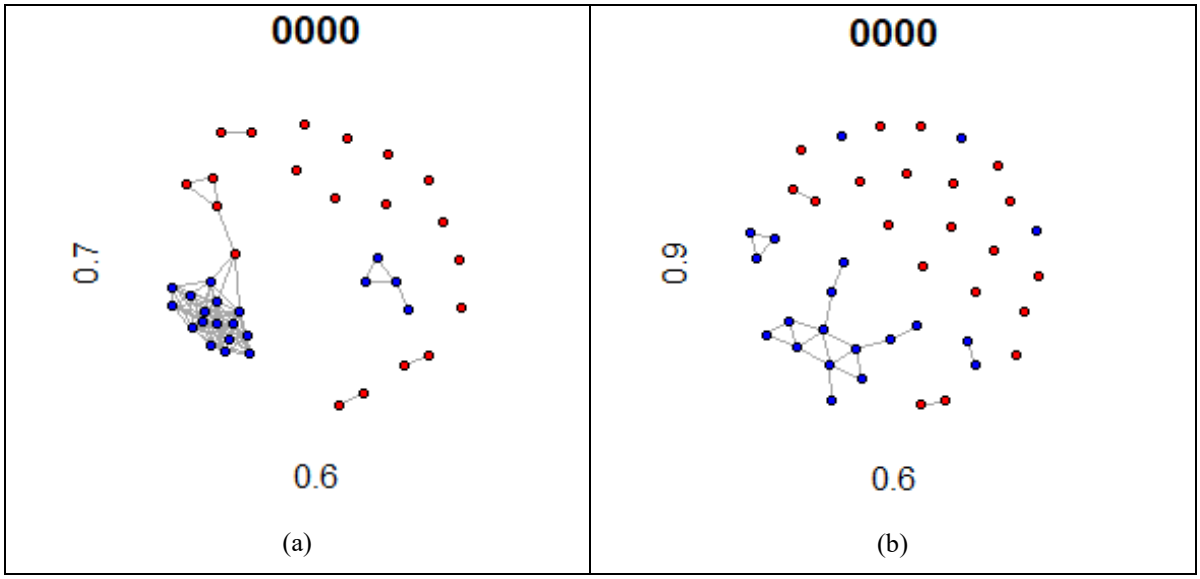
A ideia base do estudo é utilizar medidas de correlação entre as séries temporais das coordenadas de cada assinatura disponível por indivíduo, como matriz adjacência para transformar o banco de dados em uma rede. Desse modo, os elementos da rede (vértices) serão

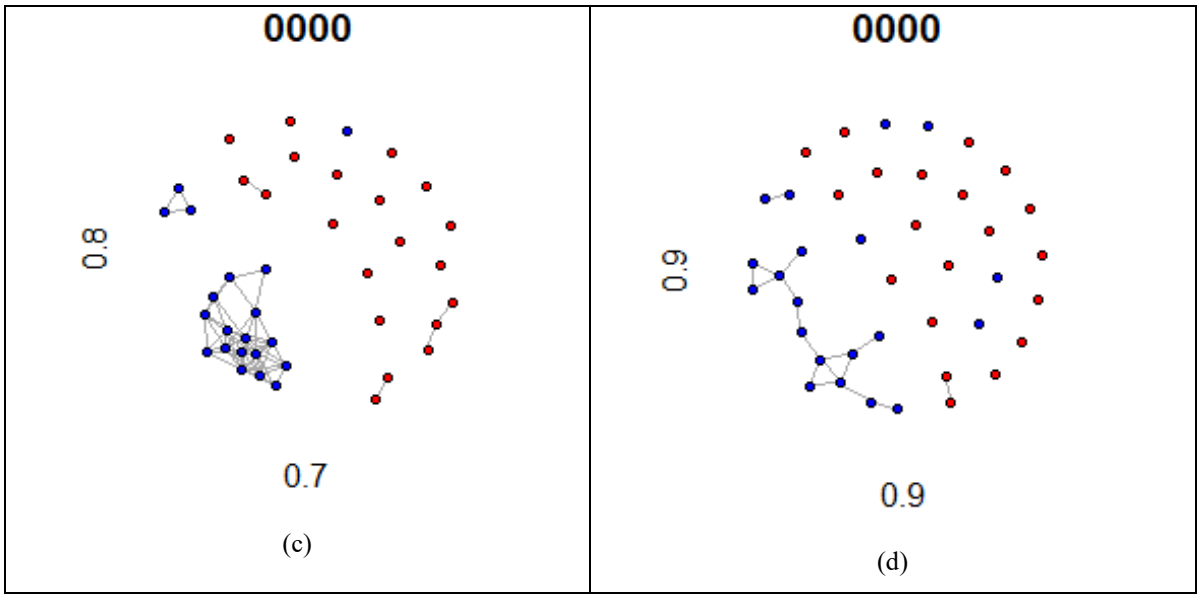
as assinaturas, a relação entre os elementos (arestas) será a correlação linear entre assinaturas e as medidas de correlação servirão para definir se essa relação é forte ou fraca.

No presente trabalho, as métricas de redes complexas serão calculadas utilizando o conjunto de funções disponíveis no pacote IGRAPH (CSARDI e NEPUSZ, 2006) – pacote computacional disponível no software livre ***R*** (R Core Team, 2020).

Primeiramente, foram selecionadas aleatoriamente 40 assinaturas para o treinamento dos modelos de classificação, sendo elas 20 falsas e 20 verdadeiras, então foram obtidas as correlações de pearson entre essas assinaturas do mesmo indivíduo, no caso duas correlações foram estabelecidas, uma para a série temporal formada pelas coordenadas x e outra para a série temporal das coordenadas y. As correlações foram utilizadas para estabelecer se existia relação entre as assinaturas, porém para isso foram estabelecidos limiares para as correlações, ou seja só existe ligação se ambas correlações estiverem acima de certos valores. Diferentes combinações de limiares geram grafos com topologias diferentes. Então, são formados grafos ponderados, se as correlações de ambas as coordenadas estiverem acima dos respectivos limiares, o peso da aresta será o da maior correlação considerando as duas coordenadas. A figura 12 ilustra alguns exemplos para as assinaturas do indivíduo 1, que na base de dados é chamado de "0000", na figura 12(a) os limiares são 0,6 e 0,7 para as coordenadas x e y respectivamente, em 12(b) são 0,6 e 0,9, em 12(c) são 0,7 e 0,8 e em 12(d) são 0,9 e 0,9. As assinaturas verdadeiras estão identificadas na cor azul e as falsas na cor vermelha.

Figura 12 - Redes do mesmo indivíduo com combinações de limiares diferentes





(c) (d)

Com o conjunto de assinaturas transformado em rede complexa, são extraídas as métricas, como o coeficiente de clusterização, as centralidades de grau, proximidade, intermediação e autovetor. Como diferentes combinações de limiares produzem resultados diferentes, foi escolhida aquela que fornecia maior correlação entre as centralidades de autovetor e os tipos de assinatura da rede gerada, pois esta é a centralidade que apresenta maior relação com o tipo de assinatura. Para isso foram testados limiares tanto para a coordenada x quanto para y num intervalo de 0,1 à 1 variando em 0,1, ou seja, testaram-se 100 combinações diferentes para cada rede.

Primeiro, são geradas as 100 redes com diferentes combinações, a partir das 40 assinaturas selecionadas aleatoriamente para treinamento, então são comparados os resultados das correlações dessas redes entre os tipos de assinatura e centralidade de autovetor. Por fim, seleciona-se a combinação de limiares que atingiu o maior resultado.

Para exemplificar a definição dos limiares, na figura 12(c) com os limiares 0,8 e 0,7 para as coordenadas x e y respectivamente, a correlação de Pearson entre as centralidades de autovetor e os tipos de assinatura é de 0,7943, ou seja, é maior que essa mesma correlação na figura 12(d) que tem limiares de 0,9 para coordenada x e 0,9 para coordenada y, e correlação de 0,5094, sendo assim os limiares escolhidos seriam os da figura 12 (c).

3.4 CLASSIFICAÇÃO

Nesta etapa, foram utilizados cinco métodos de classificação de aprendizado de máquina supervisionado para comparação de resultados. Os métodos são: Naive Bayes, Árvore de decisão, Floresta aleatória, Máquina de Vetores de Suporte e Regressão Logística.

Os dados de entrada do modelo serão as métricas resultantes da rede formada pelo conjunto das 40 assinaturas de treinamento, 20 assinaturas verdadeiras e 20 assinaturas falsas, assim as 10 assinaturas restantes de cada indivíduo são usadas como teste para medir a qualidade do modelo. Cada assinatura de teste é inserida individualmente a rede do indivíduo ao qual pertence e suas métricas são calculadas. Após isso, essa assinatura é retirada da rede e uma nova assinatura de teste é inserida. Este processo se repete até que todas as assinaturas tenham sido testadas. De um modo geral, após calculadas as métricas da rede do indivíduo 1 para diferentes combinações de limiares, escolhe-se a combinação com maior correlação entre a centralidade de autovetor e o tipo de assinatura, pois esta é a que apresenta maior correlação quando comparada com as demais, de acordo com as assinaturas selecionadas aleatoriamente. Faz-se o mesmo para os outros 99 indivíduos, assim, no final tem-se 4000 assinaturas e suas respectivas métricas que servem como conjunto de treinamento, então o modelo é ajustado com essas informações e testado nas 1000 assinaturas restantes. Esse processo foi repetido em um esquema Monte Carlo com 100 repetições para obter estimativas médias de ajuste.

Todos os experimentos foram feitos no mesmo ambiente de programação, usando pacotes dedicados para os algoritmos de classificação, garantindo que os resultados alcancem a melhor eficiência possível e a performance não seja afetada por má implementação.

O primeiro método testado é o Naive Bayes, para isso foi utilizado o pacote *e1071* no software ***R***, que calcula probabilidades condicionais a posteriori de uma variável de classe categórica dada as variáveis preditoras independentes usando a regra de Bayes. O segundo método é a árvore de decisão, aqui foi usado o pacote *rpart*, o critério utilizado para fazer as partições é o índice de Gini. O terceiro método é a floresta aleatória, onde foi usado o pacote *randomForest*. O quarto método é a máquina de vetores de suporte, também utilizando o pacote *e1071*. E o último método aplicado é a regressão logística, nesse caso estabelecendo-se um limiar de 0,5 para a probabilidade resposta da regressão, ou seja, para valores acima de 0,5 as assinaturas seriam classificadas como verdadeiras, e abaixo desse valor como falsas.

3.5 AVALIAÇÃO DO MODELO DE APRENDIZADO

Para a avaliação do modelo algumas métricas para medir os erros de classificação são utilizadas, entre elas o *False Rejection Rate* (FRR) que identifica a porcentagem de assinaturas verdadeiras que são rejeitadas pelo modelo, ou seja, os falsos negativos, o *False Acceptance Rate* (FAR) como sendo a porcentagem de assinaturas falsas aceitas, ou seja, os falsos positivos e a *Average Error Rate* (AER) que é o erro médio considerando apenas FRR e FAR, ou seja, o percentual de classificações erradas independente do erro cometido, além do *Equal Error Rate* (EER) o erro obtido quando FAR = FRR, indica o ponto onde a curva do *False Acceptance Rate* e do *False Rejection Rate* é mínimo e ótimo (HAFEMANN et al., 2017). Essas métricas fazem uma comparação entre as classificações preditas pelo modelo e a classe real do objeto analisado.

A seguir é apresentado um pseudocódigo para o modelo proposto nesta dissertação.

Pseudocódigo do modelo proposto

Para cada indivíduo no banco de dados

Selecionar assinaturas de treino

Transformar dados das coordenadas x e y em séries temporais

Matriz de correlações das coordenadas x e y

Limiar $x = 0{,}1$, Limiar $y = 0{,}1$, ótimo $= 0$

Enquanto (Limiar x < 1)

Enquanto (Limiar y < 1)

Se ((correlação $x\ (i,j) >$ Limiar x) e (correlação $y\ (i,j) >$ Limiar y))

Se (correlação $x\ (i,j) \geq$ correlação $y\ (i,j)$)

$M_{ij} =$ correlação x

Senão $M_{ij} =$ correlação y

Senão $M_{ij} = 0$

Gerar rede a partir da matriz de adjacência M_{ij}

Extrair métricas da rede

Se (ótimo < correlação (tipo de assinatura, centralidade de autovetor))

Ótimo = correlação (tipo de assinatura, centralidade de autovetor)

Limiar_ótimo_x = Limiar x, Limiar_ótimo_y = Limiar y

Limiar y = Limiar y + 0,1

Limiar x = Limiar x + 0,1

Se ((correlação $x\ (i,j) >$ Limiar_ótimo_x) e (correlação $y\ (i,j) >$ Limiar_ótimo_y))

Se (correlação $x\ (i,j) \geq$ correlação $y\ (i,j)$)

$M_{ij} =$ correlação x

Senão $M_{ij} =$ correlação y

Senão $M_{ij} = 0$

Gerar rede a partir da matriz de adjacência M_{ij}

Extrair métricas da rede

Modelo de classificação

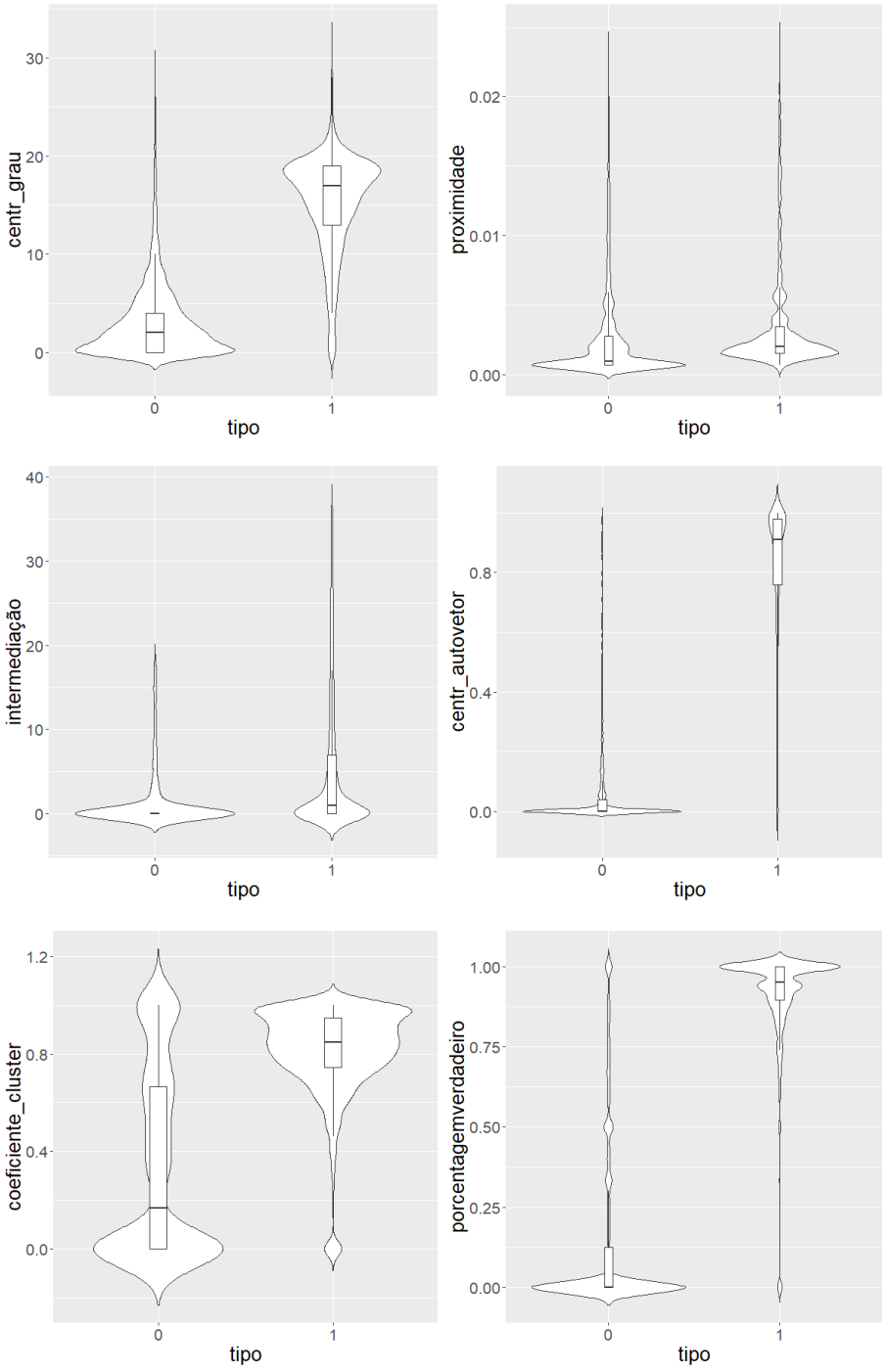
Classificação das assinaturas de teste

Avaliação de desempenho (métricas de erro)

4 ANÁLISE E RESULTADOS

Com as redes definidas, podem-se observar as relações entre as centralidades e o tipo de assinatura (falsa ou verdadeira) como mostram os gráficos de violino da figura 13.

Figura 13 - Gráficos de violino entre as métricas da rede e o tipo assinatura



O gráfico de violino é similar ao boxplot exceto que mostra também a densidade de probabilidade dos dados. Pela distribuição dos dados nota-se que a centralidade de grau, autovetor e o coeficiente de clusterização possuem relação com o tipo de assinatura. Assinaturas falsas (tipo 0) em geral têm centralidades mais baixas quando comparadas com assinaturas verdadeiras (tipo 1) que tendem a ter centralidades mais elevadas. Por outro lado, as centralidades de proximidade e de intermediação aparentemente não têm relação com o tipo de assinatura. Outro atributo que se mostrou bastante eficiente para a verificação das assinaturas foi a porcentagem de vizinhos do tipo verdadeiro, ou seja, a quantidade de assinaturas verdadeiras que fazem conexão com a assinatura analisada em relação ao total de conexões. Em geral assinaturas falsas têm pouquíssimas ligações com assinaturas verdadeiras, o que leva esta porcentagem a um nível bem baixo. Para os nós isolados o valor da porcentagem foi considerado igual a zero.

Para uma análise mais efetiva as correlações entre as centralidades e o tipo de assinatura foram estabelecidas para saber quais delas entrariam como atributos no modelo. A Tabela 2 a seguir apresenta os resultados.

Tabela 2 - Coeficiente de correlação de Pearson entre as medidas de centralidade e o tipo de assinatura

Métricas	**Correlação Pearson**
Centralidade de grau	0,7889
Centralidade de proximidade	0,0993
Centralidade de intermediação	-0,0309
Centralidade de autovetor	0,8669
Coeficiente de clusterização	0,5523
Porcentagem vizinhos verdadeiros	0,8725

Como já mencionado anteriormente, valores entre 0,10 e 0,29 podem ser considerados pequenos; escores entre 0,30 e 0,49 podem ser considerados como médios; e valores entre 0,50 e 1 podem ser interpretados como grandes. De acordo com os resultados, apenas as centralidades de grau, de autovetor, o coeficiente de clusterização e a porcentagem de vizinhos verdadeiros possuem alta correlação com o tipo da assinatura, as centralidades proximidade e intermediação têm correlação quase nula com o tipo. Desse modo, as variáveis explicativas escolhidas foram a centralidade de grau e de autovetor, o coeficiente de clusterização e a porcentagem de vizinhos verdadeiros.

De forma similar ocorre quando se utiliza a correlação de Kendall, que mensura o grau de associação entre as variáveis. Novamente as centralidades de grau, autovetor, o coeficiente de clusterização e a porcentagem de vizinhos verdadeiros possuem bastante relação com o tipo de assinatura, como mostra a tabela 3.

Tabela 3 - Coeficiente de correlação de Kendall entre as medidas de centralidade e o tipo de assinatura

Métricas	**Correlação Kendall**
Centralidade de grau	0,6345
Centralidade de proximidade	0,1934
Centralidade de intermediação	0,0865
Centralidade de autovetor	0,6613
Coeficiente de clusterização	0,3968
Porcentagem vizinhos verdadeiros	0,7447

Também foi considerado como variável regressora o quantificador de informação causal denominado de entropia de permutação baseada em Bandt-Pompe. Os valores do quantificador usados nesta dissertação foram calculados por Rosso et al. (2016). Essa medida, entretanto, funciona de forma muito local, sendo que o conjunto de assinaturas de cada indivíduo, possui informações bem diferentes. Por isso, para cada indivíduo, essas medidas foram normalizadas para que pudessem funcionar de maneira mais global no modelo. A normalização foi feita da seguinte maneira: $\big(x - min(x)\big)/\big(max(x) - min(x)\big)$, de modo que o maior valor seja 1 e o menor valor 0. A correlação entre o tipo de assinatura e a medida normalizada é de -0.4967, ou seja, é uma correlação moderada.

A tabela 4 mostra os dados de entrada selecionados para os modelos de aprendizado, com uma breve descrição para cada um deles, onde Y é a classe e X as variáveis explicativas.

Tabela 4 - Dados de entrada dos modelos de aprendizado

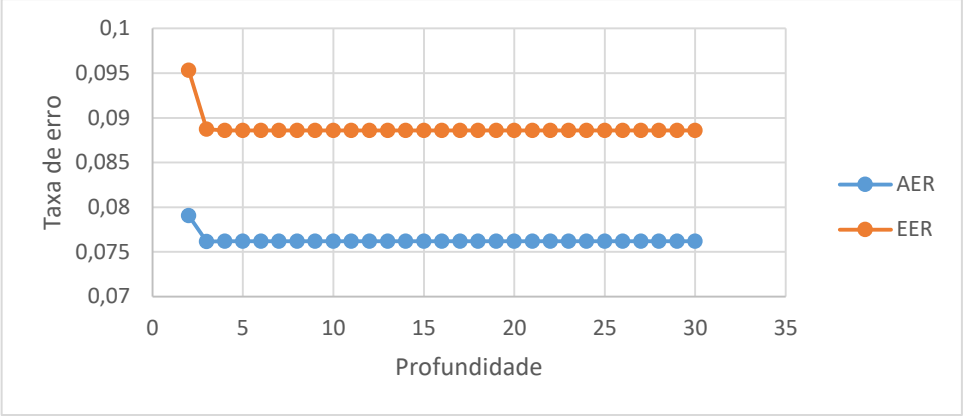
	Atributo	**Descrição**
Y	Tipo de assinatura	Se a assinatura é verdadeira ou falsa
X	Centralidade de grau	Número de arestas adjacentes ao vértice
	Centralidade de autovetor	Atribui importância para um vértice, em função da relação entre seus vizinhos
	Coeficiente de clusterização	Fração de arestas que os vizinhos de i possuem entre si e o máximo de arestas que eles poderiam possuir entre si
	Entropia	Distribuição de padrões da série temporal
	Porcentagem verdadeiras	Porcentagem de vizinhos do tipo verdadeiro

Com os atributos definidos, 100 modelos são gerados, sendo um para cada indivíduo, a partir dos dados de treinamento (40 assinaturas, sendo 20 falsas e 20 verdadeiras), cada modelo então é aplicado nos dados de teste (10 assinaturas, 5 falsas e 5 verdadeiras) e as métricas de erro são obtidas para a avaliação de desempenho.

Um fator importante para a aplicação dos métodos de classificação é a escolha dos hiperparâmetros, por isso quando necessário foi realizada uma análise quanto aos valores que seriam utilizados.

Para o método de árvore de decisão foram gerados alguns testes para saber a profundidade da árvore que retornava os melhores resultados. A figura 14 apresenta os resultados dos testes.

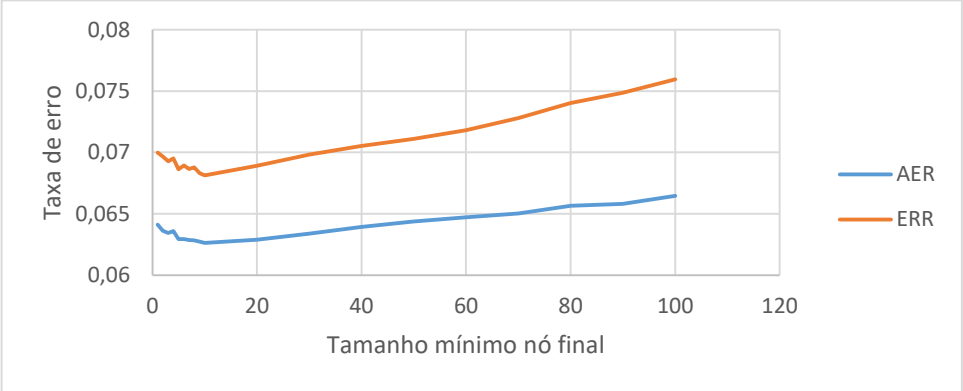
Figura 14 - Análise da profundidade para árvore de decisão



Como pode ser observado na figura 14, com profundidade 3 as taxas de erro se estabilizam, sendo assim o valor selecionado. Quanto maior o valor adotado maior o tempo computacional, e maior o risco de sobreajuste aos dados de treinamento.

Para a floresta aleatória foi realizado um estudo quanto ao tamanho mínimo dos nós finais das árvores na floresta. Valores elevados produzem árvores menores que levam menos tempo para serem geradas. A figura 15 mostra os resultados das simulações.

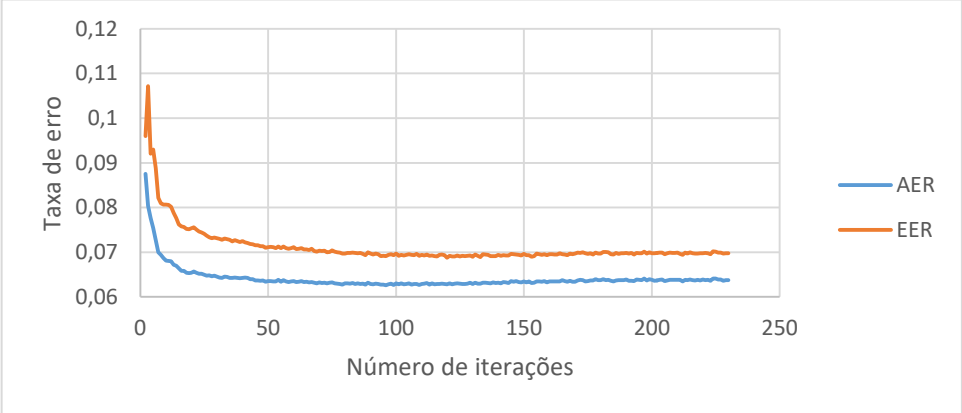
Figura 15 - Análise tamanho mínimo do nó final para floresta aleatória



O valor de 10 para o tamanho mínimo dos nós finais das árvores retorna taxas de erros mais baixas que os outros valores testados, tanto para o *equal error rate* quanto para o *average error rate*.

No caso do XGBoost foi analisado o número ideal de iterações para o modelo. Um número elevado de iterações pode levar a um sobreajuste do modelo. A figura 16 apresenta o gráfico com as taxas de erro de acordo com o número de iterações.

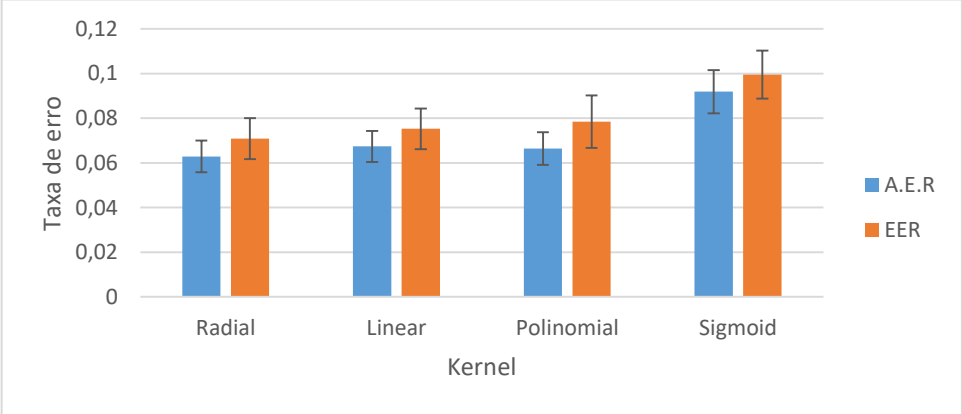
Figura 16 - Análise do número de iterações para o XGBoost



Em 96 iterações o modelo atinge as menores taxas de erro, sendo assim o valor adotado para esse parâmetro.

Já para a máquina de vetores de suporte é necessário definir a função kernel que melhor ajuste o modelo. Por isso foram testadas as quatro funções definidas anteriormente. A figura 17 mostra os resultados para cada uma das funções.

Figura 17 - Análise da função kernel para a máquina de vetores de suporte



Os melhores resultados são alcançados com o uso da função kernel do tipo radial (gaussiano). Aqui o sigma é definido como $\sigma = 1/(número\ de\ atributos)$.

A tabela 5 mostra as médias dos resultados atingidos nos diferentes métodos de classificação para um intervalo de confiança de 95%. Lembrando que cada método foi repetido 100 vezes.

Tabela 5 - Média dos erros para diferentes métodos de classificação para um intervalo de confiança de 95%

Método	FAR(%)	FRR(%)	AER(%)	EER(%)
NaiveBayes	$9{,}01 \pm 0{,}15$	$6{,}25 \pm 0{,}19$	$7{,}63 \pm 0{,}13$	$8{,}77 \pm 0{,}15$
Árvore de Decisão	$9{,}49 \pm 0{,}20$	$6{,}24 \pm 0{,}26$	$7{,}86 \pm 0{,}12$	$9{,}24 \pm 0{,}15$
Floresta Aleatória	$6{,}50 \pm 0{,}20$	$6{,}20 \pm 0{,}20$	$6{,}35 \pm 0{,}15$	$6{,}96 \pm 0{,}19$
XGBoost	$6{,}19 \pm 0{,}20$	$6{,}39 \pm 0{,}19$	$6{,}28 \pm 0{,}13$	$6{,}86 \pm 0{,}14$
SVM	$7{,}20 \pm 0{,}16$	$5{,}52 \pm 0{,}19$	$6{,}36 \pm 0{,}10$	$7{,}11 \pm 0{,}19$
Regressão Logística	$7{,}77 \pm 0{,}16$	$5{,}97 \pm 0{,}19$	$6{,}87 \pm 0{,}10$	$7{,}65 \pm 0{,}19$

As figuras 18 e 19 apresentam de forma gráfica as informações da tabela 5, com as médias e desvios padrão do *Average Error Rate* e *Equal Error Rate* respectivamente, para os diferentes métodos de classificação ajustados. Assim, observa-se que o classificador XGBoost foi o que obteve o melhor resultado paras as duas taxas de erro, com média de 6,28% e desvio padrão de $\pm$0,661% para o AER, e média de 6,86% e desvio padrão de $\pm$0,715% para o EER. Seguindo bem próximo estão os métodos de classificação de SVM e floresta aleatória.

Figura 18 - Average Error Rate para cada método de classificação

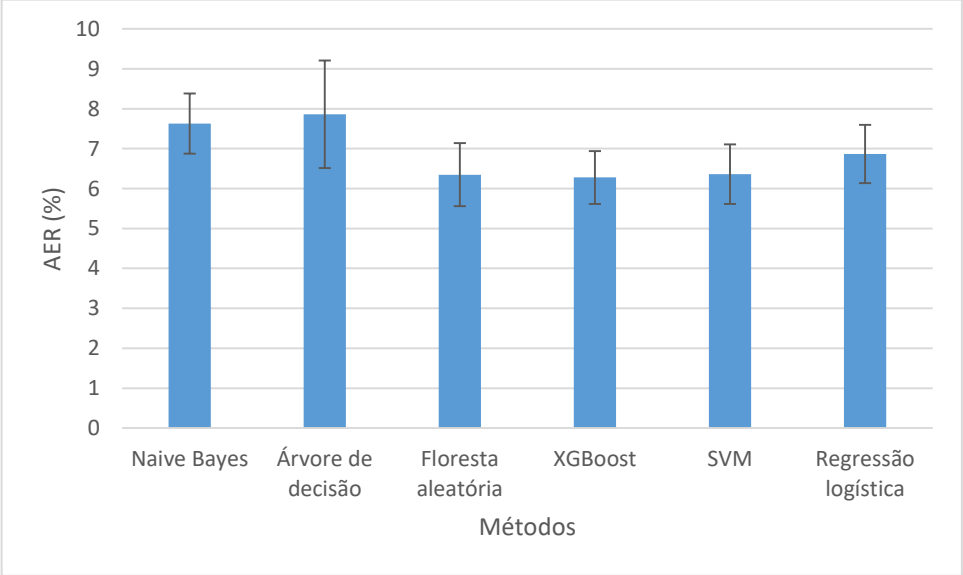
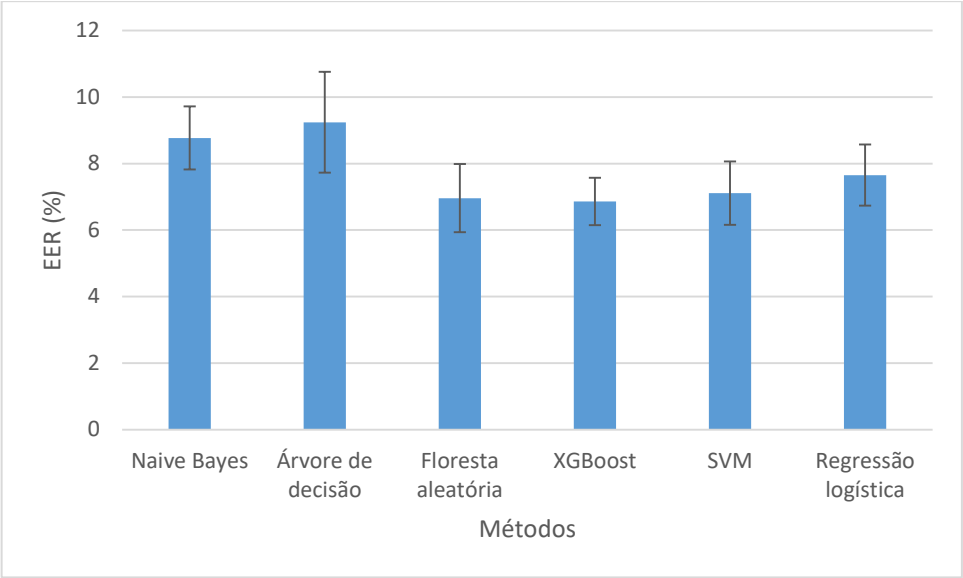
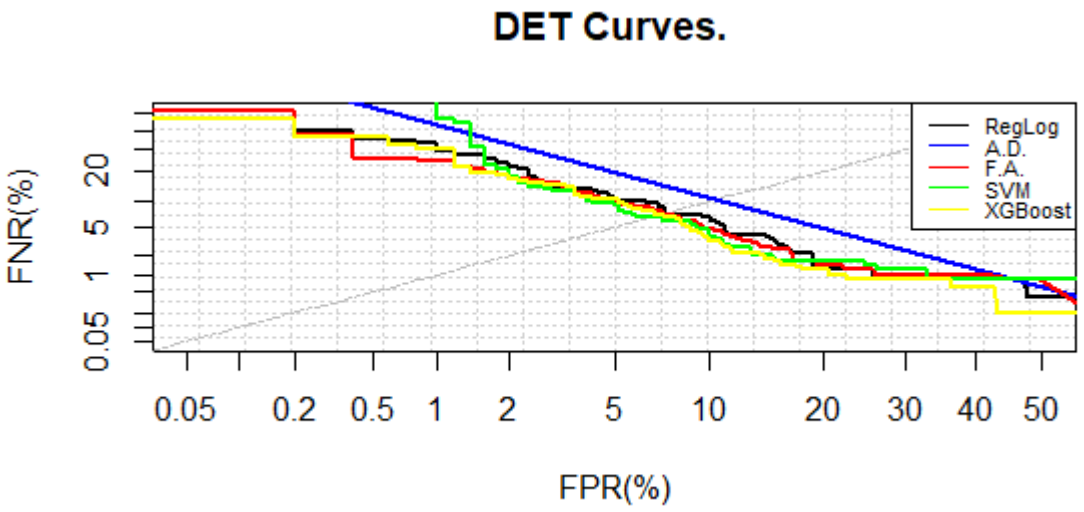


Figura 19 - Equal Error Rate para cada método de classificação



Os resultados do experimento também estão expressos nas detection error trade-off (DET) curves apresentadas na figura 20. É uma plotagem gráfica das taxas de erro para sistemas de classificação binária. Essas curvas retratam a correlação do FAR e FRR. O DET funciona variando os limiares das probabilidades respostas nos modelos e retornando os valores das taxas de falso positivo e falso negativo para esses limiares (MARTIN et al, 1997). Pela figura 20 se vê que no limiar adotado de 0,5 para a probabilidade resposta os modelos possuem resultados próximos.

Figura 20 - Detection error trade-off curves: Reglog (Regressão Logística), NBayes (Naive Bayes), A.D (Árvore de decisão), F.A. (Floresta aleatória). FPR - False Positive Rate (FAR), FNR – False Negative Rate (FRR)



Como cada grafo de assinaturas possui um limiar tomado como ótimo podem-se tirar algumas medidas deles, a média dos limiares para a coordenada x é 0,64 com desvio padrão de 0,22, já a média da coordenada y é 0,455 com desvio padrão de 0,21. A figura 21 apresenta o boxplot para os limiares das coordenadas x e y. E a tabela 6 mostra as respectivas medianas e quartis.

Figura 21 - Boxplot dos limiares das coordenadas x e y

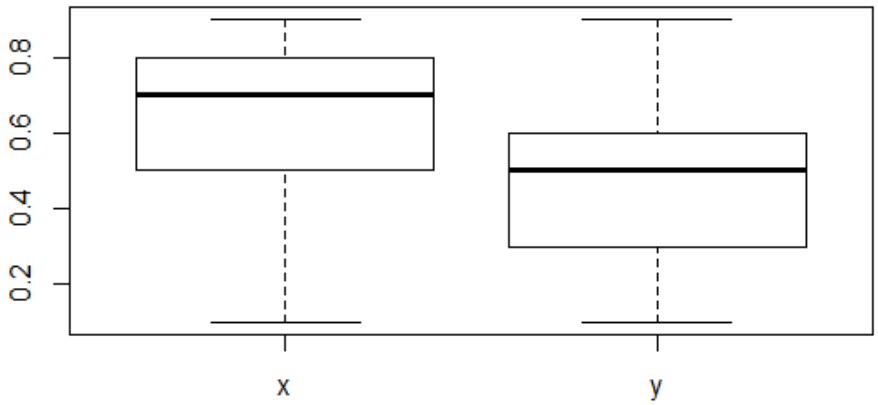


Tabela 6 – Medidas de tendência dos limiares das coordenadas x e y

	Mínimo	**1º Quartil**	**Mediana**	**3º Quartil**	**Máximo**
Coordenada x	0,10	0,50	0,70	0,80	0,90
Coordenada y	0,10	0,30	0,50	0,60	0,90

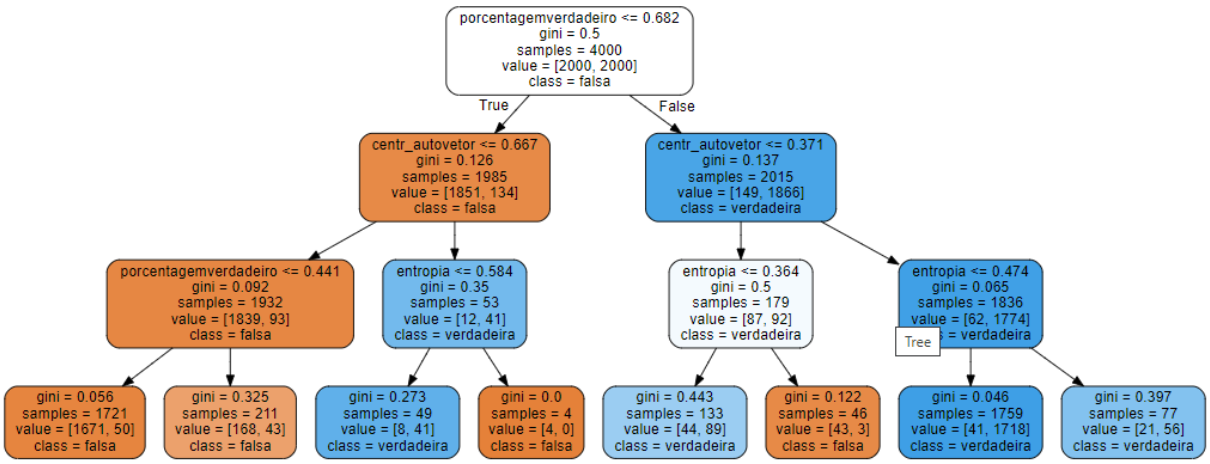
No modelo de regressão logística estabelecem-se os parâmetros para cada atributo utilizado. Como são gerados 100 modelos, a média e o desvio padrão dos parâmetros são apresentados na tabela 7.

Tabela 7 - Média e desvio padrão dos parâmetros do modelo de regressão logística

	Intercepto	Centr. Grau	Centr. Autovetor	Coef. cluster	Entropia	Porc. Verd.
Média	-2.2664	-0.2134	8.7625	-0.5422	-4.4239	3.3149
Desvio padrão	0,13212	0,0288	0,6263	0,1822	0,2758	0,2347

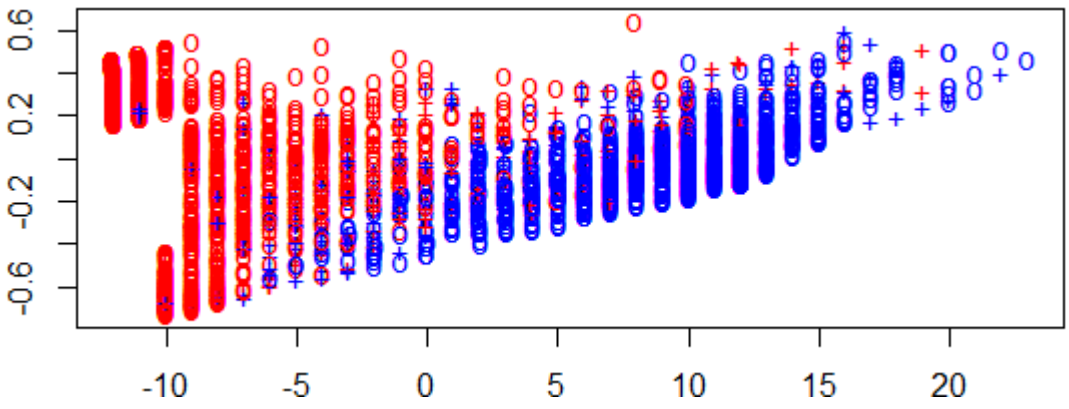
A figura 22 mostra uma das 100 árvores de decisão geradas a partir dos atributos das 4000 assinaturas analisadas, com profundidade três, ou seja, três níveis de decisão. Em cada etapa divide-se o grupo até que todos as classes sejam definidas.

Figura 22 - Árvore de decisão para o modelo de assinaturas



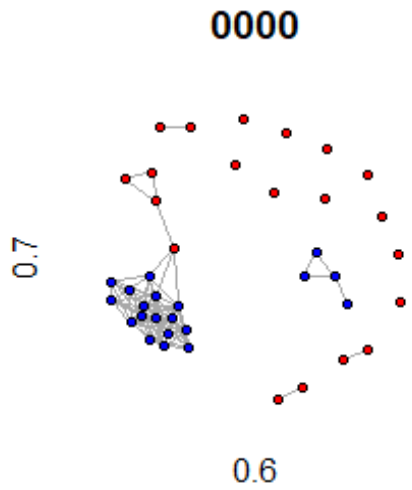
No modelo de SVM foram necessários em média 463 pontos de suporte para definir os hiperplanos. A figura 23 ilustra um exemplo da separação de classes encontrada pelo modelo. Como o modelo trata de um problema multidimensional, para uma visualização gráfica é necessário reduzir para 2 dimensões, reescalando então os atributos com o uso de *principal coordinates analysis*.

Figura 23 - Separação das classes pelo modelo de SVM



Os pontos em vermelho representam as assinaturas falsas e em azul as assinaturas verdadeiras, o símbolo de “+” representa os pontos dos vetores de suporte.

Figura 24 - Exemplo de rede formada por um conjunto de assinaturas

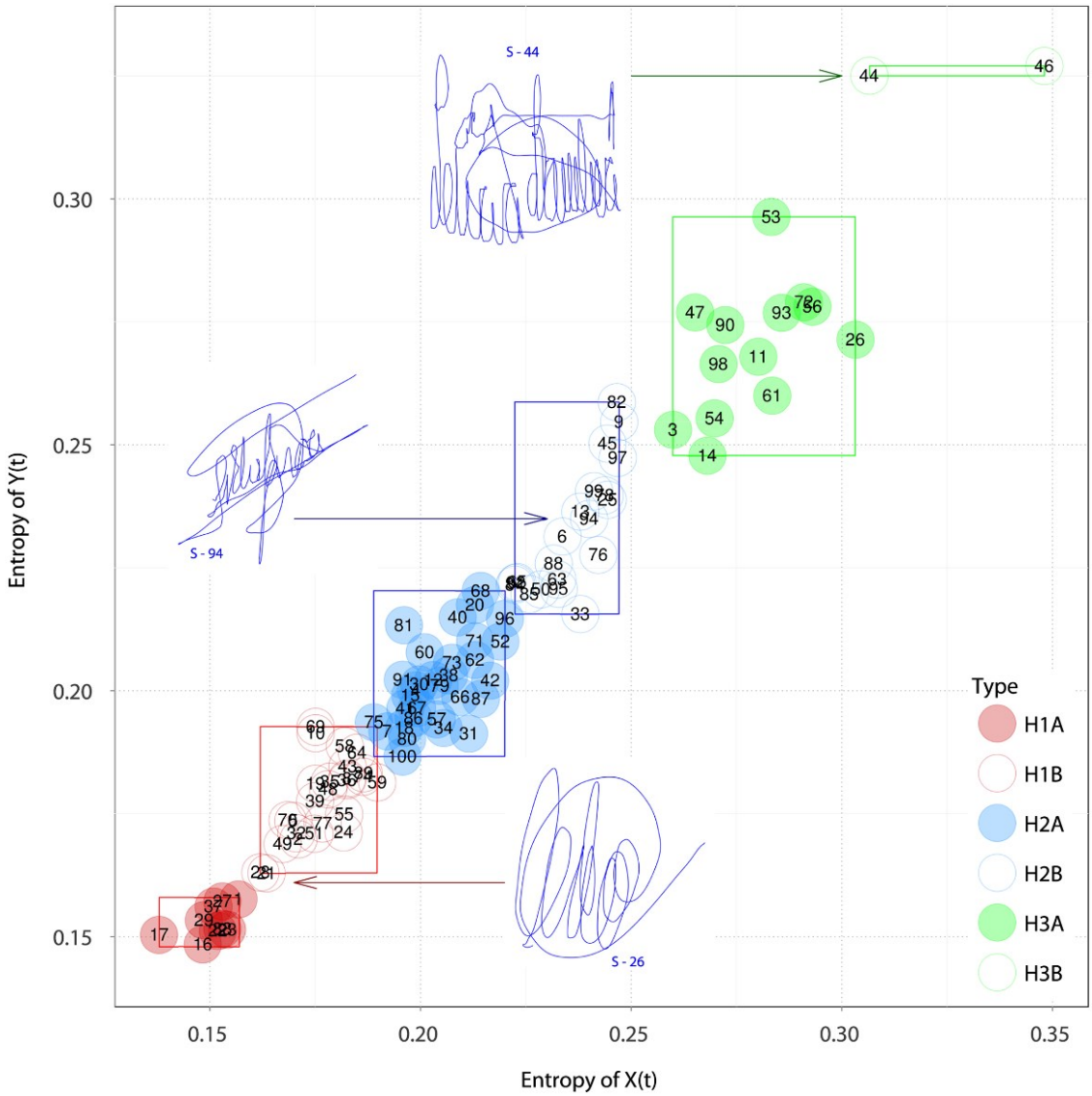


Analisando a figura 24, é possível perceber que os vértices das assinaturas falsas, em sua maioria, possuem pouca ou nenhuma conexão. Por esse motivo, o número médio de componentes das redes é de 17,35. Por outro lado, as assinaturas verdadeiras são bem conectadas, com um diâmetro médio de 4,89 por rede. O número médio de graus dentro das redes é 7,65, porém se considerarmos apenas os vértices que representam as assinaturas verdadeiras, esse número aumenta para 11,93. Outro dado interessante é a densidade da rede, ao analisar a rede completa tem-se uma densidade média de 0,15, excluindo as assinaturas falsas, a densidade sobe para 0,49.

Levando em consideração a proximidade média das redes, tem-se um valor médio de 0,0012, lembrando que a proximidade é o inverso da distância, assim para vértices desconexos é assumida uma distância infinita. Quando é usado apenas as assinaturas verdadeiras, ou seja, desconsiderando a grande maioria dos vértices desconexos, a proximidade média aumenta para 0,018. Um comportamento similar é observado para a centralidade de autovetor média, como os vértices falsos em sua maioria não possuem vizinhos, sua centralidade de autovetor é 0. Sendo assim analisando a rede por completo, tem-se uma centralidade de autovetor médio de 0,34, entretanto utilizando somente os vértices verdadeiros, a centralidade de autovetor médio passar a ser de 0,63.

Para uma análise um pouco mais detalhada quanto à forma das assinaturas, foi utilizado um método de classificação das assinaturas, a partir de suas entropias, elaborada por Rosso et al. (2016). Para a mesma base de dados aplicada neste estudo, o método classifica as assinaturas em 3 grupos, sendo cada grupo subdividido em 2 subgrupos, como é mostrado na figura 25.

Figura 25 - Classificação das assinaturas verdadeiras a partir da entropia



Fonte: Rosso et al. (2016)

O grupo H1 é composto pelos subgrupos H1A e H1B. O grupo H1A é formado exclusivamente por assinaturas muito simplificadas feitas por meros loops sem letras identificáveis. Embora o grupo H1B seja formado por assinaturas simplificadas, traços e letras ou curvas mais complexas aparecem e as diferenciam do grupo H1A.

O grupo H2 é subdividido nos grupos H2A e H2B. As assinaturas que formam o grupo H2A possuem traços que se assemelham a letras, mas não são perfeitamente identificáveis, e incluem traços circulantes de tamanho grande ou moderado. É como se as assinaturas desse grupo fossem enquadradas por esses loops. O grupo H2B é similar ao anterior, porém com letras mais identificáveis, nomes e sobrenomes são mais legíveis que nos grupos prévios.

O grupo H3, por sua vez, é formado pelos subgrupos H3A e H3B. O primeiro é composto por assinaturas caligráficas, onde traços verticais predominam sobre traços horizontais. O último é composto por assinaturas com letras cursivas bem definidas com separação entre nome e sobrenome.

Os mesmos testes que foram realizados para o banco de dados como um todo são replicados, desta vez, separando a base de dados nos grupos anteriormente definidos. Os resultados desses testes estão demonstrados nas tabelas 8-13.

Tabela 8 - Métricas de erro do método Naive Bayes para diferentes grupos de assinaturas

Naive Bayes				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$15{,}11 \pm 4{,}23$	$9{,}27 \pm 3{,}67$	$12{,}19 \pm 3{,}05$	$14{,}63 \pm 3{,}58$
H1B	$7{,}68 \pm 1{,}57$	$6{,}78 \pm 1{,}55$	$7{,}23 \pm 1{,}18$	$8{,}24 \pm 1{,}35$
H2A	$8{,}02 \pm 1{,}91$	$6{,}58 \pm 1{,}65$	$7{,}30 \pm 0{,}86$	$8{,}30 \pm 1{,}19$
H2B	$6{,}44 \pm 1{,}68$	$5{,}52 \pm 1{,}74$	$5{,}98 \pm 1{,}08$	$6{,}96 \pm 1{,}34$
H3A	$7{,}20 \pm 7{,}91$	$6{,}00 \pm 7{,}37$	$6{,}60 \pm 4{,}46$	$9{,}22 \pm 6{,}12$
H3B	$6{,}27 \pm 2{,}28$	$4{,}61 \pm 2{,}01$	$5{,}44 \pm 1{,}12$	$6{,}87 \pm 1{,}68$

Tabela 9 - Métricas de erro para o método Árvore de decisão para diferentes grupos de assinaturas

Árvore de decisão				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$13{,}22 \pm 3{,}28$	$10{,}83 \pm 3{,}69$	$12{,}02 \pm 2{,}31$	$14{,}11 \pm 3{,}14$
H1B	$7{,}97 \pm 2{,}03$	$5{,}98 \pm 2{,}17$	$6{,}98 \pm 1{,}25$	$8{,}28 \pm 1{,}75$
H2A	$7{,}92 \pm 2{,}12$	$5{,}98 \pm 2{,}47$	$6{,}95 \pm 1{,}11$	$8{,}25 \pm 1{,}49$
H2B	$4{,}16 \pm 2{,}09$	$7{,}72 \pm 3{,}07$	$5{,}94 \pm 1{,}42$	$7{,}63 \pm 1{,}85$
H3A	$4{,}00 \pm 7{,}07$	$1{,}60 \pm 3{,}74$	$2{,}80 \pm 3{,}55$	$4{,}91 \pm 6{,}09$
H3B	$6{,}15 \pm 2{,}10$	$7{,}22 \pm 2{,}94$	$6{,}67 \pm 1{,}27$	$8{,}54 \pm 2{,}12$

Tabela 10 - Métricas de erro para o método SVM para diferentes grupos de assinaturas

SVM				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$13{,}44 \pm 3{,}46$	$8{,}83 \pm 3{,}39$	$11{,}13 \pm 2{,}21$	$12{,}74 \pm 2{,}06$
H1B	$6{,}49 \pm 1{,}56$	$5{,}18 \pm 1{,}71$	$5{,}84 \pm 1{,}24$	$6{,}96 \pm 1{,}51$
H2A	$7{,}25 \pm 1{,}76$	$5{,}65 \pm 1{,}50$	$6{,}45 \pm 1{,}03$	$7{,}65 \pm 1{,}11$
H2B	$4{,}00 \pm 1{,}49$	$5{,}28 \pm 1{,}77$	$4{,}64 \pm 0{,}96$	$5{,}68 \pm 1{,}45$
H3A	$7{,}20 \pm 6{,}53$	$3{,}20 \pm 9{,}98$	$5{,}20 \pm 3{,}95$	$8{,}37 \pm 7{,}32$
H3B	$5{,}17 \pm 1{,}65$	$5{,}35 \pm 1{,}85$	$5{,}26 \pm 0{,}99$	$6{,}40 \pm 1{,}56$

Tabela 11 - Métricas de erro para o método Floresta Aleatória para diferentes grupos de assinaturas

Floresta Aleatória				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$12{,}05 \pm 3{,}54$	$9{,}77 \pm 3{,}13$	$10{,}91 \pm 2{,}45$	$12{,}74 \pm 2{,}69$
H1B	$6{,}24 \pm 2{,}18$	$6{,}82 \pm 2{,}56$	$6{,}52 \pm 1{,}65$	$7{,}56 \pm 2{,}11$
H2A	$6{,}71 \pm 1{,}61$	$6{,}48 \pm 1{,}90$	$6{,}59 \pm 1{,}33$	$7{,}64 \pm 1{,}62$
H2B	$4{,}12 \pm 1{,}99$	$5{,}44 \pm 1{,}85$	$4{,}78 \pm 1{,}45$	$5{,}94 \pm 1{,}52$
H3A	$4{,}00 \pm 7{,}07$	$3{,}60 \pm 9{,}96$	$3{,}80 \pm 5{,}30$	$5{,}73 \pm 6{,}57$
H3B	$4{,}00 \pm 2{,}59$	$6{,}09 \pm 2{,}35$	$5{,}04 \pm 1{,}67$	$6{,}65 \pm 2{,}21$

Tabela 12 - Métricas de erro para o método Regressão Logística para diferentes grupos de assinaturas

Regressão Logística				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$13{,}11 \pm 3{,}79$	$8{,}83 \pm 2{,}98$	$10{,}97 \pm 2{,}03$	$13{,}04 \pm 1{,}99$
H1B	$6{,}56 \pm 1{,}36$	$5{,}82 \pm 1{,}76$	$6{,}19 \pm 1{,}22$	$7{,}16 \pm 1{,}54$
H2A	$7{,}02 \pm 1{,}61$	$5{,}88 \pm 1{,}46$	$6{,}54 \pm 0{,}84$	$7{,}61 \pm 1{,}78$
H2B	$5{,}00 \pm 1{,}35$	$4{,}76 \pm 1{,}55$	$4{,}88 \pm 0{,}88$	$5{,}84 \pm 1{,}25$
H3A	$5{,}20 \pm 6{,}64$	$4{,}40 \pm 4{,}82$	$4{,}80 \pm 3{,}56$	$7{,}10 \pm 5{,}28$
H3B	$4{,}49 \pm 1{,}91$	$6{,}76 \pm 1{,}76$	$5{,}63 \pm 1{,}16$	$7{,}25 \pm 2{,}03$

Tabela 13 - Métricas de erro para o método XGBoost para diferentes grupos de assinaturas

XGBoost				
	FAR (%)	**FRR (%)**	**AER (%)**	**EER (%)**
H1A	$10{,}04 \pm 5{,}17$	$9{,}95 \pm 4{,}16$	$10{,}00 \pm 2{,}95$	$12{,}26 \pm 3{,}08$
H1B	$6{,}52 \pm 2{,}17$	$6{,}59 \pm 2{,}01$	$6{,}56 \pm 0{,}93$	$7{,}69 \pm 1{,}44$
H2A	$6{,}78 \pm 2{,}14$	$7{,}17 \pm 1{,}97$	$6{,}98 \pm 1{,}08$	$8{,}00 \pm 1{,}64$
H2B	$4{,}76 \pm 2{,}20$	$5{,}08 \pm 2{,}61$	$4{,}92 \pm 1{,}44$	$6{,}22 \pm 1{,}77$
H3A	$2{,}00 \pm 5{,}00$	$1{,}60 \pm 3{,}74$	$1{,}80 \pm 3{,}18$	$2{,}88 \pm 4{,}93$
H3B	$5{,}29 \pm 3{,}99$	$5{,}91 \pm 3{,}57$	$5{,}60 \pm 2{,}61$	$7{,}29 \pm 3{,}48$

Pelas tabelas 8-13 percebe-se que aquelas assinaturas que possuem letras mais bem definidas levam a menos erros de verificação, já as assinaturas formadas predominantemente por simples traços e curvas levam a maiores taxas de erros. De um modo geral, assinaturas mais elaboradas possuem taxas de erros menores, o inverso ocorre para assinaturas simplificadas. Em outras palavras, quanto menor a entropia das assinaturas maior as taxas de erro.

Por exemplo, assinaturas do grupo H1A, que têm entropia em torno de 0,15, possuem um EER de 12,74% em média, já assinaturas do grupo H3B, com entropia de aproximadamente

0,25, têm um EER de 6,65% em média, segundo o modelo de floresta aleatória. Em todos os modelos de aprendizado testados o mesmo comportamento se repete. O grupo H3A apresenta a menor taxa de erro, EER de 2,88% pelo modelo de XGBoost, e a maior entropia, acima de 0,30, porém vale ressaltar que neste grupo há assinaturas de apenas dois indivíduos, sendo assim poucos elementos na amostra analisada.

As figuras 26 e 27 mostram em uma representação gráfica as taxas de erro (AER e EER) entre os diferentes grupos.

Figura 26 - Comparação do AER entre os diferentes grupos

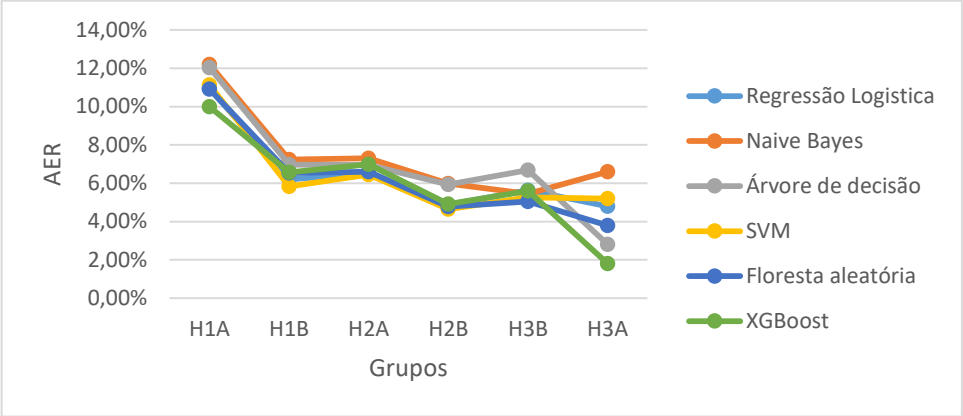
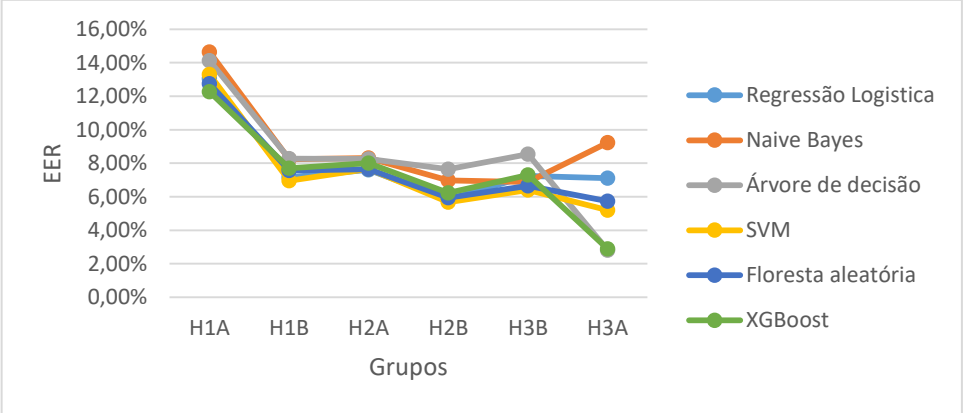


Figura 27 - Comparação do EER entre os diferentes grupos



Fica evidente que há uma tendência de queda na taxa de erro, tanto para o AER quanto para o EER, com o aumento da entropia dos grupos. Lembrando que as assinaturas com maior entropia eram aquelas mais elaboradas. Mostrando novamente que existe maior dificuldade em falsificar assinaturas complexas. Pode-se notar também que o comportamento dos resultados é bastante similar para os diferentes modelos de aprendizado utilizados.

Observando as métricas das redes de acordo com os grupos na tabela 14, apesar de serem pequenas as diferenças, os grupos classificados como HA (simplificadas), em geral, possuem maior centralidade de grau e autovetor. Também possuem densidade e tamanho da componente principal maiores, além de menor número de componentes, o que implica em redes mais conectadas e, por sua vez, demonstra que é mais difícil diferenciar assinaturas falsas de verdadeiras. Já o inverso ocorre com as assinaturas dos grupos HB (elaboradas), que mostram menor centralidade de grau e autovetor, componente principal com menos elementos e redes com menores densidades, e também maior número de componentes, assim temos redes menos conectadas, implicando em uma maior diferenciação entre assinaturas forjadas e genuínas. A única exceção se dá para os grupos H3A e H3B, pois neste caso o primeiro, apesar de não ter letras tão bem definidas quanto o segundo, apresenta maior entropia.

Tabela 14 - Métricas de rede para diferentes grupos de assinatura

	Média (desvio padrão) das métricas das redes						
	Grau	**Autovetor**	**Densidade**	**Número de Componentes**	**Componente Principal**	**Cluster coefficient**	**Diâmetro**
Todas assinaturas	11,72 (0,31)	0,4389 (0,0064)	0,2392 (0,0063)	8,84 (0,30)	38,05 (0,41)	0,5988 (0,0067)	4,48 (0,08)
H1A	11,78 (1,05)	0,4379 (0,0232)	0,2404 (0,0214)	7,93 (1,05)	40,61 (1,19)	0,5993 (0,0203)	5,35 (0,26)
H1B	11,22 (0,44)	0,4318 (0,0094)	0,2291 (0,0089)	9,25 (0,57)	36,62 (1,07)	0,6107 (0,0095)	4,41 (0,18)
H2A	11,89 (0,55)	0,4391 (0,0111)	0,2427 (0,0113)	8,44 (0,54)	38,44 (0,94)	0,5944 (0,0098)	4,12 (0,13)
H2B	11,84 (0,46)	0,4392 (0,0085)	0,2418 (0,0095)	9,26 (0,57)	38,25 (0,75)	0,5822 (0,0130)	4,54 (0,13)
H3A	9,97 (1,34)	0,4016 (0,0264)	0,2034 (0,0273)	10,68 (1,41)	35,18 (2,72)	0,5834 (0,0251)	5,91 (1,28)
H3B	12,23 (0,61)	0,4526 (0,0115)	0,2496 (0,0125)	9,06 (0,73)	38,18 (1,44)	0,6016 (0,0149)	4,59 (0,20)
H1A + H2A + H3A	11,88 (0,38)	0,4394 (0,0073)	0,2426 (0,0077)	8,25 (0,39)	38,98 (0,84)	0,5989 (0,0058)	4,45 (0,12)
H1B + H2B + H3B	11,59 (0,38)	0,4375 (0,0076)	0,2365 (0,0078)	9,29 (0,35)	37,38 (0,63)	0,5985 (0,0088)	4,52 (0,09)

As figuras 28 a 33 mostram a distribuição das métricas dos grupos através de gráficos boxplot. Vale dizer que os grupos apresentam tamanhos diferentes, por exemplo o grupo H3A é formado por apenas 2 conjuntos de assinaturas, já o grupo H1B possui 25 conjuntos de assinaturas. Analisando essas distribuições infere-se resultados semelhantes aos anteriores, ou seja, assinaturas simplistas tendem a ser mais facilmente forjadas que assinaturas complexas.

Figura 28 - Boxplot centralidade de grau

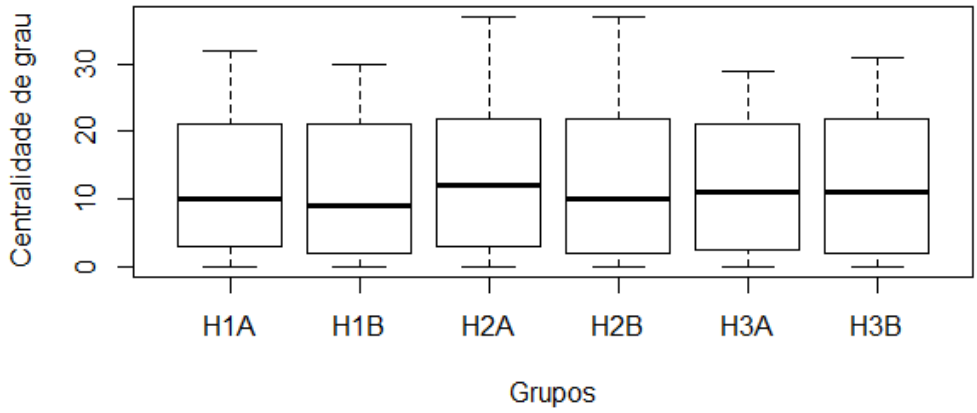
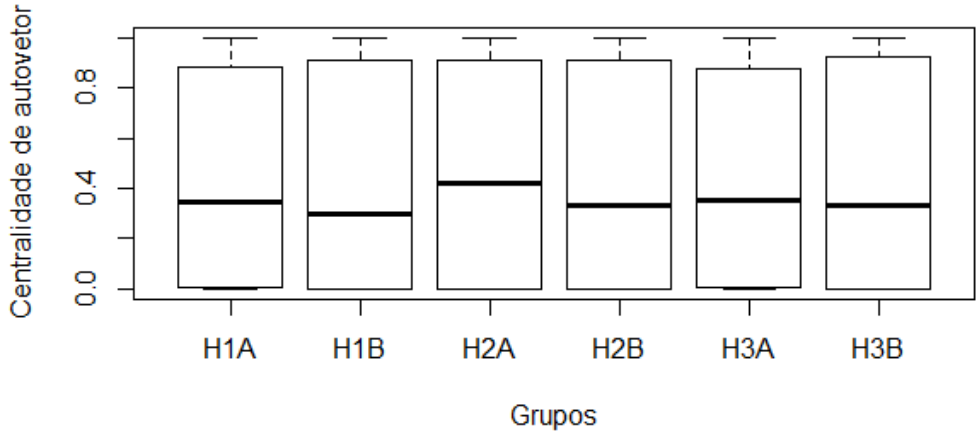
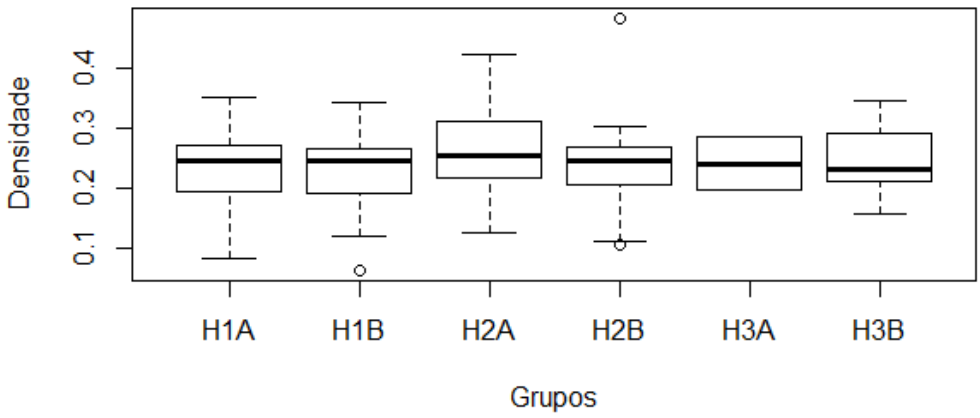


Figura 29 - Boxplot centralidade de autovetor



Observando as figuras 28 e 29, nota-se que assinaturas do grupo HB tendem a ter centralidade de grau e autovetor menores que as assinaturas do grupo HA, ou seja, estabelecem menos conexões, evidenciando a diferença entre assinaturas falsas e verdadeiras.

Figura 30 - Boxplot densidade das redes



Menos conexões do grupo HB implica em menor densidade das redes pertencentes a esse grupo, em contraponto com as redes do grupo HA que tendem a ser mais densas.

Figura 31 - Boxplot número de componentes das redes

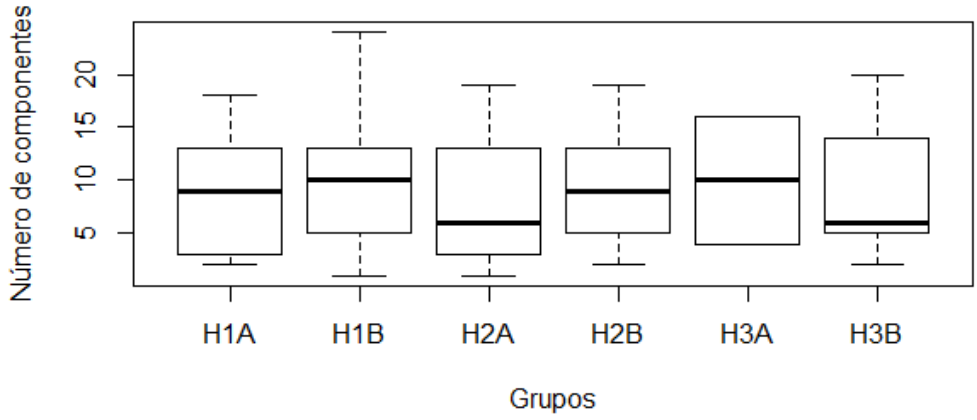
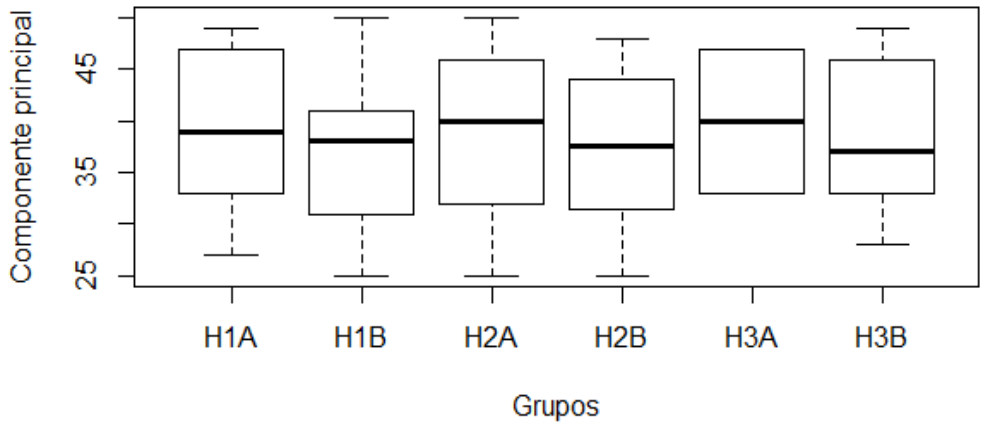
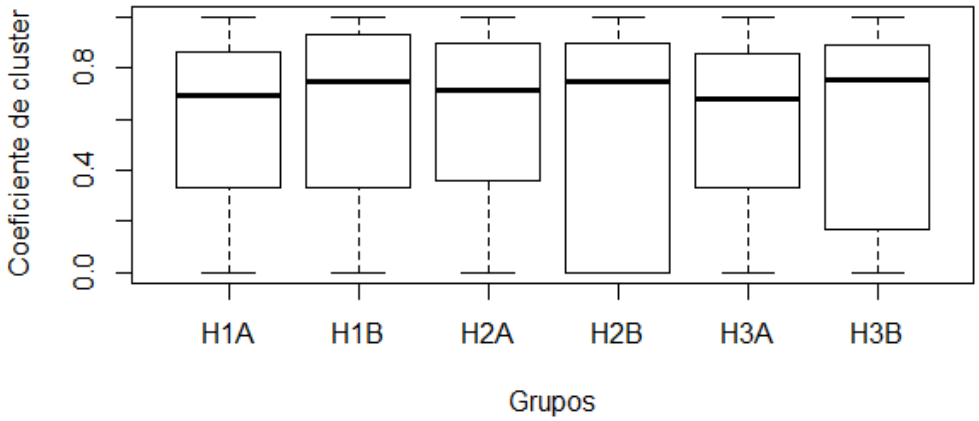


Figura 32 - Boxplot tamanho da componente principal



A figura 31 mostra que as redes das assinaturas do grupo HB têm um número maior de componentes do que as redes de assinaturas do grupo HA, consequentemente na figura 32 se vê que a componente principal das redes de grupo HB são menores que as componentes principais de HA.

Figura 33 - Boxplot coeficiente de clusterização



A distribuição dos coeficientes de clusterização por grupo está representada na figura 33. Nela, percebe-se que o grupo HB possui coeficientes mais elevados que os do grupo HA. Isso ocorre porque, em geral, as componentes do grupo HB são mais conexas que as componentes do grupo HA, onde assinaturas falsas frequentemente se encontram nas mesmas componentes que assinaturas verdadeiras, porém com poucas conexões entre si, o que diminui o coeficiente de clusterização.

Uma outra análise interessante é que, ao se retirar as métricas de rede dos modelos de aprendizado de máquina, os erros aumentam consideravelmente, como é apresentado na tabela 15, o que evidencia a importância do uso das métricas como variáveis explicativas do modelo.

Tabela 15 - Métricas de erro para os modelos sem as medidas de centralidade

Método	FAR (%)	FRR (%)	AER (%)	EER (%)
NaiveBayes	$45{,}95 \pm 1{,}77$	$10{,}55 \pm 0{,}35$	$28{,}25 \pm 0{,}71$	$33{,}48 \pm 1{,}05$
Árvore de Decisão	$39{,}20 \pm 4{,}24$	$15{,}05 \pm 2{,}47$	$27{,}12 \pm 0{,}88$	$31{,}03 \pm 1{,}11$
Floresta aleatória	$36{,}93 \pm 1{,}56$	$32{,}20 \pm 0{,}32$	$34{,}56 \pm 0{,}64$	$36{,}67 \pm 0{,}98$
XGBoost	$39{,}99 \pm 1{,}52$	$14{,}80 \pm 0{,}32$	$27{,}40 \pm 0{,}62$	$31{,}95 \pm 0{,}86$
SVM	$37{,}35 \pm 1{,}48$	$16{,}30 \pm 0{,}28$	$26{,}82 \pm 0{,}60$	$30{,}49 \pm 0{,}83$
Regressão Logística	$36{,}25 \pm 1{,}48$	$17{,}20 \pm 0{,}56$	$26{,}72 \pm 0{,}46$	$37{,}25 \pm 0{,}81$

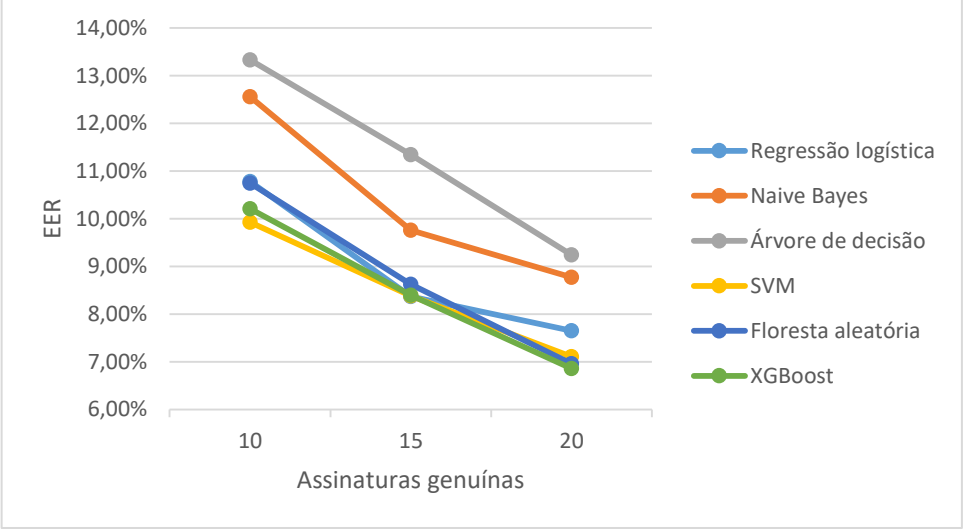
Por fim foi realizado uma análise de sensibilidade quanto ao número de assinaturas genuínas que são utilizadas para o treinamento do modelo. Cada modelo de classificador foi treinado com as mesmas assinaturas. Para todos os casos analisados foram testadas a mesma quantidade de assinaturas, sendo 5 verdadeiras e 5 falsas. A tabela 16 apresenta os resultados dos experimentos para o *Equal Error Rate*.

Tabela 16 - Análise de sensibilidade quanto ao número de assinaturas genuínas para treinamento do modelo

Nº assinaturas	Naive Bayes	Árvore de Decisão	Floresta aleatória	XGBoost	SVM	Regressão Logística
10	12,56%	13,33%	10,75%	10,21%	9,92%	10,78%
15	9,76%	11,34%	8,62%	8,40%	8,38%	8,37%
20	8,77%	9,24%	6,96%	6,86%	7,11%	7,65%

No modelo de regressão logística, por exemplo, a taxa de erro cai 2,41% quando se passa de 10 para 15 assinaturas verdadeiras de treino e 0,72% mudando de 15 para 20 assinaturas. Já no modelo de floresta aleatória a taxa de erro tem uma queda de 2,13% de 10 para 15 assinaturas e 1,66% adicionando mais cinco assinaturas genuínas. Esse mesmo comportamento de queda na taxa de erro com o aumento do número de assinaturas verdadeiras para treinamento pode ser observado em todos os modelos testados, como mostra a figura 34.

Figura 34 - Análise de sensibilidade quanto ao número de assinaturas genuínas para treinamento do modelo



Uma comparação do método proposto com resultados de verificação de assinaturas reportados na literatura é dada na tabela 17. É válido lembrar que o banco de dados utilizado é gratuito e público, então a comparação com resultados de outros autores é válida.

Tabela 17 - Comparação com outras abordagens propostas da literatura

Abordagem proposta	AER(%)	EER(%)
Este trabalho (2021)	6,28	6,86
Teymourzadeh et al. (2013)	---	8,43
Diaz et al. (2018)	---	3,76
Doroz et al. (2018)	3,19	---
Maiorana et al. (2010)	---	10,29
Lumini et al. (2009)	4,50	---

A partir da tabela 17 vê-se que a performance do modelo aqui proposto alcançou resultados semelhantes ao do estado da arte, apresentando melhoria com relação a uns modelos, porém inferior a outros.

O modelo deste trabalho permite o uso de vários métodos de classificação, contudo, os resultados experimentais mostraram que de todos os métodos testados, os melhores resultados foram alcançados com a XGBoost.

5 CONCLUSÕES

Este estudo apresentou uma nova estratégia para a verificação de assinaturas, na qual é empregada uma análise de redes complexas para o processo de verificação. O conjunto de dados de cada indivíduo é transformado em uma rede complexa e dela são extraídas métricas que são utilizadas como atributos para o método de classificação de aprendizado de máquina. A acurácia do modelo foi verificada com experimentos práticos utilizando um banco de dados reais.

A relação temporal local (entre vizinhos de cada série) usando as coordenadas de cada assinatura podem ser facilmente identificadas. Enquanto que as relações globais relevantes e de longo alcance são estabelecidas ao transformar o conjunto de séries temporais em uma grande rede complexa de interação, permitindo que os elementos sejam analisados de maneira mais ampla.

Os tipos de assinaturas se diferenciam bastante com relação às métricas estabelecidas nas redes, em geral assinaturas falsas são esparsas ou até mesmo isoladas, com pouquíssima conexão entre si. Assim a centralidade de grau, autovetor e proximidade são pequenas, além do número de componentes ser elevado. O oposto ocorre com as assinaturas verdadeiras, que são bastante conectadas levando à uma densidade elevada, assim como grandes centralidades de grau, autovetor e proximidade, e na maioria das vezes apresentam apenas uma componente.

Comparando os métodos de aprendizagem utilizados no estudo, tem-se que o XGBoost apresenta melhores resultados em todas as métricas de erro estabelecidas, assim, levando vantagem em relação as demais. O modelo de floresta aleatória vem logo em seguida reportando um AER 1,11% maior e um EER apenas 1,45% mais elevado que o XGBoost. Por outro lado, o modelo de árvore de decisão obteve o pior desempenho com o AER 25,16% acima da mesma métrica do XGBoost e o EER 34,69% maior.

Também foi possível perceber que separando os conjuntos de dados de acordo com a entropia das assinaturas, aquelas que eram classificadas como assinaturas simples (baixa entropia) de um modo geral apresentavam piores resultados, indicando uma maior facilidade para forja-las. Por outro lado, assinaturas mais complexas (entropia elevada) e com letras mais legíveis tinham melhores resultados, mostrando a dificuldade em falsificação destas.

Com os resultados comparados com outros modelos da literatura, tem-se que o uso de métricas de redes complexas como atributos em métodos de classificação para verificação de assinaturas são bastante promissores, gerando resultados competitivos com outros métodos propostos.

5.1 TRABALHOS FUTUROS

Como trabalho futuro busca-se implementar outros classificadores no processo de verificação, Redes neurais de convolução. Também a validação do modelo proposto utilizando outros bancos de dados gratuitos, como o SVC (YEUNG et al., 2004), para analisar sua performance com ainda mais modelos do estado da arte. Outro direcionamento pode ser o uso da entropia e outros quantificadores de informação das séries temporais como atributos dos nós nas redes e que possam ser incorporados aos cálculos das métricas de centralidade conforme proposto por Andrade e Rêgo (2018).

REFERÊNCIAS

ALPAYDIN, E.; **Introduction to machine learning**. MIT press, 2020.

ALVES, I.; OLIVEIRA, C.; BRITO, J.; Um estudo do problema de detecção de comunidades em redes, **Sistemas & Gestão**, v. 9, p. 566-574, 2014.

AMORIM, R. M. de. **Uma Análise das Redes de Colaboração entre Bolsistas de Produtividade e entre Programas de Doutorado em Estatística no Brasil**. 2014. 86f. Dissertação. (Mestrado – Pós-graduação em estatística / UFPE), Recife, 2014.

ANDRADE, R. L. ; REGO, LEANDRO C. . The use of nodes attributes in social network analysis with an application to an international trade network. **Physica A-Statistical Mechanics And Its Applications**, v. 491C, p. 249-270, 2018.

ARIF, T.; The Mathematics of Social Network Analysis: Metrics for Academic Social Networks, **International Journal of Computer Applications Technology and Research**, v. 4, n. 12, p. 889 - 893, 2015.

BELGIU, M.; DRAGU ¸T, L. Random forest in remote sensing: A review of applications and future directions. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 114, p. 24–31, 2016.

BONETT, D. G.; WRIGHT, T. A.; Sample size requirements for estimating pearson, kendall and spearman correlations, **Psychometrika**, v. 65, n. 1, p. 23–28, 2000.

BOX, G. E. P.; JENKINS, G. M. **Time Series Analysis: Forecast and Control**, New York, Prentice Hall, 1994.

BRAMER, M.; **Principles of data mining**, London, Springer, p. 96-100, 2007.

BRANDES, U. A Faster Algorithm for Betweenness Centrality. **Journal of Mathematical Sociology**, v. 25, 2, p. 163-177, 2001.

BREIMAN, L. Random forests. **Machine learning**, v. 45, n. 1, p. 5–32, 2001.

BRISSAUD, J.-B. The meanings of entropy. **Entropy, Molecular Diversity Preservation International**, v. 7, n. 1, p. 68–96, 2005.

CAMPBELL, C.; An introduction to kernel methods. In R. **Radial Basis Function Networks: Design and Applications**, p. 155–192, Springer Verlag, Berlin, 2000.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. ACM. **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**, p. 785–794, 2016.

COETZER, J.; HERBST, B.; DU PREEZ, J.; Off-line signature verification using the discrete radon transform and hidden markov model, **Journal on Applied Signal Processing**, v. 4, p. 559-571, 2004.

COHEN, J.; **Statistical power analysis for the behavioral sciences**. Hillsdale, NJ, Erlbaum, 1988.

CSARDI, G.; NEPUSZ, T.; The igraph software package for complex network research. **InterJournal Complex Systems**, p. 1695, 2006.

DIAZ, M.; FISCHER, A.; FERRER, M. A.; PLAMONDON, R.. Dynamic Signature Verification System Based on One Real Signature. **IEEE Transactions on Cybernetics**, v. 48, n. 1, p. 228-239, 2018.

DOROZ, R; KUDLACIK, P; PORWIK, P. Online signature verification modeled by stability oriented reference signatures. **Information Sciences**, v. 460, p. 151–171, 2018

EL-HENAWY, I. M.; RASHAD, M. Z.; NOMIR, O.; AHMED, K.; Online signature verification: state of the art. **International Journal of Computers and Technology**, v. 4, n. 1, p. 664–678. 2013.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma Introdução Sucinta à Teoria dos Grafos. 2011. Disponível em http://www.ime.usp.br/~pf/teoriadosgrafos/. Acesso em: 18 jun. 2020.

FIGUEIREDO, D.R.; Introdução a redes complexas, 303-358, 2012. Disponível em https://www.cos.ufrj.br/~daniel/JAI-RC/JAI-RC.pdf. Acesso em: 22 jun. 2020.

FRANCO, N. B. **Cálculo Numérico**. 1. ed. São Paulo: Prentice Hall, 2006.

FREEMAN, L. C. Centrality in Social Networks Conceptual Clarification. **Social Networks**, v. 1, n. 3, p. 215-239, 1979.

FREITAS, L.; **Medidas de centralidade em grafos**. 2010. 103f. Dissertação. (Mestrado em Engenharia de Produção), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

FURTADO, A. L. **Teoria dos Grafos: Algoritmos**. Livros Técnicos e Científicos. Rio de Janeiro, 1973.

GAMA, J. A. Functional trees. **Machine Learning**, v. 55, p. 219–250, 2004

GAO, Z. K.; JIN, N. D.; A directed weighted complex network for characterizing chaotic dynamics from time series. **Nonlinear Analysis: Real World Applications**, v. 13, n. 2, p. 947–952, 2012.

GROSSI, A. A. D. **Comparação e avaliação de técnicas de aprendizado de máquina para indicação de biópsia para o câncer de próstata**. 2013. TCC. (Bacharelado em Ciência da Computação), Universidade Estadual de Londrina, Londrina, 2010.

GROVER, P. Gradient boosting from scratch. **Retrieved from Medium**, 2017.

HAFEMANN, G.; SABOURIN, R.; OLIVEIRA, S.; Offline handwritten signature verification – Literature review, In **Proceedings of the International Conference on Image Processing Theory, Tools and Applications (IPTA)**, p. 1-8, 2017.

HAIR, J.F.; BLACK, W.C.; BABIN, B.J.; ANDERSON, R.E. E; TATHAM, R.L.; **Análise Multivariada de Dados**, 6ª ed., Bookman, 2009.

HAMILTON, J. D.; **Time Series Analisis**, New York, Princeton University Press, 1994.

HE, L.; TAN, H.; HUANG, Z.C.; Online handwritten signature verification based on association of curvature and torsion feature with hausdorff distance. **Multimedia Tools and Applications**, v. 78, n. 14, p. 19253– 19278, 2019.

HILTON, O.; Signatures, review and a new view. **Journal of Forensic Sciences**, v. 37, n. 1, p. 125–129, 1992.

HOSMER, D.W.; LEMESHOW, S.; **Applied logistc regression**, New York: Wiley, 2000.

LORENA, A.; CARVALHO, A.; Uma Introdução às Support Vector Machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LUMINI, A.; NANNI, L.; Ensemble of on-line signature matchers based on OverComplete feature generation, **Expert Systems with Applications**, v. 36, n. 3, p. 5291-5296, 2009.

MAIORANA, E.; CAMPISI, P.; FIERREZ, J.; ORTEGA-GARCIA, J.; NERI, A.; Cancelable templates for sequence-based biometrics with application to online signature recognition, **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, v. 40, n. 3, p. 525–538, 2010.

MARTIN, A.; DODDINGTON, A., G.; KAMM, T.; ORDOWSKI, M.; PRZYBOCKI, M. The DET Curve in Assessment of Detection Task Performance, In **Proceedings of the European Conference on Speech Communication and Technology 97**, v. 4, p. 1895-1898, 1997.

MARTINEZ-DIAZ, M.; FIERREZ, J.; O-GARCIA, J.; Universal Background Models for Dynamic Signature Verification, In: **2007 IEEE Conference on Biometrics: Theory, Applications and Systems, BTAS. Proceedings**, p. 1-6, 2007.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A.; **Foundations of machine learning.** MIT press, 2018.

MOORE, D. S.; **The Basic Practice of Statistics**. New York, Freeman, 2007.

MÜLLER, K. R.; MIKA, S.; RÄTSCH, G.; TSUDA, K.; SCHÖLKOPF, B.; An introduction to kernel-based learning algorithms. **IEEE Transactions on Neural Networks**, v. 12, n. 2, p. 181 - 201, 2001.

NARASHIMA, N.; SUSHEELA, V.; **Pattern Recognition: An Algorithmic Approach**, Springer, 2011.

NEWMAN, M.; GIRVAN, M.; Finding and evaluating community structure in networks, **Physical Review E**, v. 69, n. 2, p 026113, 2004.

OKAWA, M.; Online signature verification using single-template matching with time-series averaging and gradient boosting, **Pattern Recognition**, v. 102, p. 107227, 2020.

OLIVEIRA, A. R. **Comparação de algoritmos de aprendizagem de máquina para construção de modelos preditivos de diabetes não diagnosticada**. 2016. Dissertação. (Mestrado em Ciência da Computação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

ORTEGA-GARCIA, J.; FIERREZ-AGUILAR, J.; SIMON, D.; GONZALEZ, J.; FAUNDEZZANUY, M.; ESPINOSA, V.; SATUE, A.; HERNAEZ, I.; IGARZA, J.-J.; VIVARACHO, C. et al. Mcyt baseline corpus: a bimodal biometric database. **IEE Proceedings-Vision, Image and Signal Processing**, IET, v. 150, n. 6, p. 395–401, 2003.

ORTEGA-GARCIA, J.; BIGUN, J.; REYNOLDS, D.; GONZALEZ-RODRIGUEZ, J.; Authentication gets personal with biometrics. **IEEE Signal Processing Magazine**, v. 21, n. 2, p. 50–62, 2004.

PLAMONDON, R.; LORETTE, G.; Automatic signature verification and writer identification - the state of the art, **Pattern Recognition**, v. 1, n. 2, p. 107–131, 1989.

R CORE TEAM; R: A language and environment for statistical computing. **R Foundation for Statistical Computing**, Viena, Austria, 2020. Disponível em: https://www.r-project.org/ .

RICHIARDI, J.; KRYSZCZUK, K.; DRYGAJLO, A.; Static models of derivative-coordinates phase spaces for multivariate time series classification: An application to signature verification, **Advances in biometrics: 3rd International conference**, Proc. ICB, p.1200-1208, 2009.

ROSSO, O. A.; MICCO, L. D.; PLASTINO, A.; LARRONDO, H. A. Info-quantifiers' map-characterization revisited. **Physica A: Statistical Mechanics and its Applications**, v. 389, n. 21, p. 4604–4612, 2010.

ROSSO, O. A.; OLIVARES, F.; ZUNINO, L.; MICCO, L. D.; AQUINO, A. L.; PLASTINO, A.; LARRONDO, H. A. Characterization of chaotic maps using the permutation bandt-pompe probability distribution. **The European Physical Journal B**, v. 86, n. 4, p. 116, 2013.

ROSSO, O.; A.; OSPINA, R.; FRERY, A. C.; Classification and verification of handwritten signatures with time causal information theory quantifiers. **PloS one**, v. 11, n. 12, p. 1-19, 2016.

RUSSELL, S.; NORVIG, P.; **Artificial Intelligence: A Modern Approach**, 4th Edition, Pearson, 2020.

SANTOS, Y.; RÊGO, L.; OSPINA, R.; Verificação de assinaturas através de análise redes complexas, 2020, **Anais do LII Simpósio Brasileiro de Pesquisa Operacional**; João Pessoa, Paraíba, Brasil, Campinas, Galoá, 2020.

SANTOS, Y.; RÊGO, L.; OSPINA, R.; Signature verification via complex network analysis, **INnovation for Systems Information and Decision: Modelos and Applications, 2nd International Meeting, Local Proceedings**, pp 261 – 264, 2020.

SHANNON, C. E. A mathematical theory of communication. **Bell system technical journal**, v. 27, n. 3, pp 379–423, 1948.

SHARIF, M.; KHAN, M. A.; FAISAL, M.; MUSSARAT, Y.; FERNANDES, S. L.; A framework for offline signature verification system: Best features selection approach, **Pattern Recognition Letters**, v.139, p. 50-59, 2018.

SIGARI, M-H.; POURSHAHABI, M. R.; POURREZA, H. R.; Offline Handwritten Signature Identification and Verification Using Multi-Resolution Gabor Wavelet, **International Journal of Biometrics and Bioinformatics**, v. 5, n. 4, p. 234-248, 2011.

SMOLA, A. J. et al. **Introduction to large margin classifiers**, Morgan Kauffman, p. 1–28, 1999.

SMOLA, A. J.; SCHÖLKOPF, B. **Learning with Kernels**. Cambridge, The MIT Press, MA, 2002.

TEYMOURZADEH, R.; WAIDHUBA, M. K.; KOK, W. C.; MOK, V. H.; Smart analytical signature verification for DSP applications, **IEEE Conference on Systems, Process & Control**, p. 301-305, 2013.

VAPNIK, V. N.; **The nature of statistical learning theory**. 2nd Edition, Springer, 2000.

VERIKAS, A.; GELZINIS, A.; BACAUSKIENE, M. Mining data with random forests: A survey and results of new tests. **Pattern recognition**, v. 44, n. 2, p. 330–349, 2011.

YEUNG, D. Y.; CHANG, H.; XIONG, Y.; GEORGE, S.; KASHI, R.; MATSUMOTO, T.; RIGOLL, G.; SVC2004: First International Signature Verification Competition, **Proceedings of the International Conference on Biometric Authentication** (ICBA), Hong Kong, p. 16-22, 2004.

WEST, D. B. **Introduction to Graph Theory**, 2nd ed., Englewood Cliffs, Prentice-Hall, 2000.

ZHANG, J.; LUO, Y.; Degree Centrality, Betweenness Centrality, and Closeness Centrality in Social Network, **Advances in Intelligent Systems Research**, v. 132, p. 300-303, 2017.