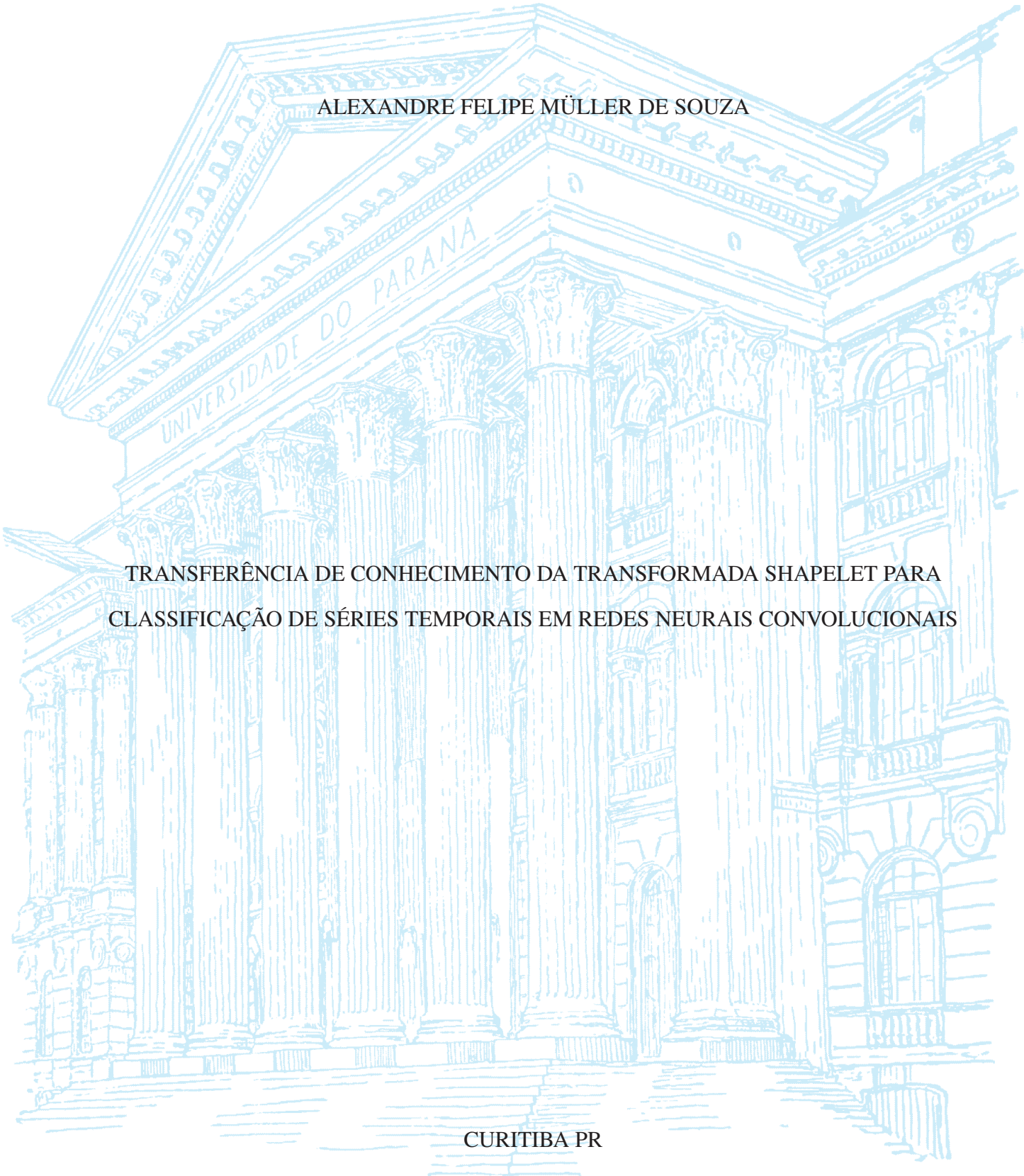
UNIVERSIDADE FEDERAL DO PARANÁ

ALEXANDRE FELIPE MÜLLER DE SOUZA



TRANSFERÊNCIA DE CONHECIMENTO DA TRANSFORMADA SHAPELET PARA CLASSIFICAÇÃO DE SÉRIES TEMPORAIS EM REDES NEURAIS CONVOLUCIONAIS

CURITIBA PR

2021

ALEXANDRE FELIPE MÜLLER DE SOUZA

TRANSFERÊNCIA DE CONHECIMENTO DA TRANSFORMADA SHAPELET PARA CLASSIFICAÇÃO DE SÉRIES TEMPORAIS EM REDES NEURAIS CONVOLUCIONAIS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

CURITIBA PR

2021

CATALOGAÇÃO NA FONTE – SIBI/UFPR

S729t Souza, Alexandre Felipe Müller de

Transferência de conhecimento da transformada shapelet para classificação de séries temporais em redes neurais convolucionais [recurso eletrônico]/ Alexandre Felipe Müller de Souza - Curitiba, 2021.

Dissertação (Mestrado) apresentada ao Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná. Área de concentração: Ciência da Computação.
Orientador: Prof. Dr. Fabiano Silva.

1. Inteligência artificial. 2. Mineração de dados. I. Silva, Fabiano. II. Título. III. Universidade Federal do Paraná..

CDD 001.535

Bibliotecária: Vilma Machado CRB9/1563

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **ALEXANDRE FELIPE MULLER DE SOUZA** intitulada: **Transferência de Conhecimento da Transformada Shapelet para Classificação de Séries Temporais em Redes Neurais Convolucionais**, sob orientação do Prof. Dr. FABIANO SILVA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 16 de Abril de 2021.

Assinatura Eletrônica
16/04/2021 13:28:19.0
FABIANO SILVA
Presidente da Banca Examinadora (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
16/04/2021 14:44:20.0
DAVID MENOTTI GOMES
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
19/04/2021 08:51:23.0
HUEI DIANA LEE
Avaliador Externo (UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ)

À Juliane Feitosa Sanches de Souza
À Eunice Müller de Souza

AGRADECIMENTOS

Em primeiro lugar, não poderia deixar de agradecer a minha esposa, Juliane, por todo apoio ao longo desse trajeto.

Ao pequeno Arthur, que mesmo ao passar por essa época conturbada, segue feliz como sempre.

A minha mãe pedagoga, Eunice, e meu pai Francisco, pois cada um em sua peculiaridade defende incondicionalmente a educação.

Ao meu orientador, Fabiano, pelo acompanhamento em todos os momentos e à Leticia, pela ajuda desde o começo dessa jornada. Assim, também ao departamento de informática UFPR e a todos os laboratórios que forneceram a infraestrutura necessária LIAMF e C3SL.

RESUMO

O crescente acúmulo na aquisição de dados, por mais diversos dispositivos, torna necessário o uso de ferramentas automatizadas para a descoberta de padrões. Dentre estas ferramentas automatizadas, a classificação de séries temporais, com aplicações em inúmeras atividades, melhora continuamente devido às bases de dados de séries públicas que permitem um marco comparativo. Também entre as diversas formas de se classificar uma série temporal, duas são foco deste trabalho: a classificação com símbolos (em especial shapelets) e as redes neurais convolucionais. A primitiva shapelet é um descritor de subsequências representativas para uma classe. A extração prévia dessa representação pode melhorar a capacidade de classificação de uma rede neural convolucional através da transferência de conhecimento. Neste trabalho, mostra-se um experimento onde a extração destas representações, antes do início do aprendizado da rede neural, muda o comportamento destes classificadores para bases específicas e, consequentemente, representa um ganho de representação em diversas bases.

Palavras-chave: Inteligência Artificial, Shapelets, séries temporais, transferência de conhecimento, mineração de dados

ABSTRACT

Nowadays the evolution of sensor data acquisition needs the use of automatic tools to discover patterns. One of those automatic tools is the *time series classification*, with widely possible applications, that improves continuously due to public datasets that becomes a benchmark in researches. Among many ways to classify time series two, in particular, are explored here: the symbolic (specially Shapelet primitive) and the neural network classification. The shapelet primitive is a descriptor of subsequences that represents a class. The previous extraction of this representation can improve the classification capacity in a convolutional neuronal network using transfer learning. Furthermore, is shown an experiment where symbolic extraction, before the network training, changes the result of a classifier for most bases and improves the classification accuracy.

Keywords: Artificial intelligence, Shapelets, temporal series, transfer learning, data mining

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

KNN	K nearest neighbor
1NN	1-nearest neighbor
DTW	Dynamic Time Warping
MLP	Multi Layer Perceptron
ED	Distância Euclidiana
SVM	Support Vector Machine
CNN	Convolutional Neural Network
ReLu	Rectified Linear Unit
LTS	Learning time-series shapelets
UCR	University California Riverside
HIVE-COTE	Algoritmo The Hierarchical Vote Collective of Transformation-based Ensembles
BOSS	Bag-of-SFA-Symbols
ELIS	Efficient Learning Interpretable Shapelets
LTS	Learning Time-series Shapelets
ECG	Eletrocardiograma
YK	Ye Keogh
FCN	Fully Convolutional Network
BoW	Bag of Words

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da tecnologia e a diminuição de dispositivos, quanto a redução dos seus custos, tem ocorrido a popularização de sistemas que geram dados em tempo real. Os dados gerados crescem continuamente ao longo do tempo, sendo necessárias atribuições de significados para se extrair as informações. Além de dados de sensores, diversos outros tipos de dados podem ser representados em estruturas que chamamos de séries temporais e tratados da mesma maneira, tais como formas de figuras que podem ser projetadas em um plano unidimensional ou espectrogramas que podem representar análises químicas (Dau et al., 2018). Dessa forma, chama-se de série temporal qualquer cadeia de valores numéricos, cuja ordem forma uma sequência (Längkvist et al., 2014). Naturalmente, o processamento dessas séries temporais possui aplicações nas mais diversas áreas (Gamboa, 2017).

Como consequência desse contexto, toda análise dos dados de séries temporais precisa cumprir uma cobertura dessa variedade de representações. Por outro lado, no contexto científico se faz necessário comparar as avaliações entre si e definir em que situações houveram ganhos e em quais não houveram. Pensando nisso, criou-se um dos principais repositórios de séries temporais usado em pesquisas dessa área (Dau et al., 2018), que tem aplicações diversas, como na medicina e na indústria. Com o crescimento da disponibilidade dos dados, esses repositórios se tornam um importante marco comparativo devido a sua diversidade. O uso dessas bases públicas facilita o trabalho dos pesquisadores, visto que não é necessário coletar e tratar os dados, mas também possibilita um comparativo, com cobertura adequada, com outros trabalhos desenvolvidos no mesmo contexto.

Entre os tipos de análise possíveis em séries temporais, o escopo deste trabalho se resume ao problema da classificação automatizada: as entradas são divididas em duas ou mais classes a serem determinadas. Em especial, o foco nessa análise é a classificação supervisionada. Isto é, dada uma série cuja classe não é conhecida a priori, é inferida a classe a qual ela pertence. Isso é realizado através de outros exemplos previamente classificados entre essas classes, não exatamente iguais, porém semelhantes. Naturalmente, o repositório das séries (Dau et al., 2018) já disponibiliza esses exemplos pré-classificados (e outros não classificados) pelo conhecedor do domínio.

Para maioria dos dados, uma das necessidades (embora secundária) na classificação automatizada é a inteligibilidade das classificações. Uma vez que tratamos de classificações automatizadas, a resposta sobre pertencer a uma classe ou outra pode ser apenas uma parte da informação necessária. Ou seja, além de saber a qual classe pertence a série, também se justificar essa classificação (Zalewski, 2015). Essa justificativa não é fornecida por todas as formas diferentes de classificadores automatizados.

Por mais que existam diversos métodos propostos para a classificação de séries temporais, iremos abordar mais a fundo dois deles: as representações simbólicas (em particular a primitiva Shapelet) e as redes neurais (em particular as convolucionais).

No método Shapelet, são extraídas as subsequências da série temporal, que são representações parciais e representativas (portanto, simbólicas) para a classe a qual pertencem. A ideia é extrair características de níveis mais altos que permitam fazer a classificação com base nessas características.

Além da classificação usando esse paradigma de representações simbólicas, há também classificadores baseados em redes neurais. Esses classificadores se utilizam de um modelo de dados chamado de rede neural, onde a memória do modelo está representada em pesos de uma

rede. Classificadores neurais são conhecidos por sua eficiência, uma vez que seu modelo já foi treinado, porém o aprendizado é baseado puramente no ajuste dos pesos da rede. Esse ajuste de pesos é dado em um algoritmo de minimização de erros, que pode dar resultados diversos dependendo de muitos fatores, como arquitetura da rede, número de camadas, filtros, entre outros. Além disso, não apresentam um modelo de aprendizado legível e interpretável pelos conhecedores do domínio.

Os classificadores baseados em redes neurais convolucionais possuem grande difusão de aplicações, inclusive cunharam o termo "Deep Learning" (LeCun et al., 2015). Devido a esse enorme sucesso, faz-se necessário analisar este tipo de classificador. Não por acaso o termo "Deep" (profundo) remete a um aprendizado com uma representação em múltiplas camadas aplicadas em bases de dados extensas.

Neste trabalho, é proposto um método de pré-treinamento de uma rede neural convolucional para gerar um classificador. Normalmente, ao iniciar um classificador utilizando-se de uma rede neural, inicia-se a rede com pesos aleatórios como na Figura 1.1. Porém, de forma diferente, a proposta é executar uma fase inicial que separa as representações simbólicas do resto da série, gerando duas novas bases. Estas bases então são usadas para treinar as partes mais relevantes em separado para gerar um modelo. Só então, gera-se um classificador final com todos os dados, porém iniciando a rede com essa informação já aprendida anteriormente, também mostrado na Figura 1.1 ao lado direito. A motivação dessa abordagem remete a dualidade entre esses classificadores simbólicos e neurais e como eles podem ser melhorados trocando informações entre eles.

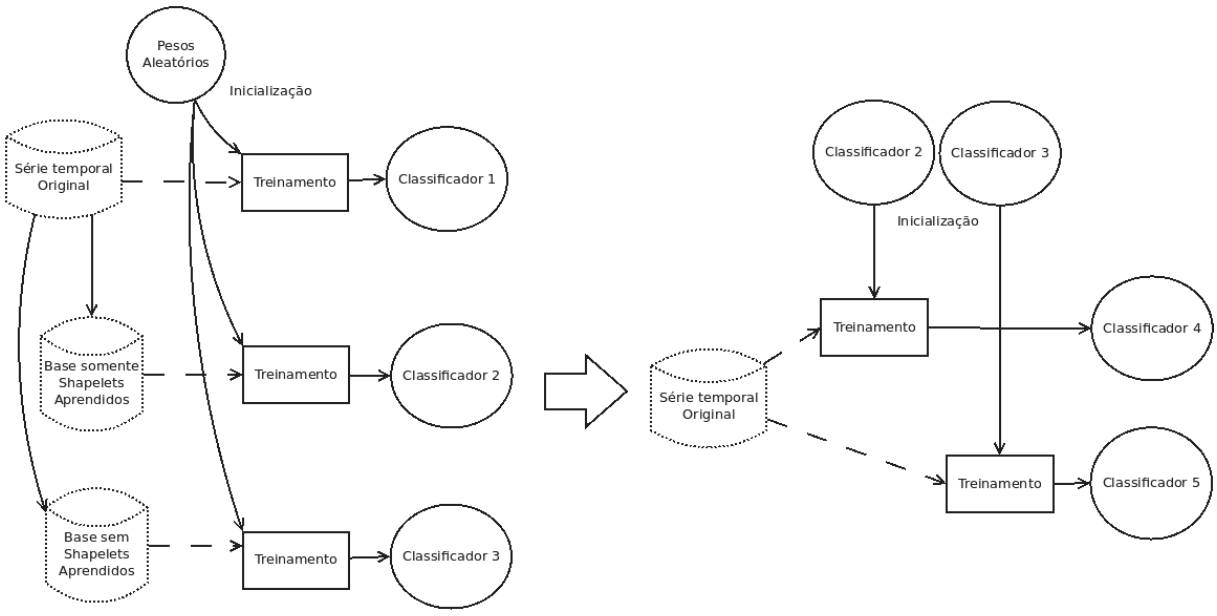


Figura 1.1: Diagrama dos experimentos

Através desse estudo foi possível observar que a pré-inicialização da rede aumenta a capacidade de acertos na classificação final, para grande parte das bases testadas. Há dois métodos de melhoria. O primeiro consiste em extrair as representações simbólicas, nesse caso shapelets, e treinar só uma parte da série original e transferir o modelo, o que permite colocar em evidência as representações simbólicas. Esse caso está sendo chamado de classificador 4 na Figura 1.1. Outro método retira os shapelets e faz treinamento somente com a parte ruidosa que é, então, transferida para a série original, chamado de classificador 5 na Figura 1.1

Sendo assim, evita-se o ajuste do modelo somente aos exemplos e se observa uma capacidade de generalização que melhora a taxa de acertos. Essa fase inicial representa justamente

a extração das informações usadas pelos conhecedores do domínio do dado a ser tratado e constitui a parte computacionalmente mais custosa. Ao final, temos os experimentos cujos resultados se destacam como sendo classificadores melhores que o marco inicial do treinamento puramente iniciado do zero.

O restante deste trabalho está organizado da seguinte maneira:

2. Fundamentação Teórica

Nesse capítulo, é apresentada toda a fundamentação teórica necessária para o desenvolvimento da ideia principal da pesquisa, com a apresentação dos conceitos de aprendizado de máquina, séries temporais, shapelets e redes neurais convolucionais.

3. Trabalhos relacionados

Nesse capítulo são apresentados os trabalhos que se relacionam diretamente ao contexto abordado nesta pesquisa. Adicionalmente, são elencados os elementos que embasaram e inspiraram a proposta do presente estudo.

4. Material e Método

Nesse capítulo é apresentado a metodologia da pesquisa e a forma como ela foi conduzida, bem como quais foram os dados testados, os experimentos conduzidos e o que eles representam.

5. Resultados e Discussão

Esse capítulo apresenta os resultados obtidos assim como as discussões sobre sua relevância para a área de pesquisa. Adicionalmente, são apresentados e analisados três estudos de caso e o que os resultados destes representam.

6. Conclusão

Por fim, o capítulo de fechamento contém as conclusões e conjecturas sobre os resultados e o trabalho como um todo.

2 FUNDAMENTAÇÃO TEÓRICA

Para que seja possível compreender esse trabalho, alguns conceitos precisam ser explorados de forma mais detalhada. Esses conhecimentos são fundamentais para a compreensão do desenvolvimento do texto. Serão abordados, a seguir, conceitos básicos de aprendizado de máquina, o que são as séries temporais e quais são os modelos dos dados alvo dessa investigação.

2.1 APRENDIZADO DE MÁQUINA

Aprendizado de máquina é uma forma de solução de problemas usando computador, sem se basear na escrita de regras para estes problemas através da programação de computadores propriamente dita, mas por meio da criação de um modelo em um processo de aprendizado (Goldberg e Holland, 1988). Desta forma, torna-se possível aprender novos conhecimentos com base em exemplos. É evidente que qualquer implementação em computador depende do algoritmo (ou escrita de regras), porém a construção do conhecimento no aprendizado de máquina se dá pelos exemplos. Esses exemplos, por sua vez, geram um modelo que representa um ou mais conhecimentos. Esse modelo pode ser usado, por exemplo, para uma classificação, regressão ou qualquer outro tipo de objetivo. Nesse texto, foca-se na classificação, ou seja, dado um exemplo cuja classe é desconhecida, é realizada a classificação de acordo com um padrão de aprendizado.

Sob o ponto de vista do aprendizado de máquina, pode-se dividir o processo em aprendizado supervisionado e aprendizado não supervisionado. No processo supervisionado, há um especialista do domínio que classifica previamente os exemplos de treinamento. Através dessa base já rotulada, gera-se um classificador que consegue determinar se um novo exemplo desconhecido pertence (ou não) a uma determinada classe. No aprendizado não supervisionado, não há classificação prévia e, portanto, o aprendizado se dá pela separação dos exemplos em uma dada representação. Isto é, no aprendizado não supervisionado, o aprendizado acontece apenas com base nos exemplos naturalmente não-rotulados.

Dentro do aprendizado supervisionado podemos construir algoritmos de aprendizado de máquina para a classificação de dados, dentre eles, as séries temporais. Isso se torna um aspecto interessante, uma vez que os algoritmos tradicionais de classificação foram desenvolvidos para tratar dados sem considerar a existência de relações de ordem ou de tempo. Comumente, trata-se em aprendizado de máquina a representação dos dados a serem trabalhados como um vetor de características. Isso é, através dos dados são extraídos representações a serem trabalhadas. Em se tratando de série temporal, há uma distinção relevante que considera que cada elemento da série não é uma característica isolada, mas sim parte de uma sequência de valores amostrados ao longo do tempo (Ehlers, 2007).

2.1.1 Paradigmas de classificação

Podemos separar os algoritmos de classificação em paradigmas de classificação, segundo Rezende (2003), os quais podem ser: simbólicos, estatísticos, baseados em exemplos, conexionista ou evolutivos.

No **paradigma baseado em exemplos**, baseia-se em procurar exemplos conhecidos mais próximos de acordo com algum critério de similaridade, podemos citar por exemplo o KNN. Esse algoritmo é simples, porém pode ser muito eficiente, inclusive para séries temporais. Seleciona-se os “K” exemplos mais próximos, baseado num “K” parametrizado e se escolhe

a que classe pertence através de uma votação pelos “K” exemplos mais próximos. O 1NN é a implementação do algoritmo do vizinho mais próximo (especialmente muito usado em séries temporais). Ou seja, por vizinho mais próximo se entende que, entre todos os exemplos da base de treinamento, é procurado aquele exemplo cuja distância é mínima. A definição de distância (que é a similaridade) em séries temporais será explorada mais adiante.

O **paradigma estatístico** se baseia em ferramentas estatísticas para classificação a qual classe pertencente. Por exemplo, no uso do SVM (máquina de vetores de suporte), procura-se um plano ou hiperplano que torna os exemplos linearmente separáveis em um espaço. O SVM procura por exemplos limites entre as classes que são chamados de vetores de suporte, então, define-se um hiperplano n-dimensional (sendo "n"o número de características) que separa as classes. Eventualmente, utiliza-se também o aumento da dimensionalidade com função kernel para tentar tornar os exemplos separáveis. Também nesse paradigma, podemos citar o método Bayesiano. Nesse método, o algoritmo Naive Bayes procura a soma das probabilidades em cada uma das características conhecidas anteriormente para tentar determinar a qual classe o exemplo pertence. A soma das probabilidades de pertencer a classe é o resultado da classificação.

O **paradigma evolutivo** se inspira no sistema de evolução hereditária existente na natureza, onde as soluções são tratadas como indivíduos capazes de formar cruzamentos e aplicar seleções. Os classificadores são avaliados por sua performance e selecionados, sendo que os resultantes cruzam entre si para gerar novos descendentes. Esse tipo de classificador se popularizou muito na década de 90 (Goldberg e Holland, 1988).

O **paradigma simbólico** trata subséries de uma série temporal como símbolos que podem classificar uma série. Diferentemente de todos os outros abordados anteriormente, baseados em ajustes estatísticos que conferem uma baixa representação em alto nível do conhecimento, esse apresenta uma resposta considerada inteligível. Como expoente desse paradigma, para séries temporais, cita-se a primitiva shapelet (Ye e Keogh, 2009). Ela foi proposta como um descritor de características morfológicas locais, que possibilita melhor compreensão dos conceitos, devido a sua maior proximidade com a percepção humana para a identificação de padrões em séries temporais (Ye e Keogh, 2009). Algoritmos baseados em aprendizado simbólico permitem a construção de estruturas baseadas em regras com a presença dos símbolos podendo ser tratada com, por exemplo, árvores de decisão. Consequentemente, a escolha baseada em presença ou não de padrões é intrínseca dos conhecedores do domínio do problema.

No **paradigma conexionista** procura se utilizar de redes neurais artificiais para aprendizado, nesse caso em particular, classificação das séries. Em geral, trata-se a fase de aprendizado como ajustes de pesos das conexões sinápticas de uma rede de neurônios artificiais e a classificação como uma propagação de entradas e saídas. As entradas, comumente, são os dados, e as saídas dos resultados esperados. Os pesos sinápticos representam, então, o modelo do aprendizado. Apesar do aprendizado exigir o ajuste dos erros ou pesos, o uso do modelo costuma ser relativamente rápido, uma vez que só precisa calcular o valor dos neurônios para se obter a saída.

2.1.2 Paradigma Simbólico X Paradigma Conexionista

A análise de séries temporais por meio de características morfológicas, como shapelets, que permitem descrever eventos e comportamentos, apresentam maior proximidade com a percepção humana (Zalewski, 2015). Por outro lado, as redes neurais também podem representar padrões morfológicos nos pesos da rede, uma vez que aplicado a imagens captam justamente as ocorrências de objetos na imagem. Essa característica intrínseca de ambos os classificadores nos leva a crer que o relacionamento entre ambos também é possível. Entretanto, em um paradigma conexionista, a classificação passa por ativações em neurônios, muitas vezes de

camadas escondidas, cujo significado da classificação não é trivial. Dessa forma, parece haver a possibilidade de usar o paradigma simbólico, o qual pode facilitar a compreensão do conhecimento por seres humanos para incrementar o modelo conexionista.

Tendo exposto todos os paradigmas de construção dos classificadores em aprendizado de máquina, pode-se olhar, especificamente, para os dois que se fazem necessário focar: paradigma simbólico, em especial, a primitiva Shapelet e conexionista, em especial, as CNNs. Porém antes, se faz necessário reduzir ainda mais o foco às séries temporais.

2.2 SÉRIES TEMPORAIS

Uma série temporal consiste em um conjunto ordenado de observações de um determinado fenômeno que ocorre de modo sequencial ao longo do tempo, não espaçados de modo necessariamente igual (Ehlers, 2007). Além do mais, trata-se não só do mesmo fenômeno, mas também do seu acontecimento e sua variação ao longo do tempo. Isso pode ser traduzido nas seguintes definições:

Definição 1 *(Série temporal) Uma série temporal* $T = \{ t_1, ..., t_i , t_j, ..., t_m\}$ *consiste em um conjunto de m valores discretos ordenados, sendo* $m \geq 2$*, tal que para qualquer i e j se* $i < j$*, então* t_i *ocorre cronologicamente antes que* t_j*.*

Entre outras palavras, a série temporal se limita a um conjunto de comprimento de, no mínimo, dois elementos e define que eles são ordenados cronologicamente. O tipo de dado temporal mais comum é a série temporal, a qual pode ser entendida como um conjunto ordenado de observações registradas cronologicamente (P. A. Morettin, 2006). As aplicações das séries temporais podem ser muito variadas: na economia, na medicina, na engenharia e em inúmeras outras áreas. O que venha, eventualmente, a se chamar de *série temporal* ao longo desse trabalho, trata-se de um conjunto de instâncias, ou de séries do mesmo fenômeno. Por exemplo: chamamos de uma série temporal, se estamos medindo a variação de temperatura ao longo do dia, por outro lado, nosso conjunto pode possuir diversos dias, cada qual é uma instância e, por definição, é uma série. Porém, no contexto da computação, chamamos de série temporal um conjunto de medições separadas pertencentes ao mesmo fenômeno. Tendo posta a definição de série temporal, também se faz necessário formalizar uma subsequência:

Definição 2 *(Subsequência) Uma subsequência* $S = \{ t_p, ..., t_{p+n-1} \}$ *consiste em um subconjunto contínuo de "n" valores de T (sendo T da Definição 1 contendo "m" elementos) com início na posição p, tal que* $2 \leq n \leq m$ *e* $1 \leq p \leq m - n + 1$*.*

A subsequência é nada mais que uma parte contínua qualquer da série toda. O conceito de subsequência também é importante, pois as representações simbólicas que serão apresentadas a seguir são construídas a partir de subsequências. Esse conceito de subsequência nos permite analisar eventos independentes, os quais são chamados de componentes da série temporal. Os componentes individuais da série temporal se classificam pela sua natureza. Esses servem para fazer algumas análises importantes sobre a série. Nesse sentido, os principais componentes de uma série temporal são: tendências, sazonalidade e resíduo (como mostra a Figura 2.2(a)) (Ferrero, 2009). A tendência corresponde a uma média central da série. É uma medida suave que indica para onde a série, como um todo, está indo. A sazonalidade, por sua vez, se caracteriza por expressar um evento cíclico que ocorre repetidas vezes num período de tempo, por exemplo, se a medida fosse temperatura, seria as estações do ano. O resíduo, particularmente, é o componente mais instável, é aquele que sobra quando se retira a tendência e a sazonalidade.

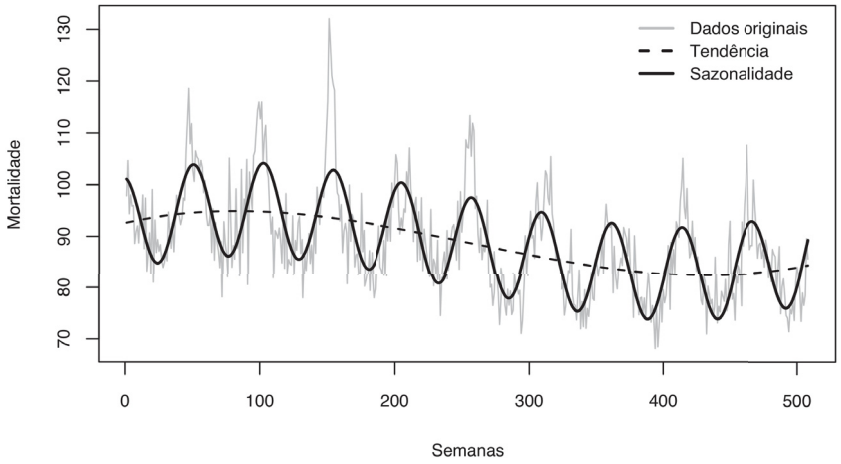


Figura 2.1: Diversos componentes das séries temporais.

Fonte: (Ferrero, 2009)

Um conceito importante para séries temporais é o de distância entre duas séries (Definição 3)

Definição 3 *(Distância entre duas séries) A distância de uma série S de uma série T é um valor d, que representa a distância de S e T, tal que:*

- $d > 0$
- *Distância (S, T) = Distância (T, S)*
- *Distância (T, T) = 0*

Na prática, a distância é uma medida de dissimilaridade, ou seja para uma subsequência ou para a série toda. O conceito de distância para a maioria dos classificadores é uma métrica que determina as tomadas de decisão. A distância, no caso da Euclidiana no espaço unidimensional das séries temporais, é a soma das diferenças dos elementos da série um a um, isto é, a soma das distâncias entre todos os pontos da série. Nesse caso, para a distância Euclidiana, mede-se a distância (no caso, a diferença) entre cada valor da série com seu correspondente na mesma posição.

A distância *Dynamic Time Warping* (DTW), mede a distância entre os pontos da série, porém tratando o alinhamento dos pontos (mesmo localmente) como mostra a Figura 2.4(a). É uma medida que consegue alinhar os elementos da série mesmo que haja distorção na dimensão temporal Senin (2008).

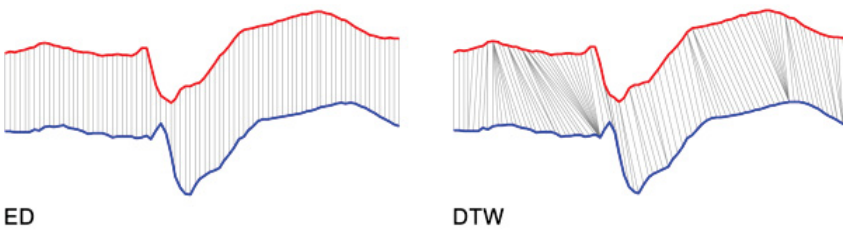


Figura 2.3: A distância Euclidiana de uma série temporal em comparação à DTW.

Fonte: (Senin, 2008)

2.3 SHAPELETS

Apresentada pela primeira vez de forma completa por Ye e Keogh (2009), a primitiva Shapelet foi, justamente, posta como descritora de características. Cada shapelet nada mais é do que uma sequência parcial dentro da série temporal e descritiva da classe a qual pertence. Ela é particularmente útil, porque detecta formas locais da série que podem se repetir. Enquanto alguns classificadores são baseados em medidas de similaridade entre as séries, as representações simbólicas consistem em extrair características (que são os símbolos). Uma classificação baseada em shapelet usa a similaridade entre uma shapelet específica e a série como uma característica discriminatória de uma classe a ser predita. Um dos resultados do uso da distância DTW é o shapelet se tornar invariante à escala (Cui et al., 2016). Um benefício da abordagem da shapelet é que as shapelet são compreensíveis e podem oferecer discernimento do domínio do problema.

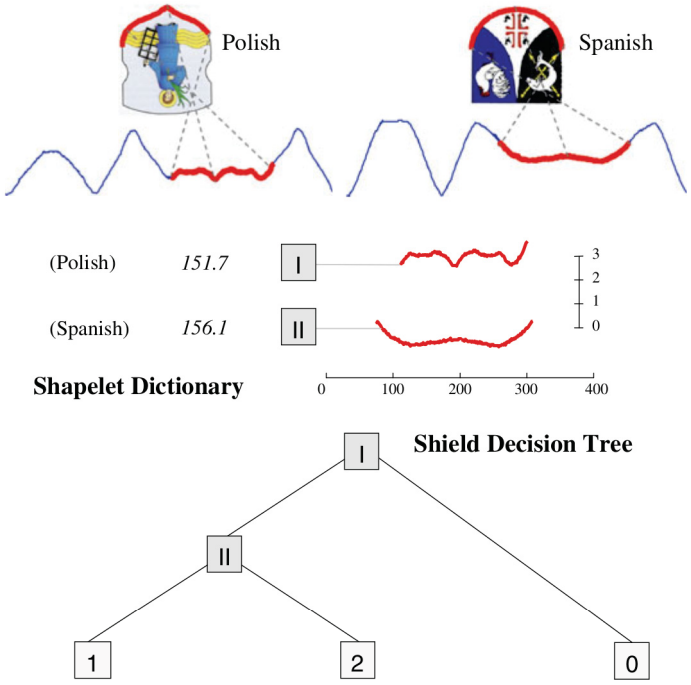


Figura 2.5: Shapelets paras classes de escudos: Polonês e Espanhol.

Fonte: (Ye e Keogh, 2011)

Como dito anteriormente, um dos primeiros trabalhos a implementar as shapelets como descritoras de características das séries foi Ye e Keogh (2009). Esse artigo traz, além de toda a formalização, diversos usos interessantes em séries cujas características locais são muito discriminantes, tais como o formato de folhas de plantas e a classificação de formas de escudos [Figura 2.6(a)]. Nesse exemplo há três classes a serem preditas: Escudo Polonês e Escudo Espanhol. O treinamento identificou pelo menos duas formas na base do escudo: I (escudo redondo) e II (escudo pontiagudo), sendo o classificador final gerado é uma árvore de decisão (cujas variáveis representam a presença ou não das shapelets). As folhas da árvore, por sua vez, representam a escolha da classe dentre as duas opções ou nenhuma delas.

Posteriormente a Ye e Keogh (2009), em Lines et al. (2012) foi proposta uma forma de extração dos k shapelets mais relevantes de uma série temporal. O processo de identificação de shapelet, de acordo com algoritmos propostos pelos dois autores (Ye e Keogh, 2009), baseia-se no conceito de janela deslizante. Nesse algoritmo são consideradas quatro etapas principais:

- Gerar as subsequências;

- Calcular as distâncias;
- Determinar o ganho de informação;
- Escolher shapelets por um critério de qualidade.

A forma mais simples de pensar na geração das subsequências é como uma força bruta através de uma janela deslizante, conforme a definição a seguir:

Definição 4 *(Janela Deslizante) "Seja uma série temporal T de tamanho m e seja n o tamanho das subsequências possíveis. Uma janela deslizante de tamanho n consiste em um conjunto formado por todas as subsequências distintas de tamanho n que podem ser extraídas de T" (Zalewski, 2015).*

Através de uma janela deslizante são geradas todas as possíveis subsequências candidatas. Essa lista, então, é atualizada com o cálculo do ganho de valor de cada subsequência. O ganho é calculado com base em uma somatória das distâncias em todas as séries. As shapelets, então tornam-se características muito discriminantes da série temporal e servem para suportar a classificação de qual classe pertencem. Por isso, são usadas como símbolos que representam características, diminuindo a dimensionalidade dos dados.

A seguir será explicado o algoritmo YK, que foi proposto como extrator das shapelets num conjunto de séries temporais. Posteriormente, é apresentado conceitos básicos sobre Redes Neurais e as Redes Neurais Convolucionais.

2.3.1 Algoritmo YK

O algoritmo YK foi proposto pela primeira vez por Ye e Keogh (2009). Ainda, segundo os autores, trabalhos anteriores a este foram discutidos com a mesma ideia, porém a viabilidade da implementação só foi possível pelo método proposto. Nele, o primeiro passo é extrair todas as subsequências possíveis através do algoritmo de geração dos candidatos, conforme mostrado na Tabela 2.1.

Algoritmo 1 Gera candidatos

Data: M: conjunto de dados das séries, min: comprimento mínimo, max: comprimento máximo
Result: C: lista de candidatos

```
i ← max;
passo ← 1;
while i ≥ min do
    for each Série T in M do
        j ← 0
        while j ≤ comprimento(T) do
            C ← C ∪ T[i : i + j]
            j ← j+1
    i ← i − passo
Return C
```

Tabela 2.1: Algoritmo geração de candidatos

Após gerar todos os possíveis candidatos, o próximo passo é realizar o cálculo dos valores dessas subsequências. Naturalmente no entanto, é necessário definir o conceito de distância de uma subsequência a uma série:

Definição 5 *(Distância de subsequência) A distância de uma subsequência S de uma série T é um valor d não negativo que representa a menor distância de S a todas as subsequências de T com comprimento igual ao comprimento de S.*

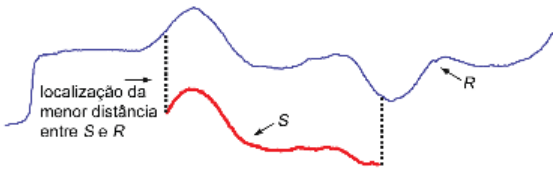


Figura 2.7: Menor distância encontrada de um shapelet.

Fonte: (Ye e Keogh, 2011)

O cálculo da distância do segmento na sequência é uma busca pelo menor valor, isto é, o lugar onde existe o melhor casamento de padrões, conforme mostra a Figura 2.8(a). Tendo definido a distância de uma subsequência, no próximo passo do algoritmo da Tabela 2.2, será calcular todas as distâncias de todas as subsequências ilustrado no algoritmo da Tabela 2.1

Algoritmo 2 Calcula distâncias

Data: M: conjunto de séries, C: lista das subsequências candidatas
Result: D: lista das distâncias
$D \leftarrow []$
for ***each*** *Série T in M* **do**
  for ***each*** *SubSequência SC in C* **do**
    D← D ∪ Distância SubSequência(T,SC)
return D

Tabela 2.2: Algoritmo do cálculo das distâncias.

Sendo calculadas todas as distâncias da série, é gerado um histograma (com as distâncias para cada valor da base) usado para obter o ganho de informação. O histograma é somente um vetor com distâncias. Porém, é preciso separar o conjunto de dados entre as classes, mas a proporção pode não ser totalmente equilibrada, por isso torna-se necessário calcular a entropia. A entropia é a medida de quão bem a base pode ser dividida entre as classes, definida a seguir:

Definição 6 *(Entropia) Dado um conjunto de séries M que possui exemplos classificados em duas classes A e B, sendo que as proporções dos objetos em M de A e B são p(A) e p(B), a entropia é: -p(A)log(p(A)) -p(B)log(p(B)) (Ye e Keogh, 2011).*

Definida a entropia, ela servirá como base para calcular o ganho de informação:

Definição 7 *(Ganho de informação) Dado uma divisão sp escolhida que divide M em dois M1 e M2, a entropia antes e depois da divisão E(M) e E'(M). Então, o ganho de informação é:*

Ganho(sp) = E(M) - E'(M)
Sendo a entropia calculada com fração de objetos em M "f"
Ganho(sP) = E(M) - (f(M1)E(M1) + f(M2)E(M2))

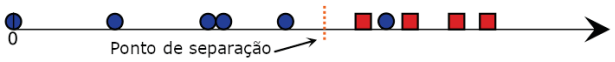


Figura 2.9: Ponto de separação da base baseado nas distâncias da subsequência

Fonte: (Ye e Keogh, 2011)

A medida do ganho de informação permite colocar um peso na separação da base, sendo separados pelo limiar (chamado comumente de *threshold*) e colocando os exemplos em cada lado da separação. Essa divisão é como um hiperplano, de forma a saber quantos verdadeiros ou falsos, positivos ou negativos, pertencem às classes, como na Figura 2.10(a). Até o presente momento, para cada candidato a Shapelet foi calculada sua distância para cada elemento da série. Então, são ordenados os elementos pelas distâncias e encontrado o Ponto de Interseção Ótimo (*Optimal Split Point*):

Definição 8 *(Ponto de interseção (Optimal Split Point - OSP)) Tem-se um conjunto de séries temporais M com duas classes A e B. Para um candidato a shapelet SC, escolhe-se o limiar dth a dividir M em M1 e M2, de forma que toda série T_1i em M1,*
Distância SubSequencia(T_1i, SC) < dth,
e para toda série T2 em M2
Distância SubSequencia(T_2i, SC) ≥ dth.

Um ponto de interseção ótimo é onde:
Ganho (SC, dOSP(D, SC)) ≥ Ganho(SC, dth)
Fonte: (Ye e Keogh, 2009)

Dessa forma, a distância de uma shapelet é usada como regra de divisão, sendo a shapelet uma subsequência que tem proximidade de uma classe, porém tem distância relativamente alta das outras classes, explicado na Definição 9. Essa divisão "OSP" é o que garante uma boa classificação utilizando árvores de decisão, uma vez que há uma variável binária para escolha de ser encontrado (ou não) aquele símbolo.

Definição 9 *(Shapelet) Dado um conjunto de séries M com duas classes A e B, shapelet(M) é a subsequência, que corresponde ao ponto de interseção, Ganho(shapelet(M), dOSP(D, shapelet(D))) ≥ Ganho(S, dOSP(D, S)).*

Ao final, o algoritmo YK aborda todos esses conceitos para extrair um único shapelet mais relevante, como no algoritmo da Tabela 2.3:

Algoritmo 3 Selecionar a melhor shapelet

```
Data: M: conjunto de dados das séries
Result: shapelet: subsequência de melhor qualidade identificada
shapelet ← [];
 melhor ← 0;
 SC ← GerarSubsequênciasCandidatas (M , min, max);
 for each Série T in M do
    DS ← CalcularDistâncias (SC,M);
     qualidade ← DeterminarQualidade (DS);
     if qualidade > melhor then
        melhor ← qualidade
         shapelet ←S
return shapelet
```

Tabela 2.3: Algoritmo que seleciona a melhor shapelet

2.4 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais são modelos de dados computacionais inspirados em redes neurais biológicas. Uma de suas características mais marcantes, é dado a entrada, a capacidade de fazer a classificação rapidamente, calculando o valor dos neurônios. Elas também possuem uma grande capacidade de se ajustar aos erros, corrigindo o modelo continuamente. A seguir, serão apresentados alguns conceitos fundamentais sobre redes neurais, a começar pelo Perceptron:

2.4.1 Perceptron

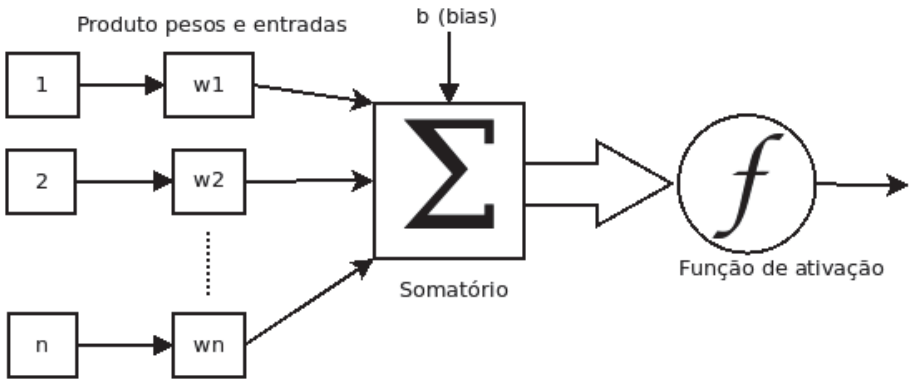


Figura 2.11: Modelo de um perceptron (neurônio)

O perceptron é uma das primeiras tentativas de reproduzir o comportamento de um neurônio biológico em uma máquina digital. Seu funcionamento é baseado em um modelo matemático com várias entradas e pelo menos uma saída (Figura 2.11). As entradas (1 a n) são multiplicadas com seus pesos sinápticos (w1 a wn), que passam por uma soma e uma função de ativação, gerando a saída. A soma resultante é, então, ajustada pelo bias (o bias nada mais é que um valor ajustável para cada neurônio que permite melhorar o desempenho da rede) e, logo em seguida, passada pela função de ativação. O cálculo destes produtos, com ajuste, permite a medição do erro dada entrada e a diferença do valor esperado na saída.

O processo de aprendizado consiste em recalcular pesos sinápticos que reduzam esse erro. Por outro lado, o objetivo principal da função de ativação é mudar o comportamento linear da saída para colocá-la dentro de um critério de escolha. Dessa forma, pode-se tornar a saída mais sensível a determinados valores ou filtrar outros. A função de ativação mais comum é chamada de *Rectified Linear Unit* (ReLu (Nair e Hinton, 2010)), e só considera a parte positiva do número de entrada. Sua saída é o próprio valor e retorna o valor 0 em caso de entrada com valor negativo. Por ser muito leve computacionalmente e muito simples, além de ter bons resultados, é um recurso muito comum nas MLPs.

Além disso, o Perceptron é um classificador linear, ou seja, aquele que classifica exemplos de um problema linearmente separável em um espaço multidimensional. Destaca-se, ainda, que os algoritmos de classificação baseados em redes neurais costumam ser chamados de "caixa preta" devido ao modelo gerado de difícil interpretação humana (Boger e Guterman, 1997).

É importante explicar o algoritmo Perceptron porque é fundamental para outras estruturas, um exemplo é quando se junta diversos Perceptrons em uma rede com saídas de uma camada ligado a entradas de outros Perceptrons, chama-se de Perceptron de múltiplas camadas (MLP).

2.4.2 Algoritmo Back-propagation

Quando se trata de um único Perceptron, o ajuste dos pesos é simples de ser definido pelo cálculo do erro naquele "neurônio". Porém, ao se tratar de uma rede neural artificial interligada com diversas camadas escondidas, há o problema da atribuição de créditos. Ou seja, o ajuste de quais pesos devem ser atualizados numa arquitetura de rede mais complexa não é trivial. A implementação apresentada de forma mais eficiente foi o algoritmo *back-propagation*. O *back-propagation* utiliza o erro que é diferença entre valor calculado e a saída (lembrando que o processo é supervisionado, portanto no aprendizado se conhece a saída) para atualizar os pesos da última camada, e por conseguinte, vai propagando essa atualização com cálculo do erro até a entrada. O back-propagation acontece em duas fases: uma fase propagando cálculo da entrada até a saída (*forward pass*) e outra fase replicando os erros para trás (*back pass*).

O processo de aprendizagem como um todo, geralmente começa com pesos escolhidos aleatoriamente ou zerados. O *forward pass* é o cálculo da propagação desses pesos até se obter um resultado de saída. Naturalmente, é de se esperar que nas primeiras iterações a previsão esteja errada ou com pouca certeza (isto é, com um valor de erro grande). No *backward pass* é aplicada a regra delta para computar o gradiente da função de perda relacionada a entrada. É chamado *backward pass* porque usa o erro de saída para atualizar os pesos para trás. Justamente na base desse passo está a regra delta, sendo esse um processo de otimização de gradiente, ou seja, da minimização da função de erro.

O cálculo do vetor gradiente da função de erro permite determinar o caminho de decréscimo a se percorrer de forma encontrar um mínimo local e como o vetor gradiente fornece a direção, há também o ajuste do tamanho do passo (taxa de aprendizado) ao se percorrer a função de erro. Um tamanho do passo muito grande pode passar direto por um mínimo local, e um muito pequeno pode tornar a busca muito demorada. Com isso, esse ajuste geralmente depende da aplicação e se dá de forma empírica.

Depois de ajustado o modelo para cada exemplo utilizando otimização do gradiente, passa-se ao próximo exemplo da base de dados. Ao fim de todos os exemplos da base, é dito que passou uma época. O número de épocas de treinamento também é ajustado empiricamente. Em geral, quando número de épocas de treinamento chega ao limite, o modelo entra em declínio dos acertos, isso porque entra em ajuste aos exemplos de treinamento, piorando o resultado.

2.5 CNN

Redes Neurais Convolucionais (CNN) têm se tornado comuns na procura por padrões, principalmente em imagens. Elas são conhecidas por serem invariantes ao espaço e ao deslocamento da imagem. Esse sucesso também é relacionado à boa performance da classificação, uma vez treinado o modelo. Uma CNN é uma versão específica de uma MLP gerada por uma operação chamada de convolução (Figura 2.13(a)).

A convolução é um processo típico do processamento de imagens, comumente utilizada como filtro. Convolução é uma janela (ou chamada "kernel") aplicada e deslocada ao longo da imagem ou da série temporal. Essa, nada mais é que um quadro centrado em um pixel (e seus vizinhos) sendo utilizado como a entrada todos os "pixels" da imagem original para gerar uma outra imagem. A ideia da convolução é extrair dos vizinhos atributos que possam ser calculados para gerar uma nova imagem. A janela é como a entrada de um filtro (conceito similar ao de imagens). O conceito de filtro coincide exatamente com o conceito de janela deslizante, quando falamos de um dado unidimensional. Geralmente as CNN trabalham com etapas em camadas: camada de convolução, camada de ativação e camada de agregação (ou *pooling*). Ao contrário de separar as implementações da extração das características (representação) e sua classificação, as CNN são muito utilizadas porque extraem os padrões que geram automaticamente uma representação da imagem, além de gerar o classificador na sua saída. Dessa forma, utiliza-se a convolução para empregar a própria imagem como entrada, agregação para reduzir a dimensionalidade e a rede neural para classificação final.

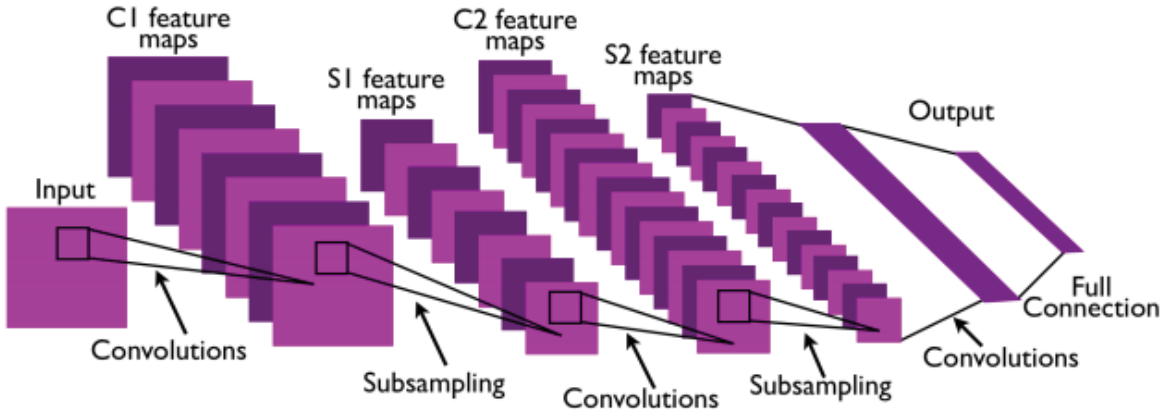


Figura 2.12: Exemplo de arquitetura de uma CNN.

Fonte: (LeCun et al., 2010)

Uma convolução é a primeira camada de uma CNN, nela um filtro é aplicado a um pequeno pedaço de uma imagem, sendo deslizado pela imagem toda e, portanto, gerando um resultado. Nesse passo podem ser utilizados vários filtros como: detecção de bordas, desfocagem, média, entre outros. Depois de geradas as imagens obtidas pelos filtros, o processo de agregação, por sua vez, reduz essa informação de forma mais compacta. O processo de uma CNN unidimensional (usado em série temporal) pode ser visto na Figura 2.15(a). O processo de agregação ou pooling visa gerar uma representação menor a cada iteração através de outra operação sobre os diversos componentes gerados no primeiro passo como média, soma ou maior valor. Esse processo de convolução e *pooling* é aplicado diversas vezes até gerar uma representação menor da imagem que é vetor de características. Esse processo permite extrair características discriminantes e de alto nível da imagem processada. De forma geral, esse processo é a extração de características propriamente dita. Assim, o objetivo principal da camada

de ativação é colocar os valores dentro de um limite e, quanto a função de ativação, a mais comum (*ReLu*) visa tornar o aprendizado mais rápido e generalista.

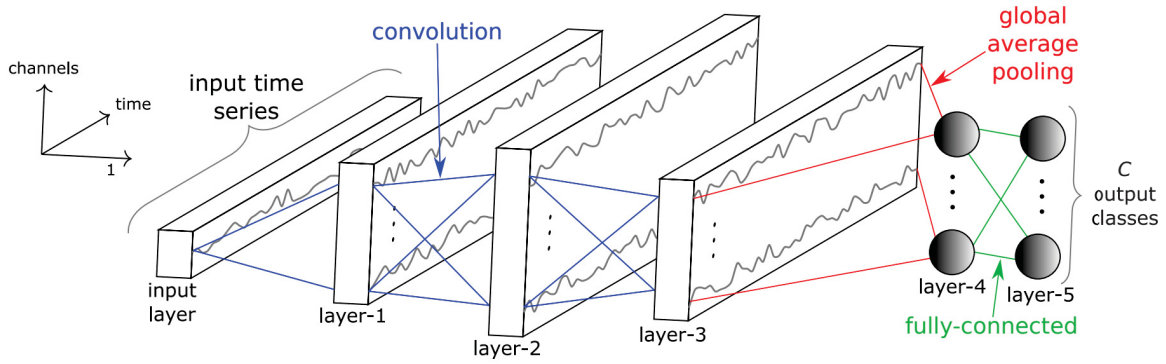


Figura 2.14: Arquitetura da CNN unidimensional.

Fonte: Fawaz et al. (2018)

A última parte é uma MLP, uma camada totalmente conectada, onde todos os neurônios de uma camada possuem conexões com todos os outros neurônios nas camadas posteriores, sendo o número entradas do tamanho da representação gerada pelas camadas anteriores e de saídas iguais aos números de classes a serem preditos. Um aspecto fundamental a ser entendido é que a CNN em imagens tem uma capacidade de aprender representações simbólicas naturalmente (LeCun et al., 2015).

2.6 TRANSFERÊNCIA DE CONHECIMENTO

A transferência de conhecimento é uma técnica que consiste em treinar uma "base fonte" com o objetivo de gerar um modelo de treinamento e transferir esse modelo a outra "base alvo" para aumentar o poder de classificação (Fawaz et al., 2018). O resultado dessa transferência é um aumento na generalização e na acurácia final da classificação em diversos casos (dependendo da base fonte e alvo). Isso se torna particularmente útil em casos em que não existem exemplos suficientes na base de treinamento e também para melhorar o resultado final dos classificadores em geral. As transferências de conhecimento podem ser: positivas (casos sem que ambas as bases possuem mesmo espaço de características), negativas (as quais as características importadas causam detrimento da informação) ou heterogêneas (quando as bases possuem espaços de características distintas) (Pan e Yang, 2009).

2.7 CONTEXTUALIZAÇÃO DA REVISÃO

Foram apresentados nas seções anteriores diferentes paradigmas de classificação e diversas características das séries temporais. Por exemplo, em Wang et al. (2013) foi verificado por meio de uma extensa avaliação empírica, que o algoritmo de classificação 1-vizinho mais próximo (1NN) em combinação com a medida de distância DTW (Senin, 2008) apresenta bom desempenho, em termos de acurácia para a maioria dos domínios de séries temporais avaliados e foi apontado como o método estado-da-arte por volta dessa época por (Mueen et al., 2011). Entretanto, a inteligibilidade é pouco presente nesses paradigmas de exemplos ou estatísticos, como no algoritmo 1NN-DTW, pois a única informação disponível se refere ao grau de similaridade entre as séries temporais consideradas semelhantes. Sendo assim, a estratégia mais comum para a construção de classificadores inteligíveis tem sido a utilização de algoritmos

de aprendizado de máquina simbólica, tais como árvores ou regras de decisão apresentadas em Zalewski (2015). A seguir, são explorados trabalhos que se dedicam a estudar séries temporais sob os mais diversos aspectos e como se relacionam sob o ponto de vista desse trabalho.

3 TRABALHOS RELACIONADOS

Visando obter as condições necessárias para se comparar trabalhos de classificação, pesquisadores utilizam frequentemente uma base de séries temporais da UCR (Dau et al. (2018)) disponível abertamente. Esse é um popular *benchmark* de séries temporais devido sua diversidade. Ademais, é graças a essa base de dados que se torna possível comparar algoritmos e trabalhos diferentes e saber onde há ganhos em acurácia na classificação e onde não há.

A classificação de séries temporais se divide em diversos paradigmas de geração dos modelos, sendo eles simbólicos ou estatísticos. Os trabalhos baseados em paradigmas simbólicos, em maioria recaem ao uso das shapelets, possuem um apelo na questão de inteligibilidade da classificação. Sobre a importância da legibilidade da classificação, e como as representações simbólicas ajudam, há um detalhamento contextualizado em Zalewski (2015), utilizando-se de uma árvore decisão algébrica. Por consequência, também como uma demonstração de mais resultados em Abel (2018).

Segundo Ye e Keogh (2009), os shapelets podem prover resultados que devem ajudar os conhecedores do domínio a entender melhor seus próprios dados. Em Li et al. (2020a) é apresentado uma interface para visualização dos shapelets extraídos de forma interativa, podendo ser usado também pelo menos outros 4 métodos distintos de extração. O primeiro método seria dos mesmos autores (Li et al., 2020b) se baseia em representações simbólicas intermediárias chamadas *symbolic aggregate approximation* (SAX). Esse conceito é bastante explorado, apresentado pela primeira vez em (Lin et al., 2007), consiste em discretizar a série em faixas que se tornam símbolos de um alfabeto e por consequência cada subsequência geraria uma palavra. No trabalho de Li et al. (2020b) as SAX words são então convertidas em shapelets através de uma estrutura de bitmaps ponderados das SAX words para determinar a qualidade dos candidatos a shapelets, transformando o problema de seleção das Shapelets em um processo de busca heurística. Dessa forma segundo os autores deixando a descoberta dos Shapelets mais eficiente.

Os principais métodos de classificação baseados em Shapelets, citados em Li et al. (2020a) são: HIVE-COTE de Lines et al. (2018), ELIS de Fang et al. (2018) e LTS de Grabocka et al. (2014). Além disso o método BOSS de Schäfer (2015) também é referenciado por Wang et al. (2016) e Fawaz et al. (2019). A seguir será explicado com detalhes cada um destes:

Um dos algoritmos de classificação de séries temporais mais referenciado é chamado de BOSS (*Bag of SFA Symbols*). Nele, é gerado um conjunto de representações simbólicas SFA (*Symbolic Fourrier Approximation*) com diferentes tamanhos de janela deslizante e criado um *ensemble* baseado em um modelo parecido com "*Bag of Words*". Esse modelo (BoW) nada mais é que um conjunto com contagem das ocorrências dos símbolos, desconsiderando outros elementos como por exemplo, a ordem que eles aparecem. Em comparação com as shapelets, essas representações são extremamente curtas e de tamanhos variados. Sua classificação se baseia em um histograma das ocorrências ao invés de em uma árvore de decisão. Os ganhos desse método são a tolerância a ruídos e invariância de escala e posição das representações simbólicas.

Um dos trabalhos que fundamentou diversos outros na área de classificação de séries temporais tem sido o uso de uma coleção de classificadores chamado COTE (Bagnall et al., 2015). Nesse tipo de abordagem há 35 classificadores (Lines et al., 2018). Cada classificador é treinado individualmente e depois os resultados são combinados atribuindo pesos aos classificadores (Lines e Bagnall, 2015). Uma variação posterior dele *Hierarchical Vote Collective of Transformation based Ensembles* (HIVE-COTE) baseia-se numa votação hierárquica entre

módulos dos classificadores de mesma natureza. Nesse método, há ao todo 5 módulos, por exemplo um módulo seria Shapelet e outro módulo por exemplo seria *Bag-of-Words*. Ao fazer essa votação hierárquica é conseguido melhores resultados. Apesar de ambos estes trabalhos exporem resultados positivos eles utilizam classificadores de diversos paradigmas como baseado em exemplos, simbólico, estatístico e conexionista. Dessa forma, tornando a classificação final pouco inteligível.

Outro trabalho relacionado a este, pois apresenta um classificador de séries temporais usando shapelets, é o de Fang et al. (2018), apelidado de *Efficient Learning Interpretable Shapelets* (ELIS), e possui um método de extração mais rápido que o tradicional e apresenta bons resultados. Esse método funciona em duas fases: descoberta do shapelet e ajuste dos shapelets. No primeiro estágio ocorre a geração das subsequências em um processo chamado PAA (*Piecewise Aggregation Aproximation*, em português Aproximação de Agregação por Partes). Esse processo é como uma discretização que reduz a resolução da série. Para cada classe são rankeadas as palavras PAAs e essa informação ajuda a ajustar os números de shapelets a serem extraídos automaticamente. É criado um modelo que tenta ajustar o shapelet a melhor forma. A ideia principal é que Shapelets verdadeiros não devem aparecer exatamente na melhor forma no conjunto de treinamento (Fang et al., 2018). Por fim, no segundo estágio é construído vários classificadores cada um com seus shapelets específicos. Então é usado um processo de regressão para obter melhor ajuste do shapelet na série e gerar um classificador.

Esse método conhecido como ELIS possui limitações em bases de dados pequenas, assim como na quantidade de parâmetros que precisam ser ajustados, (Zhang et al., 2021). Justamente para corrigir essas limitações, os autores propuseram recentemente um método ELIS++ (Zhang et al., 2021), que além de usar *data augmentation* (aumento de dados), também ajusta seus parâmetros automaticamente.

Outro trabalho muito relevante na área dos shapelets, inclusive por citações, é Grabocka et al. (2014). Nele, os autores obtêm os shapelets não através de cálculos de candidatos, mas por meio de um algoritmo de descida de gradiente (método conhecido como *Learning Time-series Shapelet* – LTS). Dessa forma, o processo se torna muito parecido com ajustes de pesos de uma rede neural.

Apesar de eficientes, todos estes classificadores não substituem completamente o uso de um classificador baseado em uma CNN. Um trabalho que conseguiu levar comparativos de CNN em séries temporais foi Fawaz et al. (2019). Nesta revisão de classificação de séries temporais os autores reproduziram as principais implementações e compararam métodos de deep learning com outro métodos. Segundo os autores HIVE/COTE é atualmente considerado o algoritmo estado da arte para classificação de séries temporais quando avaliado nos 85 conjuntos de dados do arquivo UCR. Através do estudo empírico realizado pelos autores ResNet foi que chegou mais próximo do COTE/HIVE em termos de acurácia.

Por outro lado, ao se falar de séries temporais, há um consenso de que faltam pesquisas sobre o uso de CNN nesse contexto (Cui et al., 2016) (Fawaz et al., 2018). Diante disso, temos uma importante área a ser estudada, visto que as representações simbólicas aprendidas (tais como são em imagens) oferecem uma importante inteligibilidade que pode ser usada também para as séries temporais. A implementação da rede convolucional apresentada nesta dissertação foi derivada de Wang et al. (2016), que já obteve resultados próximos do estado da arte em 2016.

Analisando o aprendizado de máquina de forma geral, o uso de CNN para classificação e segmentação em imagens é amplamente explorado atualmente, isso se dá principalmente graças às competições como ILSVRC [1], onde todos os anos, com redes cada vez mais profundas e treinadas com mais exemplos, os competidores conseguem melhorar suas classificações.

[1] http://www.image-net.org/challenges/LSVRC/

Diante disso, parece haver uma necessidade de correlacionar um classificador baseado em CNN e Shapelets. A motivação dessa proposta não é inédita. Vários autores em artigos se debruçam sobre essa temática, apesar de cada um ter uma metodologia e uma apresentação própria. Por exemplo, em Cui et al. (2016) os autores supõem que aprendizado de shapelets LTS é um caso particular de aprendizado em redes convolucionais. Os autores do artigo explicam que a medida de distância dos shapelets pode ser transformada em uma forma convolucional e, dessa forma, há uma relação. Entretanto, esse estudo específico apenas visa propor um classificador multivalorado baseado em CNN. Dessa forma, o estudo justifica o seu resultado positivo pela capacidade de aprender essas representações similares aos shapelets. Também nesse trabalho, os autores apresentam uma formalização teórica, correlacionando as camadas de uma CNN com shapelets e cálculo da distância Euclidiana. Apesar de os autores apresentarem essa formalização teórica, eles não aprofundam essa relação, evidenciando-a experimentalmente.

Apesar de não abordar diretamente, alguns outros conhecimentos têm envolvimento com essa questão. Outro artigo relevante a ser citado, Fawaz et al. (2018), explora a questão da transferência de conhecimento. Esse conceito se concretiza quando se utiliza os pesos de uma rede previamente treinada em outra série como pesos iniciais do treinamento, ao invés de pesos atribuídos aleatoriamente. Esse processo de aprendizado é mais rápido e com melhor generalização, o que se deve à transferência de representações entre às séries originais e as que receberam o conhecimento. O artigo cita, inclusive, séries baseadas em símbolos, como um caso particular de boa originadora de modelos. Dessa forma, evidencia-se que, ao transferir o conhecimento dessa série, a convergência da rede neural é mais rápida, reforçando a hipótese de que o modelo da rede neural aprende representações simbólicas (formas). Esse experimento influencia o desenvolvimento dessa proposta, conforme veremos no Capítulo 4.

4 MATERIAL E MÉTODO

Neste capítulo, são apresentados todos os detalhes das experimentações que foram realizadas, assim como quais são os resultados esperados dadas as referências bibliográficas e o método seguido. Tanto o classificador através dos códigos apresentados quanto os métodos utilizados estão disponíveis para consulta [1]. O presente trabalho serve como base para trabalhos derivados, seja reprodução dos procedimentos, análises ou inspiração para outros classificadores baseados em redes neurais convolucionais.

O início do experimento que gera a classificação foi a escolha das séries que compõem a base de dados. A escolha das 20 séries representadas na Tabela 4.1 se deu pela diversidade, assim como pelos resultados em artigos anteriores de classificadores baseados em representações simbólicas. Os dados dessa base se apresentam em arquivos de texto com formatos diversos, porém já normalizados e padronizados. A base de dados escolhida já possui exemplos de treinamento e teste, ou seja, cada base tem exemplos pré-classificados em quantidades pré-determinadas para serem usadas na geração e validação do modelo.

#	Conjunto de dados	Treinamento	Validação	Comprimento	Classes	Tipo
1	BeetleFly	20	20	512	2	IMAGE
2	BirdChicken	20	20	512	2	IMAGE
3	ECGFiveDays	23	861	136	2	ECG
4	ECG200	100	100	96	2	ECG
5	CBF	30	900	128	3	SIMULATED
6	FaceFour	24	88	350	4	IMAGE
7	FacesUCR	200	2050	131	14	IMAGE
8	Gun_Point	50	150	150	2	MOTION
9	ItalyPowerDemand	67	1029	24	2	SENSOR
10	Lightning7	70	73	319	7	SENSOR
11	Lightning2	60	61	637	2	SENSOR
12	MoteStrain	60	61	637	2	SENSOR
13	OliveOil	30	30	570	4	SPECTRO
14	DiatomSizeReduction	16	306	345	4	IMAGE
15	Coffee	28	28	286	2	SPECTRO
16	Symbols	25	995	398	6	IMAGE
17	Beef	30	30	470	5	SPECTRO
18	SyntheticControl	300	300	60	6	SIMULATED
19	Trace	100	100	275	4	SENSOR
20	TwoLeadECG	23	1139	82	2	ECG

Tabela 4.1: Séries temporais de Dau et al. (2018) selecionadas

[1] https://github.com/alexandrefelipemuller/timeseries_shapelet_transferlearning

4.1 EXTRAÇÃO DAS SHAPELETS

Esse passo é o diferencial deste trabalho em relação a um treinamento convencional utilizando um classificador baseado em uma rede neural. A extração das shapelets é uma tarefa que usualmente demanda muito tempo de processamento e treinamento de uma rede neural. Encontrar essas subsequências é um passo adicional que permite fazer transferência destas para a série original.

Dado o conjunto de séries, foram extraídas as shapelets mais relevantes de cada uma e, para isso, foi utilizado o método tradicional com janela deslizante e dicionário de candidatos de Ye e Keogh (2009) de tamanhos de 5 a 18 valores. Uma descrição mais detalhada pode ser encontrada no capítulo 2. Essa escolha de tamanho foi ajustada para equilibrar a proporção em valores do que era shapelet do que não era. Nesse método, é gerada uma lista com todas as representações com o cálculo do ganho. Essa lista, então, é ordenada pelo ganho. Se fosse para escolher apenas um shapelet por classe, a escolha do primeiro elemento dessa lista seria o suficiente. Nesse caso, são separados os 3 mais relevantes, descartando-se interseções. Esse descarte é feito olhando a posição em que o shapelet foi extraído na série original: se houver conflito da posição (por mais que se trate de exemplos diferentes), ocorre a exclusão. Desses candidatos, foram escolhidos de 1 a 3 shapelets mais relevantes. O critério de escolha se baseia no ganho e em não haver intersecção para a maioria das séries.

Os shapelets resultantes servem para gerar duas bases extras de apoio, sendo três bases ao todo, nomeadas a seguir:

- Série Original;
- Série Shapelet: onde foi recortada da série original somente a parte correspondente às shapelets e todo resto substituído pelo valor central da série. Entende-se por valor central o valor médio dos extremos da série toda;
- Série Inversa: onde foi recortada da série original somente a parte não correspondente às shapelets, restando a parte correspondente da série shapelet substituído pelo valor central.

O processo visa a geração dessas outras duas bases de apoio. Naturalmente, ao passo que se obtém as Shapelets já são geradas a Série Shapelet e a Série Inversa como no exemplo da Figura 4.1. Ambas as bases juntas, Série Shapelet e Série Inversa, correspondem ao dado original.

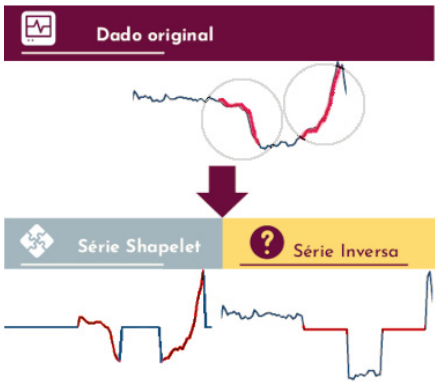


Figura 4.1: Exemplo da geração da série shapelet e inversa com 2 shapelets

4.2 TREINAMENTO DA REDE NEURAL

Como descrito no Capítulo 3, a partir dessas três bases foi executado o processo de convolução unidimensional. A implementação desse treinamento inicial é baseada na implementação de Wang et al. (2016), que não se utiliza de shapelet em momento algum. O único pré-processamento realizado foi o embaralhamento da base de treinamento para melhorar o treinamento da rede neural.

A arquitetura de rede é mostrada na 4.2. Ela foi ajustada empiricamente para 3 camadas unidimensionais. Primeira camada convolucional da rede tem tamanho de 32 filtros e kernel de comprimento 8 (conceito equivalente a janela), a segunda camada tem 64 filtros e kernel de comprimento 5 e a terceira camada convolucional tem 32 filtros kernel de comprimento 3. Ao final há uma camada de Pooling e a saída corresponde ao número de classes, obtendo então, a probabilidade da classificação de cada uma. Em relação a Wang et al. (2016) foi feito a remoção do "dropout". Essa técnica consiste na remoção aleatória de alguns neurônios para melhorar a generalização e assim evitar *overfitting* (Srivastava et al., 2014).

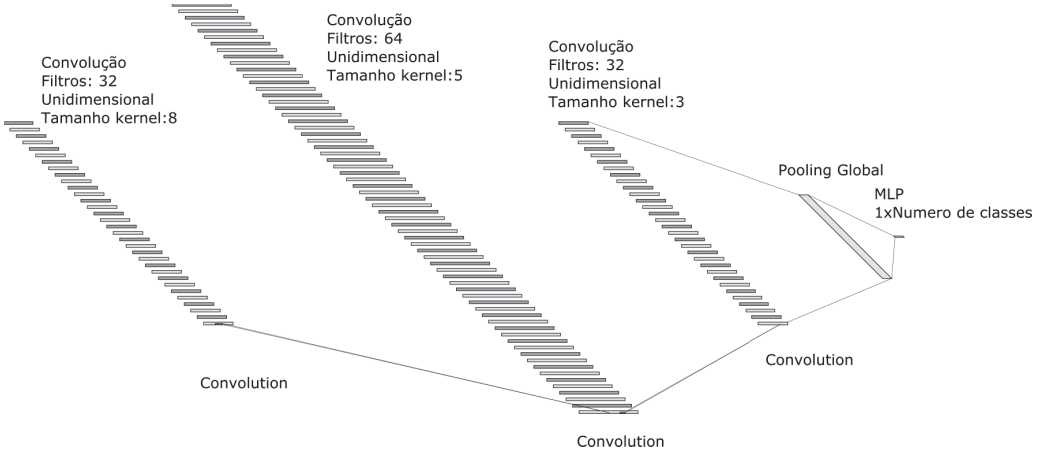


Figura 4.2: Modelo proposto de arquitetura CNN unidimensional

A partir da implementação com esse classificador, foram executados alguns experimentos:

0. Processo que antecede o treinamento: extração dos Shapelets e geração das bases de apoio;
1. Treinamento da série original iniciada com pesos zerados;
2. Treinamento da série shapelet iniciada com pesos zerados;
3. Treinamento da série original iniciada com modelo importado do experimento 2;
4. Treinamento da série inversa iniciada com pesos zerados;
5. Treinamento da série original iniciada com modelo importado do experimento 4;
6. Validação cruzada, treinamento da série shapelet, importando modelo da 4;
7. Validação cruzada, treinamento das séries inversa, importando modelo da 2.

O objetivo do item 1 é ter um marco comparativo para os testes seguintes, dessa maneira os resultados deste item devem ser semelhantes aos obtidos em Wang et al. (2016). Os itens 2 e 4, por sua vez, têm como objetivo obter um modelo de treinamento, sendo esse modelo somente os shapelets ou o restante da série sem shapelets (inversa). Os itens 3 e 5 são os mais importantes, pois visam mostrar a relevância da transferência do modelo previamente treinado. Como todas as séries possuem mesmo comprimento preenchido com valor central da série, a troca do modelo inicial é possível. Por fim, os itens 6 e 7 servem para elucidar e ratificar a importância da transferência de conhecimento em cada situação. Como essa classificação se baseia em uma rede neural, os pesos sofrem de perda de informação ao longo do treinamento, isso se a mesma informação não estiver sendo colocada em treinamento novamente. O que significa que se o experimento 6 apresentar melhores resultados que o 7, os shapelets são a parte mais relevante da informação.

Dessa forma, apesar de todos os experimentos possuírem a mesma arquitetura de rede, somente os experimentos 1, 3 e 5 utilizam o dado completo, ficando todos os outros com dados parciais.

4.3 TREINAMENTO E TRANSFERÊNCIA DE CONHECIMENTO

A primeira fase do treinamento no sentido amplo, a extração das representações simbólicas, é um algoritmo determinístico, portanto, não altera o seu resultado em diversas rodadas. Por sua vez, o treinamento da rede neural pode variar por ter fatores aleatórios que são os ajustes dos pesos. Sendo assim, os experimentos resultantes do treinamento da rede neural precisam ser avaliados estatisticamente.

As bases selecionadas da Tabela 4.1 são separadas em treinamento e validação (ou teste). Durante o treinamento da rede neural, os pesos são ajustados com exemplos da base de treinamento. Ao final de cada época (que é o fim do ajuste dos pesos), é validada a acurácia do modelo na base de validação, porém esse resultado é somente uma consulta. Durante todo o processo de treinamento, naturalmente, só é usada a base de treinamento. Isto é, enquanto o treinamento ocorre, o erro continua sendo ajustado e diminuído, porém esse é o erro da base de treinamento, enquanto o erro na base de validação aumenta. Essa diferença basicamente é uma forma de sobre-ajuste ou *overfitting*.

Dado o tamanho das bases de séries temporais, foi determinado o aprendizado em 300 épocas, ficando o melhor modelo entre estas épocas salvo para a etapa de validação. Para a maioria das séries testadas, o sobre-ajuste começa a acontecer entre 150 e 250 épocas, ficando o melhor modelo salvo. Como os modelos posteriores são descartados, por isso a escolha por rodar 300 épocas. Via de regra, o modelo validado somente na base de treinamento atinge 100% de acurácia dessa forma, o melhor modelo é dado por um *"checkpoint"*, ou seja, enquanto o treinamento continua e o erro aumenta, o melhor modelo fica salvo até que outro melhor seja encontrado. É como se estivessem sendo guardados modelos de todas as épocas e, ao final, sendo escolhido o melhor. Portanto, não há critério de parada nessa faixa de 250 épocas, porque já é salvo o melhor modelo.

Como se trata da transferência do modelo dentro do próprio domínio é esperado que todo experimento que receba um modelo pré-treinado evolua rapidamente em número de épocas para atingir um resultado satisfatório. O comportamento nessa situação de transferência de conhecimento é a evolução de acurácia do modelo de forma mais rápida (Fawaz et al. (2018)). Por esta razão, a maior expectativa na definição dos experimentos seria que o experimento 3, além de evoluir rapidamente, teria melhor acurácia. Essa expectativa também se deve ao marco teórico exposto no Capítulo 3.

Neste capítulo foi apresentada a forma com que os experimentos foram conduzidos, além de quais e como os dados foram testados, bem como quais são os parâmetros para a extração dos shapelets e quais e como são os experimentos rodados. Assim, o próximo capítulo apresenta e discute os resultados obtidos a partir dessa pesquisa.

5 RESULTADOS E DISCUSSÃO

A discussão dos resultados foi subdividida entre as Seções 5.2, 5.3 e 5.4, que comparam pares entre Série Shapelet e Série Inversa. Ao final, todo objetivo se concentra nos experimentos 1, 3 e 5 que são os classificadores finais. Após a apresentação de um panorama geral das séries, são destacados três estudos de caso que devem ser analisados particularmente.

5.1 EXPERIMENTO 1

A média da taxa de acerto do experimento 1 foi semelhante a implementação de Wang et al. (2016) naquelas bases que foram executadas (Confrontando Tabela 1 de Wang et al. (2016) com Tabela 5.1).

5.2 EXPERIMENTO 2 E 4, RESULTADOS E DISCUSSÃO

O primeiro resultado a se considerar é a acurácia de validação média (entre todas as séries temporais) entre experimentos que iniciaram com pesos zerados para série Shapelet, que seria o experimento 2, e Inversa, que seria o experimento 4. No experimento 2, foi 75%, contra o experimento 4, que foi 82%. Isso leva a crer que, numa média, há ainda muita informação relevante para classificação que está compreendida fora dos shapelets. Em partes, isso se deve ao tamanho das shapelets extraídas. Destaca-se que essas representações são curtas, os shapelets gerados têm comprimentos absolutos de no máximo 18 valores, que para maioria das bases é um comprimento pequeno. Um aumento nesse comprimento ajustaria esse resultado, porém a um custo de não fazer sentido colocar em evidência um segmento muito grande. Como o experimento extrai o segmento da série original, aumentar demais o segmento do shapelet tornaria a premissa inicial de colocar shapelet em evidência comprometida, pois o shapelet poderia ser quase todo comprimento da série. Dessa forma, há um impasse entre ajuste dos parâmetros: de tamanho dos shapelets extraídos, quantidade dos shapelets extraídos e a representatividade do shapelet. Além disso, a quantidade de informação dentro e fora dele influenciam no resultado final.

5.3 EXPERIMENTO 6 E 7, RESULTADOS E DISCUSSÃO

Outro indicador da qualidade dos shapelets são estes dois experimentos de validação cruzada: Treinamento 6 (modelo importado da inversa e série shapelet sendo treinado por último) e 7 (modelo importado da série shapelet e série inversa treinada por último). A evolução foi descrita na figura 5.1, onde é possível notar que somente com a transferência de conhecimento, sem todos os dados juntos, não é possível superar o treinamento da série toda iniciada do zero (experimento 1). Mesmo assim, aquele que mais se aproximou foi justamente o teste iniciando no modelo inverso e terminando com os shapelets. Como esperado, o que é treinado por último é a informação colocada mais em evidência. Por outro lado, precisamos considerar o quão incompleta essa análise pode ser, dado que representa uma média de vários testes com várias bases. Por esse motivo, a próxima seção traz resultados mais precisos sobre a distribuição entre as diversas séries da base.

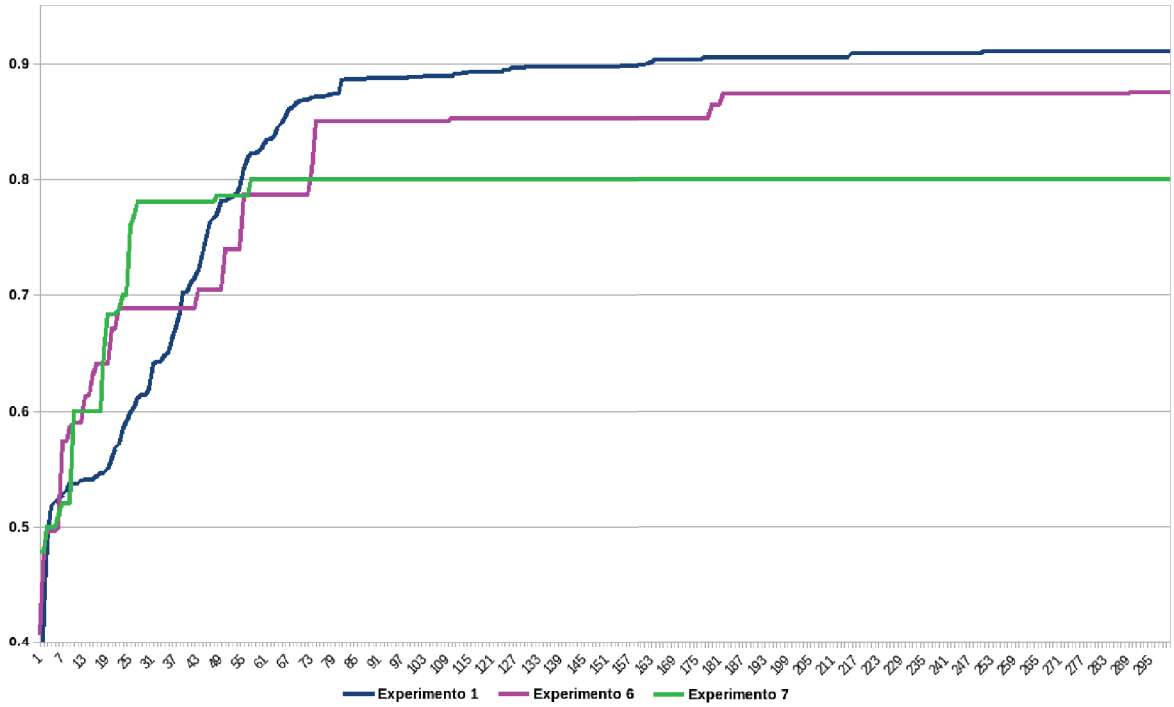


Figura 5.1: Épocas X Acurácia de treinamento do modelo nos experimentos 6 e 7

5.4 EXPERIMENTO 3 E 5, RESULTADOS E DISCUSSÃO

Estes dois experimentos de treinamento se dão sobre a série original, porém com transferência de conhecimento das séries shapelet e inversa, constituem o objetivo final. O resultado da acurácia na base de validação para os experimentos 1, 3 e 5 está descrito na tabela 5.1 para cada série. Para comparação foram escolhidos 4 métodos de validação em evidência no momento, como descrito no capítulo 3. A tabela 5.1 está expressa em médias (de 30 medições), cada série de medições possui variação padrão na ordem de 10^{-5}. O teste t student das séries tanto entre experimentos 1 com 3 e separadamente experimentos 1 com 5 obtiveram p-valor na ordem de cada um 10^{-15}, portanto é um grau de confiabilidade bastante elevado de que as séries apresentam diferenças estatisticamente relevantes.

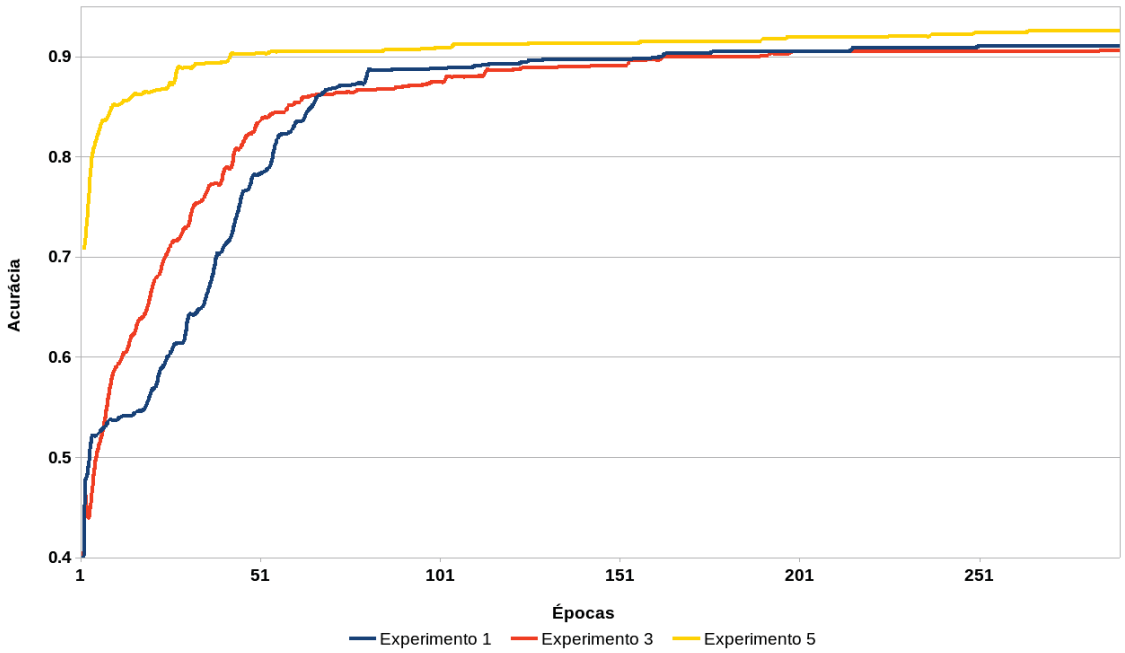


Figura 5.2: Épocas X Acurácia das séries ao longo das épocas

5.5 RESULTADOS POR BASE DE DADOS

Como já explicado na seção 5.4, ao todo foram realizadas 30 rodadas de treinamento. Para comparação com outros trabalhos está expressa as médias na Tabela 5.1.

#	Conjunto de Dados	Expr 1	Expr 3	Expr 5	LTS	ELIS	BOSS	HIVE-COTE
1	BeetleFly	0.9500	0.9500	**1.0000**	**1.0000**	0.8500	0.9490	0.9590
2	BirdChicken	1.0000	1.0000	**1.0000**	**1.0000**	0.9000	0.9840	0.9505
3	ECGFiveDays	0.9978	0.9954	**1.0000**	0.9954	**1.0000**	0.9830	0.9895
4	ECG200	0.9100	**0.9300**	0.9000	0.9200	-	0.8900	0.8819
5	CBF	0.9933	0.9956	0.9978	0.9967	-	0.9980	**0.9994**
6	FaceFour	0.9318	0.9091	0.9659	0.9432	0.9545	**0.9960**	0.9495
7	FacesUCR	0.9268	0.9273	0.9210	0.9434	-	0.9510	**0.9836**
8	Gun_Point	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9333	0.9940	0.9967
9	ItalyPowerDemand	0.9718	**0.9974**	0.9689	-	0.9757	0.8660	0.9678
10	Lightning7	0.8904	0.8493	0.8630	**0.9178**	0.8082	0.8100	0.8111
11	Lightning2	0.7869	0.7869	**0.8197**	0.7869	-	0.6660	0.7970
12	MoteStrain	0.9393	0.9377	0.9090	0.9361	0.8978	0.8460	**0.9468**
13	OliveOil	0.8000	0.7000	0.6667	**0.9667**	-	0.8700	0.8977
14	DiatomSizeRed	0.6536	0.5850	0.5850	-	0.8987	0.9390	**0.9419**
15	Coffee	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9643	0.9890	0.9982
16	Symbols	0.9598	0.9545	0.9618	**0.9889**	0.7829	0.9610	0.9650
17	Beef	0.6333	0.7000	0.7000	**0.9330**	0.6333	0.6150	0.7227
18	SyntheticControl	0.9933	0.9933	0.9900	-	-	0.9680	**0.9996**
19	Trace	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
20	TwoLeadECG	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9982	0.9850	0.9935

Tabela 5.1: Resultado da acurácia dos principais métodos, com melhor resultado em negrito

Um resultado importante é que dessas 20 séries, pelo menos 15 (nos experimentos 3 e 5) foram melhores quando comparadas com o experimento 1 (base). Outro fator relevante é que a acurácia da classificação superou o que equivale ao estado da arte em metade dos casos. Conforme é possível ver na figura 5.2, a transferência de conhecimento dos shapelets já no inicialização dos pesos faz a série convergir mais rápido, porém causando sobre-ajuste (*overfitting*). Ao final da classificação tem-se uma média inferior ao experimento 1 (marco comparativo). Por outro lado, a transferência de conhecimento das informações alheias aos shapelets faz com que a convergência seja ainda maior e com resultado final médio superior ao experimento 1. Tal comportamento se deve ao fato de que, quando shapelets são aprendidos e depois transferidos, o modelo se torna tendencioso. Ou seja, é mais complicado se ajustar ao resto da série, tendo já a parte mais relevante ajustada no modelo.

5.6 ESTUDOS DE CASO

Apesar do quadro geral apresentado sobre séries temporais, alguns exemplos foram destacados como relevantes e serão abordados a seguir.

5.6.1 ECG200, o melhor caso

Dentre as diversas séries testadas, uma das que apresentou melhor desempenho, se comparada ao estado da arte, foi a ECG200. Essa base foi apresentada por Olszewski (2001) como parte de uma tese. Cada série representa a atividade elétrica gerada por uma única batida de coração. As duas classes presentes representam as condições: batimento normal ou infarto agudo do miocárdio. As patologias cardíacas são diagnosticadas pelos conhecedores do domínio por meio de deformações na forma de áreas da onda. Em especial, a miocardiopatia é diagnosticada por uma elevação do segmento ST, como mostra a Figura 5.4(a). Esse tipo de deformação de onda é o caso mais típico de aplicação dos shapelets. A Figura 5.5 representa quais foram os shapelets extraídos para cada classe, coincidentemente compatíveis com uma elevação no segmento específico.

Essa série apresenta certa complexidade e superou o estado da arte com 93% de acurácia contra 89% a 92% de trabalhos anteriores. Não por acaso, todas as séries do tipo ECG tiveram um bom desempenho (séries 3, 4 e 20), pois elas se baseiam em representações morfológicas.

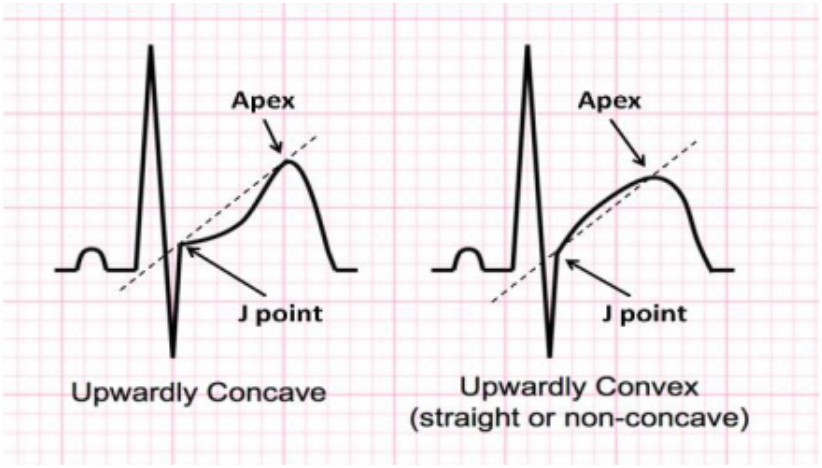


Figura 5.3: Elevação do segmento ST do eletrocardiograma mostrando possível cardiopatia.

Fonte: (Abdushi et al., 2008)

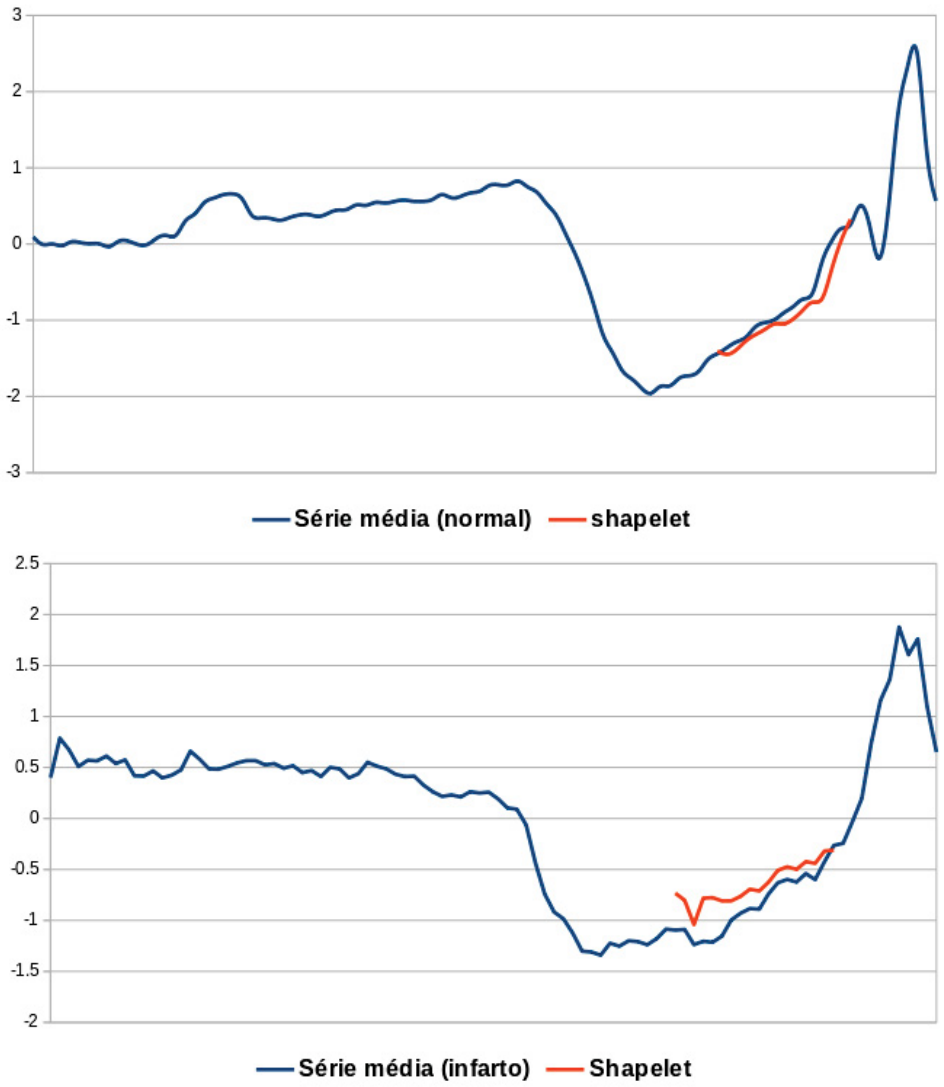


Figura 5.5: Shapelets aprendidos para cada classe de ECG200

5.6.2 OliveOil, o pior caso

Essa base representa uma espectroscopia de amostras de azeite de oliva, apresentada pela primeira vez em Bagnall et al. (2012). As aplicações práticas são na classificação da qualidade e origem de cada amostra. Cada uma das 4 classes correspondem a um azeite vindo de um país diferente. Esta base teve resultado bem inferior ao desejável, tanto no experimento 3 (70%) quanto no experimento 5 (67%). Uma análise mais aprofundada revela que os Shapelets extraídos tiveram alta similaridade entre as classes, ficando muito próximos do valor médio da série. Ou seja, indica que foram extraídas representações de baixa qualidade.

Esse gráfico representa um histograma, ou seja, uma dispersão de frequências de classes químicas. Por se tratar de um histograma, as representações simbólicas podem não fazer sentido aos espectros de comida (como por exemplo número 17 [Beef]) também tiveram baixo desempenho.

5.6.3 FaceFour, casos inesperados

Tendo sido apresentado o melhor caso (subseção 5.6.1) e o pior caso (subseção 5.6.2), há situações em que os resultados merecem uma discussão por apresentarem características

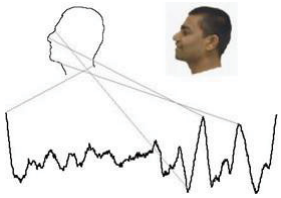


Figura 5.6: Representação da série FaceFour.

Fonte: (Dau et al., 2018)

particulares. É o caso da série FaceFour, que representa o formato de rostos de perfil mostrado na Figura 5.7(a). Essa série possui 4 classes (sendo cada uma delas relacionada a um indivíduo diferente). No teste cujo modelo inicial era aquela sem as Shapelets essa base teve um bom desempenho (96,6%). Na figura 5.8 é possível ver os Shapelets extraídos que representam segmentos muito característicos do rosto de cada indivíduo. A exemplo a figura 5.9, que representa a média da base como um todo, os segmentos extraídos ficam distantes como esperado. Apesar de ser um caso muito característico do uso de representações simbólicas, a escolha por apenas 3 representações deixou parte da informação de fora. Dessa forma o rosto como um todo tem também outras informações sobre os indivíduos. Por consequência, o rosto sem essas partes serve como base melhor para começar o treinamento, ficando as partes mais características para a parte final.

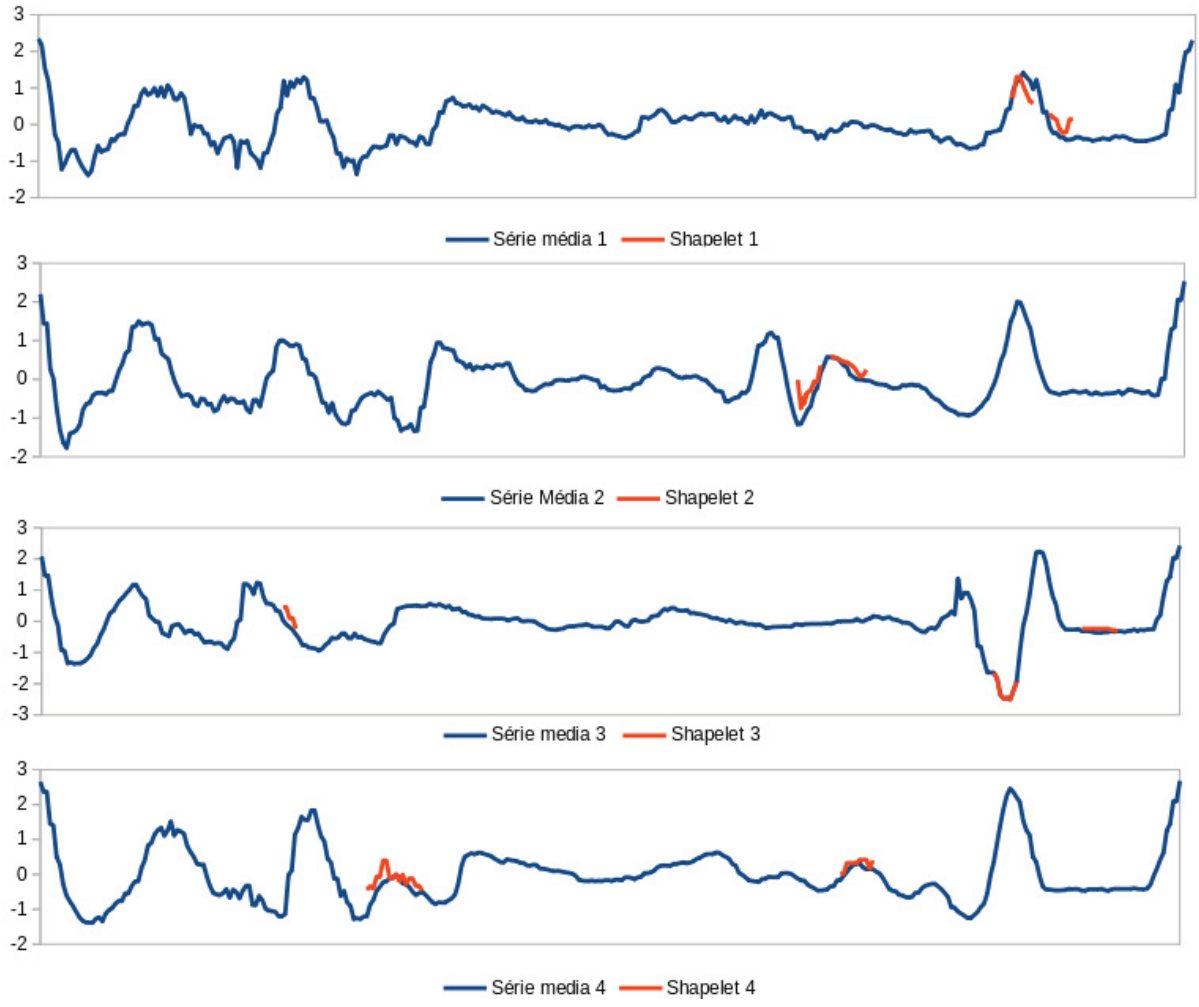


Figura 5.8: Representação da média das classes FaceFour com os shapelets extraídos para as quatro classes

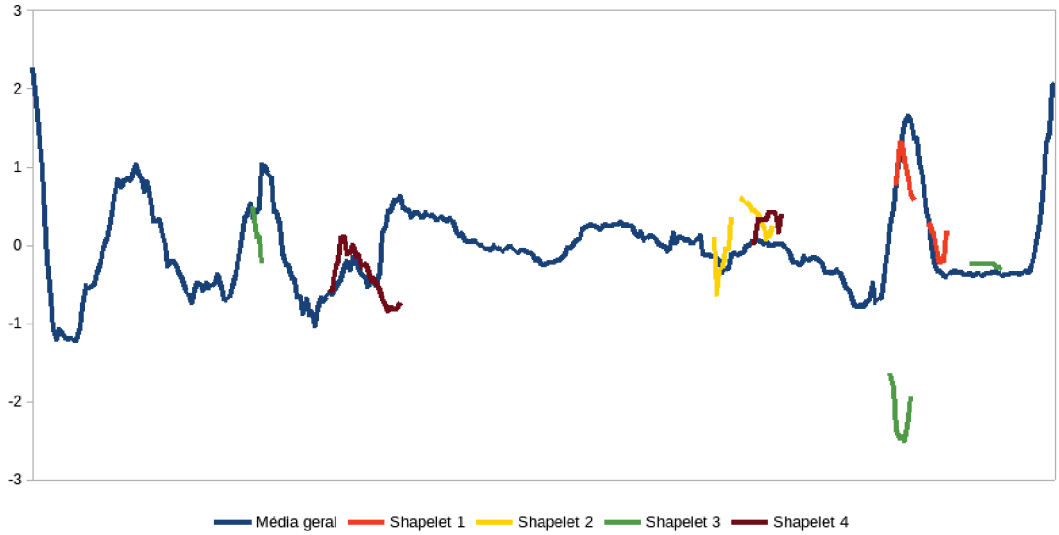


Figura 5.9: Representação da média da série FaceFour com os shapelets extraídos para as quatro classes. Conforme esperado, os shapelets estão fora da média

Nesse capítulo foram apresentados os resultados coletados para todos os sete experimentos, comparando-os em pares com a Série Shapelet e Série Inversa. Os resultados foram comparados, também, com os melhores classificadores do estado-da-arte, superando-os em alguns casos. Apresentamos 3 exemplos de forma mais detalhada, cada um deles contando com características e resultados diferentes. Toda a parte experimental foi resultado de uma evolução da análise sobre os dados, porém tornou-se evidente que o uso de transferência de conhecimento ocasionou ganho de acurácia na maior parte das séries temporais utilizadas, resultado da transferência de conhecimento. Por outro lado, ainda é possível inferir que a extração da representação dos shapelets é passível de ajustes e melhorias que podem refinar ainda mais os resultados. Todas essas questões são tratadas no próximo capítulo.

6 CONCLUSÃO

O método de melhoria proposto neste estudo consistia em extrair as representações simbólicas, gerar uma nova série, executar o treinamento só uma parte da série original e transferir o modelo aprendido. Tal processo permite colocar em evidência as representações simbólicas, sendo treinadas por primeiro ou por último. Assim, evita-se o ajuste do modelo somente aos exemplos de treinamento e se obtém uma capacidade mais generalista que melhora a taxa de acertos. Ao final, os objetivos deste trabalho se concretizaram, visto que os experimentos executados apresentaram resultados melhores que o marco inicial do treinamento puramente iniciado do zero.

Ao explorar as shapelets, obtivemos resultados importantes para classificação de séries temporais na maioria das bases testadas. Por fim, fica evidente a importância de extrair representações simbólicas em classificadores de séries temporais e como elas podem ser úteis mesmo em classificadores baseados em redes convolucionais.

Ao avaliar o estado da arte sobre a temática de representações simbólicas, era esperado que o modelo treinado com série Shapelets fosse similar ao modelo do treinamento da série original iniciada do zero. Entretanto, houveram situações em que a transferência de conhecimento evidenciou resultados superiores quando iniciada a partir dos shapelets e em alguns outros casos em que o resultado final foi melhor quando o treinamento foi terminado pelos shapelets. Nesses, parece que há um conhecimento externos frente às representações simbólicas que suportam uma classificação mais precisa e com menos sobre-ajuste (*overfitting*).

Em diversos casos, os resultados obtidos em termos de acurácia do classificador de séries temporais superam os trabalhos de referência. Os mesmos resultados no simples treinamento das bases, usando a implementação de Wang et al. (2016), seriam repetidamente semelhantes. Porém, ao incluir as representações simbólicas (que, na maioria dos casos, são os critérios de classificação dos conhecedores do domínio), os pesos se ajustam mais rapidamente (o que é esperado) e melhora a generalização do aprendizado, dois ganhos relevantes.

O aprendizado baseado em redes neurais não é trivial, pois necessita que o analista de dados ajuste seus experimentos com diversos parâmetros. Os parâmetros são variáveis de configuração que são otimizados (ou seja, ajustados manualmente) pelo processo de treinamento. Somente a continuidade do treinamento do modelo, sem mudanças dos parâmetros, invariavelmente levaria ao aumento do erro. O método proposto, portanto, traz uma grande vantagem nesse tipo de classificador.

Todo processo de extração dos shapelets revela informações inteligíveis sobre os dados estudados. Esse processo continua sendo importante, pois permite conhecer melhor os dados. Ao final dos experimentos, revela-se que as informações contidas na transferência de conhecimento fazem com que os dados treinados por último sejam colocados em evidência e, por esse motivo, resultem em uma melhor classificação média.

6.1 LIMITAÇÕES

No decorrer deste trabalho, já foram abordados alguns pontos identificados como limitações. Dentre eles, estão experimentos, cujos resultados ficaram aquém dos desejados, como séries temporais que representam espectros de análises químicas. Além disso, parece haver necessidade de sanar uma dúvida ainda não completamente respondida: há componentes não simbólicos relevantes para classificação que podem ser capturados por uma rede neural? O fator

de qualidade dos shapelets extraídos ainda precisa ser separado do fator de informações externas a toda representação simbólica.

6.2 TRABALHOS FUTUROS

Ao apontar as limitações deste trabalho, evidencia-se que estudos derivados ainda são necessários. Os dados dos experimentos nos levam a crer que a qualidade dos shapelets extraídos são passíveis de melhorias, tornando os resultado ainda mais refinados. Além disso, devido à quantidade de parâmetros, os testes precisam ser replicados a fim de confirmar se os resultados persistem em diferentes contextos, como arquiteturas da rede neural.

A análise sobre o modelo já treinado em uma rede neural convolucional também é um desdobramento necessário. Dada a complexidade de um conjunto de pesos já ajustado, o aprendizado não supervisionado pode ser aplicado para segmentar os conceitos no modelo aprendido, como, por exemplo, identificar nos pesos ativados de uma rede neural onde estão as representações simbólicas. Isso daria uma perspectiva não experimental, mais analítica, do que ocorre durante o processo de transferência de conhecimento. Esse processo pode levar à supressão do estágio de treinamento das séries shapelet usando uma rede neural e criando um novo modelo do zero.

REFERÊNCIAS

Abdushi, S., Veseli, A., Abdushi, S. e Zenelaj, F. (2008). Differential diagnosis of st segment elevation on ecg.

Abel, M. U. (2018). Classificação de séries temporais por meio da transformada shapelet. Dissertação de Mestrado, UFPR.

Bagnall, A., Davis, L., Hills, J. e Lines, J. (2012). Transformation based ensembles for time series classification. Em *Proceedings of the 2012 SIAM International Conference on Data Mining*, páginas 307–318. SIAM.

Bagnall, A., Lines, J., Hills, J. e Bostrom, A. (2015). Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.

Boger, Z. e Guterman, H. (1997). Knowledge extraction from artificial neural network models. Em *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 4, páginas 3030–3035. IEEE.

Cui, Z., Chen, W. e Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification. *CoRR*, abs/1603.06995.

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A. e Batista, G. (2018). The ucr time series classification archive. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

Ehlers, R. S. (2007). *Análise de Séries Temporais*. UFPR.

Fang, Z., Wang, P. e Wang, W. (2018). Efficient learning interpretable shapelets for accurate time series classification. páginas 497–508.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L. e Muller, P. (2018). Transfer learning for time series classification. *CoRR*, abs/1811.01533.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L. e Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.

Ferrero, C. A. (2009). Algoritmo knn para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia. Dissertação de Mestrado, Universidade de São Paulo, USP.

Gamboa, J. C. B. (2017). Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.

Goldberg, D. E. e Holland, J. H. (1988). Genetic algorithms and machine learning.

Grabocka, J., Schilling, N., Wistuba, M. e Schmidt-Thieme, L. (2014). Learning time-series shapelets. Em *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 14, páginas 392–401. ACM.

Längkvist, M., Karlsson, L. e Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24.

LeCun, Y., Bengio, Y. e Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

LeCun, Y., Kavukcuoglu, K. e Farabet, C. (2010). Convolutional networks and applications in vision. Em *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, páginas 253–256.

Li, G., Choi, B., Bhowmick, S. S., Wong, G. L.-H., Chun, K.-P. e Li, S. (2020a). Visualet: Visualizing shapelets for time series classification. Em *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, páginas 3429–3432.

Li, G., Choi, B. K. K., Xu, J., Bhowmick, S. S., Chun, K.-P. e Wong, G. L. (2020b). Efficient shapelet discovery for time series classification. *IEEE Transactions on Knowledge and Data Engineering*.

Lin, J., Keogh, E., Wei, L. e Lonardi, S. (2007). Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144.

Lines, J. e Bagnall, A. (2015). Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592.

Lines, J., Davis, L. M., Hills, J. e Bagnall, A. (2012). A shapelet transform for time series classification. Em *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, páginas 289–297. ACM.

Lines, J., Taylor, S. e Bagnall, A. (2018). Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Trans. Knowl. Discov. Data*, 12(5):52:1–52:35.

Mueen, A., Keogh, E. e Young, N. (2011). Logical-shapelets: An expressive primitive for time series classification.

Nair, V. e Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. Em *ICML*.

Olszewski, R. T. (2001). *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. Tese de doutorado, Citeseer.

P. A. Morettin, C. M. T. (2006). *Análise de Séries Temporais*. Edgard Blecher.

Pan, S. J. e Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Rezende, S. O. (2003). *Sistemas Inteligentes: Fundamentos e Aplicações*. Barueri, Brasil: Manole.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530.

Senin, P. (2008). Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. e Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Wang, J., Liu, P., She, M. F., Nahavandi, S. e Kouzani, A. (2013). Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634–644.

Wang, Z., Yan, W. e Oates, T. (2016). Time series classification from scratch with deep neural networks: A strong baseline. *CoRR*, abs/1611.06455.

Ye, L. e Keogh, E. (2009). Time series shapelets: A new primitive for data mining. KDD '09, páginas 947–956, New York, NY, USA. ACM.

Ye, L. e Keogh, E. (2011). Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data mining and knowledge discovery*, 22(1):149–182.

Zalewski, W. (2015). Modelagem simbólica de padrões morfológicos para a classificação de séries temporais. Dissertação de Mestrado, UFPR.

Zhang, H., Wang, P., Fang, Z., Wang, Z. e Wang, W. (2021). Elis++: a shapelet learning approach for accurate and efficient time series classification. *World Wide Web*, 24(2):511–539.