



Previsão Inteligente do consumo de energia com base em modelos Deep Learning

RICARDO DE SÁ DALTE SILVA BÁRTOLO

Outubro de 2018

Previsão Inteligente do consumo de energia com base em modelos Deep Learning

Ricardo Bártolo

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de Informação e Conhecimento

Orientador: Isabel Praça
Co-Orientador: Eugénia Vinagre

Júri:

Presidente:

Dr. Nuno Silva, Professor, DEI/ISEP

Vogais:

Dr. Susana Nicola, Professor, DEI/ISEP

Dr. Nuno Escudeiro, Professor, DEI/ISEP

Resumo

Devido ao contínuo aumento do consumo de eletricidade e a possibilidade do esgotamento de recursos não renováveis, surge uma necessidade de uma melhor gestão do setor energético. Assim, com a inserção de fontes renováveis de energia na rede elétrica, uma das tarefas mais importantes revela-se ser a previsão do consumo de energia. Como tal, a obtenção de uma previsão do consumo de energia confiável é essencial para o controlo do consumo e a manutenção do equilíbrio nas redes de distribuição. Esta tese apresenta uma comparação entre soluções de Machine Learning tradicionais e Deep Learning para o problema da previsão do consumo de energia quando aplicadas a diferentes casos. Estas abordagens são apoiadas pela utilização da plataforma TensorFlow como fonte dos algoritmos utilizados. O objetivo deste estudo é explorar se o potencial fornecido pelo Deep Learning é capaz de superar a previsão de consumo dos métodos tradicionais, e fazer-lo quase em tempo-real. Os resultados mostram que apesar do Deep Learning ter capacidade para conseguir uma melhor precisão de previsão, necessita de grandes quantidade de dados como suporte.

Abstract

Due to the increasing amount of today's electricity consumption and the possibility of the depletion of non-renewable resources, a necessity arose for better management of the energy sector. As such, with the insertion of renewable sources of energy into the grid, one of the most important tasks became the prediction of the energy consumption. This way, having a trustable forecast of the energy consumption is essential to control the consumption and maintain the balance in energy distribution networks. This thesis presents a comparison between traditional Machine Learning and Deep Learning solutions for the energy consumption forecasting problem, when applied to different scenarios. These approaches are supported by the usage of the TensorFlow platform as source for the algorithms used. The goal of this study is to explore whether the potential that Deep Learning provides can be used to surpass the forecasting of traditional Machine Learning, while doing so in near real-time execution. The results show that while Deep Learning can provide better precision of forecasting, it needs great amounts of data to feed and support it.

Conteúdo

Lista de Figuras	v
1 Introdução	1
2 Contexto e Estado da arte	3
2.1 Smart Grid	3
2.1.1 Características	3
2.1.2 Microgrid	4
2.1.3 O problema de períodos de pico	5
2.1.4 Demand Response	6
2.1.5 Desenvolvimento	7
2.1.6 Previsão de Consumos	8
2.2 Machine Learning	9
2.2.1 Resolução de tarefas	10
2.2.2 Tipos de Aprendizagem	11
2.2.3 Algoritmos	12
2.2.4 Ferramentas	13
2.3 Deep Learning	14
2.3.1 Modelos	14
2.3.2 Áreas de Aplicação	16
2.3.3 Contributos Recentes	17
2.3.4 Ferramentas	17
3 Análise de Valor	21
3.1 The new concept development model (NCD)	21
3.2 Valor	22
3.3 Proposta de Valor	23
3.4 Cadeia e Rede de Valor	25
3.5 Método AHP	25
4 Design	31
4.1 Arquitetura adotada	31
4.1.1 Casos de Estudo	34
4.2 Arquitetura alternativa	35
4.2.1 Casos de Estudo alternativos	36
4.3 Dados e Datasets	37
5 Implementação	41
5.1 Linear Regression	44
5.2 Deep Neural Network	46
5.3 Comparação com trabalhos previamente existentes	51

6	Experimentação e Avaliação	57
6.1	Teste de Friedman	58
7	Conclusões	59
7.1	Trabalho Futuro	60
	Bibliografia	61
A	Experiências e dados adicionais	65
A.1	Experiências configuração DNN	65
A.2	Experiências configuração DNN base de dados completa	74
A.3	Dados adicionais do caso de estudo de 5.3	78

Lista de Figuras

3.1	Árvore hierárquica de decisão	26
3.2	Comparação Par a Par	26
3.3	Comparação Normalizada	26
3.4	Matriz paritária para Precisão	27
3.5	Matriz paritária para Tempo de Pré-Processamento	27
3.6	Matriz paritária para Tempo de Execução	27
3.7	Matriz paritária para Precisão Normalizada	28
3.8	Matriz paritária para Tempo de Pré-Processamento Normalizada	28
3.9	Matriz paritária para Tempo de Execução Normalizada	28
3.10	Matriz com prioridades globais	28
3.11	Prioridades Compostas	29
3.12	Cálculo vetor próprio	29
3.13	Tabela de valores IR para matrizes quadradas de ordem n	30
4.1	Arquitetura TensorFlow	32
4.2	Arquitetura TensorFlow Simplificada	33
4.3	Arquitetura TensorFlow Distribuída	34
4.4	Big Data	35
4.5	Extensão Big Data	36
4.6	Vista aérea Edifício N	37
4.7	Exemplo Query BD	38
4.8	Exemplo Query Consumo	39
5.1	Classe Estimator	42
5.2	Previsão do Consumo de Energia	43
5.3	Execução Linear Regression Inicial	44
5.4	Execução Linear Regression Sockets	45
5.5	Execução Linear Regression Lights	45
5.6	Execução Linear Regression HVAC	46
5.7	Execução Linear Regression Total	46
5.8	Execução DNN Sockets	47
5.9	Execução DNN Lights	47
5.10	Execução DNN Sockets	48
5.11	Execução DNN Total	48
5.12	Comparação Layers DNN	49
5.13	Exemplo BoxPlot com Outliers	49
5.14	Exemplo BoxPlot sem Outliers	50
5.15	Comparação Layers DNN Base de Dados completa	50
5.16	Resultados caso estudo Linear Regression	52
5.17	Resultados caso estudo DNN	53
5.18	Resultados caso estudo Linear Regression sem Outliers	54

5.19	Resultados caso estudo DNN sem Outliers	55
5.20	Resultados paper MOGUL	56
A.1	Teste configuração 10,10	65
A.2	Teste configuração 20,10	66
A.3	Teste configuração 20,20,20	67
A.4	Teste configuração 20,15,10	68
A.5	Teste configuração 50,50,50	69
A.6	Teste configuração 1024,512,256,128,64,32,16,8	70
A.7	Teste configuração 20,20,20,20,20	71
A.8	Teste configuração 50,50,50,50,50	72
A.9	Teste configuração 25,25,25,25,25	73
A.10	Teste configuração 50,40,30,20,10	74
A.11	Teste configuração 50,50,50,50,50 base de dados completa	75
A.12	Teste configuração 50,50,50,50,50,50,50,50,50,50 base de dados completa	76
A.13	Teste configuração 100,100,100,100,100,100,100,100,100,100 base de dados completa	77
A.14	Resultados caso estudo Linear Regression para HVAC	78
A.15	Resultados caso estudo Linear Regression para Lights	79
A.16	Resultados caso estudo Linear Regression para Sockets	80
A.17	Resultados caso estudo DNN para HVAC	81
A.18	Resultados caso estudo DNN para Lights	82
A.19	Resultados caso estudo DNN para Sockets	83

Capítulo 1

Introdução

Tem-se verificado uma crescente procura de energia, impulsionada pelos avanços tecnológicos, ao mesmo tempo que nos deparamos com a grande possibilidade do esgotamento de recursos não renováveis. Por razões económicas, ecológicas e mesmo políticas impostas pela União Europeia, em que impõem claramente a redução de emissão de CO₂ a cada país membro, surgiu a necessidade de repensar a forma de gerir o setor energético. Um dos principais objetivos foca-se na necessidade de reaproveitamento da energia proveniente de fontes renováveis e a sua gestão de forma cada vez mais eficiente e flexível. Assim surge um novo conceito, **Smart Grids**, (i.e. redes de energia inteligentes) sendo o seu desenvolvimento promissor para se atingirem os objetivos propostos na área de energias.

A flexibilidade e consciencialização do consumo é atualmente um dos grandes desafios na sua otimização. A previsão do consumo e produção de energia é fundamental para que as **Smart Grids** se transformem numa verdadeira realidade. Antever com a máxima brevidade e exatidão o valor destas variáveis (i.e. consumo e produção) é um dos maiores desafios das **Smart Grids** visto que a maioria das atividades e processos que as envolvem dependem destes valores, e só com esse conhecimento será possível otimizar os recursos. A previsão do consumo é de extrema complexidade e está relacionado com inúmeros fatores que dependem diversos contextos, desde as condições atmosféricas e ambientais, a fatores sociais, hora do dia, entre outras. A sua complexidade tende a aumentar em ambientes heterogéneos. A avaliação em tempo real do contexto envolvente é essencial para que as decisões a tomar ao nível da gestão de energia possam ser as mais adequadas.

No âmbito deste grande desafio a ser solucionado no contexto das Smart Grids (i.e. previsões em (quase) tempo real com a maior exatidão possível) é proposto o principal objetivo do presente projeto. Pretende-se o estudo, análise e implementação de metodologias avançadas de **Machine Learning** (i.e. **Deep Learning**) aplicadas à previsão de consumo energético.

Deep Learning é uma subárea de **Machine Learning** que estuda como solucionar problemas complexos e intuitivos. As metodologias propostas permitem, com recurso a meios computacionais, que as máquinas aprendam e compreendam o mundo em determinados contextos a partir de experiências anteriores e que, com base na hierarquia de conceitos, possam compreender conceitos mais complexos de forma a solucionarem eficientemente a mais variadíssima gama de problemas.

Os objetivos do presente trabalho são a utilização de abordagens **Deep Learning** quando comparadas com abordagens de **Machine Learning** tradicionais, com o intuito de demonstrar que as abordagens **Deep Learning** apresentam melhores resultados, bem como a exploração de diversas abordagens **Deep Learning** diferentes de modo a encontrar uma que apresente o desempenho adequado para o problema de previsão descrito.

A solução proposta para além de pretender proporcionar métodos de previsão de consumo de energia mais rápidos e eficientes, visa contribuir para o impulsionamento e implementação de sistemas de gestão de energia inteligentes. Assim, potenciará indiretamente a otimização de recursos, otimização da produção renovável, dinamização de mercados energéticos, entre outros.

A abordagem preconizada, que envolve a implementação de algoritmos de previsão de consumos de energia segundo as metodologias existentes na área de Deep Learning, será efetuada com recurso a uma plataforma Big Data disponível no GECAD (*Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development*), centro de investigação onde irão decorrer os trabalhos referentes á presente tese (Pinheiro et al. 2017) (Vinagre, Pinto et al. 2017). Esta plataforma disponibilizará os dados e ferramentas necessárias para o desenvolvimento do trabalho a ser realizado.

O presente documento encontra-se dividido em vários capítulos. O capítulo 2 apresenta um levantamento do estado da arte relevante ao contexto em que se insere o do problema identificado, abordando algumas soluções/abordagens existentes. No capítulo 3 é realizada uma análise de valor do trabalho de um ponto de vista de negócio. O capítulo 4 expõe o design da solução preconizada e finalmente no capítulo 5 é apresentada a metodologia de avaliação e experimentação para a solução preconizada.

Capítulo 2

Contexto e Estado da arte

Neste capítulo irão ser abordados os temas de maior relevo da presente tese (i.e., **Smart Grids**, **Machine Learning** e **Deep Learning**), onde são demonstrados alguns exemplos da sua implementação e aplicações.

2.1 Smart Grid

A indústria de energia elétrica encontra-se num processo de transformação de uma rede centralizada, controlada pelos produtores, para uma menos centralizada e mais interativa com o consumidor. A transição para uma rede mais inteligente (**Smart Grid**) traz uma mudança no modelo de negócio da indústria no geral, bem como as relações com os *stakeholders*, *utilities* envolvidas, reguladores, fornecedores de serviços de energia, tecnologia e promotores de automação, bem como todos os consumidores de energia elétrica.

Uma rede mais inteligente torna esta transformação possível ao introduzir filosofias, conceitos e tecnologias que permitiram a utilização de comunicações bi-direcionais nas *utilities* e na rede elétrica. Mais importante, possibilita alcançar o potencial de algumas das melhores ideias de atualização/modernização da rede.

Muitas destas ideias já se encontram em desenvolvimento. No entanto, grande parte delas apenas se tornaram possíveis com a implementação de comunicação digital bi-direcional, bem como capacidades de automação e controlo remoto. Uma vez que a maior parte destas interações ocorrem “em segundo plano” com intervenção humana mínima, isto potencia novas abordagens de gestão de recursos de energia de forma mais inteligente e autónoma.

Este tipo de implementação foi conceptualizada no passado, mas sem as ferramentas que a **Smart Grid** oferece com a implementação de tecnologias de ponta, interoperabilidade baseada em padrões e comunicação e eletrónica de baixo custo, não possuía o potencial que hoje se verifica. (United States Department of Energy 2008)

2.1.1 Características

De acordo com o departamento de energia dos Estados Unidos da América, cinco tecnologias são responsáveis por impulsionar a **Smart Grid**:

- Comunicações integradas que conectam componentes para uma arquitetura aberta, permitindo troca de informação e controlo em tempo-real
- Tecnologias de deteção e medição para fornecer suporte mais rápido e com maior precisão em monitorização remota, gestão preço aquando a sua utilização, bem como gestão do lado do consumidor

- Componentes avançados para utilização com as investigações mais recentes em supercondutividade, armazenamento, eletrónica de potência e diagnóstico
- Métodos de controlo avançados, como ferramenta para monitorizar componentes essenciais, permitir diagnóstico rápido e soluções precisas e apropriadas para qualquer evento
- Interfaces melhoradas e apoio à decisão de forma a aperfeiçoar a tomada de decisão humana, disponibilizando uma nova visão do sistema aos operadores e gestores da rede

Desta forma a **Smart Grid** difere da rede elétrica tradicional de acordo com a seguinte tabela:

Rede elétrica tradicional	Smart Grid
Subjetiva - consumo e produção calculados de maneira comparativa	Objectiva – consumo e produção calculados de forma numérica
Comunicação uni-direcional	Comunicação bi-direcional
Produção centralizada	Produção descentralizada
Comunicação em parte da rede	Comunicação com a rede toda
Gestão do saldo de energia com base na produção	Gestão do saldo de energia com base no consumo
Consumidor passivo	Consumidor activo

Tabela 2.1: Tabela comparativa Smart Grid vs Rede Tradicional (CSS et al. 2015)

2.1.2 Microgrid

As **Microgrids** são geralmente consideradas sistemas de gestão de energia que possuem capacidade de produção da sua própria energia e não necessitam de se encontrar ligadas à rede pública, podendo funcionar de modo isolado. Podem também ser consideradas como "os blocos" com que se construí uma **Smart Grid**. Enquanto que a **Smart Grid** é uma evolução da rede elétrica tradicional, tendo em vista a gestão de energia a nível nacional, as **Microgrids** tiram proveito de fontes de energia renováveis para conseguirem alcançar independência energética. Um exemplo de uma **Microgrid** seria um edifício ou grupo de edifícios geograficamente próximos equipados com painéis solares e um gerador a diesel de reserva munidos de sistemas de controlo inteligentes ligados a controladores de carga.

Algumas das características das **Microgrids** são:

- Inteligente – capacidade de deteção das condições do sistema e reconfigurar operações de dispositivos e topologia do sistema tendo em conta fiabilidade de fornecimento, custos de operação e minimização de emissões.

- Eficiente – baseadas majoritariamente em fontes de energia limpas e renováveis e cargas inteligentes de um ponto de vista de eficiência energética
- Resiliente – capaz de reconfiguração de dispositivos para evitar falhas e colapso total do sistema. Capaz de garantir qualidade de fornecimento quando há problemas na rede geral
- Dinâmica – em constante mudança para garantir os objetivos anteriores
- Integração de cargas – integração inteligente de cargas locais com geração, correspondendo oferta e procura maximizando a utilização da infraestrutura.
- Flexível – introdução fácil de novas cargas e geradores, e de adoção de novas tecnologias de comunicação sem requerer mudanças significativas na infraestrutura

Uma das vantagens da capacidade de isolamento das microgrids, para além de resistir a falhas que possam ocorrer na rede, é a sua possível intervenção na recuperação dessas mesmas falhas, através do fornecimento de energia para garantia do serviço, por exemplo (Sioshansi 2011).

2.1.3 O problema de períodos de pico

Os períodos de pico de consumo de energia surgem no sistema como consequência do comportamento coletivo dos consumidores, influenciado por inúmeros fatores externos. Um exemplo de um comportamento em conjunto pode acontecer quando um grupo relativamente grande de consumidores liga o ar condicionado em casa como resultado de temperaturas altas. Neste exemplo o comportamento em conjunto é fácil de prever uma vez que o fator da temperatura afeta a população num todo o que pode levar a um período de pico. No entanto, existem outros fatores que suscetíveis de influenciar o consumo e, assim sendo, não é simples prever o nível de consumo e, por sua vez, picos de consumo com antecedência.

Picos de consumo podem causar grandes dificuldades aos fornecedores de eletricidade, pois causa a necessidade de sobre dimensionamento da rede de forma a suportar a carga alta de consumo. Em casos mais excessivos, em que a produção é inferior ao consumo, os operadores de rede vêm-se obrigados a trazer online ativos de geração denominadas estações "peaker", para garantir fiabilidade e produção necessária para cobrir a procura no período de pico. A gestão destes picos de consumo é crucial para os fornecedores, uma vez que a escassez de energia pode levar a consequências graves, como por exemplo cortes de energia que podem levar à desativação de sistemas vitais (p.e. um hospital), bem como afetar os preços da energia de forma irregular e radical (o preço da eletricidade aumenta durante período de pico).

Ao tornar uma rede que responde em tempo-real uma realidade, a **Smart Grid** possibilita reduzir os custos altos de responder à procura em períodos de pico. Oferece aos operadores da rede maior visibilidade do sistema a um nível de detalhe mais alto, permitindo que possam controlar cargas de forma a minimizar a necessidade de capacidades de pico altas. Para além de reduzir custos, pode ainda eliminar a necessidade de estações "peaker" completamente (Gajowniczek, Nafkha e Zabkowski 2017) (United States Department of Energy 2008).

2.1.4 Demand Response

Uma vez conhecidas as capacidades da **Smart Grid**, surge a possibilidade de tomar medidas que possam reduzir o impacto e a frequência dos períodos de pico, nomeadamente através da *Demand Response*.

Demand Response refere-se a um conjunto de estratégias que podem ser utilizadas nos mercados competitivos de eletricidade de forma a aumentar a participação dos consumidores em estabelecer preços e aliviar o mercado. Quando os clientes são expostos a preços em tempo-real, estes podem responder ao alterar a altura do dia em que procuram energia para uma com menor *peak*, ou reduzir a quantidade total de energia que necessitam empregando medidas de eficiência energética ou auto-geração. Alternativamente existe sempre a hipótese de escolherem não tomar nenhuma medida e pagar o preço de mercado da eletricidade. Na medida em que os consumidores respondem, o perfil de procura vai estabilizando, que por sua vez vai afetar os preços, diminuindo *peaks* significativamente e baixando os preços. O efeito final de *Demand Response* é aliviar restrições do sistema e gerar segurança e vantagens económicas para o mercado na sua totalidade.

A gestão de um sistema de eletricidade seguro e fiável depende de corresponder o fornecimento disponível com a procura a qualquer altura. Uma vez haver variações na procura de eletricidade, sejam elas sazonais, meteorológicas ou outras, e visto que eletricidade não é facilmente armazenada em grandes quantidades, é necessário planejar a disponibilidade de fornecimento de acordo com a previsão mais alta, acrescentando de uma margem de erro. Caso contrário, interrupções do serviço resultando em apagões seriam algo comum e causariam consideráveis danos económicos.

Nos mercados de eletricidade verticalmente integrados os recursos são monopolizados por um fornecedor que detém a responsabilidade de garantir que a capacidade de geração disponível é adequada. Os preços são geralmente regulamentados e por esse motivo não desempenham nenhum papel ao sinalizar escassez de eletricidade. Em contraste, nos mercados liberalizados, o equilíbrio de procura e oferta é realizado em tempo-real, normalmente através de um mercado de eletricidade onde a informação sobre o saldo é indicada pelos preços da eletricidade. As políticas do mercado de eletricidade liberalizado foram introduzidas com o intuito de criar um setor de eletricidade fiável e economicamente eficiente.

De uma forma geral, os preços eficientes de mercado são estabelecidos por interações entre o lado da oferta (vendedores) e o lado da procura (consumidores). No entanto, nos mercados de eletricidade liberalizados, quase todos os consumidores vêem-se expostos a preços fixos durante longos períodos de tempo independentemente do saldo entre oferta e procura no mercado. Sob estas condições, os consumidores não têm qualquer incentivo para variar os seus hábitos de consumo em resposta às condições que de fato se verificam no mercado, nem conseguem demonstrar a sua resposta face à avaliação do fornecimento de energia em tempo-real. Isto pode resultar em volatilidade excessiva de preços.

A procura de eletricidade, e o custo do seu fornecimento, podem variar substancialmente de hora para hora, resultando em variações de preços exponenciais. É considerado normal uma certa volatilidade de preços resultante do equilíbrio procura-oferta em tempo-real da eletricidade. No entanto, volatilidade excessiva pode ser criada ou agravada por restrições de capacidade, interrupções planeadas ou não-planeadas, transmissões inibidas, *peaks* de procura e potencialmente o exercício de poder de mercado de produtores ou comerciantes. A operação convencional da rede tem-se focado maioritariamente em soluções do lado do

fornecimento, por exemplo aumento de geração e infraestrutura de transmissão, de modo a restringir preços ineficientes.

Mercados para *Demand Response* em sistemas liberalizados podem operar de duas maneiras:

- Gerido por sistema, onde um operador de sistema, ou agregador de serviço ou agente, sinaliza os consumidores de que existe necessidade de redução ou mudança de carga. Preços são estabelecidos pelo mercado ou operador de sistema (mercados grossistas).
- Gerido por mercado, onde o consumidor responde diretamente ao preço de mercado, causando mudanças de comportamento e consumo constantes. Preços são estabelecidos por mecanismos de mercado (interações entre mercados grossistas e retalhistas).

Como nota final é de notar a diferença com um termo relacionado, sendo este *Demand Side Management* (DSM). Enquanto que *Demand Response* refere-se ao uso de preços baseados no mercado para influenciar o momento e nível de procura, *Demand Side Management* refere-se ao um conjunto de medidas com o intuito de aumentar a eficiência no lado do consumidor e/ou ajustar a carga de peak, mas normalmente não são utilizadas estratégias de preços baseadas no mercado para as realizar. As duas abordagens não conflitem uma com a outra, aliás o aumento de *Demand Response* pode estabelecer incentivos económicos que podem apoiar a implementação de atividades de *Demand Side Management*, como isolamento doméstico, iluminação energeticamente eficiente e programas de conservação (Energy Agency 2003) (Ul Hassan et al. 2015).

2.1.5 Desenvolvimento

De seguida listam-se alguns dos progressos ainda em desenvolvimento que contribuem diretamente para que a implementação da **Smart Grid** se torne uma realidade:

- Zero-net energy buildings/ Nearly Zero-net energy buildings:
Edifícios equipados com fontes de energia renovável e equipados com tecnologias de **Smart Grid** são capazes de equilibrar o consumo e a produção do edifício de forma a que a energia consumida é igual ou aproximadamente igual à energia produzida.
- Cabos de alimentação super-condutores:
Capazes de reduzir perdas e de transportar 3 a 5 vezes mais energia de forma mais eficiente do que os cabos de cobre tradicionais.
- Armazenamento de energia:
Apesar de eletricidade não poder ser economicamente armazenada, com a utilização de tecnologias, a energia pode. Armazenamento térmico, às vezes conhecido como ar condicionado híbrido, apresenta potencial para afetar de forma positiva as cargas de *peak*.
- Sensores avançados:
Monitorizar e reportar condições de linhas de energia em tempo-real permite a circulação de mais energia pelas linhas já existentes (United States Department of Energy 2008).

2.1.6 Previsão de Consumos

Load forecasting (previsão de carga) é uma ferramenta importante para garantir que a energia fornecida pelas *utilities* é suficiente para cobrir a carga da procura e a energia perdida pelo sistema (Soliman e Al-Kandari 2010).

Load forecasting, neste contexto *electric load forecasting*, é por definição “uma estimativa realista da procura de energia, normalmente através de uma série temporal (num determinado período de tempo) (Matijaš 2013).

Esta pode ser dividida em três categorias, nomeadamente:

1. Long-term forecasting (previsão a longo termo), é utilizado para previsão de cargas a longo prazo, nomeadamente até 50 anos no futuro para que seja possível planear expansão.
2. Medium-term forecasting (previsão a médio termo), é utilizado para previsão de cargas de pico semanal, mensal ou anualmente até 10 anos de forma a poder-se planear operações eficientemente.
3. Short-term forecasting (previsão a curto termo), é utilizado para previsão de cargas até uma semana em avanço como medida de minimizar custos diários de execução e expedição (Soliman e Al-Kandari 2010).

Long-term load forecasting (LTLF)

O LTLF tem como principal vantagem permitir o planeamento estratégico da expansão do sistema elétrico no futuro. Oferece a possibilidade de avaliar a evolução do consumo de energia de forma a desenhar um plano eficiente de construção ou aquisição de novas unidades de produção de energia ou mesmo a desativação das mesmas. Permite visualizar a um alto nível a infraestrutura atual e perceber se será adequada para responder às necessidades dos consumidores no futuro. Um exemplo seria a adoção de energias renováveis cada vez mais prevalecente que introduz não só novas fontes de energia, mas também um fator de incerteza quando consumidores passam a ser também produtores.

Medium-term load forecasting (MTLF)

O MTLF apresenta potencial para um melhor planeamento da expansão de centrais elétricas, operações de manutenção, realizar melhorias nos sistemas ou ainda planear estratégias de aquisição de combustíveis, por exemplo. Ao realizar-se uma previsão do consumo de energia para um determinado dia e seu resultado indicar que o consumo irá ser mais reduzido que o normal, poder-se-á optar por reduzir a quantidade de energia produzida para esse dia. Deduzindo-se que o número de pessoas a usufruir do serviço é reduzido, existe ainda a possibilidade de efetuar operações ou intervenções na rede, reduzindo o número de pessoas que possam ser afetadas.

Short-term load forecasting (STLF)

O STLF é essencial na gestão de produção e fornecimento de energia de maneira eficaz. As previsões de curto prazo são utilizadas como inputs de algoritmos de produção e fornecimento tentando minimizar custos bem como gerir a entrega de energia de forma segura e eficaz, mesmo ocorrendo falhas ou comportamentos inesperados por parte dos consumidores. Este período temporal é também importante quando estamos perante mercados de energia em que previsões de curto prazo nos permitem negociar as compras e vendas de energia em função das necessidades ou excessos de produção. Uma previsão adequada permite prevenir

sobrecarga na rede reduzindo a ocorrência de falhas, uma vez que podemos tomar ações preventivas de modo a reduzir o impacto que estas possam ter no processo de fornecimento de energia.

Grande parte das metodologias e modelos de previsão que se encontram disponíveis focam-se na previsão de consumos ao nível de uma sub-rede (ao nível de uma central ou transformador). A este nível é impossível identificar perfis de consumo de clientes individuais. Por consequente torna-se também impossível de efetuar uma previsão de cada cliente de modo a aumentar a eficiência na faturação, em campanhas, na segurança de fornecimento, entre outros serviços ligados diretamente ao cliente. Esta metodologia não utiliza a potencialidade dos *smart meters* (medidores digitais que enviam os dados de consumo digitalmente aos fornecedores de energia) pois não analisa os dados ao nível de granularidade mais alto, impossibilitando assim efetuar previsões a todos os níveis de abstração da rede de distribuição – região, sub-rede, cliente individual (Lima 2015). É de notar ainda que o consumo de energia é afetado por diversos fatores e variáveis que são por vezes de difícil identificação e utilização. O perfil dos consumidores varia com a altura do dia, condições climáticas, estação do ano, dia da semana, feriados, inúmeras ocasiões sociais, entre outros, tornando-se necessário a utilização de métodos capazes de utilizar estes fatores, bem como o histórico de consumo, para efetuar previsões com elevado nível de precisão. De forma a conseguir-se atingir esse objetivo introduz-se a área de **Machine Learning**.

2.2 Machine Learning

Encontramo-nos atualmente na era da Big Data, isto é a cada momento quantidades enormes de dados são produzidos por todo o mundo, muito para além do que um humano é capaz de processar. Por exemplo, existem mais de um trilião de páginas web, a cada segundo é carregado uma hora de vídeo no Youtube resultando em 10 anos de conteúdo por dia, o Walmart processa mais de 1 milhão de transações por hora e possui bases de dados que contêm mais de 2.5 petabytes de informação, entre outros.

Resultante desta torrente de informação constante surgem métodos de análise de dados (data analysis), que é o que **Machine Learning** fornece. Em termos práticos, **Machine Learning** é definido como um conjunto de métodos que conseguem automaticamente detetar padrões em dados, e por consequência utilizar os padrões descobertos para prever o futuro ou para auxiliar o processo de tomada de decisão quando existe incerteza (Murphy s.d.).

Um algoritmo de **Machine Learning** é um algoritmo que é capaz de aprender com dados. Em (Mitchell 1997) define aprendizagem de uma forma sucinta como: “Um programa de computador aprende com experiência E, face a uma classe de tarefas T, e medida de desempenho P, se a sua performance nas tarefas T, medidas por P, melhora com a experiência E”. A experiência E, as tarefas T e o desempenho P podem tomar uma grande variedade de valores que vai depender do contexto e do âmbito do algoritmo em questão. **Machine Learning** permite-nos enfrentar tarefas que são demasiados difíceis ou complexas de resolver através de mais tradicionais. Neste contexto a tarefa não é o processo de aprendizagem em si, mas sim obter a capacidade de executar a tarefa. Por exemplo, se se quiser que um robot seja capaz de andar, então a tarefa é andar e a aprendizagem é o método para o *robot* conseguir andar.

2.2.1 Resolução de tarefas

Existem diversas tarefas que podem ser resolvidas com **Machine Learning**. Algumas das mais comuns incluem:

- Classificação

Neste tipo de tarefa, o programa é solicitado especificar qual de k categorias um certo input pertence. Um exemplo de classificação é reconhecimento de objetos, onde o input é uma imagem (normalmente representada por um conjunto de valores de brilho dos pixels), e o output é um código numérico que identifica o objeto na imagem.

- Regressão

Neste tipo de tarefa, o programa é solicitado prever um valor numérico dado um certo input. Este tipo de tarefa é semelhante à classificação, exceto que o formato do output é diferente. Um exemplo de regressão é precisamente o tema da presente tese, onde os dados de consumo são utilizados como input e o output esperado é a previsão de do consumo para um determinado período de tempo.

- Transcrição

Neste tipo de tarefa, o sistema de **Machine Learning** é solicitado que observe um representação não-estruturada de dados e transcreva a informação em forma textual. Por exemplo, em reconhecimento ótico de caracteres, o programa recebe como input uma imagem com texto e é pedido que retorne o texto em forma de uma sequência de caracteres (ex: ASCII ou Unicode). Outro exemplo é reconhecimento de fala, onde o input é uma forma de onda de áudio e o output é uma sequência de caracteres ou códigos ID de palavras que descrevem as palavras faladas no áudio.

- Tradução automática

Nesta tarefa o input já consiste numa sequência de símbolos numa certa língua, e o programa converte-o numa sequência de símbolos noutra língua. É frequentemente aplicado em linguagens naturais, como por exemplo traduzir de Português para Inglês.

- Estruturação de output

Este tipo de tarefa envolve qualquer tarefa onde o output é um vetor (ou outra estrutura de dados com múltiplos valores) com relações importantes entre diferentes elementos. Esta é uma categoria ampla e agrupa as tarefas de transcrição e tradução acima referidas, bem como muitas outras tarefas. Um exemplo comum é análise, seja ela ao mapear uma frase numa árvore que descreve a sua estrutura gramatical ao marcar nós da árvore como verbos, nomes, advérbios, etc, ou segmentação de imagens por pixel, onde o programa atribui cada pixel da imagem a uma categoria específica.

- Detecção de anomalias

Neste tipo de tarefa o programa examina um conjunto de eventos o objetos e marca alguns deles como sendo invulgares ou irregulares. Um exemplo de deteção de anomalias é deteção de fraude de cartões de crédito. Ao modelar o perfil de compras de um cliente, as empresas de cartões de crédito conseguem detetar utilização atípica do cartão. Se um ladrão roubar um cartão de crédito, as compras que este fizer virão de uma distribuição de probabilidade diferente do que a do proprietário. Ao detetar esta anomalia a empresa pode congelar a conta evitando abuso contínuo.

- Síntese e amostragem

Neste tipo de tarefa, o algoritmo é solicitado gere novos exemplos que são semelhantes aos que se encontram nos dados de treino. Síntese e amostragem por **Machine Learning** pode ser útil em aplicações media onde gerar grandes quantidades de conteúdo manualmente pose-se tornar dispendioso, entediante ou exigir demasiado tempo. Por exemplo, jogos de vídeo podem gerar texturas para objetos de grandes dimensões ou para paisagens, em vez de ser necessário um artista manualmente cria-las.

- Dedução de valores em falta

Neste tipo de tarefa, é dado ao algoritmo um novo conjunto de exemplos, mas com algumas entradas em falta. O algoritmo deverá fornecer uma previsão dos valores das entradas em falta.

- Redução de ruído

Neste tipo de tarefa, o algoritmo recebe como input um exemplo corrompido obtido através de um processo de corrupção desconhecido aplicado a exemplos íntegros. O programa deverá prever o exemplo íntegro através da versão corrompida (Goodfellow, Bengio e Courville 2016)

2.2.2 Tipos de Aprendizagem

- Aprendizagem Supervisionada - Os algoritmos de aprendizagem supervisionada são treinados utilizando exemplos legendados, onde para um *input* é conhecido o output desejado. Por exemplo, existir pontos de um processo que estão legendados como “Sucesso” ou “Insucesso”. O algoritmo de aprendizagem recebe um grupo de *inputs* acompanhados dos *outputs* corretos correspondentes, aprendendo através da comparação entre o *output* resultante e o esperado, procurando erros. Terminado o processo, o modelo é ajustado de forma a aumentar a probabilidade de sucesso. O objetivo esperado é o modelo ser capaz de classificar corretamente novos dados não-legendados.
- Aprendizagem Não-Supervisionada - A metodologia de aprendizagem não-supervisionada é utilizada em dados que não se encontram classificados (sem legenda). Não é explicitada a solução correta, é esperado que o algoritmo a deduza. O objetivo é explorar dados de forma a encontrar potenciais estruturas ou padrões neles. Esta metodologia funciona particularmente bem em dados transacionais, por exemplo identificar segmentos de clientes com características semelhantes. Estes algoritmos são também utilizados para segmentar texto em tópicos, fazer recomendações e identificar *outliers* em dados. Portanto, a diferença na aprendizagem não-supervisionada é não existir diferença entre os dados de treino e os dados de teste, espera-se que o modelo seja capaz de inferir algum padrão dos dados de treino e aplica-lo para conseguir corretamente classificar os dados de teste. (Ben-David e Shalev-Shwartz 2014) (LeCun, Bengio e Hinton 2015)

Uma vez que a aquisição de *datasets* com quantidade e qualidade de dados suficientes para utilização no processo de aprendizagem é difícil e dispendiosa, existem ainda outros tipos de aprendizagem menos utilizados.

- Aprendizagem Semi-Supervisionada - é utilizada no mesmo tipo de aplicações que a aprendizagem supervisionada. No entanto utiliza dados legendados e não-legendados. Isto ocorre do facto que dados legendados serem mais difíceis e caros de adquirir,

utilizando-se assim um conjunto pequeno de dados legendados e um conjunto de maior dimensão de dados não legendados (Olivier, Schölkopf e Zien 2006).

- Aprendizagem de Reforço - é frequentemente utilizada em robótica, jogos e navegação. Com este tipo de aprendizagem, o algoritmo aprende através de tentativa/erro quais as ações que resultam em melhores resultados. O objetivo é que o agente escolha as ações que maximizem a recompensa esperada durante um período de tempo (Sutton e Barto 2012).

2.2.3 Algoritmos

Nesta secção listam-se alguns dos algoritmos mais populares de **Machine Learning**

1. Linear Regression – é utilizado para estimar valores reais com base em variáveis contínuas
2. Logistic Regression – na realidade é um algoritmo de classificação, é utilizado para estimar valores discretos com base num conjunto de variáveis independentes
3. Decision Tree – normalmente utilizado em problemas de classificação, a população é dividida em dois ou mais conjuntos homogêneos com base nos atributos mais importantes
4. SVM (Support Vector Machine) – método de classificação, cada atributo é mapeado num vetor num espaço n-dimensional (onde n é o número de atributos). É traçado um espaço, cuja dimensão varia com o número de atributos, que divide os dados em grupos.
5. Naive Bayes – método de classificação baseado no teorema de Bayes. O classificador Naive Bayes assume que a presença de um certo atributo numa classe é independente da presença de qualquer outro atributo.
6. KNN (K- Nearest Neighbors) – pode ser utilizado para classificação e regressão. É um algoritmo simples que guarda todos os casos disponíveis e classifica-os conforme um voto de maioria pelos seus K vizinhos mais próximos.
7. K-Means – é um tipo de algoritmo não-supervisionado para problemas de *clustering*. Um *dataset* é classificado através de K *clusters* que vão sendo construídos conforme os seus elementos.
8. Random Forest – termo utilizado para descrever um conjunto de árvores de decisão. Para classificar um novo objeto baseado em atributos cada árvore fornece a sua classificação e “vota” nessa classe. A classe com mais votos é escolhida.
9. Dimensionality Reduction Algorithms – utilizado para classificação, nomeadamente para escolher quais variáveis são importantes quando se quer, por exemplo, segmentar grupos de clientes.
10. GBM (Gradient Boosting Algorithms) – é utilizado para lidar com grandes quantidades de dados e obter previsões com grande precisão. Utiliza um conjunto de algoritmos de aprendizagem mais fracos em conjunção para construir um algoritmo de previsão mais robusto (Analytics 2018).

2.2.4 Ferramentas

De seguida apresenta-se uma lista de algumas plataformas/*frameworks* de **Machine Learning** existentes:

- Amazon Machine Learning
- Azure ML Studio
- H2O
- Massive Online Analysis (MOA)
- MLlib (Spark)
- Mlpack
- Pattern
- Scikit-Learn
- Shogun
- Torch
- Accord.net Framework
- Apache Mahout
- Cloudera Oryx
- GoLearn
- Weka
- Microsoft Distributed Machine Learning Toolkit
- Microsoft Computational Network Toolkit
- Marvin

Algumas das ferramentas listadas possuem também suporte para técnicas de **Deep Learning** (KDNuggets 2018) (InfoWorld 2018) (InfoWorld_b 2018).

Para além das ferramentas listada acima existem ainda bibliotecas/*packages* para diferentes linguagens de programação que também são frequentemente utilizadas. Algumas das mais populares são:

- R (KDNuggets_R 2018)
 - e1071
 - rpart
 - igraph
 - nnet
 - randomForest
 - caret
- Matlab (MathWorks 2018)

- Statistics and Machine Learning Toolbox
- Financial Toolbox
- Risk Management Toolbox
- Neural Network Toolbox
- Computer Vision System Toolbox
- Python (TowardsDataScience 2018)
 - Tensorflow
 - Keras
 - Theano

2.3 Deep Learning

As técnicas de **Machine Learning** tradicionais eram limitadas na sua capacidade de processar dados naturais na sua forma não tratada. Durante décadas, construir um sistema de reconhecimento de padrões ou de **Machine Learning** requeria engenharia cuidadosa e um conhecimento do domínio considerável para se conseguir construir um modelo capaz de extrair representações adequadas ou vetores de características, que fossem posteriormente analisados, frequentemente por um classificador, de forma a detetar ou classificar padrões no *input* (LeCun, Bengio e Hinton 2015).

O **Deep Learning** moderno fornece uma *framework* poderosa para aprendizagem supervisionada. Ao acrescentar mais camadas ocultas e mais unidades em cada camada, uma rede “deep” consegue representar funções de complexidade cada vez mais elevada. A maioria das tarefas que consistem em mapear um vetor de input para um vetor de output, e que são de fácil resolução para um ser humano, pode ser realizado via **Deep Learning**, dados modelos suficientemente grandes munidos de *datasets* suficientemente grandes com exemplos legendados. Fundamentalmente, o propósito dos métodos **Deep Learning** é conseguir fornecer resposta para problemas demasiado complexos para seres humanos resolverem, sem ser necessário a intervenção dos mesmos na sua resolução (Goodfellow, Bengio e Courville 2016).

Tradicionalmente os modelos de **Machine Learning** tinham uma arquitetura simples de duas camadas. No entanto, se se analisar o cérebro pode-se verificar que existe vários níveis de processamento, onde cada nível aprende características ou representações a níveis crescentes de abstração. Os modelos **Deep Learning** adotam esta estrutura ao inovar sobre as redes neuronais tradicionais ao incorporar múltiplas camadas de processamento ocultas onde cada nó de cada camada possui várias ligações com os nós da camada seguinte. Um exemplo deste tipo de modelo é o córtex visual, onde na primeira camada o cérebro extrai arestas, seguido de manchas, superfícies, objetos, etc, (Murphy s.d.). Hoje em dia pode-se considerar uma rede neuronal como “superficial” se tiver mais do que duas ou dez camadas, sendo consideradas as redes “deep” as que têm na ordem das centenas (ou mais) de camadas.

2.3.1 Modelos

Os modelos “deep” têm milhões de parâmetros. Obter dados legendados suficientes para treinar estes modelos é difícil, apesar da era da Big Data fornecer cada vez mais *datasets*,

alguns deles públicos de acesso livre, nem sempre os dados disponíveis são os adequados para treino para uma determinada tarefa. Para ultrapassar esta limitação surge um foco em aprendizagem não-supervisionada. De seguida apresenta-se exemplos de alguns modelos de **Deep Learning** mais frequentemente utilizados:

- Deep Feedforward networks

Deep feedforward networks, também denominadas de multilayer perceptrons (MLPs) são o modelo de **Deep Learning** de referência. São assim chamadas uma vez a informação flui em apenas um sentido, não existe feedback onde os outputs do modelo são reintroduzidos em si próprio (Goodfellow, Bengio e Courville 2016) (Murphy s.d.) (Gulli e Pal s.d.).

- Convolutional Networks

Convolutional Networks são um tipo específico de rede neuronal especializadas em processamento de dados com uma topologia em grelha. Por exemplo, análise de séries temporais, que podem ser consideradas grelhas de uma dimensão com ao longo de vários intervalos de tempo, ou análise de imagens, que podem ser consideradas grelhas de duas dimensões de pixels (Goodfellow, Bengio e Courville 2016) (Murphy s.d.) (Gulli e Pal s.d.).

- Recurrent Neural Networks

Recurrent Neural Networks são um tipo de redes neuronais para processamento de dados sequenciais. Da mesma forma que as Convolutional Networks conseguem escalar para imagens com largura e altura amplas e por vezes variáveis, as Recurrent Networks conseguem escalar para sequências bastante mais longas do que seria prático para redes neuronais sem especialização para sequências (Goodfellow, Bengio e Courville 2016) (Murphy s.d.) (Gulli e Pal s.d.).

- Stacked Denoising Autoencoders

Conjunto de denoising autoencoders em pilha, permite especificar que a corrupção é apenas utilizada para o treino de redução de ruído inicial para cada camada, permitindo que aprenda extratores de características úteis. A função resultante é de seguida utilizada em dados limpos para aprender uma nova função. Esta nova função será utilizada para treinar um encoder de segunda camada e o processo repete-se (Vincent PASCALVINCENT e Larochelle LAROCHEH 2010).

- Deep Belief Networks Deep Belief Networks funcionam de maneira semelhante aos Stacked Denoising Autoencoders, no entanto, em vez de uma pilha de denoising encoders, trata-se de uma pilha de Boltzmann Machines (Goodfellow, Bengio e Courville 2016) (Murphy s.d.) (Gulli e Pal s.d.).

- Deep Boltzmann Machines

Deep Boltzmann Machines apresentam um funcionamento muito semelhante às Deep Belief Networks, uma vez serem compostas por uma pilha de Restricted Boltzmann Machines. No entanto, ao contrário das Deep Belief Networks, para além de um processo de inferência bottom-up, é capaz de incorporar feedback top-down permitindo lidar de forma mais robusta com inputs ambíguos (Salakhutdinov e Hinton s.d.).

Existem inúmeros modelos e arquiteturas de **Deep Learning**. Muitas delas são semelhantes uma vez terem por base, de forma geral, as redes neuronais e inovarem sobre estas de forma

a conseguirem atingir o objetivo pretendido. Ainda que sejam semelhantes, dois modelos podem ter tarefas bastante distintas.

2.3.2 Áreas de Aplicação

Deep Learning tem variadas aplicações práticas que já se encontram em uso e algumas que se encontram em desenvolvimento e se espera que se concretizem brevemente. Alguns exemplos são listados de seguida:

- Reconhecimento de fala

Tanto a nível académico como empresarial, a adoção de Deep Learning para reconhecimento de fala tem-se verificado. Xbox, Skype, Google Now e a Siri são alguns exemplos onde tecnologias de Deep Learning foram empregadas nos seus sistemas para reconhecer fala humana e padrões de voz.

- Reconhecimento de imagem

Uma aplicação prática de reconhecimento de imagem é a legendagem automática de descrição de cenário. Pode vir a ter uma contribuição crucial em investigações criminais para identificação de atividades criminais em fotografias submetidas por espetadores numa área com grande densidade populacional. Self-driving cars também podem beneficiar de reconhecimento de imagem através do uso de câmara de 360 graus.

- Processamento de linguagem natural

As redes neuronais têm sido utilizadas para processar e analisar texto escrito há vários anos. Esta técnica, que é uma especialização de text mining, pode ser utilizada para descobrir padrões em queixas de clientes, notas de médicos, reportagens de notícias, entre outros.

- Sistemas de recomendação

A Amazon e o Netflix são dois exemplos de serviços que oferecem um sistema de recomendação com base em experiência prévia. O Deep Learning pode ser utilizado para melhorar as recomendações em ambientes complexos, por exemplo, gostos musicais ou preferências de vestuário sobre várias plataformas.

- Tradução automática

As técnicas de Deep Learning têm vindo a melhorar a tarefa de tradução automática de palavras ou frases numa linguagem para outra linguagem, tanto em texto como imagens.

- *Self-driving cars*

O Deep Learning inova os serviços de assistência ao condutor transformando-os num serviço completamente automático de condução. Projeto liderado maioritariamente pela Google e pode incluir várias contribuições de diferentes tipos de Deep Learning.

- Saúde

Várias implementações na área da saúde têm-se vindo a revelar, nomeadamente diagnóstico de doenças, aplicações para monitorização remota, previsão e personalização de medicamentos, entre outros (Medium 2018).

2.3.3 Contributos Recentes

O **Deep Learning** é uma área que tem vindo a ter um grande crescimento, havendo a cada momento novas contribuições a serem publicadas e divulgadas. Algumas das implementações mais recentes de **Deep Learning** são as seguintes:

- Extração automatizada de informação pertinente a partir de relatórios patológicos de cancro (Qiu et al. 2018)
- Reconhecimento de assinaturas escritas à mão (Tolosana et al. 2018)
- Reconhecimento de emoções humanas (Zheng et al. 2018)
- Reconstrução de caras em fotografias (Zhang et al. 2018)
- Representação de biométricas cardíacas (da Silva Luz et al. 2018)

As seguintes implementações dizem respeito especificamente à tarefa de *forecasting*

- Em (Kong et al. 2018) é realizada a previsão de consumo a curto-termo baseado em aprendizagem do perfil dos residentes. É utilizada uma estrutura baseada em Recurrent Neural Networks.
- Em (Wang et al. 2018) é realizada previsão de cibercrimes no âmbito das **Smart Grids**. São utilizados Stacked AutoEncoders para realizar estimativas do estado das **Smart Grids** em determinados intervalos.
- Em (Chen et al. 2018) é realizada a previsão de perfis de geração de energia eólica e fotovoltaica. São utilizados métodos inovadores onde são conectadas duas redes neuronais *deep* baseadas em Convolutional Neural Networks.
- Em (Lee, Cisek e Yoo 2017) é realizada a previsão de perfil de vento e movimentação de nuvens para auxílio de previsão de irradiação solar. São utilizadas Convolutional Neural Networks apoiadas de algoritmo geométricos computacionais.
- em (Khare et al. 2017) é realizada a previsão de preços de ações a curto-termo. São utilizadas Feedforward Neural Networks e Recurrent Neural Networks.

2.3.4 Ferramentas

Nesta última secção é apresentada uma lista extensiva de ferramentas/*frameworks* de **Deep Learning** (DeepLearningNet 2018).

1. Theano – compilador de expressões simbólicas a nível de em python
2. Torch – ambiente para algoritmos de Machine Learning em lua
3. Pylearn2 – biblioteca projetada para facilitar investigação de Machine Learning
4. Blocks – *framework* Theano para treino de redes neuronais
5. Tensorflow – biblioteca *open-source* para computação numérica utilizando grafos de fluxo
6. MXNet – *framework* Deep Learning contruída para eficiência e flexibilidade
7. Caffe - *framework* Deep Learning com expressão, velocidade e modularidade em mente.
8. Lasagne – biblioteca leve para construir e treinar redes neuronais em Theano

9. Keras – biblioteca Deep Learning baseada e Theano
10. Deep Learning Tutorials – exemplos de como utilizar Deep Learning com Theano
11. Chainer – *framework* de redes neuronais baseada em GPU
12. Matlab Deep Learning – ferramentas de Deep Learning em Matlab
13. CNTK – Computational Network Toolkit – *toolkit* unificado de Deep Learning pela Microsoft Research.
14. MatConvNet – *toolbox* em Matlab para implementação de Convolutional Neural Networks (CNNs) para processamento de imagem
15. DeepLearnToolbox – *toolbox* em Matlab para Deep Learning
16. BigDL - biblioteca Deep Learning *open-source* distribuída para Apache Spark para escalamento eficiente para vários nós
17. Cuda-Convnet – implementação rápida em C++/CUDA de Convolutional (ou no geral Feedforward) Neural Networks
18. Deep Belief Networks - código Matlab para aprendizagem de Deep Belief Networks
19. RNNLM – toolkit de Tomas Mikolov para modelos de linguagem baseado em Recurrent Neural Network
20. RNNLIB - biblioteca de Recurrent Neural Networks para problemas de aprendizagem de sequências.
21. matrbm - versão simplificada do código de Ruslan Salakhutdinov por Andrej Karpathy (Matlab).
22. deeplearning4j – biblioteca distribuída de redes neuronais em Java e Scala para Apache 2.0
23. Estimating Partition Functions of RBM's - código Matlab para estimar funções de partições de Restricted Boltzmann Machines
24. Learning Deep Boltzmann Machines - código Matlab para treino e ajuste de Deep Boltzmann Machines
25. LUSH - linguagem e ambiente de programação para Deep Convolutional Networks
26. Eblearn.lsh - biblioteca baseada em LUSH para aprendizagem baseada em energia. Inclui código para métodos de aprendizagem não-supervisionada incluindo auto-encoders
27. deepmat – algoritmos Deep Learning baseados em Matlab
28. MShadow – biblioteca leve que permite distribuição por CPU/GPU/multi-GPU. Tem foco em simplicidade e desempenho
29. CXXNET – *framework* rápida e concisa para Deep Learning distribuído em C++/CUDA com interface simples para Python/Matlab
30. Nengo-Nengo - software gráfico baseado em script para simulação de sistemas neuronais de grande escala
31. Eblearn - biblioteca C++ de Machine Learning para aprendizagem baseada em energia, Convolutional Networks, aplicações de reconhecimento visual, entre outros

32. cudamat - matriz baseada em GPU para Python. Inclui código para treino de redes neurais e Restricted Boltzmann Machines
33. Gnumpy - módulo Python que serve faz interface idêntica ao numpy, mas executa as computações localmente no GPU. Funciona por cima de cudamat
34. The CUV Library - *framework* C++ com ligações Python para fácil utilização de funções Nvidia CUDA em matrizes. Contém implementação de Restricted Boltzmann Machines
35. 3-way factored RBM e mcRBM - código Python que invoca CUDAmat para treinar modelos de imagens naturais
36. mPoT - código Python utilizando CUDAmat e gnumpy para treino de modelos de imagens naturais
37. neuralnetworks - biblioteca java baseada em GPU para algoritmos Deep Learning
38. ConvNet - *toolbox* baseada em Matlab para Convolutional Networks
39. Elektronn - *toolkit* Deep Learning que torna redes neurais poderosas acessíveis a cientistas fora da comunidade de Machine Learning
40. OpenNN - biblioteca *open-source* de classes em C++ que implementam redes neurais
41. NeuralDesigner - ferramenta Deep Learning inovadora para análise preditiva
42. Theano Generalized Hebbian Learning
43. Apache Singa - biblioteca *open-source* para Deep Learning que fornece uma arquitetura flexível para treino distribuído escalável. Foco em aplicações na área da saúde
44. Lightnet - *framework* leve e versátil de Deep Learning, puramente baseada em Matlab. Pretende oferecer um *design* fácil de entender e utilizar para investigação de Deep Learning
45. SimpleDNN - biblioteca leve *open-source* de Machine Learning escrita em Kotlin com o objetivo de apoiar o desenvolvimento de Feedforward e Recurrent Neural Networks

Capítulo 3

Análise de Valor

Neste capítulo irá ser apresentada a análise de valor so presente trabalho. São apresentados vários conceitos de valor e a forma como estes se aplicam ao trabalho bem como modelo de negócio CANVAS e proposta de valor.

3.1 The new concept development model (NCD)

Para se conceptualizar um novo produto é necessário seguir algumas fases para todo o processo de inovação e de desenvolvimento. Para tal o *New Concept Development Model* (NCD) de Peter Koen (A. Koen et al. 2002) como metodologia para cultivar ideias e conceitos. Esta metodologia fornece uma estrutura de pensamento para as atividades de partilha e criação de novo conhecimento, quando se inicia o planeamento de um projeto. Consideram-se cinco elementos do NCD: Identificação da oportunidade; Análise da oportunidade; Geração e enriquecimento de ideias; Seleção de ideias; Definição do conceito. Um projeto pode começar de duas formas: geração de ideias, fase onde são apresentadas ideias e é a partir daí que é desenvolvido um produto; ou pela identificação da oportunidade no mercado, onde é verificada uma lacuna/necessidade e é desenvolvido um produto que colmata essa lacuna.

- Identificação da oportunidade

A oportunidade foi identificada através de uma análise do estado da arte no setor de energia, nomeadamente sistemas de gestão inteligentes. Existem várias dificuldades em produzir e distribuir energia de forma eficaz sem haver perdas nem desperdícios, fatores que causam não só prejuízos monetários mas também a possibilidade de cortes de energia fatais. Até à data a rede de eletricidade pública vê-se obrigada a suportar uma infraestrutura sobre-dimensionada para poder responder à procura de energia de difícil previsão pela parte dos consumidores.

- Análise da oportunidade

Analisando o mercado, pode-se verificar a existência de diversos projetos e soluções que visam abordar/resolver o problema identificado. A tecnologia proposta (**Deep Learning**) demonstra ser promissora na resolução do problema, quando comparada com as abordagens tradicionais. Pode-se ainda verificar que não existem ainda soluções que sejam capazes de resolver o problema dentro das limitações impostas (previsões com erro o mais próximo de 0% possível, apresentar resultados em quase tempo-real).

- Geração e enriquecimento de ideias

Foi levantado um conjunto de ferramentas e plataformas disponíveis para a realização do projeto preconizado. Verificou-se a oportunidade de utilização de ferramentas *open-source* que se enquadravam no âmbito da solução proposta, tendo sido selecionadas algumas como potenciais escolhas. Estas oferecem não só algoritmos mas também ambientes para executar os mesmos.

- Seleção de ideias

Uma vez o presente projeto ser desenvolvido no centro de investigação GECAD, as ferramentas a utilizar foram escolhidas com base na cooperação com um projeto desenvolvido paralelamente que poderá servir de suporte e infraestrutura para a execução do projeto. Assim, foi selecionado de entre várias possibilidades a utilização da plataforma Spark como ambiente principal para execução de algoritmos, pois é facilmente integrada com a plataforma Big Data que foi desenvolvida paralelamente, que disponibiliza dados de consumo para análise e permite abordar a limitação do tempo de execução.

- Definição do conceito

A previsão do consumo de energia é uma área de complexa resolução, uma vez que o consumo pode ser afetado por inúmeros fatores que não são facilmente identificados. Como não se consegue prever com precisão a quantidade de energia que será necessária para satisfazer a procura da população a qualquer dado intervalo de tempo, os fornecedores/produtores de energia têm a necessidade de investir em infraestruturas com dimensão e custos desnecessariamente altos para conseguir responder a eventuais picos de consumo que possam ocorrer a qualquer momento. Assim, de forma a reduzir estes custos e evitar que ocorram picos de consumo a solução proposta deverá utilizar dados de consumo que foram recolhidos ao longo do tempo para efetuar previsões do consumo de energia muito próximas da realidade e ao distribuir a execução dos algoritmos pela plataforma Big Data disponibilizada, estas previsões deverão ocorrer o mais próximo do tempo-real possível (de modo a que os operadores da rede tenham tempo para agir).

A implementação deste projeto contribui diretamente para a integração de novas tecnologias no quotidiano da sociedade, permitindo diminuir custos e otimizar recursos. Permite ainda suportar uma revolução dos mercados de energia, dando mais poder ao consumidor, que por sua vez pode-se tornar um *prosumer* (consumidor e produtor), tornando a gestão de recursos energéticos mais dinâmica (Santos et al. 2016).

3.2 Valor

Valor é o resultado de qualquer atividade de negócio que resulte em algo que permita ser avaliado e aceite por parte dos consumidores/clientes como tendo interesse para estes (NICOLA, E. P. FERREIRA e J. J. P. FERREIRA 2012).

O valor percebido é, como o nome indica, a perceção geral da utilidade do produto frente à sua perceção do propósito do produto, contrastando o que é dado com o que é recebido, pelo ponto de vista do cliente (Zeithaml 1988).

O valor para o cliente é definido pela avaliação do produto pelo cliente, extrapolando as suas vantagens e características do seu ponto de vista. Portanto, é o contrabalanço dos ganhos e sacrifícios que um determinado cliente faz ao adquirir um produto. Isto significa que o

valor de um produto para um cliente pode diferir radicalmente para outro cliente (Woodall 2003).

De um ponto de vista da solução proposta, pode-se assumir dois segmentos de clientes diferentes. Um dos segmentos tira proveito direto da solução, enquanto que o outro apenas indiretamente. Diretamente, existe o segmento de clientes que desenvolve ou está de alguma forma envolvido em projetos ou investigações no ramo de previsões de consumo de energia. Este segmento tem como benefício a utilização da solução ao enriquecer os seus sistemas de previsão, que por sua vez irá ter impacto noutros componentes/sistemas. Tem como sacrifícios a aquisição da solução, bem como possível serviço futuro de suporte que possa ser necessário. O segundo segmento são os utilizadores dos sistemas desenvolvidos pelo primeiro segmento. Têm como benefício a diminuição de custos energéticos, otimização dos recursos associados, prevenção e deteção de falhas em tempo-real, entre outros. Têm como sacrifícios a instalação dos sistemas em si, bem como de aparelhos/eletrodomésticos capazes de interagir com os sistemas inteligentes.

3.3 Proposta de Valor

A solução proposta pretende proporcionar métodos de previsão de consumo de energia mais rápidos e eficientes, capazes de contribuir para o impulsionamento, implementação e adoção de sistemas de gestão de energia inteligentes no dia-a-dia quotidiano.

De seguida encontra-se o modelo de negócio CANVAS que apresenta visão global da estrutura do negócio.

The Business Model Canvas

Team or Company Name:
GECAD

Date:
14/12/2016

Primary Canvas
 Alternative Canvas

<p>Key Partners</p> <ul style="list-style-type: none"> • Agregadores de recursos com data-sets de grande dimensão • P.ex. Comercializadores e distribuidores de energia elétrica, • Instituições com monitorização em tempo real <p>P.ex. GECAD, e alguns dos seus parceiros</p>	<p>Key Activities</p> <ul style="list-style-type: none"> • Implementação de Algoritmos • Operações trial and error • Assistência ao cliente 	<p>Value Proposition</p> <ul style="list-style-type: none"> • Serviços de previsão de alto desempenho e precisão • Resultante da implementação: <ul style="list-style-type: none"> • Otimização de recursos • Metodologias que tirem partido da flexibilidade das redes • Otimização da produção renovável no âmbito dos <i>prosumers</i>, gestores de edifícios, gestores de microredes • etc... 	<p>Customer Relationships</p> <ul style="list-style-type: none"> • Suporte ao Cliente • Upgrade do serviço <p>Channels</p> <ul style="list-style-type: none"> • Interação presencial • Email • Website 	
<p>Cost Structure</p> <ul style="list-style-type: none"> • Custo de Recursos Humanos • Custo com eventuais licenças de software/packages • Custo de Hardware 	<p>Revenue Streams</p> <ul style="list-style-type: none"> • Serviços de Previsão • Serviço de manutenção/suporte 			
<p>Key Resources</p> <ul style="list-style-type: none"> • Programadores • Analistas de Informação • Hardware 		<p>Customer Segments</p> <ul style="list-style-type: none"> • Atores no âmbito das <i>Smart Grids</i> <ul style="list-style-type: none"> ○ Retailers ○ Prosumers ○ <i>Microgrid Managers</i> ○ Utilities, ○ etc... • Prestadores de Serviços de Previsão 		

Source: www.businessmodelgeneration.com

3.4 Cadeia e Rede de Valor

A cadeia de valor de Porter é definida pelo conjunto de atividades que uma empresa/organização executa de modo a distribuir um produto ou serviço para o mercado. Os produtos passam por uma ordem de atividades (a cadeia) onde cada atividade adiciona valor aos produtos. Por exemplo, a atividade de corte de diamantes, apesar de ter um custo baixo, a atividade em si fornece um enorme valor, uma vez que um diamante bruto tem valor bastante mais baixo que um processado. Assim, sendo que as atividades em si são importantes para atribuir valor aos produtos, então a otimização e redução de custos e tempo das atividades irá adicionar ainda mais valor. A cadeia de valor de Porter permite modelar as atividades de uma empresa como uma sequência analisando as ligações entre atividades, procurando meios de as melhorar ou otimizar. Se na análise das atividades se verificar que uma delas fica parada devido a uma outra necessitar do mesmo recurso, então pode-se concluir-se que uma possibilidade de otimização seria a disponibilização de adicionais recursos pertinentes (Porter 1985).

A rede de valor de Verna Allee é definida por qualquer rede de relações que criam tanto valor tangível como intangível, através de trocas entre indivíduos, grupos ou organizações. O valor tangível pode ser interpretado como qualquer tipo de interação sobre a qual se consegue atribuir valor numérico. Seja a venda de bens ou serviços, pagamentos, solicitações de propostas, faturas, etc. No entanto, o valor intangível é mais direcionado para o fator humano, e assenta nas interações e troca de informação entre os intervenientes no processo de negócio, que passam conhecimento e/ou benefícios entre si. O propósito de das redes de valor é criar o máximo benefício para os atores envolvidos na rede, desde fornecedores até aos produtores e finalmente aos clientes. A rede de valor difere da cadeia de valor na medida em que aceita o valor produzido pelas inter-relações dos intervenientes, não só pelas atividades por estes realizados. Enquanto que segundo a cadeia de valor o processo de negócio pode ser otimizado apenas alterando o *modus operandi* das atividades, na rede de valor a partilha de informação e conhecimento traduz-se também num acréscimo de valor para o processo. Se um colaborador não souber executar uma tarefa, é mais viável disponibilizar alguém que possa ensinar, ou oferecer formação, do que estar sempre a despedir e contratar novos colaboradores até encontrar um que saiba executar todas as tarefas (Allee 2002).

A natureza das **Smart Grids** tem a troca de informação e valor como um dos aspetos centrais e é uma das razões pela qual a sua implementação trás diversas vantagens para todos os seus atores bem como o meio envolvente. Torna possível a troca de energia entre consumidores, a gestão dos seus consumos tanto a nível residencial como a nível da rede toda e possibilita que haja a participação conjunta da sociedade na resolução de problemas que possam ocorrer no sistema (p.e. caso haja uma falha na distribuição de energia ou rede em si, permite redirecionar energia de micro-produtores para locais críticos como hospitais).

O presente projeto visa impulsionar e possibilitar algumas das características das **Smart Grids**, influenciando todos os que nela participam, assim como qual que outro contributo fará o mesmo, tendo em conta a característica da comunicação bi-direcional que liga todas as pessoas que utilizam a rede elétrica.

3.5 Método AHP

Nesta secção é apresentado um exemplo ilustrativo da utilização do método de decisão multi-critério AHP, na escolha do melhor algoritmo para previsão de consumo.

NOTA: Os valores apresentados são arredondados, no entanto os valores reais foram utilizados para efetuar os cálculos.

Árvore hierárquica de decisão para o enunciado do objetivo é apresentada na figura 3.1.

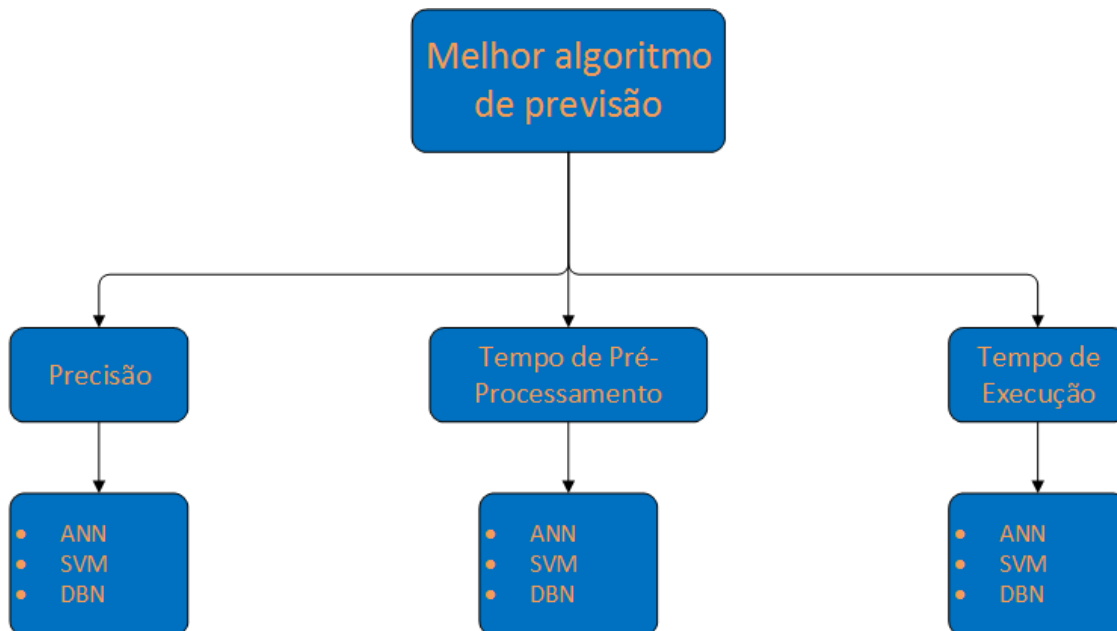


Figura 3.1: Árvore hierárquica de decisão

Na figura 3.2 é apresentada a tabela de comparação par a par entre os vários critérios apresentados na árvore hierárquica.

Critérios	Precisão	Tempo de Pré-Processamento	Tempo de Execução
Precisão	1	7	4
Tempo de Pré-Processamento	1/7	1	1/3
Tempo de Execução	1/4	3	1
Totais Coluna	1 2/5	11	5 1/3

Figura 3.2: Comparação Par a Par

Na figura 3.3 é demonstrada a prioridade relativa de cada critério (Pesos):

Critérios	Precisão	Tempo de Pré-Processamento	Tempo de Execução	PESOS
Precisão	0,7179	0,6364	0,7500	0,7014
Tempo de Pré-Processamento	0,1026	0,0909	0,0625	0,0853
Tempo de Execução	0,1795	0,2727	0,1875	0,2132

Figura 3.3: Comparação Normalizada

De seguida apresentam-se as matrizes de comparação paritárias para cada critério (Precisão, Tempo de Pré Processamento e Tempo de Execução), considerando cada uma das alternativas selecionadas (ANN, SVM e DBN).

A figura 3.4 demonstra a comparação paritária referente ao critério da Precisão.

Precisão	ANN	SVM	DBN
ANN	1	1/4	1/2
SVM	4	1	3
DBN	2	1/3	1
Total	7,0000	1,5833	4,5000

Figura 3.4: Matriz paritária para Precisão

A figura 3.5 demonstra a comparação paritária referente ao critério da Tempo de Pré-Processamento.

Tempo de Pré-Processamento	ANN	SVM	DBN
ANN	1	3	5
SVM	1/3	1	3
DBN	1/5	1/3	1
Total	1,5333	4,3333	9,0000

Figura 3.5: Matriz paritária para Tempo de Pré-Processamento

A figura 3.6 demonstra a comparação paritária referente ao critério da Tempo de Execução.

Tempo de Execução	ANN	SVM	DBN
ANN	1	2	4
SVM	1/2	1	2
DBN	1/4	1/2	1
Total	1,7500	3,5000	7,0000

Figura 3.6: Matriz paritária para Tempo de Execução

De seguida apresentam-se as matrizes de comparação paritárias normalizadas para cada alternativa.

A figura 3.7 apresenta a matriz de comparação paritária com os valores normalizados para o critério de Precisão.

Precisão	ANN	SVM	DBN	Pesos
ANN	1/7	1/6	1/9	0,1373
SVM	4/7	0,6316	0,6667	0,6232
DBN	2/7	1/5	2/9	0,2395
Total				1

Figura 3.7: Matriz paritária para Precisão Normalizada

A figura 3.8 apresenta a matriz de comparação paritária com os valores normalizados para o critério de Tempo de Pré-Processamento.

Tempo de Pré-Processamento	ANN	SVM	DBN	Pesos
ANN	2/3	2/3	5/9	0,6333
SVM	0,2174	0,2308	0,3333	0,2605
DBN	1/8	0	1/9	0,1062
Total				1

Figura 3.8: Matriz paritária para Tempo de Pré-Processamento Normalizada

A figura 3.9 apresenta a matriz de comparação paritária com os valores normalizados para o critério de Tempo de Execução.

Tempo de Execução	ANN	SVM	DBN	Pesos
ANN	4/7	4/7	4/7	0,5714
SVM	2/7	2/7	2/7	0,2857
DBN	1/7	1/7	1/7	0,1429
Total				1

Figura 3.9: Matriz paritária para Tempo de Execução Normalizada

Na seguinte figura 3.10 é apresentada a matriz com as prioridades globais para cada critério:

Previsão	Precisão	Tempo de Pré-Processamento	Tempo de execução
ANN	0,1373	0,6333	0,5714
SVM	0,6232	0,2605	0,2857
DBN	0,2395	0,1062	0,1429

Figura 3.10: Matriz com prioridades globais

Por fim é calculada a prioridade composta de cada alternativa, através da multiplicação dos valores das prioridades globais acima demonstradas, com os pesos calculados na normalização da comparação, podendo-se verificar os resultados na tabela 3.11.

ANN	0,272188923
SVM	0,520305182
DBN	0,207505895

Figura 3.11: Prioridades Compostas

Pode-se assim verificar que a alternativa SVM aparece como sendo a mais indicada, em função dos critérios definidos e respetivas importâncias.

De seguida são ainda apresentados os cálculos efetuados para o teste de consistência.

- 1º Passo - Cálculo do vetor próprio para λ_{\max}

Calculado através da multiplicação da matriz de comparação par a par com o vetor de pesos calculado na normalização, como evidenciado na figura 3.12

Matriz de comparação Par a Par				Pesos		
1	7	4	x	0,7014	=	2,15166
1/7	1	1/3	x	0,0853	=	0,25661
1/4	3	1	x	0,2132	=	0,64457

Figura 3.12: Cálculo vetor próprio

- 2º Passo - Cálculo do vetor intermédio

Os valores obtidos no passo anterior são divididos pelo respetivo valor do vetor de pesos normalizado.

$$\frac{2.15166}{0.7014} = 3.06750208$$

$$\frac{0.25661}{0.0853} = 3.00745318$$

$$\frac{0.64457}{0.2132} = 3.02277000$$

- 3º Passo - Cálculo do valor próprio (λ_{\max})

Obtido através da média dos valores obtidos no 2º Passo

$$\frac{3.06750208 + 3.00745318 + 3.02277}{3} = 3.032576371 \quad (3.1)$$

- 4º Passo - Cálculo do índice de consistência

$$IC = \frac{(\lambda_{\max} - n)}{(n - 1)} \quad (3.2)$$

Ou seja,

$$IC = \frac{(3.032576371 - 3)}{(2)} = 0.01629 \quad (3.3)$$

- 5º Passo - Cálculo da razão de consistência

$$RC = \frac{IC}{IR} \quad (3.4)$$

Onde IR é obtido da seguinte tabela, conforme o número de critérios

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Figura 3.13: Tabela de valores IR para matrizes quadradas de ordem n

Obtém-se,

$$RC = \frac{0.01629}{0.58} = 0.028083078 \quad (3.5)$$

No exemplo ilustrado $RC \leq 0.1$, logo pode-se concluir que os valores de prioridades relativas utilizados são consistentes.

Capítulo 4

Design

Neste capítulo descreve-se o design das soluções preconizadas, sendo apresentadas as arquitetura das mesmas bem como as experiências a realizar no âmbito dos casos de estudo. É ainda abordada a estrutura dos dados que são utilizados na realização dos casos de estudo.

4.1 Arquitetura adotada

A solução implementada utiliza a biblioteca *open source* de *software* de alto desempenho **TensorFlow**. O **TensorFlow** oferece um sistema de **Machine Learning** capaz de operar em grande escala e em sistemas heterogêneos. A sua arquitetura flexível permite computação distribuída com facilidade através de módulos computacionais, nomeadamente processadores *multicore* (CPUs), processadores gráfico (GPUs)s e ASICs (Application Specific Integrated Circuits) personalizados, denominados Tensor Processing Units (TPUs). Parte da sua flexibilidade advém também da liberdade dada ao programador para poder experimentar com otimizações e algoritmos de treino inovadores. O **TensorFlow** suporta uma variedade de aplicações com um forte foco em treino e inferência em redes neuronais **Deep Learning** (Abadi et al. 2016). Originalmente desenvolvido por investigadores e engenheiros da equipa Google Brain, sendo hoje utilizado por diversos serviços da Google, e desde de que foi disponibilizado como um projeto *open source* foi adotado por diversas empresas, entre as quais:

- AMD
- NVIDIA
- ebay
- Intel
- Twitter
- DropBox
- Coca-Cola

A figura 4.1 demonstra a arquitetura do **TensorFlow** e a maneira como os diversos módulos se encontram implementados e encaixam uns nos outros.

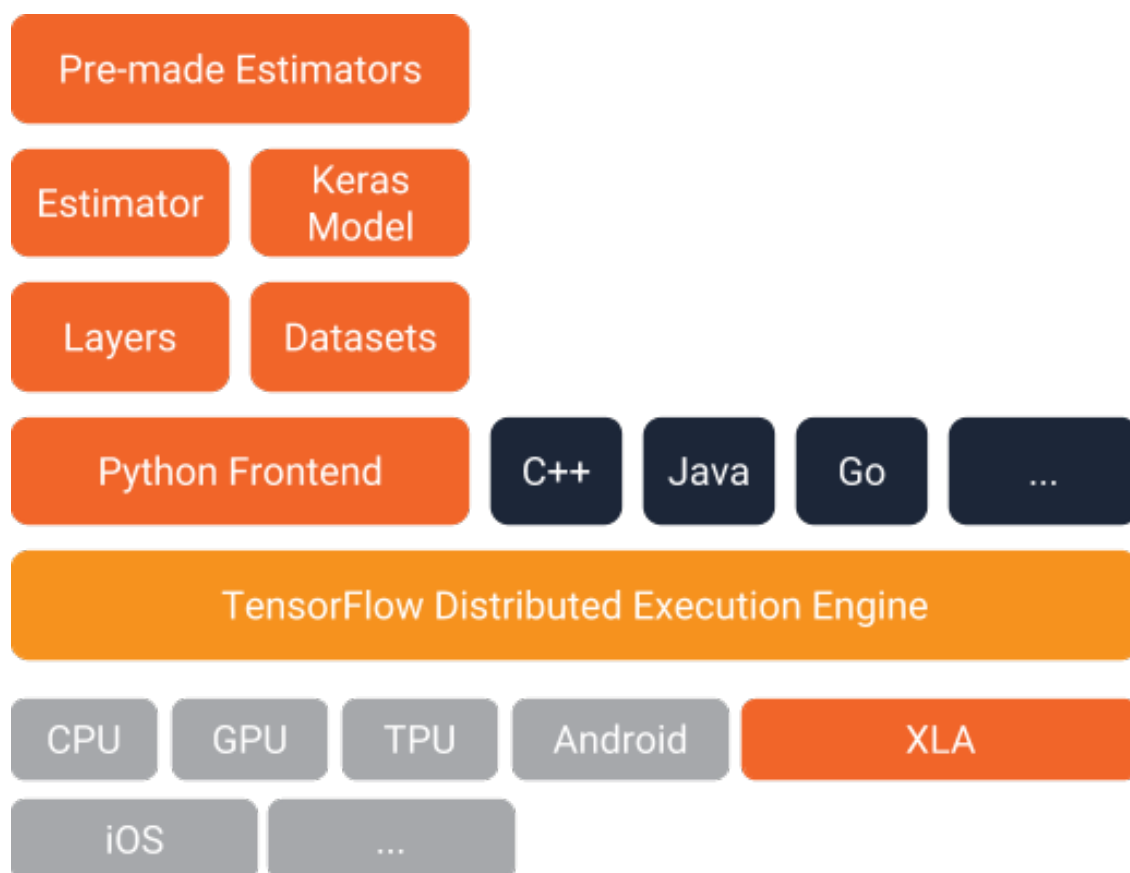
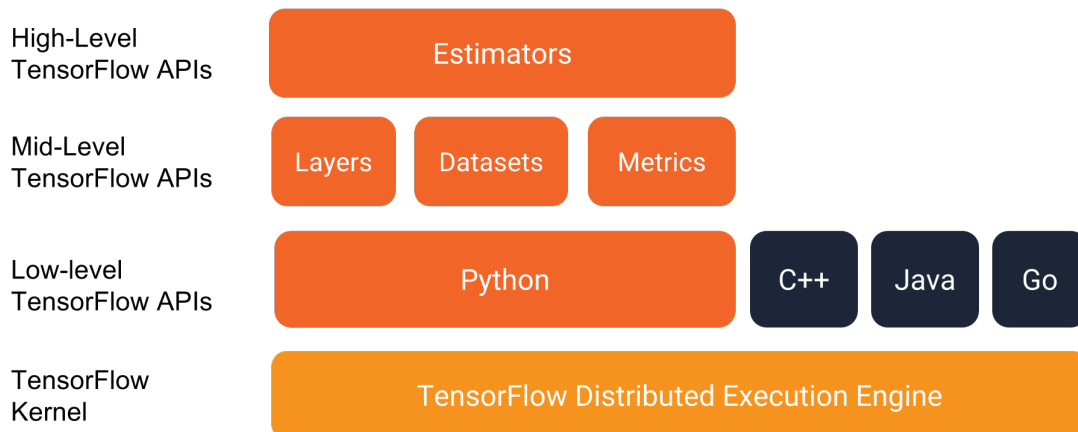


Figura 4.1: Arquitetura dos módulos do **TensorFlow** (TensorFlow Team 2018a)

Apesar de o **TensorFlow** disponibilizar os diversos módulos que podem ser observados, no âmbito do presente trabalho, o foco pode ser reduzido às APIs que permitem construir e treinar modelos de **Machine Learning** e **Deep Learning**. Na figura 4.2 pode-se observar uma arquitetura simplificada, a qual diz respeito a implementação da solução descrita no presente documento.

Figura 4.2: Arquitetura simplificada do **TensorFlow**.

O módulo *Estimators* disponibiliza algoritmos de **Machine Learning** e **Deep Learning**, os módulos *Layers*, *Datasets* e *Metrics* permitem a construção, treino e avaliação de modelos e, no caso do presente trabalho, estes foram utilizados tirando partido da linguagem Python. Por fim existe ainda o módulo de computação distribuída que é inerente ao *Kernel* do **TensorFlow**. A seguinte imagem demonstra com um pouco mais de detalhe a arquitetura que permite a execução distribuída, mostrando como se separam as camadas de interface com o utilizador e as camadas do *core* do **TensorFlow** (TensorFlow Team 2018b).

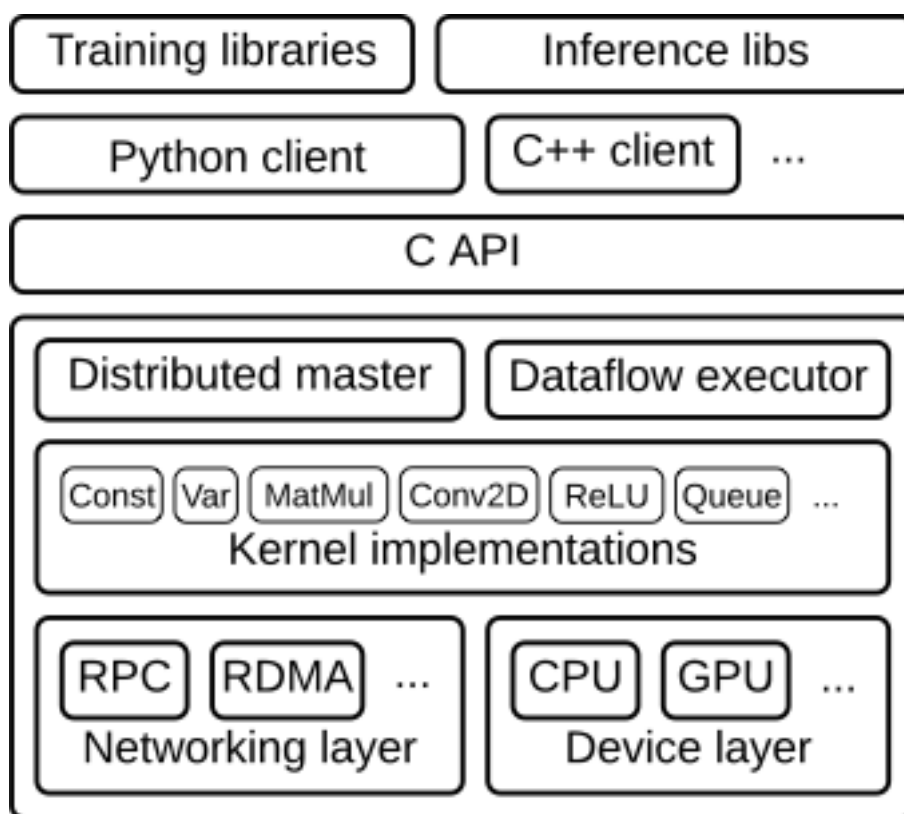


Figura 4.3: Arquitetura distribuída do **TensorFlow**.

4.1.1 Casos de Estudo

Conforme descrito na secção 2.3 "**Deep Learning**", são inúmeros os trabalhos referenciados na literatura que apontam para os bons resultados que se tem vindo a obter quando se faz uso de metodologias **Deep Learning** comparativamente aos tradicionais métodos de **Machine Learning**. Nesse sentido é proposto neste trabalho que sejam exploradas estas novas metodologias e abordagens.

Irão ser efetuados vários casos de estudos de forma a comprovar se a solução proposta é ou não viável para o problema proposto (i.e. se as metodologias **Deep Learning** respondem com sucesso à previsão de consumos no contexto descrito no capítulo 2), uma vez ser essa a tese que se pretende provar com este trabalho. Os casos de estudo descritos em 2.3, que fazem uso de metodologias tradicionais de **Machine Learning**, apontam claramente para uma melhor precisão da previsão dos algoritmos quando se faz uso de um histórico mais enriquecido (i.e Series temporais de maior dimensão).

O objetivo destes casos de estudo consiste na comparação direta com as metodologias tradicionais já aplicadas com as descritas na secção 2.3 "Deep Learning". Pretende-se, nesta fase, com ferramentas disponibilizadas pela arquitetura **TensorFlow**, descrita na sub-secção anterior, analisar o resultado dos algoritmos:

- Linear Regression quando treinada com um grande volume de dados e processados de forma distribuída. Serão utilizados diversos *Datasets* para poder observar o impacto do tamanho de dados e das variáveis de *input*. A metodologia usada será "janela deslizante".

- DNN (Deep Neural Network) quando treinadas com um grande volume de dados e processados de forma distribuída. Para além de diversos *Datasets* serão ainda realizadas diversas experiências para se conseguir encontrar uma configuração de rede neuronal ótima (menor erro possível).

Será ainda efetuada uma comparação com os resultados obtidos num trabalho previamente existente (Jozi et al. 2018), que utiliza métodos de **Machine Learning** tradicionais para realizar previsões utilizando a mesma fonte de dados.

4.2 Arquitetura alternativa

A seguinte arquitetura descrita foi originalmente concebida para a execução do presente trabalho, tendo sido posteriormente abandonada em favor do **TensorFlow**.

A solução seria implementada numa plataforma **Big Data** que foi desenvolvida no centro de investigação GECAD conforme mostra a figura 4.4.

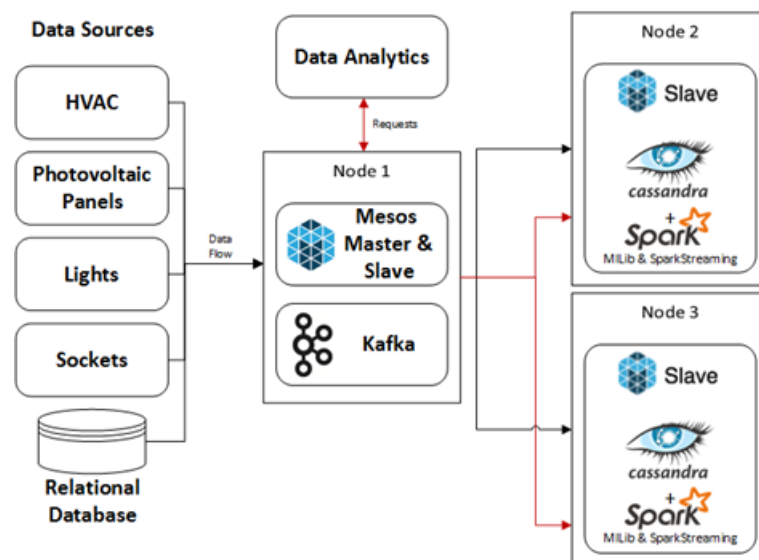


Figura 4.4: Plataforma Big Data implementada no GECAD (Pinheiro et al. 2017)

A plataforma é baseada na arquitetura SMACK conforme descrito detalhadamente em (Rouse 2014). Consiste numa arquitetura distribuída onde são utilizadas várias ferramentas Apache, entre as quais Cassandra, Kafka e Mesos. Os dados de consumo, provenientes de vários analisadores conforme descrito detalhadamente em (Vinagre, Gomes e Vale 2016), estão a ser recolhidos e armazenados em Cassandra em tempo real. A plataforma é composta por vários componentes Apache Spark dos quais, para o presente trabalho, se destaca a MLLib. A MLLib é uma biblioteca que já vem integrada com o *package* do Apache Spark e é dedicada à análise de dados. É composta por vários algoritmos de **Machine Learning** onde se incluem modelos de classificação, regressão, árvores de decisão, recomendação, *clustering*, entre outros (MLlib | Apache Spark 2017). A camada de Data Analytics, para além da MLLib pretende-se que integre outros módulos como se ilustra na figura 4.5.

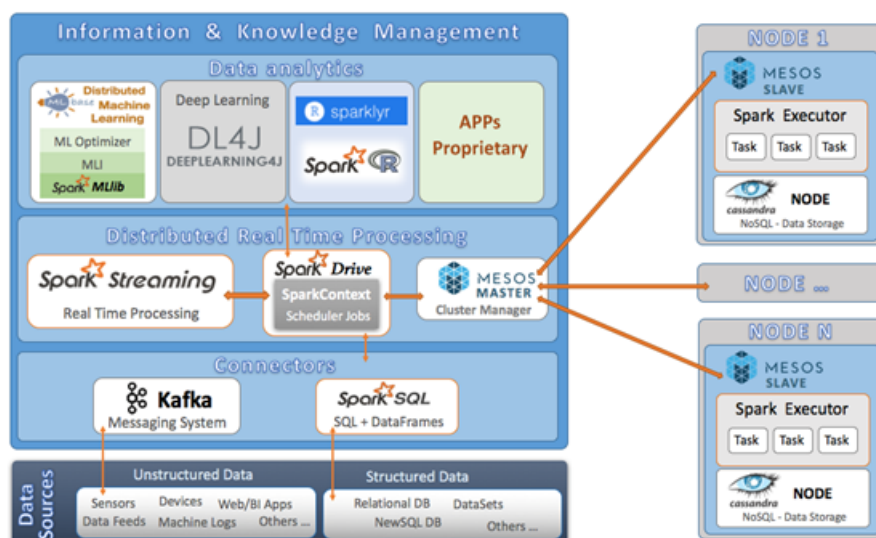


Figura 4.5: Extensão da Plataforma Big Data a implementar no GECAD (Vinagre, Pinto et al. 2017)

Os Módulos que iriam ser implementados nesta extensão, dos quais se destacam as APIs Distributed Machine Learning (DLM), DeepLearning4J (DL4J) e Sparklyr, seriam de extrema importância para a segunda fase de casos de estudo que se pretendia implementar (i.e. DLM facilita a execução distribuída de qualquer algoritmo da biblioteca MLlib, DL4J permite explorar algoritmos **Deep Learning** escritos na linguagem Java e finalmente a Sparklyr facilita a execução distribuída de qualquer código escrito na linguagem R)

Devido a dificuldades na utilização e disponibilidade da plataforma Big Data descrita, foi tomada a decisão de mudar o paradigma para o **TensorFlow** de forma a conseguir atingir os objetivos do presente trabalho.

4.2.1 Casos de Estudo alternativos

Esta sub-seção descreve os casos de estudo aplicáveis à arquitetura alternativa descrita na sub-seção anterior. A estrutura dos mesmos é semelhante, uma vez o objetivo ser o mesmo, fazer uma comparação direta entre metodologias tradicionais e metodologias **Deep Learning**.

1. Pretende-se, nesta fase, com ferramentas (i.e. Spark MLlib) disponibilizadas na plataforma Big Data, descrita detalhadamente na sub-seção anterior, analisar o resultado dos algoritmos:
 - SVM e ANN quando treinados com um grande volume de dados e processados de forma distribuída. A metodologia usada será a de “Janela deslizante”.
 - ANN quando treinadas com um grande volume de dados e processados de forma distribuída. Neste caso serão usadas variáveis externas como neurónios de input.
2. O objetivo destes casos de estudo será o de explorar outras metodologias propostas em **Deep Learning** (isoladamente ou combinada) de forma a encontrar uma melhor solução.

Deep Learning é um assunto emergente que tem despertado grande curiosidade e estudos pela comunidade científica e empresarial. Assim, é ainda de referir que em relação à

elaboração da segunda fase de casos de estudo está previsto a atualização do estado de arte em relação a esta matéria. Ao longo da elaboração da presente tese prevê-se adicionar nova informação relevante que irá surgir quer em matéria de novas metodologias em **Deep Learning** quer em relação a novas ferramentas disponíveis para a sua execução.

4.3 Dados e Datasets

De forma a facilitar a compreensão e visualização da execução dos algoritmos anteriormente mencionados, esta sub-secção reserva-se a demonstrar exemplos dos dados e *Datasets* que serão utilizados, sejam estes para treino ou os valores reais que se pretende obter da previsão.

Os dados utilizados são dados históricos do consumo de energia do edifício N onde se encontra o GECAD.



Figura 4.6: Vista aérea Edifício N

O GECAD encontra-se ativamente envolvido nos temas de previsão de consumo e produção de energia. O edifício N é onde se encontram maioritariamente os projetos e trabalhos relacionados com o setor de energia, uma vez possuir diversos analisadores que recolhem o consumo, bem como possuir painéis fotovoltaicos como demonstrado na figura 4.6, permitindo a recolha de dados de produção em tempo real. Os dados encontram-se a ser recolhidos de dez em dez segundos e são armazenados numa base de dados SQL. Na seguinte figura pode-se observar o resultado de uma *Query* à base de dados de um dos analisadores.

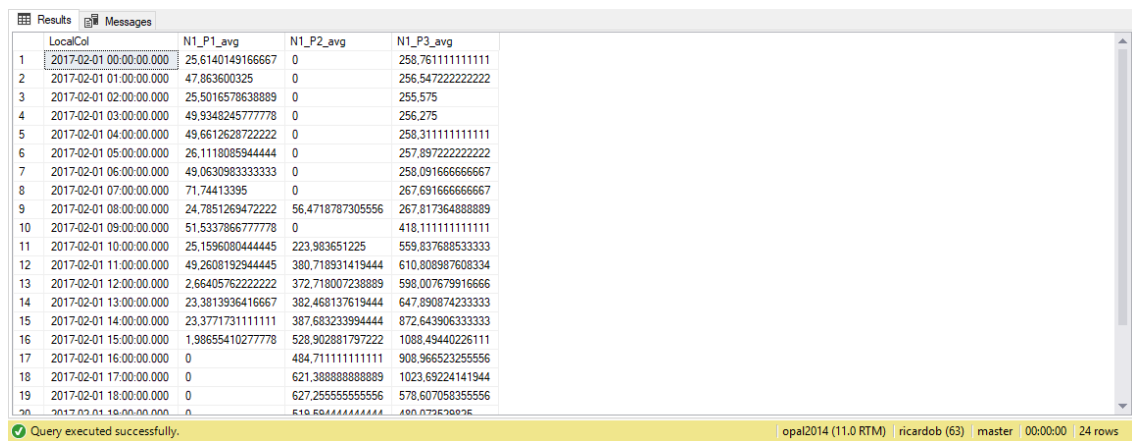
	TimeCol	MSecCol	LocalCol	UserCol	ReasonCol	N1_THD_U1	N1_U3	N1_ActEner3	N1_THD_U3N	N1_THD_U2	N1_THD_U2N	N1_THD_U3	N1_PF1
1	2015-08-13 19:54:20.6930000	692	2015-08-13 20:54:20.6930000	NULL	Time On	123.026459	2.484375	43909120	3.582757	0	3.202435	28,875	1.007388
2	2015-08-13 19:54:30.0070000	6	2015-08-13 20:54:30.0070000	NULL	Time On	0	2.5	43909120	3.526489	0	3.526489	27,5	1.000862
3	2015-08-13 19:54:40.0000000	0	2015-08-13 20:54:40.0000000	NULL	Time On	101.075966	2.515625	43909120	3.595259	0	3.2298	28,5	1.004142
4	2015-08-13 19:54:50.0100000	10	2015-08-13 20:54:50.0100000	NULL	Time On	93.212563	2.546875	43909120	3.533218	0	3.533218	28,125	1.006341
5	2015-08-13 19:55:00.0070000	7	2015-08-13 20:55:00.0070000	NULL	Time On	0	2.53125	43909120	3.59058	0	3.217911	28,375	0
6	2015-08-13 19:55:10.0030000	4	2015-08-13 20:55:10.0030000	NULL	Time On	94.284004	2.5	43909120	3.642895	0	3.642895	29	1.000426
7	2015-08-13 19:55:20.0000000	1	2015-08-13 20:55:20.0000000	NULL	Time On	97.373291	2.515625	43909120	3.553391	0	3.553391	26,5	1.004428
8	2015-08-13 19:55:30.1170000	117	2015-08-13 20:55:30.1170000	NULL	Time On	94.612671	2.515625	43909120	3.588221	0	3.223778	29	1.004725
9	2015-08-13 19:55:40.1170000	117	2015-08-13 20:55:40.1170000	NULL	Time On	109.269485	2.546875	43909120	3.628414	0	3.628414	28,25	1.001046
10	2015-08-13 19:55:50.0100000	10	2015-08-13 20:55:50.0100000	NULL	Time On	103.213715	2.5625	43909120	3.650054	0	3.280772	28,5	1.005713
11	2015-08-13 19:56:00.0000000	1	2015-08-13 20:56:00.0000000	NULL	Time On	87.665733	2.53125	43909120	3.700289	0	3.322456	29,375	1.004003
12	2015-08-13 19:56:10.0100000	11	2015-08-13 20:56:10.0100000	NULL	Time On	87.351692	2.5625	43909120	3.672261	0	3.268636	28,375	1.00106
13	2015-08-13 19:56:20.0030000	3	2015-08-13 20:56:20.0030000	NULL	Time On	88.527809	2.609375	43909120	3.568111	0	3.203238	27,5	1.003039
14	2015-08-13 19:56:30.0000000	0	2015-08-13 20:56:30.0000000	NULL	Time On	100.314049	2.53125	43909120	3.671752	0	3.255249	28,75	1.005098
15	2015-08-13 19:56:40.0070000	5	2015-08-13 20:56:40.0070000	NULL	Time On	86.598228	2.546875	43909120	3.60638	0	3.60638	29,625	0
16	2015-08-13 19:56:50.0030000	2	2015-08-13 20:56:50.0030000	NULL	Time On	105.361969	2.515625	43909120	3.453135	0	3.162661	30,125	1.000218
17	2015-08-13 19:57:00.0130000	12	2015-08-13 20:57:00.0130000	NULL	Time On	114.466805	3.234375	43909120	6.920267	0	4.735768	93,5	0
18	2015-08-13 19:57:10.0100000	9	2015-08-13 20:57:10.0100000	NULL	Time On	92.548668	2.5625	43909120	3.563781	0	3.563781	29,625	1.005185
19	2015-08-13 19:57:20.0070000	6	2015-08-13 20:57:20.0070000	NULL	Time On	113.047388	3.5	43909120	7.900584	0	7.900584	96,5	0

Query executed successfully. opal2014 (11.0 RTM) ricardob (53) GID_N 00:00:00 1000 rows

Figura 4.7: Exemplo de uma *query* à base de dados

Como se pode observar na figura 4.7, a base de dados possui dados históricos que remontam a, pelo menos, três anos, contendo assim o que se espera ser volume de dados satisfatório para se poder utilizar para treino de redes neuronais **Deep Learning**.

No contexto do presente trabalho distinguem-se três conjuntos de dados que serão utilizados para efetuar a previsões, sendo estes o consumo referente ao ar condicionado (HVAC), iluminação (Lights) e das tomadas (Sockets). Cada conjunto apresenta diferentes perfis de consumo, sendo cada um utilizado como *Dataset* para treino. No entanto, como alguns dos conjuntos apresentam perfis em que, por exemplo, existem momentos em que o consumo é zero (resultando em problemas no cálculo do erro da previsão) é ainda utilizado como *Dataset* a soma dos três conjuntos, realizando-se a previsão do consumo total. Como os dados estão a ser recolhidos a cada dez segundos, e uma vez que a previsão é realizada *Hour Ahead* (previsão do consumo para a hora seguinte), é necessário utilizar o consumo médio para cada hora anterior como dados de treino. A seguinte figura demonstra um exemplo do resultado de uma *query* onde são agrupados os dados de consumo médio para cada hora.



LocalCol	N1_P1_avg	N1_P2_avg	N1_P3_avg
2017-02-01 00:00:00.000	25.6140149166667	0	258.761111111111
2017-02-01 01:00:00.000	47.863600325	0	256.547222222222
2017-02-01 02:00:00.000	25.5016578638889	0	255.575
2017-02-01 03:00:00.000	49.9348245777778	0	256.275
2017-02-01 04:00:00.000	49.6612628722222	0	258.311111111111
2017-02-01 05:00:00.000	26.1118085944444	0	257.897222222222
2017-02-01 06:00:00.000	49.0630983333333	0	258.091666666667
2017-02-01 07:00:00.000	71.74413395	0	267.691666666667
2017-02-01 08:00:00.000	24.7851269472222	56.4718787305556	267.817364888889
2017-02-01 09:00:00.000	51.5337866777778	0	418.111111111111
2017-02-01 10:00:00.000	25.1596080444445	223.983651225	559.837688533333
2017-02-01 11:00:00.000	49.2608192944445	380.718931419444	610.808987608334
2017-02-01 12:00:00.000	2.66405762222222	372.718007238889	598.007679916666
2017-02-01 13:00:00.000	23.3813936416667	382.468137619444	647.890874233333
2017-02-01 14:00:00.000	23.3771731111111	387.68323394444	872.643906333333
2017-02-01 15:00:00.000	1.98655410277778	528.902881797222	1088.49440226111
2017-02-01 16:00:00.000	0	484.711111111111	908.966523255556
2017-02-01 17:00:00.000	0	621.388888888889	1023.69224141944
2017-02-01 18:00:00.000	0	627.255555555556	578.607058355556
2017-02-01 19:00:00.000	0	519.584444444444	480.072538825

Figura 4.8: Exemplo de uma *query* do consumo de cada conjunto

No exemplo observado na figura 4.8 a *query* executada retorna o consumo médio para o dia 1 de Fevereiro de 2017, para HVAC (N1_P1_avg), Lights (N1_P2_avg) e Sockets (N1_P3_avg).

Capítulo 5

Implementação

Neste capítulo é apresentada a implementação da solução preconizada, sendo demonstrado o ciclo básico de execução das experiências, bem como os diversos casos de estudo realizados.

De forma a apresentar uma visão global da solução implementada, apresentam-se inicialmente os requisitos funcionais e não-funcionais identificados para a resolução da mesma.

Requisitos funcionais:

- Utilizar plataforma TensorFlow
- Aceder a base de dados SQL Server
- Implementar soluções com Linear Regression e DNN
- Definir dados de treino e teste
- Definir parâmetros dos algoritmos, nomeadamente a configuração de camadas ocultas
- Construir um modelo baseado no algoritmo escolhido e dados de treino definidos
- Executar previsão através do modelo construído e dados de teste definidos
- Calcular o erro encontrado através da implementação das métricas MAPE e SMAPE

Requisitos não-funcionais:

- Utilizar a linguagem Python
- Apresentar resultados em consola ou escrever num ficheiro CSV/Excel

Tal como é mencionado no capítulo 4, os casos de uso realizados envolveram a utilização de dois algoritmos, sendo um deles tradicional de **Machine Learning** (Linear Regression) e o restante um algoritmo de **Deep Learning** (Deep Neural Network). Estes algoritmos foram escolhidos pela sua disponibilidade no **TensorFlow**, fazendo parte de uma sub-classe de Estimator, sendo esta Pre-made Estimators. A figura 5.1 demonstra a relação de herança entre as classes, sendo que a classe Estimator já fora previamente demonstrada na arquitetura do **TensorFlow** no capítulo 4.

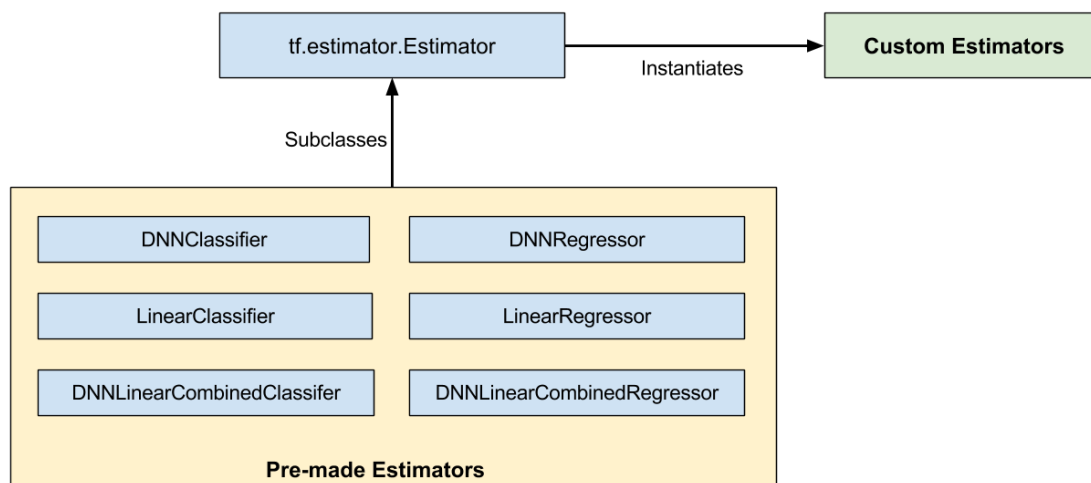


Figura 5.1: Herança de Estimator e Pre-made Estimator

Parte da razão pela qual foi utilizada a classe Pre-made Estimators passa por esta estar preparada para computação distribuída, simplificando assim a transição para a execução dos algoritmos em vários nós/recursos computacionais.

A execução dos algoritmos segue uma lógica simples:

1. Inicialização de variáveis, nomeadamente do **TensorFlow**
2. Conexão à base de dados e aquisição dos dados de treino
3. Tratamento dos dados e criação de ficheiro de treino
4. Criação e treino do modelo
5. Inserção dos dados de teste e execução da previsão
6. Cálculo do erro
7. Demonstração de resultados

O diagrama de sequência da apresentado na figura 5.2 demonstra o exemplo do fluxo acima descrito de uma execução.

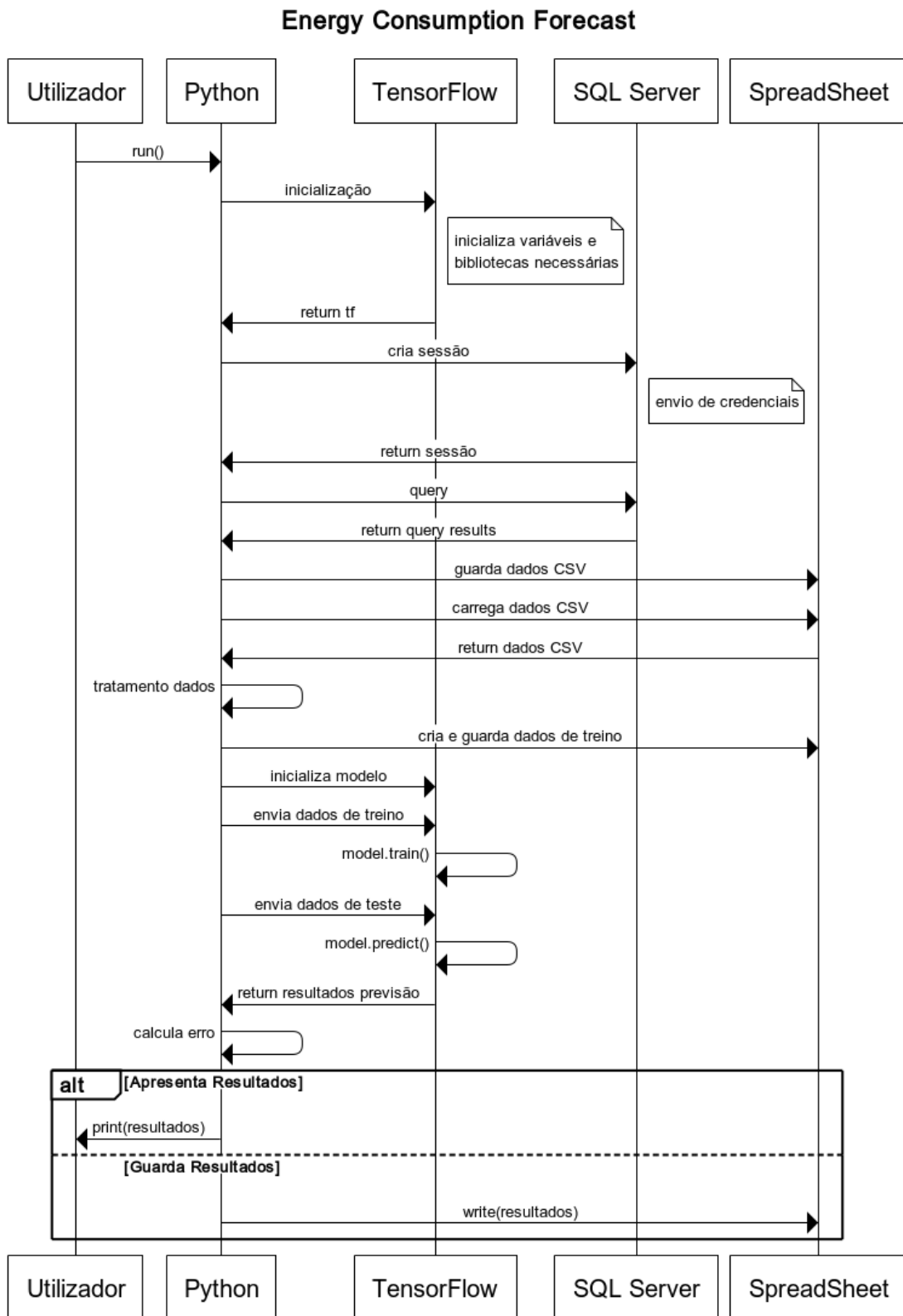


Figura 5.2: Processo de previsão do consumo de energia

5.1 Linear Regression

A primeira implementação realizada foi a Linear Regression. Sendo um algoritmo de **Machine Learning** tradicional é de mais simples implementação e execução servindo assim como referência para as experiências que se seguiram. Tal como referido no capítulo 4, a metodologia utilizada foi a de "janela deslizante", isto é, para calcular o consumo de uma determinada hora, são utilizados os valores do consumo das horas anteriores. Por exemplo, para calcular o consumo de energia para as 17:00 horas são utilizados os valores de consumo das 10:00 até 16:00 horas. No entanto, se se realizar a previsão para as 19:00 são utilizados valores de consumo das horas 12:00 até 18:00 (onde o valor das 17:00 é o valor real, não o resultado da previsão anterior). Este período de 6 horas anteriores à hora sobre a qual se quer realizar a previsão de consumo é denominada "janela deslizante".

Inicialmente o treino do modelo foi realizado com os dados do *Dataset* referente a Janeiro de 2017. Foi escolhido este *Dataset* pela integridade dos dados, uma vez não haver erros nem lacunas nos mesmos. Os dados de *input* utilizados foram o consumo de cada uma das 6 horas anteriores. As previsões realizadas são referentes as seguintes três horas diferentes do dia:

- 7:00 - hora em que o consumo se mantém constante às horas anteriores pela virtude de que não se encontra ninguém no edifício
- 13:00 - hora em se pode verificar um pico no consumo, positivo ou negativo, resultante da hora do almoço. Já contém valores mais diversos nas horas anteriores
- 19:00 - final do dia, a atividade e consumo já deve baixar apesar das horas anteriores serem, teoricamente, de atividade de trabalho normal

Como tal, as previsões executadas obtiveram os resultados observados na figura 5.3.

Linear Regression - Sockets		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução		6ª Execução	
MAPE	5,05%	MAPE	2,21%	MAPE	4,59%	MAPE	5,12%	MAPE	2,80%	MAPE	2,80%
SMAPE	4,93%	SMAPE	2,23%	SMAPE	4,48%	SMAPE	5,00%	SMAPE	2,76%	SMAPE	2,76%
Valor Previsão	245,673	Valor Previsão	266,366	Valor Previsão	355,876	Valor Previsão	252,164	Valor Previsão	258,495	Valor Previsão	258,495
Valor Real	258,091	Valor Real	260,491	Valor Real	372,194	Valor Real	265,083	Valor Real	265,722	Valor Real	265,722
Linear Regression - Sockets		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução		6ª Execução	
MAPE	0,07%	MAPE	7,24%	MAPE	14,77%	MAPE	6,75%	MAPE	3,72%	MAPE	3,72%
SMAPE	0,07%	SMAPE	6,99%	SMAPE	15,95%	SMAPE	6,53%	SMAPE	3,65%	SMAPE	3,65%
Valor Previsão	597,613	Valor Previsão	952,88	Valor Previsão	854,994	Valor Previsão	246,168	Valor Previsão	252,308	Valor Previsão	252,308
Valor Real	598,008	Valor Real	1021,879	Valor Real	728,679	Valor Real	262,764	Valor Real	261,694	Valor Real	261,694
Linear Regression - Sockets		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução		6ª Execução	
MAPE	42,12%	MAPE	27,30%	MAPE	39,67%	MAPE	8,59%	MAPE	2,65%	MAPE	2,65%
SMAPE	53,35%	SMAPE	31,61%	SMAPE	49,48%	SMAPE	8,24%	SMAPE	2,61%	SMAPE	2,61%
Valor Previsão	999,635	Valor Previsão	802,231	Valor Previsão	641,271	Valor Previsão	244,803	Valor Previsão	253,74	Valor Previsão	253,74
Valor Real	578,607	Valor Real	538,231	Valor Real	386,881	Valor Real	265,841	Valor Real	260,452	Valor Real	260,452

Figura 5.3: Execução Linear Regression Inicial

Cada execução refere-se a um dia diferente, neste caso 1ª execução = 1 de Fevereiro, 2ª execução = 2 de Fevereiro, etc, onde o mesmo modelo é utilizado para cada hora do dia. Este exemplo inicial refere-se ao consumo de Sockets. Observando os resultados pode-se concluir que o algoritmo não era capaz de criar um modelo que realizasse uma previsão com erro baixo para qualquer uma das horas. Isto é, quando o modelo conseguia uma previsão com erro mais baixo para uma das horas, nas restantes horas observava-se um aumento no erro.

Assim, foram adicionadas duas novas variáveis de *input* na expectativa de que pudesse ajudar a diminuir a variação do erro. As variáveis adicionadas foram a hora do dia e um *boolean*

que indica se o dia para o qual se está a efetuar a previsão do consumo é fim-de-semana ou não. Na figura 5.4 pode-se observar os resultados da execução com estas novas variáveis.

SOCKETS									
Linear Regression		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	1,52%	MAPE	1,86%	MAPE	1,07%	MAPE	1,33%	MAPE	0,92%
SMAPE	1,51%	SMAPE	1,85%	SMAPE	1,65%	SMAPE	1,33%	SMAPE	0,93%
Valor Previsão	254,214	Valor Previsão	255,724	Valor Previsão	368,252	Valor Previsão	361,593	Valor Previsão	266,201
Valor Real	258,091	Valor Real	260,491	Valor Real	372,194	Valor Real	265,083	Valor Real	265,722
Linear Regression		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	4,08%	MAPE	6,58%	MAPE	1,47%	MAPE	2,89%	MAPE	0,17%
SMAPE	4,17%	SMAPE	6,37%	SMAPE	1,46%	SMAPE	2,85%	SMAPE	0,17%
Valor Previsão	623,446	Valor Previsão	958,78	Valor Previsão	718,088	Valor Previsão	255,418	Valor Previsão	262,145
Valor Real	598,007	Valor Real	1021,879	Valor Real	728,679	Valor Real	262,764	Valor Real	261,694
Linear Regression		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	44,50%	MAPE	19,90%	MAPE	42,40%	MAPE	4,69%	MAPE	1,04%
SMAPE	57,23%	SMAPE	22,10%	SMAPE	53,80%	SMAPE	4,58%	SMAPE	1,04%
Valor Previsão	1042,523	Valor Previsão	728,135	Valor Previsão	671,618	Valor Previsão	244,803	Valor Previsão	263,182
Valor Real	578,007	Valor Real	538,231	Valor Real	386,881	Valor Real	265,841	Valor Real	260,452
Tempo Pré-Processamento(s)	0,709	Tempo Pré-Processamento(s)	0,758	Tempo Pré-Processamento(s)	0,551	Tempo Pré-Processamento(s)	0,515	Tempo Pré-Processamento(s)	0,709
Tempo Modelo	3,938	Tempo Modelo	3,936	Tempo Modelo	4,119	Tempo Modelo	3,848	Tempo Modelo	3,938

Figura 5.4: Execução Linear Regression Sockets

Apesar do erro médio para as 7:00 e as 13:00 ter baixado, o mesmo não se verifica para as 19:00. Para além disso, a diferença de erro calculado entre diferentes horas do mesmo dia com o mesmo modelo continua a ser elevado. É de notar ainda que é também calculado o tempo de Pré-Processamento (tempo resultante de acesso à base de dados e tratamento dos dados), e tempo do Modelo (tempo gasto para treinar o modelo e realizar a previsão).

Para poder fazer uma comparação futura, foram também executadas previsões para os mesmos dias para Lights e HVAC, como se apresentam nas seguintes figuras.

LIGHTS									
Linear Regression		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	100,00%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%
SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	7,244	Valor Previsão	7,904	Valor Previsão	7,558	Valor Previsão	7,662	Valor Previsão	7,655
Valor Real	0	Valor Real	0	Valor Real	0	Valor Real	0	Valor Real	0
Linear Regression		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	2,46%	MAPE	15,57%	MAPE	4,92%	MAPE	100,00%	MAPE	100,00%
SMAPE	2,50%	SMAPE	14,44%	SMAPE	4,80%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	382,136	Valor Previsão	690,25	Valor Previsão	409,26	Valor Previsão	7,662	Valor Previsão	7,655
Valor Real	372,718	Valor Real	797,701	Valor Real	429,394	Valor Real	0	Valor Real	0
Linear Regression		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	6,41%	MAPE	1,67%	MAPE	75,64%	MAPE	100,00%	MAPE	100,00%
SMAPE	6,21%	SMAPE	1,68%	SMAPE	121,66%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	589,469	Valor Previsão	657,612	Valor Previsão	410,294	Valor Previsão	7,662	Valor Previsão	7,655
Valor Real	627,255	Valor Real	646,627	Valor Real	99,929	Valor Real	0	Valor Real	0
Tempo Pré-Processamento(s)	0,557	Tempo Pré-Processamento(s)	0,578	Tempo Pré-Processamento(s)	0,568	Tempo Pré-Processamento(s)	0,668	Tempo Pré-Processamento(s)	0,509
Tempo Modelo	4,018	Tempo Modelo	3,737	Tempo Modelo	3,769	Tempo Modelo	4,011	Tempo Modelo	3,711

Figura 5.5: Execução Linear Regression Lights

HVAC									
Linear Regression	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	62,58%	MAPE	100,00%	MAPE	18,68%	MAPE	100,00%	MAPE	100,00%
SMAPE	47,67%	SMAPE	200,00%	SMAPE	17,08%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	30,178	Valor Previsão	4,999	Valor Previsão	45,425	Valor Previsão	4,146	Valor Previsão	5,026
Valor Real	49,063	Valor Real	0	Valor Real	53,911	Valor Real	0	Valor Real	0
Linear Regression	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	94,79%	MAPE	208,01%	MAPE	14,75%	MAPE	100,00%	MAPE	100,00%
SMAPE	180,18%	SMAPE	200,00%	SMAPE	15,92%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	51,105	Valor Previsão	-2,143	Valor Previsão	51,724	Valor Previsão	4,146	Valor Previsão	5,026
Valor Real	2,664	Valor Real	2,315	Valor Real	44,095	Valor Real	0	Valor Real	0
Linear Regression	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	100,00%	MAPE	589,92%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%
SMAPE	200,00%	SMAPE	149,36%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	6,722	Valor Previsão	6,769	Valor Previsão	38,3	Valor Previsão	4,146	Valor Previsão	5,026
Valor Real	0	Valor Real	46,703	Valor Real	0	Valor Real	0	Valor Real	0
Tempo Pre-Processamento(s)	0,557	Tempo Pre-Processamento(s)	0,5	Tempo Pre-Processamento(s)	0,563	Tempo Pre-Processamento(s)	0,615	Tempo Pre-Processamento(s)	0,54
Tempo Modelo	4,018	Tempo Modelo	3,723	Tempo Modelo	3,862	Tempo Modelo	4,032	Tempo Modelo	4,051

Figura 5.6: Execução Linear Regression HVAC

Pode-se observar pelas figuras 5.5 e 5.6 o problema que advém quando existem valores reais de consumo iguais a zero. Para além das métricas de erro acusarem 100% (ou 200% no caso so SMAPE), causa erros desproporcionalmente elevados para casos em que o consumo real não é zero. Como tal, para ter uma referência base para poder fazer comparações entre os diversos casos foi executada a previsão para o total do consumo, isto é, utilizando a soma do consumo dos três conjuntos como dados de treino.

TOTAL - ANALYZER1_V2									
Linear Regression	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	Previsão 7:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	16,21%	MAPE	1,24%	MAPE	8,63%	MAPE	1,94%	MAPE	5,16%
SMAPE	14,99%	SMAPE	1,24%	SMAPE	8,27%	SMAPE	1,92%	SMAPE	5,03%
Valor Previsão	264,317	Valor Previsão	257,292	Valor Previsão	392,269	Valor Previsão	260,037	Valor Previsão	252,683
Valor Real	307,154	Valor Real	260,491	Valor Real	426,105	Valor Real	265,083	Valor Real	265,722
Linear Regression	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	Previsão 13:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	6,57%	MAPE	10,89%	MAPE	8,29%	MAPE	3,56%	MAPE	6,36%
SMAPE	6,79%	SMAPE	10,33%	SMAPE	8,65%	SMAPE	3,50%	SMAPE	6,17%
Valor Previsão	1041,923	Valor Previsão	1642,937	Valor Previsão	1310,874	Valor Previsão	253,749	Valor Previsão	246,041
Valor Real	973,389	Valor Real	1821,896	Valor Real	1202,169	Valor Real	262,794	Valor Real	261,694
Linear Regression	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	Previsão 19:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	27,51%	MAPE	9,18%	MAPE	53,58%	MAPE	5,54%	MAPE	5,29%
SMAPE	31,90%	SMAPE	9,62%	SMAPE	73,19%	SMAPE	5,39%	SMAPE	5,15%
Valor Previsão	1663,454	Valor Previsão	1405,601	Valor Previsão	1048,761	Valor Previsão	251,88	Valor Previsão	247,37
Valor Real	1205,862	Valor Real	1276,562	Valor Real	486,81	Valor Real	265,841	Valor Real	260,452
Tempo Pre-Processamento(s)	0,569	Tempo Pre-Processamento(s)	0,574	Tempo Pre-Processamento(s)	0,53	Tempo Pre-Processamento(s)	0,603	Tempo Pre-Processamento(s)	0,641
Tempo Modelo	4,021	Tempo Modelo	3,962	Tempo Modelo	3,927	Tempo Modelo	3,867	Tempo Modelo	3,825

Figura 5.7: Execução Linear Regression Total

Mais uma vez se pode verificar que o Linear Regression não é capaz de criar um modelo satisfatório com os dados que foram utilizados.

5.2 Deep Neural Network

Terminada a execução do algoritmo de Linear Regression para cada um dos *Datasets* relevantes, foi implementado de seguida um algoritmo de **Deep Learning**, neste caso Deep Neural Network (DNN), baseado na classe Pre-made Estimators do **TensorFlow** anteriormente referida. Sendo um algoritmo de **Deep Learning**, ou seja, baseado em redes neuronais, este pode ser executado de diversas maneiras resultando em diversos modelos. Como tal, foram efetuadas diversas experiências ajustando os parâmetros da rede neuronal. No caso do presente trabalho, foram ajustados o número de camadas ocultas de processamento da rede neuronal, bem como o número de nós de cada camada. Para efeito de se realizar uma comparação direta entre as duas implementações, a metodologia utilizada é a mesma do que em Linear Regression (janela "deslizante") bem como os mesmos *Datasets*

A classe Pre-made Estimators utilizava por defeito duas camadas com 10 nós cada. Como tal, foi efetuado a previsão inicial para cada *Dataset* com esses parâmetros.

SOCKETS									
DNN - 2 Layers 10 Nodes Each		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	2,42%	MAPE	3,95%	MAPE	2,34%	MAPE	1,47%	MAPE	1,90%
SMAPE	2,39%	SMAPE	3,97%	SMAPE	2,31%	SMAPE	1,46%	SMAPE	1,89%
Valor Previsão	352	Valor Previsão	250,598	Valor Previsão	363,701	Valor Previsão	269,045	Valor Previsão	260,771
Valor Real	258,091	Valor Real	260,491	Valor Real	372,194	Valor Real	265,083	Valor Real	265,722
DNN - 2 Layers 10 Nodes Each		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
MAPE	5,81%	MAPE	12,23%	MAPE	2,26%	MAPE	1,39%	MAPE	3,10%
SMAPE	5,64%	SMAPE	11,52%	SMAPE	2,28%	SMAPE	1,38%	SMAPE	3,05%
Valor Previsão	565,196	Valor Previsão	910,535	Valor Previsão	745,498	Valor Previsão	259,194	Valor Previsão	253,827
Valor Real	598,007	Valor Real	1021,879	Valor Real	728,679	Valor Real	262,764	Valor Real	261,694
DNN - 2 Layers 10 Nodes Each		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
MAPE	39,59%	MAPE	22,11%	MAPE	39,78%	MAPE	2,39%	MAPE	1,93%
SMAPE	49,36%	SMAPE	24,86%	SMAPE	49,66%	SMAPE	2,36%	SMAPE	1,91%
Valor Previsão	957,789	Valor Previsão	691,02	Valor Previsão	642,477	Valor Previsão	259,631	Valor Previsão	255,531
Valor Real	578,007	Valor Real	538,231	Valor Real	386,881	Valor Real	265,841	Valor Real	260,452
Tempo Pre-Processamento(s)	0,576	Tempo Pre-Processamento(s)	0,628	Tempo Pre-Processamento(s)	0,563	Tempo Pre-Processamento(s)	0,667	Tempo Pre-Processamento(s)	0,563
Tempo Modelo	3,714	Tempo Modelo	3,587	Tempo Modelo	3,57	Tempo Modelo	3,555	Tempo Modelo	3,61

Figura 5.8: Execução DNN Sockets

LIGHTS									
DNN - 2 Layers 10 Nodes Each		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução	2ª Execução	3ª Execução	4ª Execução	5ª Execução	6ª Execução	7ª Execução	8ª Execução	9ª Execução	10ª Execução
MAPE	100,00%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%
SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	13,035	Valor Previsão	9,347	Valor Previsão	14,61	Valor Previsão	10,784	Valor Previsão	8,865
Valor Real	0	Valor Real	0	Valor Real	0	Valor Real	0	Valor Real	0
DNN - 2 Layers 10 Nodes Each		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
MAPE	19,56%	MAPE	11,13%	MAPE	26,35%	MAPE	100,00%	MAPE	100,00%
SMAPE	21,69%	SMAPE	10,54%	SMAPE	23,28%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	463,349	Valor Previsão	717,804	Valor Previsão	339,843	Valor Previsão	10,784	Valor Previsão	8,865
Valor Real	372,718	Valor Real	797,701	Valor Real	429,394	Valor Real	0	Valor Real	0
DNN - 2 Layers 10 Nodes Each		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
MAPE	6,31%	MAPE	1,00%	MAPE	70,43%	MAPE	100,00%	MAPE	100,00%
SMAPE	6,12%	SMAPE	1,00%	SMAPE	108,70%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	590,007	Valor Previsão	653,144	Valor Previsão	337,885	Valor Previsão	10,784	Valor Previsão	8,865
Valor Real	627,255	Valor Real	646,627	Valor Real	99,929	Valor Real	0	Valor Real	0
Tempo Pre-Processamento(s)	0,579	Tempo Pre-Processamento(s)	0,528	Tempo Pre-Processamento(s)	0,562	Tempo Pre-Processamento(s)	0,549	Tempo Pre-Processamento(s)	0,586
Tempo Modelo	3,687	Tempo Modelo	3,994	Tempo Modelo	4,107	Tempo Modelo	3,558	Tempo Modelo	3,573

Figura 5.9: Execução DNN Lights

HVAC									
DNN - 2 Layers 10 Nodes Each		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	33,87%	MAPE	100,00%	MAPE	0,44%	MAPE	100,00%	MAPE	100,00%
SMAPE	49,06%	SMAPE	200,00%	SMAPE	0,44%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	33,871	Valor Previsão	6,575	Valor Previsão	53,677	Valor Previsão	5,794	Valor Previsão	5,824
Valor Real	49,063	Valor Real	0	Valor Real	53,911	Valor Real	0	Valor Real	0
DNN - 2 Layers 10 Nodes Each		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	93,90%	MAPE	331,01%	MAPE	12,87%	MAPE	100,00%	MAPE	100,00%
SMAPE	173,47%	SMAPE	200,00%	SMAPE	13,75%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	37,5	Valor Previsão	-1,002	Valor Previsão	50,607	Valor Previsão	5,794	Valor Previsão	5,026
Valor Real	2,664	Valor Real	2,315	Valor Real	44,095	Valor Real	0	Valor Real	0
DNN - 2 Layers 10 Nodes Each		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	100,00%	MAPE	465,43%	MAPE	100,00%	MAPE	100,00%	MAPE	100,00%
SMAPE	200,00%	SMAPE	139,89%	SMAPE	200,00%	SMAPE	200,00%	SMAPE	200,00%
Valor Previsão	5,078	Valor Previsão	8,26	Valor Previsão	70,225	Valor Previsão	5,794	Valor Previsão	5,026
Valor Real	0	Valor Real	46,703	Valor Real	0	Valor Real	0	Valor Real	0
Tempo Pre-Processamento(s)	0,593	Tempo Pre-Processamento(s)	0,572	Tempo Pre-Processamento(s)	0,534	Tempo Pre-Processamento(s)	0,567	Tempo Pre-Processamento(s)	0,54
Tempo Modelo	3,605	Tempo Modelo	3,646	Tempo Modelo	3,698	Tempo Modelo	3,58	Tempo Modelo	4,051

Figura 5.10: Execução DNN HVAC

Como se pode verificar nas imagens 5.8, 5.9 e 5.10, quando comparados os resultados com os obtidos com o Linear Regression, o DNN, quando treinado com duas camadas de 10 nós cada, apresenta um erro médio aproximadamente semelhante. Observa-se apenas casos isolados como a previsão para as 7:00 do HVAC, em que o DNN apresenta um erro médio notavelmente mais baixo. No entanto mantém-se o entrave de previsões para horas em que o consumo real é zero, sendo portanto incapaz de criar um modelo que consiga ter em conta as horas em que não existe consumo. De forma a poder-se efetuar experiências sem ter de ser necessário ponderar ignorar dados com valores de erro médio de 100%, passa-se a utilizar como referência a previsão do consumo total, que se apresenta na figura 5.11.

TOTAL - ANALYZER1_V2									
DNN - 2 Layers 10 Nodes Each		Previsão 7:00		Previsão 7:00		Previsão 7:00		Previsão 7:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	15,22%	MAPE	6,34%	MAPE	15,51%	MAPE	6,89%	MAPE	10,87%
SMAPE	14,15%	SMAPE	6,15%	SMAPE	14,39%	SMAPE	6,66%	SMAPE	10,31%
Valor Previsão	266,572	Valor Previsão	244,954	Valor Previsão	368,903	Valor Previsão	248,007	Valor Previsão	239,66
Valor Real	307,154	Valor Real	260,491	Valor Real	426,105	Valor Real	265,083	Valor Real	265,722
DNN - 2 Layers 10 Nodes Each		Previsão 13:00		Previsão 13:00		Previsão 13:00		Previsão 13:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	3,25%	MAPE	10,46%	MAPE	10,41%	MAPE	9,38%	MAPE	13,28%
SMAPE	3,31%	SMAPE	9,94%	SMAPE	10,99%	SMAPE	8,96%	SMAPE	12,46%
Valor Previsão	1006,106	Valor Previsão	1649,431	Valor Previsão	1341,928	Valor Previsão	240,256	Valor Previsão	231,006
Valor Real	973,389	Valor Real	1821,896	Valor Real	1202,169	Valor Real	262,794	Valor Real	261,694
DNN - 2 Layers 10 Nodes Each		Previsão 19:00		Previsão 19:00		Previsão 19:00		Previsão 19:00	
1ª Execução		2ª Execução		3ª Execução		4ª Execução		5ª Execução	
MAPE	29,82%	MAPE	3,14%	MAPE	52,20%	MAPE	11,21%	MAPE	11,35%
SMAPE	35,05%	SMAPE	3,19%	SMAPE	70,64%	SMAPE	10,61%	SMAPE	10,74%
Valor Previsão	1718,307	Valor Previsão	1317,96	Valor Previsão	1018,512	Valor Previsão	239,053	Valor Previsão	247,37
Valor Real	1205,862	Valor Real	1276,562	Valor Real	486,81	Valor Real	265,841	Valor Real	260,452
Tempo Pre-Processamento(s)	0,549	Tempo Pre-Processamento(s)	0,518	Tempo Pre-Processamento(s)	0,525	Tempo Pre-Processamento(s)	0,435	Tempo Pre-Processamento(s)	0,511
Tempo Modelo	3,718	Tempo Modelo	3,639	Tempo Modelo	3,472	Tempo Modelo	3,553	Tempo Modelo	3,929

Figura 5.11: Execução DNN Total

Observando-se os resultados da figura anterior, verifica-se que a diferença do erro médio entre Linear Regression e DNN não é muito elevada, apesar de se observar que o Linear Regression tem um desvio padrão menor.

De forma a tirar partido das vantagens do **Deep Learning**, pretende-se aumentar o número de camadas ocultas bem como o número de nós até se conseguir melhores resultados. Para tal, foram realizadas várias experiências com diferentes configurações das camadas ocultas, com o intuito de encontrar a que resultasse em melhores valores de erro. A seguinte figura apresenta o erro médio resultante de 30 execuções de cada configuração testada.

07:00			13:00			19:00			
Layers	MAPE	SMAPE	Layers	MAPE	SMAPE	Layers	MAPE	SMAPE	
[10,10]	22,90289	20,46196	[10,10]	7,148389	7,470893	[10,10]	26,15513	30,18441	Tempo médio: 3,572s
[20,10]	19,28903	17,52112	[20,10]	10,40532	11,01775	[20,10]	30,49361	36,05931	Tempo médio: 3,851s
[20,20,20]	17,48958	15,9339	[20,20,20]	7,140193	7,476664	[20,20,20]	27,39683	31,88402	Tempo médio: 3,728s
[20,15,10]	20,38783	18,37519	[20,15,10]	8,236222	8,717738	[20,15,10]	28,72349	33,64002	Tempo médio: 4,059s
[50,50,50]	11,01514	10,31875	[50,50,50]	8,344101	8,871961	[50,50,50]	24,73094	28,48311	Tempo médio: 3,964s
[1024,512,256,128,64,32,16,8]	18,96414	17,15777	[1024,512,256,128,64,32,16,8]	7,833763	8,226515	[1024,512,256,128,64,32,16,8]	23,90411	27,27461	Tempo médio: 18,64s (sem outliers)
[1024,512,256,128,64,32,16,8]	8848,801	29,32677	[1024,512,256,128,64,32,16,8]	28008,05	21,00495	[1024,512,256,128,64,32,16,8]	34712,01	38,78442	Tempo médio: 18,64s (com outliers)
[20,20,20,20,20]	14,48986	13,35296	[20,20,20,20,20]	6,879038	7,268691	[20,20,20,20,20]	26,78605	31,11815	Tempo médio: 4,504s
[50,50,50,50,50]	11,93074	11,04288	[50,50,50,50,50]	6,676022	6,964985	[50,50,50,50,50]	6,202963	6,411594	Tempo médio: 4,903s
[25,25,25,25,25]	10,38108	9,686194	[25,25,25,25,25]	7,445573	7,824636	[25,25,25,25,25]	24,05882	27,55897	Tempo médio: 4,641s
[50,40,30,20,10]	17,34473	15,804	[50,40,30,20,10]	5,372668	5,560192	[50,40,30,20,10]	26,01732	30,0035	Tempo médio: 4,811s (sem outliers)
[50,40,30,20,10]	348,6147	21,81257	[50,40,30,20,10]	1064,07	11,99981	[50,40,30,20,10]	1337,712	35,63636	Tempo médio: 4,811s (com outliers)

Figura 5.12: Comparação configurações Layers DNN

Na figura 5.12 pode-se observar destacado a vermelho a configuração que resultou no erro médio mais baixo, neste caso sendo a configuração de 5 camadas com 50 nós cada. É de notar ainda que este resultado foi obtido sem ser necessário excluir *outliers*. Os resultados apresentados em que foram excluídos *outliers*, foram resultado de uma análise prévia para os conseguir. Como se pode ver na seguinte figura, isto foi conseguido através do desenho de *Boxplots* para os dados obtidos, identificando assim os fortes *outliers* que apresentavam valores com grande discrepância relativamente à média e mediana correspondente.

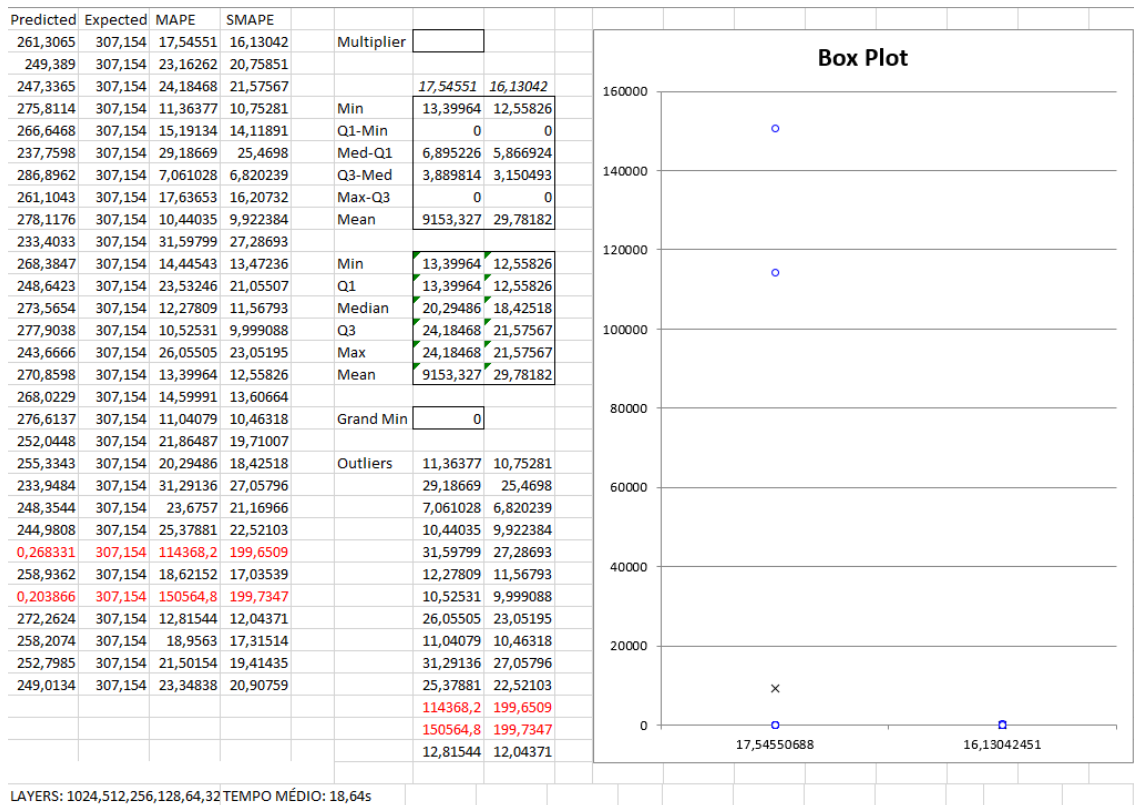


Figura 5.13: Exemplo BoxPlot com Outliers

A figura 5.13 demonstra o exemplo de um *Boxplot* que foi construído para se poder identificar os *outliers* numa das configurações de camadas ocultas testadas. A vermelho encontra-se destacada os *outliers* identificados, e que foram removidos para recalculer o erro médio. A seguinte figura 5.14 apresenta o resultado do *Boxplot* depois de eliminados os *outliers*.

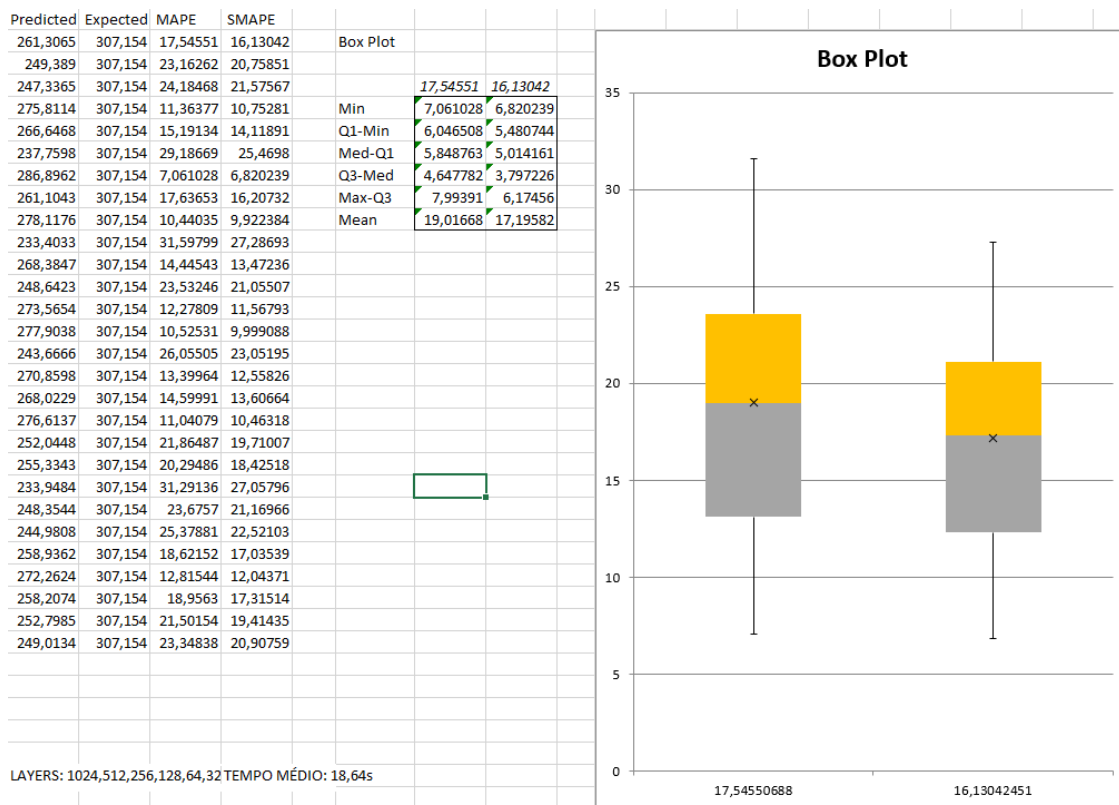


Figura 5.14: Exemplo BoxPlot sem Outliers

Para conseguir obter os resultados obtidos nas figuras 5.13 e 5.14, a ferramenta utilizada para construir os *Boxplots* foi a *Real Statistics Resource Pack* para Excel.

Por fim, a modo de tirar partido de uma das razões principais que levou à emergência do **Deep Learning**, foram realizadas experiências para a previsão do consumo, com um *Dataset* composto por todos os dados históricos existentes na base de dados até ao mês de Fevereiro de 2017 para treino do modelo, em vez de se utilizar apenas o mês de Janeiro. Uma vez que é geralmente aceite que quanto maior a quantidade de dados utilizados para treino, melhores previsões que se conseguem obter, o objetivo foi tentar observar se tal se verificava com os dados disponíveis. Inicialmente foi utilizada a configuração de camadas ocultas com os melhores resultados encontrados (neste caso 5 camadas com 50 nós cada), tendo sido posteriormente ajustada a configuração face aos resultados obtidos.

07:00	Média		13:00	Média		19:00	Média		
Layers	MAPE	SMAPE	Layers	MAPE	SMAPE	Layers	MAPE	SMAPE	
[50,50,50,50,50]	12,1809	13,06854	[50,50,50,50,50]	35,12702	27,46053	[50,50,50,50,50]	31,75963	38,10296	Tempo médio: 5,184s (sem outliers)
[50,50,50,50]	828,5968	43,10787	[50,50,50,50,50]	1449,27	60,18051	[50,50,50,50,50]	453,0304	57,53458	Tempo médio: 5,184s (com outliers)
[50,50,50,50,50,50,50,50,50,50]	18,62941	20,70614	[50,50,50,50,50,50,50,50,50,50]	11,25792	10,26506	[50,50,50,50,50,50,50,50,50,50]	24,65471	28,92951	Tempo médio: 7,102s (sem outliers)
[50,50,50,50,50,50,50,50,50]	13407,12	103,7097	[50,50,50,50,50,50,50,50,50]	28991,09	104,4709	[50,50,50,50,50,50,50,50,50]	30329,36	105,8879	Tempo médio: 7,102s (com outliers)
[100,100,100,100,100,100,100,100,100]	13,24081	14,08387	[100,100,100,100,100,100,100,100,100]	12,07785	11,92874	[100,100,100,100,100,100,100,100,100]	24,6271	28,68191	Tempo médio: 8,453s (sem outliers)
[100,100,100,100,100,100,100,100,100]	519,2613	25,6493	[100,100,100,100,100,100,100,100,100]	579,6011	23,68714	[100,100,100,100,100,100,100,100,100]	780,0415	40,18655	Tempo médio: 8,453s (com outliers)

Figura 5.15: Comparação Layers DNN Base de Dados completa

Na figura 5.15 pode-se observar destacado a vermelho a configuração que resultou no erro médio mais baixo, neste caso sendo a configuração de 10 camadas com 100 nós cada. Ao contrário das configurações para o *Dataset* de Janeiro de 2017, cada uma das configurações testadas apresentava valores de erro muito longe do esperado. O número de resultados que

apresentava erros de MAPE acima de 100% era demasiado alto para se poder afirmar que o modelo tinha boas capacidades para previsão, mesmo ignorando estes "outliers", uma vez que havia casos em que mais de metade dos valores da amostra teriam de ser ignorados para o cálculo do erro. A configuração destacada, para além de apresentar o erro médio mais baixo, apresentava também o menor número de outliers (neste caso apenas 2), mostrando uma aparente correlação entre o tamanho da rede neuronal (número de camadas ocultas e respetivos nós) e a quantidade de dados de treino. Resultante da observação de vários valores de MAPE muito acima dos 100%, foi realizada uma análise aos dados disponíveis na base de dados, com o intuito de perceber o que poderia estar a causar estes valores anormais. Verificou-se assim que os dados mais antigos tinham valores falaciosos e lacunas que resultavam numa distorção do perfil de consumo

5.3 Comparação com trabalhos previamente existentes

Tal como é referido na secção 4.2, pretende-se comparar os resultados da previsão obtidos utilizando metodologias **Deep Learning** com os resultados obtidos através de métodos tradicionais. Para este efeito foram realizadas previsões utilizando os mesmos dados e metodologia referidos em (Jozi et al. 2018). Neste *paper* são realizadas previsões para cada hora do dia utilizando como dados de treino os dados históricos de consumo de cada hora nas 14 semanas anteriores. Ou seja, ao contrário da janela "deslizante" que usa um determinado número de horas imediatamente anteriores à hora que se quer efetuar a previsão, esta metodologia usa o consumo correspondente à mesma hora, no mesmo dia, mas das 14 semanas anteriores. No caso do *paper* mencionado, os dados de treino utilizados são dados históricos de Setembro de 2016 até Novembro de 2016, pretendendo-se fazer a previsão do consumo para cada hora do dia 16 de Novembro. É de mencionar ainda que os dados utilizados são provenientes da mesma base de dados, realizando-se a previsão para HVAC, Lights e Sockets, apresentando o erro para o consumo do total dos três.

Os dados fornecidos foram os mesmos utilizados pelos autores do *paper*, pelo que se torna assim possível fazer uma comparação direta entre os resultados obtidos na execução deste trabalho com os publicados. Nas seguintes tabelas apresentam-se os resultados das execuções dos algoritmos Linear Regression e DNN para o caso de estudo pertinente ao artigo mencionado, onde a configuração de DNN é 5 camadas com 50 nós cada.

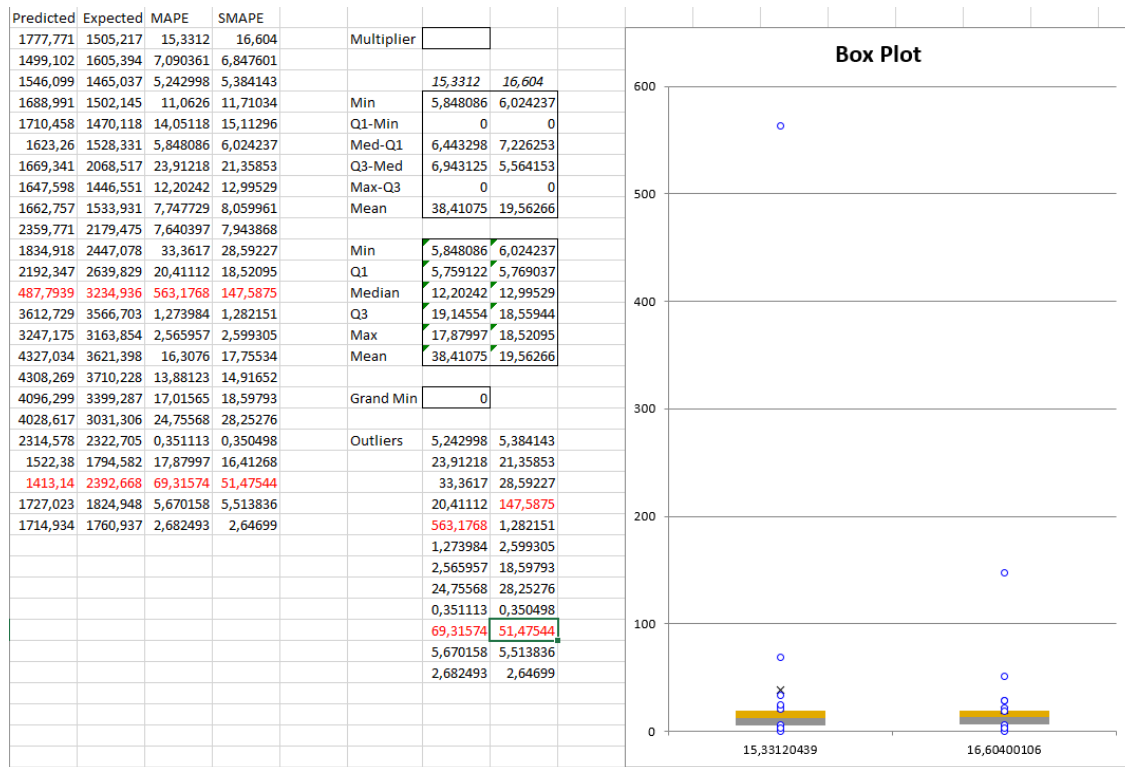


Figura 5.16: Resultados caso estudo Linear Regression

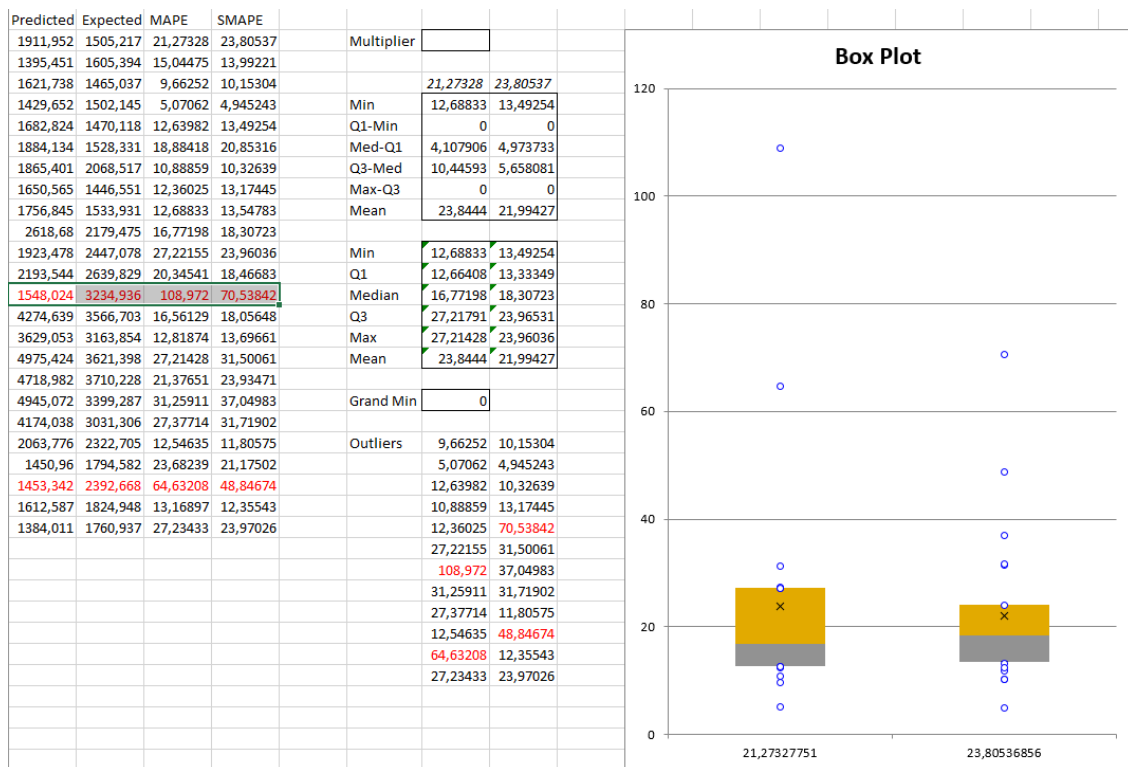


Figura 5.17: Resultados caso estudo DNN

Observando as figuras 5.16 e 5.17 pode-se verificar que os modelos produzidos resultaram em dois outliers, curiosamente nas mesmas horas (12:00 e 21:00). Tal como anteriormente foram eliminados os outliers e recalculado o erro médio como se pode ver nas seguintes figuras.

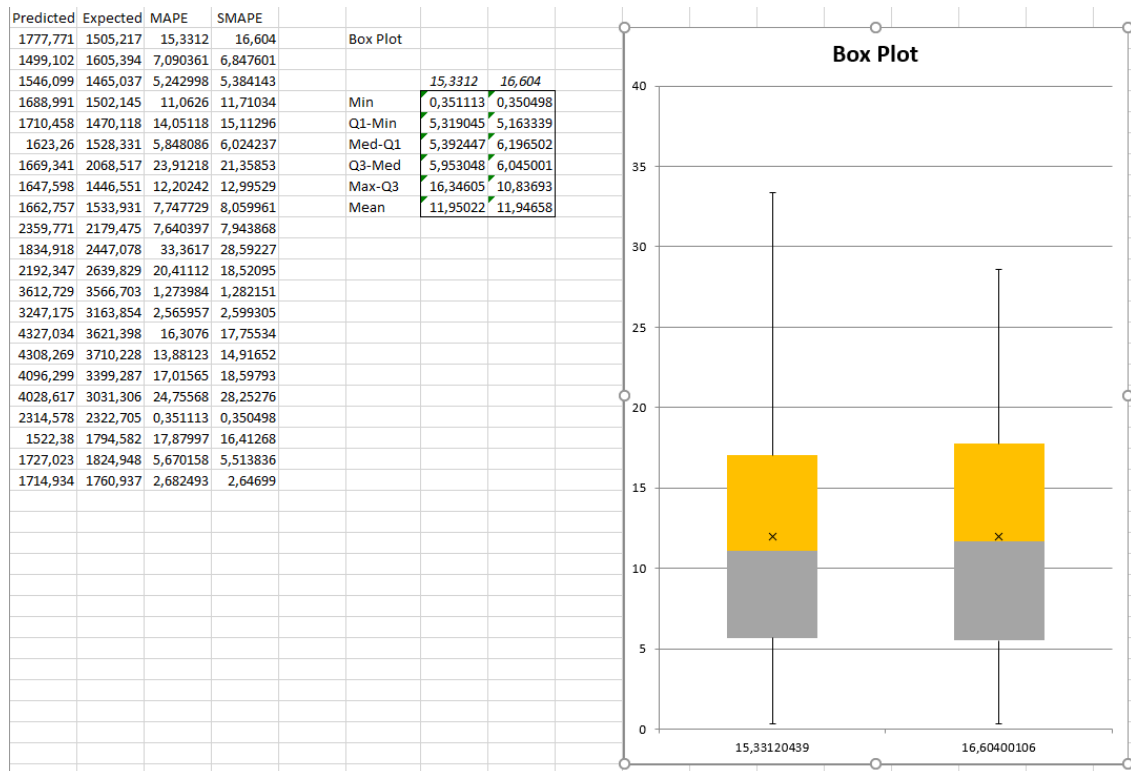


Figura 5.18: Resultados caso estudo Linear Regression sem Outliers

Hour	HyFIS	WM	Mogul
0:00	12.43%	12.03%	12.10%
1:00	15.73%	15.73%	11.39%
2:00	9.73%	9.50%	18.26%
3:00	26.75%	25.93%	16.91%
4:00	25.16%	25.16%	7.17%
5:00	5.71%	5.48%	15.99%
6:00	19.65%	19.59%	24.68%
7:00	0.26%	1.18%	24.85%
8:00	6.18%	6.18%	4.19%
9:00	19.89%	16.72%	7.96%
10:00	21.05%	20.03%	2.38%
11:00	23.81%	23.81%	7.35%
12:00	38.64%	42.11%	1.59%
13:00	28.47%	28.47%	9.39%
14:00	13.46%	14.28%	12.40%
15:00	14.69%	14.69%	7.14%
16:00	52.04%	52.04%	0.32%
17:00	24.64%	24.64%	5.47%
18:00	16.33%	16.82%	10.88%
19:00	9.33%	9.22%	7.91%
20:00	23.12%	23.12%	3.52%
21:00	28.71%	28.61%	15.31%
22:00	3.98%	3.21%	9.69%
23:00	12.48%	12.47%	3.29%
Average	18.84%	18.79%	10.01%

Figura 5.20: Resultados paper MOGUL

Pelos resultados apresentados na figura 5.20 pode-se verificar que apesar do Linear Regression e DNN apresentarem um erro menor do que os métodos HyFIS e WM quando se ignora os *outliers*, o mesmo não acontece para o MOGUL. O Linear Regression consegue um erro perto do erro encontrado no artigo pelo MOGUL, mas acaba por ser maior com uma diferença inferior a 2%. Isto poderá ser devido a um historial pouco enriquecido que não permite tirar partido das capacidades do **Deep Learning**

Para fins científicos e como contribuição para possíveis trabalhos futuros relacionados com o tema da presente tese, encontra-se anexado no apêndice A os resultados de todas as experiências realizadas. Isto compreende resultados obtidos não só em testes das configurações das camadas ocultas, mas também de execuções que foram realizadas, por exemplo, para HVAC, Lights e Sockets mas não foram abordadas neste capítulo.

Capítulo 6

Experimentação e Avaliação

De forma a ser possível avaliar a solução preconizada é necessário definir os parâmetros que vão ser avaliados. No caso abordado neste documento tomam-se como importantes as seguintes características:

- Tempo de Pré-Processamento

Consiste no tempo aplicado na obtenção e preparação dos dados.

- Precisão da previsão

Existem varias métricas para quantificar o erro da precisão. O mais comum em previsões é o MAPE (*Mean Absolute Percentage Error*). O MAPE é uma medida estatística, normalmente utilizada para medir a precisão de métodos de previsão. É representada por uma percentagem e define-se pela fórmula:

$$M = \frac{100}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (6.1)$$

Onde A_t é o valor real, F_t é o valor resultante da previsão e n é o número de valores observados.

Nos casos de estudos contemplados na primeira fase esta será a métrica usada para facilitar a comparação com as experiências já efetuadas (i.e. experiências executadas com metodologias de **Machine Learning**). Na segunda fase prevê-se a abordagem desta e outras métricas (e.g. NMSE, MSE, R_2 , RRMSE (Fagiani et al. 2015)).

- Tempo de execução

Consiste no tempo de execução do modelo. Esta métrica é de grande importância para a validação da melhor solução uma vez que é expectável que as previsões sejam executadas o mais próximo possível do tempo real.

A hipótese que pretende ser testada é a capacidade de os algoritmos implementados serem capazes de prever o consumo de energia, fazendo-o com melhor precisão e em menor tempo que abordagens previamente existentes.

A metodologia de avaliação a utilizar irá ser efetuada através da utilização de grupos de teste, onde para cada algoritmo implementado irão ser executadas um número de experiências, recolhendo os dados referentes a cada uma das características acima identificadas. Os dados recolhidos através das métricas apontadas, irão ser utilizados para verificar qual das

implementações apresenta melhores resultados face à hipótese que se pretende testar. Para tal irá ser utilizado o método estatístico seguinte.

6.1 Teste de Friedman

Um dos testes estatísticos frequentemente utilizado para comparação de vários (i.e mais do que dois) algoritmos no âmbito de **Machine Learning** é o *repeated-measures* ANOVA. No entanto, este é um teste paramétrico que parte de alguns pressupostos que raramente se podem verificar quando se analisa o desempenho de algoritmos de **Machine Learning**. O primeiro pressuposto é que as amostras para testar seguem uma distribuição normal. Regra geral não se consegue garantir que a distribuição dos resultados de precisão seja normal. O segundo pressuposto é a verificação de homogeneidade de variância, o que requer que as variáveis aleatórias escolhidas para teste tenham variância igual. Mais uma vez este pressuposto é difícil de verificar tendo em conta a natureza dos algoritmos de aprendizagem e dos *data sets*. Assim, foi decidido utilizar o teste de Friedman, que se pode dizer que é equivalente ao teste ANOVA, sendo no entanto não-paramétrico. Este teste avalia e classifica os algoritmos testados para cada *data set*, ordenando-os numa escala onde o primeiro classificado é considerado o melhor.

Depois de se verificar que existe uma diferença significativa entre algoritmos, poder-se-à utilizar o teste de Nemenyi, um teste *post-hoc*, para se identificar onde se encontra a diferença entre os algoritmos testados (Demšar 2006).

Infelizmente não foi possível realizar o teste de Friedman durante a realização do presente trabalho, tendo as comparações feitas no capítulo 5 sido conseguidas através da comparação direta do valor médio do MAPE e SMAPE calculados através dos resultados obtidos.

Capítulo 7

Conclusões

Neste capítulo são apresentadas as conclusões e as hipóteses resultantes da elaboração do presente trabalho, bem como uma abordagem de potencial trabalho futuro que poderá ser realizado no âmbito do tema e objetivos enunciados.

Na realização do trabalho descrito na presente tese foram recolhidos dados relativos a diversas implementações e diferentes casos de estudo, sobre os quais se pode observar e retirar algumas conclusões. Dada a natureza de incerteza do trabalho relativamente à escolha de algoritmos/modelos e configurações do mesmos, pode-se ainda ponderar algumas hipóteses relativamente a correlações entre algumas das observações obtidas.

Conclusões retiradas:

1. DNN é capaz de erro médio menor que o Linear Regression com a configuração de camadas ocultas certa

Como se pode verificar na secção 5.2, quando a configuração ótima de camadas ocultas é encontrada o erro médio resultante consegue ser bastante mais baixo do que o conseguido por Linear Regression, com destaque em especial para a previsão das 19:00 onde o LR obtém um valor de MAPE de 27.51% enquanto que o DNN obtém 6.2%.

2. De forma geral, quanto maior o número de camadas e respetivos nós menor o erro

Mais uma vez se pode observar na secção 5.2 e secção de apêndice A.1, durante as experiências para encontrar a melhor configuração de camadas ocultas, sempre que o número de camadas e número de nós total aumenta, o erro encontrado tem tendência a diminuir.

3. A quantidade de dados disponíveis não é suficientemente grande para tirar partido do potencial do **Deep Learning**

Uma das particularidades do **Deep Learning** é a capacidade de tirar proveito de grandes quantidades de dados. A literatura refere frequentemente a tempos de treino na ordem das horas ou mesmo dias. No entanto, durante a execução do presente trabalho, mesmo quando utilizados todos os dados disponíveis, os tempos de treino nunca chegaram aos 20 segundos, tendo o maior valor de tempo médio observado sido a configuração [1024,512,256,128,64,32,16,8] de camadas ocultas. Pondo-se a hipótese de que tal acontece devido a configurações com número de camadas/nós insuficientes, a mesma é invalidada quando os resultados obtidos com mais de 5 camadas apresentarem um número proporcionalmente elevado de previsões com valores de MAPE superiores a 100%.

Hipóteses retiradas:

1. Maior número de dados necessita de maior número de camadas/nós

Quando foram realizadas experiências com todos os dados disponíveis na base de dados, verificou-se que a configuração anteriormente testada apresentava vários valores com MAPE acima de 100%. Em contrapartida, quando se aumentou o número de camadas/nós o número destes valores diminuiu.

2. Execução distribuída poderá não ter impacto positivo

Uma vez que nos resultados obtidos o tempo de treino do modelo nunca foi uma limitação ou uma variável agravante, não se prevê que a execução distribuída dos algoritmos utilizados neste contexto irá trazer benefício em termos temporais. Poderá no entanto ter impacto no erro conseguido.

7.1 Trabalho Futuro

Durante a realização da presente tese foi necessária a mudança de paradigma da arquitetura inicialmente concebida para a atualmente implementada. Como tal, devido a limitações de tempo e das diferenças entre a arquitetura inicial e a adotada não foi possível executar alguns dos casos de estudo conceptualizados. Deste modo, sugere-se como trabalho futuro:

- Implementação de outros algoritmos de **Machine Learning** e **Deep Learning**

De forma a se poder fazer uma comparação clara entre o desempenho de algoritmos tradicionais e algoritmos "deep", sugere-se a implementação de outros algoritmos de **Machine Learning**, nomeadamente o SVM e de **Deep Learning**, por exemplo Recurrent Neural Network (RNN) e Convolutional Neural Network (CNN).

- Execução distribuída dos algoritmos

Apesar de ser normalmente uma medida com o intuito de diminuir o tempo de treino ou processamento dos algoritmos, seria interessante verificar se a execução distribuída dos algoritmos tem impacto na precisão dos mesmos.

- Utilização de mais dados de *input* para treino

A grande parte dos dados de treino são dados históricos e numéricos de consumo, sendo as únicas exceções a hora do dia e indicação se a previsão se trata de um dia da semana ou fim-de-semana. No entanto, se se utilizar dados ambientais (p.e temperatura, previsão atmosférica, exposição solar, estação do ano) bem como indicação sobre férias/feriados e outras variáveis externas, poderá resultar em previsões com maior precisão e demonstrar uma diferença notável entre **Deep Learning** e metodologias de **Machine Learning** tradicionais.

Bibliografia

- A. Koen, Peter et al. (2002). *Fuzzy Front End: Effective Methods, Tools, and Techniques*. Ed. por Paul. Belliveau, Abbie. Griffin e Stephen. Somermeyer, p. 480. isbn: 978-0-471-20611-8.
- Abadi, Martín et al. (2016). *TensorFlow: A system for large-scale machine learning*. Rel. téc. arXiv: 1605.08695v2. url: <https://arxiv.org/abs/1605.08695>.
- Allee, Verna (2002). «A Value Network Approach for Modeling and Measuring Intangibles». Em: November.
- Analytics (2018). *Essentials of Machine Learning Algorithms (with Python and R Codes)*. url: <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/> (acedido em 11/02/2018).
- Ben-David, Shai e Shai Shalev-Shwartz (2014). *Understanding Machine Learning: From Theory to Algorithms*, p. 409. isbn: 9781107057135. doi: 10.1017/CB09781107298019. url: <http://www.cs.huji.ac.il/~7B~%7Dshais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>.
- Chen, Yize et al. (2018). «Model-Free Renewable Scenario Generation Using Generative Adversarial Networks». Em: *IEEE Transactions on Power Systems*, pp. 1–1. issn: 0885-8950. doi: 10.1109/TPWRS.2018.2794541. url: <http://ieeexplore.ieee.org/document/8260947/>.
- CSS et al. (2015). *D3.1. State of the Art*. Rel. téc., pp. 1–88.
- da Silva Luz, Eduardo Jose et al. (mai. de 2018). «Learning Deep Off-the-Person Heart Biometrics Representations». Em: *IEEE Transactions on Information Forensics and Security* 13.5, pp. 1258–1270. issn: 1556-6013. doi: 10.1109/TIFS.2017.2784362. url: <http://ieeexplore.ieee.org/document/8219706/>.
- DeepLearningNet (2018). *Software links « Deep Learning*. url: http://deeplearning.net/software%7B%5C_%7Dlinks/ (acedido em 13/02/2018).
- Demšar, Janez (2006). «Statistical Comparisons of Classifiers over Multiple Data Sets». Em: *Journal of Machine Learning Research* 7, pp. 1–30.
- Energy Agency, International (2003). «The Power to Choose – Demand Response in Liberalised Electricity Markets». Em: url: https://www.schneider-electric.us/documents/solutions1/demand-response-solutions/powertochoose%7B%5C_%7D2003.pdf.
- Fagiani, Marco et al. (jun. de 2015). «Short-term load forecasting for smart water and gas grids: A comparative evaluation». Em: *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*. IEEE, pp. 1198–1203. isbn: 978-1-4799-7993-6. doi: 10.1109/EEEIC.2015.7165339. url: <http://ieeexplore.ieee.org/document/7165339/>.
- Gajowniczek, Krzysztof, Rafik Nafkha e Tomasz Zabkowski (set. de 2017). «Electricity peak demand classification with artificial neural networks». Em: pp. 307–315. doi: 10.15439/2017F168. url: <https://fedcsis.org/proceedings/2017/drp/168.html>.
- Goodfellow, Ian, Yoshua Bengio e Aaron Courville (2016). *Deep Learning*. MIT Press. url: <http://www.deeplearningbook.org>.

- Gulli, Antonio e Sujit Pal (s.d.). «Deep Learning with Keras». Em: (). url: <http://www.zhanjunlang.com/resources/tutorial/Deep%20Learning%20with%20Keras.pdf>.
- InfoWorld (2018). *11 open source tools to make the most of machine learning | InfoWorld*. url: <https://www.infoworld.com/article/2853707/machine-learning/11-open-source-tools-machine-learning.html> (acedido em 12/02/2018).
- InfoWorld_b (2018). *13 frameworks for mastering machine learning | InfoWorld*. url: <https://www.infoworld.com/article/3026262/machine-learning/13-frameworks-for-mastering-machine-learning.html> (acedido em 12/02/2018).
- Jozi, Aria et al. (2018). «Day ahead electricity consumption forecasting with MOGUL learning model». Em: *World Congress on Computational Intelligence* 641794.641794, p. 6.
- KDNuggets (2018). *Top 15 Frameworks for Machine Learning Experts*. url: <https://www.kdnuggets.com/2016/04/top-15-frameworks-machine-learning-experts.html> (acedido em 11/02/2018).
- KDNuggets_R (2018). *Top 20 R Machine Learning and Data Science packages*. url: <https://www.kdnuggets.com/2015/06/top-20-r-machine-learning-packages.html> (acedido em 24/02/2018).
- Khare, Kaustubh et al. (mai. de 2017). «Short term stock price prediction using deep learning». Em: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, pp. 482–486. isbn: 978-1-5090-3704-9. doi: 10.1109/RTEICT.2017.8256643. url: <http://ieeexplore.ieee.org/document/8256643/>.
- Kong, Weicong et al. (jan. de 2018). «Short-Term Residential Load Forecasting Based on Resident Behaviour Learning». Em: *IEEE Transactions on Power Systems* 33.1, pp. 1087–1088. issn: 0885-8950. doi: 10.1109/TPWRS.2017.2688178. url: <http://ieeexplore.ieee.org/document/7887751/>.
- LeCun, Yann, Yoshua Bengio e Geoffrey Hinton (2015). «Deep learning». Em: *Nature* 521.7553, pp. 436–444. issn: 0028-0836. doi: 10.1038/nature14539. arXiv: arXiv:1312.6184v5. url: <http://dx.doi.org/10.1038/nature14539>.
- Lee, Daniel, Daniel Cisek e Shinjae Yoo (ago. de 2017). «Sensor network-based wind field estimation using deep learning». Em: *2017 New York Scientific Data Summit (NYSDS)*. IEEE, pp. 1–3. isbn: 978-1-5386-3161-4. doi: 10.1109/NYSDS.2017.8085047. url: <http://ieeexplore.ieee.org/document/8085047/>.
- Lima, Carlos (2015). «Previsão de consumo de Energia Elétrica em contexto de Smart Grids». Em:
- MathWorks (2018). *Machine Learning in MATLAB - MATLAB & Simulink*. url: <https://www.mathworks.com/help/stats/machine-learning-in-matlab.html> (acedido em 24/02/2018).
- Matijaš, Marin (2013). «ELECTRIC LOAD FORECASTING USING MULTIVARIATE META-LEARNING». Tese de doutoramento. url: https://bib.irb.hr/datoteka/636228.Marin%7B%5C_%7DMatijas%7B%5C_%7D-%7B%5C_%7DPhD%7B%5C_%7Dthesis.pdf.
- Medium (2018). *Top 15 Deep Learning applications that will rule the world in 2018 and beyond*. url: <https://medium.com/@vratulmittal/top-15-deep-learning-applications-that-will-rule-the-world-in-2018-and-beyond-7c6130c43b01> (acedido em 13/02/2018).
- Mitchell, Tom M (1997). «Machine Learning». Em: url: http://personal.disco.unimib.it/Vanneschi/McGrawHill%7B%5C_%7D-%7B%5C_%7DMachine%7B%5C_%7DLearning%7B%5C_%7D-Tom%7B%5C_%7DMitchell.pdf.
- MLlib | Apache Spark* (2017). url: <https://spark.apache.org/mllib/> (acedido em 20/02/2017).

- Murphy, Kevin P (s.d.). «Machine Learning A Probabilistic Perspective». Em: (). url: https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Machine%20Learning%7B%5C_%7D%20A%20Probabilistic%20Perspective%20%7B%5C%7D5BMurphy%202012-08-24%7B%5C%7D5D.pdf.
- NICOLA, SUSANA, EDUARDA PINTO FERREIRA e J J PINTO FERREIRA (2012). «A NOVEL FRAMEWORK FOR MODELING VALUE FOR THE CUSTOMER, AN ESSAY ON NEGOTIATION». Em: *International Journal of Information Technology & Decision Making* 11.03, pp. 661–703. doi: 10.1142/S0219622012500162. url: <http://www.worldscientific.com/doi/abs/10.1142/S0219622012500162>.
- Olivier, Chapelle, Bernhard Schölkopf e Alexander Zien (2006). *Semi-Supervised Learning*. Vol. 1. 2, p. 524. isbn: 9780262033589. doi: 10.1007/s12539-009-0016-2. url: <http://www.ncbi.nlm.nih.gov/pubmed/21243728>.
- Pinheiro, Gil et al. (2017). «Smart Grids Data Management: A Case for Cassandra». Em: *PAAMS'17 - Special Session on "Intelligent and Secure Management towards Smart Buildings and Smart Grids*.
- Porter, Michael E (1985). *Competitive Advantage - Creating and Sustaining Superior Performance*. doi: 10.1182/blood-2005-11-4354.
- Qiu, John X. et al. (jan. de 2018). «Deep Learning for Automated Extraction of Primary Sites From Cancer Pathology Reports». Em: *IEEE Journal of Biomedical and Health Informatics* 22.1, pp. 244–251. issn: 2168-2194. doi: 10.1109/JBHI.2017.2700722. url: <http://ieeexplore.ieee.org/document/7918552/>.
- Rouse, Margaret (2014). *Big Data SMACK*. isbn: 9781484221747. doi: 2024194500. url: <http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data>.
- Salakhutdinov, Ruslan e Geoffrey Hinton (s.d.). «Deep Boltzmann Machines». Em: (). url: <http://proceedings.mlr.press/v5/salakhutdinov09a/salakhutdinov09a.pdf>.
- Santos, Gabriel et al. (2016). «House management system with real and virtual resources: Energy efficiency in residential microgrid». Em: *2016 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 1–6. doi: 10.1109/GIIS.2016.7814943. url: <http://ieeexplore.ieee.org/document/7814943/>.
- Sioshansi, Fereidoon P. (Fereidoon Perry) (2011). *Smart grid : integrating renewable, distributed & efficient energy*. Elsevier/Academic Press, p. 510. isbn: 0123864534. url: <https://books.google.pt/books?id=hqkb9WsB- eoC%7B%5C%7Ddq=smart+grid+vs+microgrid>.
- Soliman, S A e A M Al-Kandari (2010). *Electrical Load Forecasting. Modeling and Model Construction*. url: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.720.3658%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- Sutton, Richard S. e Andrew G. Barto (2012). «Reinforcement learning». Em: 3.9, p. 322. issn: 18726240. doi: 10.1109/MED.2013.6608833. arXiv: 1603.02199. url: <https://books.google.com/books?id=CAFR6IBF4xYC%7B%5C%7Dpgis=1%7B%5C%7D5Cnhttp://incompleteideas.net/sutton/book/the-book.html%7B%5C%7D5Cnhttps://www.dropbox.com/s/f4tnuhipchpkgoj/book2012.pdf>.
- TensorFlow Team (2018a). *Google Developers Blog: Introduction to TensorFlow Datasets and Estimators*. url: <https://developers.googleblog.com/2017/09/introducing-tensorflow-datasets.html> (acedido em 05/10/2018).
- (2018b). *TensorFlow Architecture | TensorFlow*. url: <https://www.tensorflow.org/extend/architecture> (acedido em 05/10/2018).

- Tolosana, Ruben et al. (2018). «Exploring Recurrent Neural Networks for On-Line Handwritten Signature Biometrics». Em: *IEEE Access*, pp. 1–1. issn: 2169-3536. doi: 10.1109/ACCESS.2018.2793966. url: <http://ieeexplore.ieee.org/document/8259229/>.
- TowardsDataScience (2018). *Best Python libraries for Machine Learning and Data Science : (Part -1)*. url: <https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-data-science-part-1-f18242424c38> (acedido em 24/02/2018).
- Ul Hassan, Naveed et al. (nov. de 2015). «Customer Engagement Plans for Peak Load Reduction in Residential Smart Grids». Em: *IEEE Transactions on Smart Grid* 6.6, pp. 3029–3041. issn: 1949-3053. doi: 10.1109/TSG.2015.2404433. url: <http://ieeexplore.ieee.org/document/7055261/>.
- United States Department of Energy (2008). *The Smart Grid: An Introduction*. Rel. téc. url: https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE%7B%5C_%7DSG%7B%5C_%7DBook%7B%5C_%7DSingle%7B%5C_%7DPages%7B%5C%7D281%7B%5C%7D29.pdf.
- Vinagre, Eugenia, Luis Gomes e Zita Vale (2016). «Electrical energy consumption forecast using external facility data». Em: *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015* 12004, pp. 659–664. doi: 10.1109/SSCI.2015.101.
- Vinagre, Eugenia, Tiago Pinto et al. (2017). «Big Data in efficient Smart Grids management». Em: *PAAMS'17 - Doctoral Consortium*.
- Vincent PASCALVINCENT, Pascal e Hugo Larochelle LAROCHEH (2010). «Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol». Em: *Journal of Machine Learning Research* 11, pp. 3371–3408. url: <http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf>.
- Wang, Huaizhi et al. (2018). «Deep Learning Based Interval State Estimation of AC Smart Grids against Sparse Cyber Attacks». Em: *IEEE Transactions on Industrial Informatics*, pp. 1–1. issn: 1551-3203. doi: 10.1109/TII.2018.2804669. url: <http://ieeexplore.ieee.org/document/8288611/>.
- Woodall, Tony (2003). «Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis». Em: *Academy of Marketing Science Review* 12.5, pp. 1–42. issn: 14705931.
- Zeithaml, Valarie A. (1988). «Consumer Perceptions A Means-End Value : Quality , and and Model Synthesis of Evidence». Em: 52.July, pp. 2–22.
- Zhang, Shu et al. (mar. de 2018). «DeMeshNet: Blind Face Inpainting for Deep Mesh-Face Verification». Em: *IEEE Transactions on Information Forensics and Security* 13.3, pp. 637–647. issn: 1556-6013. doi: 10.1109/TIFS.2017.2763119. url: <http://ieeexplore.ieee.org/document/8067496/>.
- Zheng, Wei-Long et al. (2018). «EmotionMeter: A Multimodal Framework for Recognizing Human Emotions». Em: *IEEE Transactions on Cybernetics*, pp. 1–13. issn: 2168-2267. doi: 10.1109/TCYB.2018.2797176. url: <http://ieeexplore.ieee.org/document/8283814/>.

Apêndice A

Experiências e dados adicionais

A.1 Experiências configuração DNN

Nesta secção são apresentados os resultados de cada uma das experiências realizadas para encontrar a configuração com melhores resultados, como referido na sub-secção 5.2 deste documento. Os testes foram realizados utilizando o *Dataset* de Janeiro de 2017 como dados de treino, tendo sido efetuado a previsão para o dia de 1 de Fevereiro de 2017 para as 7:00, 13:00 e 19:00.

Predicted	Expected	MAPE	SMAPE		Predicted	Expected	MAPE	SMAPE		Predicted	Expected	MAPE	SMAPE
244,407	307,154	25,67318	22,75253		1078,105	973,389	9,712936	10,20872		1717,493	1205,862	29,7894	35,00299
244,8337	307,154	25,45413	22,58032		1021,987	973,389	4,755229	4,871044		1490,348	1205,862	19,08856	21,10265
262,2022	307,154	17,14397	15,79042		1122,093	973,389	13,25238	14,19282		1656,27	1205,862	27,19411	31,47359
246,9399	307,154	24,3841	21,73425		1030,711	973,389	5,561364	5,720431		1625,717	1205,862	25,82584	29,6552
253,6061	307,154	21,11458	19,09831		1071,205	973,389	9,131426	9,568287		1690,499	1205,862	28,66829	33,46524
248,3337	307,154	23,68597	21,17788		1074,363	973,389	9,398503	9,86194		1712,549	1205,862	29,5867	34,72346
251,0353	307,154	22,35488	20,10739		1054,666	973,389	7,706411	8,015256		1552,834	1205,862	22,34443	25,15477
259,4901	307,154	18,3683	16,82323		1066,394	973,389	8,72141	9,119066		1612,209	1205,862	25,20437	28,83867
245,1107	307,154	25,31238	22,4687		1084,196	973,389	10,22018	10,77057		1676,34	1205,862	28,06578	32,64712
245,7495	307,154	24,98665	22,21167		1037,251	973,389	6,156805	6,352356		1580,654	1205,862	23,71118	26,90038
242,7035	307,154	26,55522	23,4426		1049,963	973,389	7,292998	7,569002		1654,842	1205,862	27,13131	31,3895
267,0197	307,154	15,03047	13,97985		1079,718	973,389	9,847851	10,35786		1678,83	1205,862	28,17246	32,79156
266,2628	307,154	15,35748	14,26231		1047,166	973,389	7,045398	7,302649		1721,152	1205,862	29,93867	35,20927
242,1177	307,154	26,86143	23,68091		1051,105	973,389	7,393741	7,677571		1751,839	1205,862	31,16592	36,91898
258,2442	307,154	18,93937	17,30102		1004,208	973,389	3,068969	3,116796		1656,919	1205,862	27,22263	31,5118
259,1825	307,154	18,50876	16,94098		1012,519	973,389	3,864617	3,940765		1639,768	1205,862	26,4614	30,49627
273,6617	307,154	12,23857	11,53284		1095,064	973,389	11,11123	11,76484		1677,397	1205,862	28,1111	32,70845
240,7588	307,154	27,5775	24,2357		1093,08	973,389	10,94987	11,58409		1464,476	1205,862	17,65914	19,36937
253,3188	307,154	21,25198	19,21066		1043,738	973,389	6,740137	6,975207		1712,159	1205,862	29,57067	34,70139
249,323	307,154	23,19524	20,78471		1028,182	973,389	5,329082	5,474964		1653,79	1205,862	27,08492	31,32743
259,2023	307,154	18,49973	16,93342		1070,858	973,389	9,101946	9,535923		1725,931	1205,862	30,13265	35,47786
249,6014	307,154	23,05781	20,67429		1042,014	973,389	6,585797	6,810044		1672,287	1205,862	27,89146	32,41148
244,0793	307,154	25,84188	22,88493		1029,008	973,389	5,405136	5,555271		1643,885	1205,862	26,64558	30,74116
241,6774	307,154	27,09255	23,86036		997,5569	973,389	2,422707	2,452415		1503,081	1205,862	19,77399	21,94355
239,4067	307,154	28,29799	24,7904		1011,588	973,389	3,776097	3,848763		1645,82	1205,862	26,73182	30,856
240,8	307,154	27,55567	24,21884		1013,895	973,389	3,995084	4,076514		1655,815	1205,862	27,17413	31,44683
257,945	307,154	19,07735	17,41608		1136,961	973,389	14,38677	15,50188		1686,317	1205,862	28,49137	33,22442
242,1464	307,154	26,84643	23,66926		1011,381	973,389	3,756429	3,828333		1503,192	1205,862	19,77991	21,95084
224,7176	307,154	36,68445	30,99861		973,2608	973,389	0,013172	0,013171		1523,316	1205,862	20,83967	23,26371
255,6664	307,154	20,1386	18,29629		1055,141	973,389	7,74799	8,060243		1612,037	1205,862	25,19639	28,82823
LAYERS: 10,10					TEMPO MÉDIO: 3,572s								

Figura A.1: Teste configuração [10,10]

A.3 Dados adicionais do caso de estudo de 5.3

Nesta secção são apresentados os resultados das previsões efetuadas para HVAC, Lights e Sockets utilizando a metodologia e *Dataset* referente ao caso de estudo abordado na secção 5.3 do presente documento. Os dados são referentes à execução do Linear Regression e do DNN (5 camadas com 50 nós cada) para previsão do consumo de cada hora do dia 16 de Novembro de 2016.

Predicted	Expected	MAPE	SMAPE
490,0307	419,3433	14,4251	15,54639
471,9769	454,347	3,73533	3,806421
418,9219	397,6938	5,06733	5,199057
454,4146	429,1154	5,56742	5,726839
513,9378	393,2239	23,48804	26,61354
450,5097	394,6209	12,40569	13,22608
444,4587	418,1624	5,916474	6,096833
507,0731	394,6423	22,17251	24,93709
426,5911	467,9229	9,688868	9,241185
430,2839	395,0489	8,188789	8,538384
473,8661	454,5986	4,066014	4,150392
516,5629	468,0507	9,391339	9,854052
835,2544	452,3167	45,84683	59,48218
456,7926	468,5461	2,573061	2,540378
604,7703	558,9467	7,577024	7,875384
325,8045	559,0352	71,5861	52,71706
209,6717	543,6287	159,2761	88,66502
383,2456	520,4203	35,7929	30,35961
481,5925	453,5656	5,819625	5,99404
446,4098	452,8967	1,453112	1,442631
431,3412	467,9732	8,492577	8,146647
377,6498	443,3589	17,39949	16,00693
437,3118	435,0428	0,518843	0,520192
431,6007	425,1402	1,496875	1,508163

Figura A.14: Resultados caso estudo Linear Regression para HVAC

Predicted	Expected	MAPE	SMAPE
10,3777	0	100	200
11,80393	0	100	200
3,426442	0	100	200
0,002557	0	100	200
0,000881	0	100	200
0,000806	0	100	200
0,000784	0	100	200
0,011179	0	100	200
-215,731	0	100	200
120,9236	535,6353	342,9535	126,3289
-310,571	690,7359	322,4086	200
598,1033	684,6788	14,47501	13,49808
626,1655	712,0501	13,71597	12,8357
511,3813	727,1075	42,18499	34,837
138,0342	701,6064	408,2846	134,2413
659,4188	978,5169	48,39082	38,96346
1010,431	1223,199	21,05723	19,05138
33,15718	1042,19	3043,18	187,6664
373,1967	1020,014	173,3181	92,85277
346,1992	484,9233	40,07062	33,38236
46,74963	0	100	200
1210,715	0	100	200
3,655335	0	100	200
1,205258	0	100	200

Figura A.15: Resultados caso estudo Linear Regression para Lights

Predicted	Expected	MAPE	SMAPE
1199,545	1085,874	9,476184	9,947507
1232,97	1151,047	6,644382	6,872706
1072,463	1067,343	0,477379	0,478521
1565,165	1073,029	31,44304	37,30851
1469,47	1076,894	26,71547	30,83422
1000,624	1133,71	13,30023	12,4709
1189,416	1650,354	38,75335	32,46308
1184,849	1051,909	11,21995	11,8868
1104,615	1066,008	3,495091	3,557256
1738,591	1248,791	28,17224	32,79126
1437,679	1301,743	9,455223	9,924411
1661,042	1487,1	10,47188	11,05047
1534,053	2070,569	34,97375	29,76822
1454,559	2371,05	63,00818	47,91347
2127,258	1903,301	10,52796	11,11295
1924,135	2083,846	8,300407	7,96965
2548,981	1943,4	23,75776	26,96035
1508,367	1836,677	21,76591	19,62963
1443,722	1557,726	7,896535	7,596601
1043,195	1384,885	32,7542	28,14488
1537,767	1326,608	13,73148	14,74375
1094,593	1949,309	78,08523	56,15921
1441,818	1389,906	3,600509	3,666516
1616,637	1335,797	17,37187	19,02431

Figura A.16: Resultados caso estudo Linear Regression para Sockets

Predicted	Expected	MAPE	SMAPE
427,3058	419,3433	1,863424	1,880949
470,5364	454,347	3,440608	3,500833
443,55	397,6938	10,33846	10,90201
449,2341	429,1154	4,478436	4,581015
432,3684	393,2239	9,053508	9,482769
462,8595	394,6209	14,74283	15,91607
433,6274	418,1624	3,566429	3,631181
500,854	394,6423	21,20613	23,72132
442,5194	467,9229	5,740667	5,580488
391,6098	395,0489	0,878193	0,874353
400,1409	454,5986	13,60963	12,74253
530,2505	468,0507	11,73027	12,46113
521,6464	452,3167	13,29056	14,23662
464,5509	468,5461	0,860017	0,856335
550,083	558,9467	1,611333	1,598455
541,0873	559,0352	3,317007	3,262892
310,6743	543,6287	74,98351	54,53673
403,8497	520,4203	28,86484	25,22435
726,3784	453,5656	37,55794	46,24164
348,1306	452,8967	30,09389	26,15792
422,7787	467,9732	10,68987	10,14749
371,6565	443,3589	19,29266	17,59535
385,1924	435,0428	12,9417	12,15516
390,4598	425,1402	8,881949	8,504276

Figura A.17: Resultados caso estudo DNN para HVAC

Predicted	Expected	MAPE	SMAPE
13,76127	0	100	200
61,07009	0	100	200
0,681337	0	100	200
-3,73E-09	0	100	200
0,001674	0	100	200
1,38E-07	0	100	200
2,24E-08	0	100	200
0,012978	0	100	200
240,0521	0	100	200
102,3064	535,6353	423,5597	135,8522
335,1601	690,7359	106,0914	69,32006
721,7459	684,6788	5,135752	5,271107
249,2963	712,0501	185,6241	96,27204
52,43518	727,1075	1286,679	173,0944
140,4371	701,6064	399,5878	133,2875
484,7532	978,5169	101,8588	67,4877
723,9526	1223,199	68,96126	51,2797
121,6233	1042,19	756,9001	158,1984
697,1227	1020,014	46,31773	37,60812
276,5687	484,9233	75,33556	54,72272
4,145196	0	100	200
816,2243	0	100	200
1,112133	0	100	200
0,587592	0	100	200

Figura A.18: Resultados caso estudo DNN para Lights

Predicted	Expected	MAPE	SMAPE
1114,534	1085,874	2,571476	2,604969
1409,697	1151,047	18,34795	20,2012
1181,552	1067,343	9,66602	10,1569
1313,566	1073,029	18,31176	20,15734
1440,229	1076,894	25,22757	28,86905
956,7125	1133,71	18,50056	16,93411
1092,835	1650,354	51,0158	40,64748
1020,49	1051,909	3,07879	3,032113
1009,017	1066,008	5,648192	5,493063
1322,513	1248,791	5,574395	5,734219
1392,681	1301,743	6,529687	6,750067
1519,212	1487,1	2,113762	2,136341
1504,104	2070,569	37,66134	31,69328
1772,658	2371,05	33,75675	28,88195
2343,98	1903,301	18,80047	20,75113
2505,14	2083,846	16,81719	18,36111
2530,182	1943,4	23,19127	26,23318
1371,998	1836,677	33,8688	28,96393
1210,205	1557,726	28,71582	25,11048
1015,042	1384,885	36,43626	30,82121
1299,957	1326,608	2,050168	2,029365
1029,909	1949,309	89,27	61,72089
1227,962	1389,906	13,18799	12,37217
1697,473	1335,797	21,30671	23,84724

Figura A.19: Resultados caso estudo DNN para Sockets