Universidade do Minho
Escola de Engenharia

Tiago Azevedo Lima

HTM approach to Image Classification, Sound Recognition and Time Series Forecasting

February 2021

Universidade do Minho
Escola de Engenharia

Tiago Azevedo Lima

HTM approach to Image Classification, Sound Recognition and Time Series Forecasting

Master dissertation
Master Degree in Biomedical Engineering

Dissertation oriented by
Professor Doutor José Manuel Ferreira Machado

February 2021

DECLARAÇÃO

Nome: Tiago Azevedo Lima

Titulo da Dissertação: HTM approach to Image Classification, Sound Recognition and Time Series Forecasting

Mentor: Professor Doutor José Manuel Ferreira Machado

Supervisor: Professor Doutor José Manuel Ferreira Machado

Ano de Conclusão: 2021

Designação do Curso: Mestrado Integrado em Engenharia Biomédica

Ramo de Mestrado: Informática Médica

Eu declaro que concedo à Universidade do Minho e aos seus agentes uma licença não exclusiva para arquivar e disponibilizar através do seu repositório, nas condições indicadas abaixo, a minha dissertação, como um todo ou parcialmente, em apoio digital.

Eu declaro que autorizo a Universidade do Minho a arquivar mais de uma cópia da dissertação e, sem alterar o seu conteúdo, a converter a dissertação em qualquer formato ou suporte, para fins de preservação e acesso.

Além disso, eu retenho todos os direitos autorais relacionados com a dissertação e o direito de usá-lo em trabalhos futuros.

Eu autorizo a reprodução parcial desta dissertação para fins de investigação por meio de uma declaração escrita da pessoa interessada ou entidade.

Universidade do Minho, _______/_______/_________

Assinatura: ____

ACKNOWLEDGEMENTS

This project not only represents a final step in a long journey of my learning process in the Integrated Masters in Biomedical Engineering at the University of Minho, but also the beginning of new paths for my academic and professional life. I conclude this journey with a sense of completeness, pride and joy after many long days and nights of work, effort and dedication. This feeling is accentuated when I realize that I could reconcile this project with a full-time job, working in two fronts that gave me a lot of knowledge in two completely different branches of Informatics.

However, this journey could not be concluded on my own. Some acknowledgments are required to the ones that stood with me. Firstly, I need to thank my project advisor, Professor Doutor José Machado, for the sharing of his knowledge, guiding and motivating me to research and explore the theory at study.

To my friends and colleagues, that walked this path along with me, everyday for five years, sharing ideas, concerns and, most importantly, laughs. To my work colleagues, that in this last year, even though many work days were tiresome, always motivated me to conclude this project and to not give up to fatigue.

Lastly, but not least, to the most important people in my life, my family and my soulmate. To my love, I appreciate all the loving, caring and understanding that you gave me for the last five years. To my family, for supporting me everyday of my life, for giving me a home, safety and peace, for everything you taught me and much more; without you, I would not have walked this path. For these special people, thank you for believing in me, I will do my best not to disappoint you and always fight for my dreams, with you by my side. I will always be with you.

ABSTRACT

The introduction of Machine Learning (ML) on the orbit of the resolution of problems typically associated within the human behaviour has brought great expectations to the future. In fact, the possible development of machines capable of learning, in a similar way as of the humans, could bring grand perspectives to diverse areas like healthcare, the banking sector, retail, and any other area in which we could avoid the constant attention of a person dedicated to the solving of a problem; furthermore, there are those problems that are still not at the hands of humans to solve - these are now at the disposal of intelligent machines, bringing new possibilities to the humankind development.

ML algorithms, specifically Deep Learning (DL) methods, lack a bigger acceptance by part of the community, even though they are present in various systems in our daily basis. This lack of confidence, mandatory to let systems make big, important decisions with great impact in the everyday life is due to the difficulty on understanding the learning mechanisms and previsions that result by the same - some algorithms represent themselves as ”black boxes”, translating an input into an output, while not being totally transparent to the outside. Another complication rises, when it is taken into account that the same algorithms are trained to a specific task and in accordance to the training cases found on their development, being more susceptible to error in a real environment - one can argue that they do not constitute a true Artificial Intelligence (AI).

Following this line of thought, this dissertation aims at studying a new theory, Hierarchical Temporal Memory (HTM), that can be placed in the area of Machine Intelligence (MI), an area that studies the capacity of how the software systems can learn, in an identical way to the learning of a human being. The HTM is still a fresh theory, that lays on the present perception of the functioning of the human neocortex and assumes itself as under constant development; at the moment, the theory dictates that the neocortex zones are organized in an hierarchical structure, being a memory system, capable of recognizing spatial and temporal patterns. In the course of this project, an analysis was made to the functioning of the theory and its applicability to the various tasks typically solved with ML algorithms, like image classification, sound recognition and time series forecasting. At the end of this dissertation, after the evaluation of the different results obtained in various approaches, it was possible to conclude that even though these results were positive, the theory still needs to mature, not only in its theoretical basis but also in the development of libraries and frameworks of software, to capture the attention of the AI community.

Keywords: Hierarchical Temporal Memory; Machine Intelligence; Neocortex; Hebbian Learning; Image Classification; Sound Recognition; Time Series Forecasting; Artificial Intelligence

RESUMO

A introdução de ML na órbita da resolução de problemas tipicamente dedicados ao foro humano trouxe grandes expectativas para o futuro. De facto, o possível desenvolvimento de máquinas capazes de aprender, de forma semelhante aos humanos, poderia trazer grandes perspetivas para diversas áreas como a saúde, o setor bancário, retalho, e qualquer outra área em que se poderia evitar o constante alerta de uma pessoa dedicada a um problema; para além disso, problemas sem resolução humana passavam a estar à mercê destas máquinas, levando a novas possibilidades no desenvolvimento da humanidade.

Apesar de se encontrar em vários sistemas no nosso dia-a-dia, estes algoritmos de ML, especificamente de DL, carecem ainda de maior aceitação por parte da comunidade, devido à dificuldade de perceber as aprendizagens e previsões resultantes, feitas pelos mesmos - alguns algoritmos apresentam-se como "caixas negras", traduzindo um input num output, não sendo totalmente transparente para o exterior - é necessária confiança nos sistemas que possam tomar decisões importantes e com grandes impactos no quotidiano; por outro lado, os mesmos algoritmos encontram-se treinados para uma tarefa específica e de acordo com os casos encontrados no desenvolvimento do seu treino, sendo mais susceptíveis a erros em ambientes reais, podendo se discutir que não constituem, por isso, uma verdadeira Inteligência Artificial.

Seguindo este segmento, a presente dissertação procura estudar uma nova teoria, HTM, inserida na área de MI, que pretende dar a capacidade aos sistemas de software de aprenderem de uma forma idêntica à do ser humano. Esta recente teoria, assenta na atual percepção do funcionamento do neocórtex, estando por isso em constante desenvolvimento; no momento, é assumida como uma teoria que dita a hierarquização estrutural das zonas do neocórtex, sendo um sistema de memória, reconhecedor de padrões espaciais e temporais. Ao longo deste projeto, foi feita uma análise ao funcionamento da teoria, e a sua aplicabilidade a várias tarefas tipicamente resolvidas com algoritmos de ML, como classificação de imagem, reconhecimento de som e previsão de séries temporais. No final desta dissertação, após uma avaliação dos diferentes resultados obtidos em várias abordagens, foi possível concluir que apesar dos resultados

positivos, a teoria precisa ainda de maturar, não só a nível teórico como a nível prático, no desenvolvimento de bibliotecas e frameworks de software, de forma a capturar a atenção da comunidade de Inteligência Artificial.

Palavras-chave: Hierarchical Temporal Memory; Machine Intelligence; Neocórtex; Aprendizagem Hebbiana; Classificação de Imagem; Reconhecimento de Som; Previsão de Séries Temporais; Inteligência Artificial

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AAE Absolute Average Error. 27, 32, 34
AI Artificial Intelligence. v, vi, 1–3, 5
ANN Artificial Neural Network. 13
ARIMA Autoregressive integrated moving average. 13

CNN Convolutional Neural Network. 13
CSV Comma-separated values. 15

DL Deep Learning. v, vii, 1, 2, 6, 12, 13, 31, 37

FFT Fast Fourier Transform. 18, 20

GRU Gated Recurrent Unit. 13

HTM Hierarchical Temporal Memory. v, vii, xi, 1–4, 6, 8, 9, 12–17, 19, 22, 26, 27, 31–33, 35, 37, 38, 45

IoT Internet of Things. 12

kNN K-Nearest Neighbor. 13, 16

LSTM Long Short-Term Memory. 13

MAPE Mean Average Percentage Error. 13, 27, 32–34
MFCC Mel-Frequency Cepstral Coefficients. xi, 18–20
MI Machine Intelligence. v, vii, 15
ML Machine Learning. v, vii, 1–3, 5, 6, 14, 32, 38

NAB Numenta Anomaly Benchmark. 14
NuPIC Numenta Platform for Intelligent Computing. 4, 15–18, 31, 32

RMSE Root Mean Square Error. 27, 32, 34
RNN Recurrent Neural Network. 13

SDR Sparse Distributed Representation. 2, 9–12, 16, 17, 19, 22, 37, 38

SIFT Scale-Invariant Feature Transform. 13
SP Spatial Pooler. xiii, 10, 13, 16, 17, 19–21, 26, 28, 31
SVM Support Vector Machine. 13

TM Temporal Memory. xiii, 11, 20, 21, 26, 28, 38

WHO World Health Organization. 23, 28, 32

1

INTRODUCTION

The following dissertation project describes the HTM theory and its connection to the biological human neocortex, as well how it can be used in the resolution of tasks commonly approached by ML algorithms; as the use of ML and more specifically DL techniques is booming in the healthcare area, it is expected that any conclusions taken from this theory can be used in medical environments. The work is framed in the dissertation of the masters in Medical Informatics, of the Integrated Masters in Biomedical Engineering in the University of Minho. In this chapter, it is presented the contextualization of this project and its motivation; the objectives are also pointed as a guide to the work proposed and the answers that are supposed to be obtained, with no conflicts of interest. The last section of this chapter pretends to give an overall idea of how the dissertation is structured.

1.1 CONTEXTUALIZATION AND MOTIVATION

The XX century digital revolution brought some major perspectives of intelligent machines capable of solving problems, otherwise, typically solved by humans; it was the beginning of AI. AI is a branch of computer science that looks to mimic human behaviours, in which learning is one of them [1]. For a system to learn a determined task without being explicitly programmed to, meaning, learning by experience and inference, the area of ML was developed and largely studied [2]. ML technics are used in different areas such as computer vision, speech recognition, natural language processing, recommendation of contents, amongst others [3, 4]. These technics brought some good results, although not the ones expected, being limited when used on raw data, with no feature extraction giving meaning to the data [5]; in the beginning of the second decade of the XXI century and with the development of technology, introducing new and faster

CPU's and GPU's [6], a branch of ML, denominated DL, emerged with significant impact [3]. These ML methods allowed to transform the data representation, from its most raw state to a more abstract [5]. However, this classic approach of AI ends up trying to build an intelligence based on rules and structural data of human knowledge, solving specific problems for which the system was built, being very limited in cases not supported in its training environment; this learning does not offer a response to the question: how to create a true artificial intelligence? With this goal to mind, it turns imperative to first understand on how the human neocortex - part of the brain involved in perception, cognition, motor skills and more - works. Only then it will be possible to create machines that can learn and adapt similarly to the human brain.

In search for a true AI, the HTM theory, first tries to describe, biologically, how the neocortex works; with this knowledge, it pretends to convert it into a way of creating intelligent machines. Since the study of the neocortex is still incomplete, the theory is itself in evolution. HTM is built in three main features of the neocortex: it is a memory system, with temporal patterns and its regions are organized in an hierarchical structure. There are many biological details that the theory simply does not aboard in case it has no relevance for learning. In short, this approach includes Sparse Distributed Representation (SDR)s, its semantical and mathematical operations, neurons along the neocortex capable of learning sequences and enabling predictions [7, 8, 9]; these systems learn in a continuous way, with new inputs through time and with flows of information top-down and bottom-up between its hierarchical layers, making them efficient in detecting temporal anomalies. The theory relies on the fact that by mimicking the neocortex, through the encoding of data in a way that gives it a semantic meaning, activating neurons sparsely in an SDR through time, will give these systems a power to generalize and learn, not achieved to date on other classic approaches of AI; it is expected to achieve better results and conclusions, while being an intelligence with higher flexibility when put up against adverse contexts.

1.2 OBJECTIVES

The idea of this thesis project was born from this scope, having the objective to study applications of the HTM theory, still largely unknown to the pattern learning and recognition community; the applications in study range from audio recognition, image classification and time series forecasting with public datasets, that may someday help in anomaly detections in medicine, hospital management or to act in case of urgency matters. In order to have the confidence to use these systems daily, there is the need for the introduction of new technologies, supported by an AI system with an higher generalization capacity to the ones already in place. Having this in mind, the objectives for this dissertation are the following:

- Investigate and understand the Hierarchical Temporal Memory theory;
- Test and analyse the applications of the HTM theory;
- Compare the HTM theory results against traditional ML technics in terms of:
  - Accuracy and other classification or regression metrics;
  - Computing power/time required;
  - Amount and type of data required;
  - Noise robustness of the algorithms;
  - Possibility to justify the obtained results.

1.3 Dissertation's structure

The dissertation is structured in five chapters: Introduction, State of the Art, Methodology and Development Tools, Results and Discussion and Conclusions and Future Work. The purpose of each chapter is described bellow.

Chapter 1 – Introduction: The first chapter of this dissertation pretends to contextualize and show the motivations behind its writing as well as to define the objectives and questions that it pretends to investigate and answer.

Chapter 2 – State of the Art: The State of the Art chapter gives a comprehension to the reader of the ML domain and how it differs from the principles of the HTM, the theory in study. Some of the concepts presented include how the human neocortex works and what is Hebbian Learning and how they relate to the HTM theory. At the end of this chapter some definitions, properties and real life applications of this evolving theory are presented.

Chapter 3 – Methodology and Development Tools: After presenting the key perceptions of the theory, the third chapter aims at describing how the theory is put into practice using the Python library, Numenta Platform for Intelligent Computing (NuPIC); also, the networks developed for the three tasks proposed are described - firstly, for the Image Classification task, that uses the MNIST dataset; then for the Sound Recognition challenge, where it is used the Spoken Digit dataset; lastly, it is described the network and methodology for the Time Series Forecasting task, that uses a Stock Market dataset obtained through Yahoo Finance.

CHAPTER 4 – RESULTS AND DISCUSSION: The fourth chapter, Results and Discussion, shows the results obtained in the tasks proposed in the previous chapter; these results are also discussed, analyzed and compared to other methods.

CHAPTER 5 – CONCLUSIONS AND FUTURE WORK: The final chapter, summarizes the conclusions taken from the study of the HTM theory and its applications in the various tasks. It is also shown the prospects for future work and how the theory can still evolve and help in the development of intelligent machines.

2

STATE OF THE ART

2.1 MACHINE LEARNING

Machine Learning may be described as the discipline focused on how to create computer systems capable of learning and automatically improve through experience. These systems differ from hard computing, making it tolerant to imprecision, uncertainty, partial truth and approximations [10]. For the past two decades, ML became the method of choice to develop AI systems for computer vision, speech recognition, natural language processing and others [4]. These systems are trained by showing it examples of desired input-output behaviours and not by anticipating and computing the desired response for all possible inputs – soft computing against hard computing. In these systems, the machine learns with experience, inferring plausible models to explain the observed data [11]. The most widely used ML methods can be categorized as supervised learning methods. The systems that use these methods are typically applied in spam classifiers, image recognition, medical diagnostics or even in the prediction of stock market prices [4, 5]. These methods form their predictions via a learned mapping function, which produces an output (or a probability distribution) for a given input – there are a variety of proposed learning algorithms to achieve the best learning mapping. Each of these algorithms will provide different trade-offs between visualization and explanation, computational complexity, amount of data or accuracy of the learned mapping (in relation to a test dataset). Unsupervised learning methods are part of another category, which involve the analysis of unlabeled data, giving it some meaning through its structural properties. Examples of these methods include dimension reduction, where the number of variables is reduced, or clustering, where the observed data is partitioned into clusters formed by semantically identical data; nowadays large amounts of data are generated on a daily basis in the health sector (amongst other sectors), mostly unlabeled data, but most of the time, this data is not used, not translating into knowledge [12, 13, 14, 15] - unsupervised learning methods can be useful for a data mining approach, like in a nutritional follow-up [16] or in predicting early stages of chronic kidney disease [17]. The third paradigm category is reinforcement learning, where in contrast to supervised learning, the training data does not indicate the correct output for a given input but it will indicate if the action is correct or not, reinforcing correct actions and penalizing incorrect ones.

These technics brought some good results, although not the ones expected, being limited when used on raw data, with no feature extraction giving meaning to the data, or even when fed with features extracted manually by experts from raw data [5].

2.2 DEEP LEARNING

In the beginning of the second decade of the XXI century and with the development of technology, introducing new and faster CPU's and GPU's [6], a branch of ML, denominated DL emerged with significant impact [3]. These DL methods are representation-learning methods allowing to transform the data representation, from its most raw to a more abstract state [5]. A key aspect of DL is that these more abstract layers are learnt from the data using a learning procedure, without human intervention, needed for manual feature extraction. While some positive aspects can be taken from this, DL systems struggle with how to explain its results to the engineer who built them, delaying the use of these systems in a more general way. Nonetheless, Deep Learning is making major advances in the artificial intelligence community, with great performances in image recognition [18, 19], speech recognition [20], knowledge representation [21] and natural language processing [22], clearly surpassing other ML methods.

2.3 THE NEOCORTEX

The functioning of the neocortex is the foundation for the HTM theory. The human neocortex is a tissue with a surface area of 2600 cm2 and a thickness of 3-4 mm, containing up to 28 109 neurons and approximately the same number of glial cells - non-neuronal cells that do not produce electrical impulse [23, 24]. The cortex is organized horizontally into six laminae, and vertically into groups of cells linked by synapses across the horizontal laminae; its basic unit is the minicolumn, a narrow chain of around 80-100 neurons. The bound of many minicolumns by short-range horizontal connections makes the formation of the cortical columns. The most common excitatory neuron in the neocortex is the pyramidal cell; a simple model of the cortical processing, according to the review in [25], describes a patch of superficial pyramidal neurons that receive feedforward excitatory input from subcortical, interareal and intra-areal sources. Besides this feedforward mechanism with their close neighbours of the patch, the pyramidal cells also receive feedback from deep pyramidal cells beneath their patch and from other close patches in the superficial layers. All of these inputs seem to be processed by the dendrites of the superficial pyramids; these participate in a selection network, with a soft winner-take-all or soft MAX mechanism, important elements used on many neuronal network models; the outputs will also feedback to adapt the pattern of vertical smooth cell activation. Furthermore, in [26] it is purposed three zones of synaptic integration on a neuron - proximal, basal and apical - the proximal zone receives the feedforward input and it is defined as the basic receptive field response of the neuron; the basal zone receives contextual input, mostly from nearby cells in the same cortical region, learning transitions in sequences, representing the prediction that the cell will become active shortly; the apical zone receives the feedback input, invoking a top-down expectation, having a similar effect as a basal dendrite, by recognizing patterns and forming predictions.

Our brains learn about the outside world by processing our sensory inputs and movements, receiving a sensorimotor sequence. The cortical areas that are traditionally viewed as sensory areas are known to integrate the motor stream into their processing; proposed in [27], the neocortex processes a sensorimotor sequence by converting it into a sequence of sensory feature at object-centric locations; it learns and recognizes objects as sets of sensory features at locations in a specific reference frame of the object, predicting sensory input by referring to these learned object models – this approach integrates movement into object recognition, although, leaving open the neural mechanisms for computing such a model. More recently, [28] extended this approach by using an analogue to grid cells, proposing that every neocortical column contains a variant of the model proposed; grid cells are present in the entorhinal cortex and represent locations of a body in an environment [29, 30]; although there is still not a consensus on this subject, some properties can be taken into account when speaking of grid cells: a set of grid cell modules can unambiguously represent locations in an environment and these locations can be path integrated via movement; in [28] it is proposed that equivalent grid cells exist in the neocortex defining a unique location space around each object; as a sensor moves, populations of grid cells representing each sensory patch's location will integrate the path through unique relative locations, a potential cue for disambiguation and recognition of objects.

The figure 1 represents the pyramidal neuron, the most typical neuron in the neocortex and how it is translated into an HTM neuron, with the various inputs: feedforward, context and feedback.

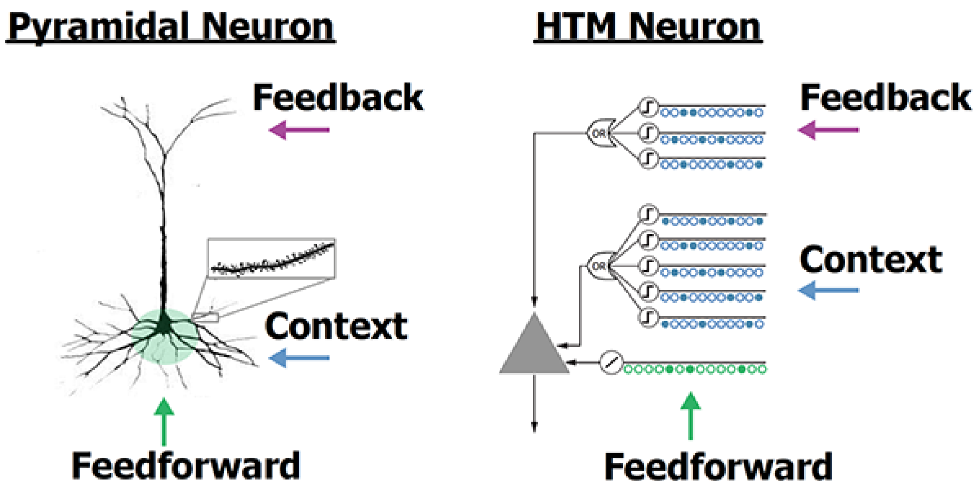

Figure 1: Pyramidal neuron and its relation to the HTM neuron; adapted from [31].

2.4 HEBBIAN LEARNING

As stated in the previous section, the Hierarchical Temporal Memory theory relies on the understanding of how the neocortex works and its connection to the mechanism of learning. The brain is constantly receiving signals from millions of receptor cells, making sense of the objects in the environment - it is capable of transforming these high dimensional patterns into symbolic representations; following the idea presented in [32], objects/features/concepts are collections or conjunctions of highly correlated properties, relatively independent from other such conjunctions. The process of translating or mapping the signals received into symbolic representations is called learning. The learning process is based on the correlation in the firing activity of the pre- and the post-synaptic neurons [33] and it relies in two different mechanisms [34, 35]: activitydependent synaptic modification along the lines proposed by Hebb (hence the named Hebbian learning) and a mechanism that forces competition between different synapses.

The learning process present in the building blocks of the HTM theory involves the establishment of connections between cells, synapses [7] - this process relies on Hebbian Learning, which is an unsupervised learning method; fundamentally, the permanency of synapses is measured by the rules similar to the Hebbian Learning, meaning that synapses that are active, contributing to the cell being active, will have their permanence value increased, while inactive synapses will have their permanence value decreased.

2.5 Hierarchical Temporal Memory

2.5.1 *Definitions and Network*

Hierarchical Temporal Memory is a theory that was born from the idea of creating a truly Machine Intelligence, machines capable of learning, supporting itself on the way the human neocortex works [8]. The HTM theory is built based on three main characteristics of the neocortex: it presents itself as a system with memory, organized into regions following an hierarchical structure and generating temporal patterns with the input given. All the nodes/neurons present in the regions implement the same learning and inference algorithms, only differing in the information gained during the learning phase [36].

The first region will be responsible for the sensory action of the algorithm – comparable to the human sensory organs; it is named the Encoder region, and its main function is to receive the data in its raw form and to convert it into a binary vector. Although this first region should not be assumed as part of the HTM algorithm, it is an essential region, in order to create an SDR, a foundation representation for the theory. An SDR corresponds, in biology terms, to the active neurons of the neocortex and it is represented as an array of bits, with the bit 1 being an active neuron and the 0 an inactive neuron. The mechanism of transforming the raw data into a set of bits must ensure that the semantic characteristics of the data are preserved in order to lead to a successful learning process. This leads to a very important feature of the theory, that says that similar data entries should create overlapping SDRs, when submitted to the encoding process – the 1s and 0s should have a high percentage of overlapping, when the input is similar. Another major feature related to the SDRs is their similar dimensionality and sparsity (ratio between the number of 1 bits and the total number of bits) – a certain percentage of sparsity, typically around 2%, will result in a better ability of the system to handle noise, undersampling and to reduce over-fitting. [37]; the encoding binary vector may be denser, since it represents our sensations, and it is not a representation of the neocortex neurons.

The next region, Spatial Pooler, is responsible for assigning mini-columns, where each one of them corresponds to a dendritic segment of the neuron; this process is responsible for creating the proximal dendritic connections mentioned on the previous section 2.3, The Neocortex. A mini-column connects to a local area of the input vector created by the Encoder region and has a set of synapses that can be initialized at random, with a permanence value. Some of these columns will be active, when its synaptic permanence value is higher than a stimulus threshold; when the mini-column is connected to a 1 bit (overlapping), the synapses become active, increasing its permanence value. Inhibition is introduced within the other columns in the vicinity, leading to only a small fraction of the SP mini-columns being active in a local area (following the Hebbian Learning where k-winners-take-all); active synapses will have their permanence value increased and will inhibit inactive synapses, decreasing its permanence value. In the process of learning, the mini-columns will learn to recognize the important features of the spatial input, meaning that different columns will be more sensible to certain features of the input space. A boost factor can also be applied differently to all columns, in order to multiply the overlap score of a column before the inhibition phase; this allows for less active columns to express themselves and increase the granularity of how the SP region recognizes the input space. Another important feature is that a mini-column is composed by many cells, where each one of them share the same proximal dendritic connections to the input space. In the following figure, 2, is demonstrated the connections between the Encoding and the Spatial Pooler regions, transforming the input space into an SDR.

At the output of the SP region is presented an SDR of active columns, according to the process explained previously. This representation will be the input for the next region, Temporal Memory; this region is responsible for receiving and learning the previous SDR and try to predict the next active columns – the next spatial pattern. Prior to the learning, when the algorithm cannot predict the next time step, since no cells are in a predictive state, all the cells of the active columns remain active (the input is unexpected) – process denominated bursting; however a winner-cell is chosen

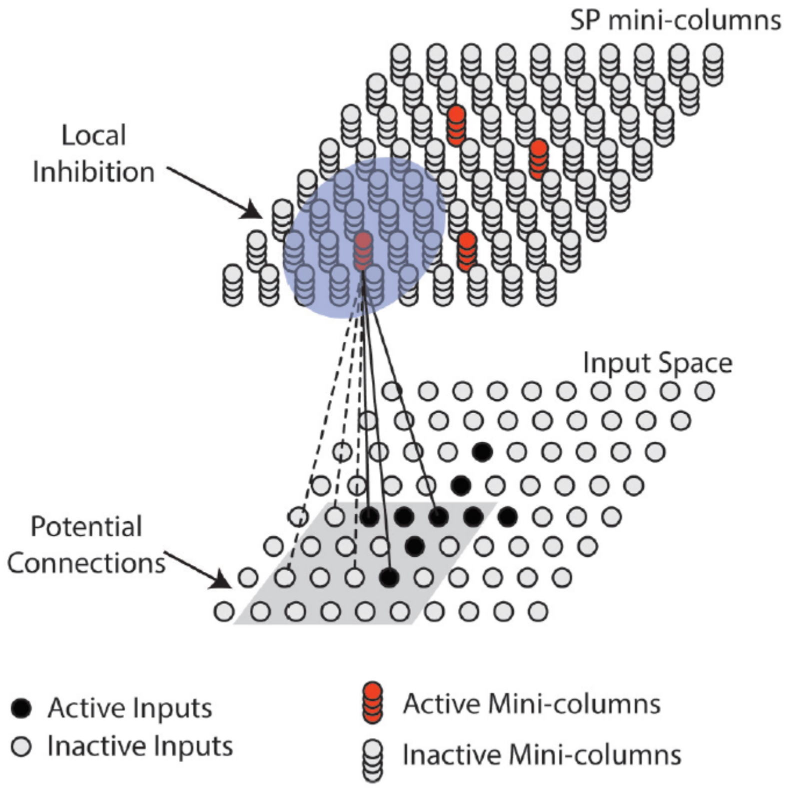

Figure 2: Synaptic connections between the Encoding and the Spatial Pooler regions; adapted from [31].

(randomly or by the lowest amount of distal connections). After learning, the algorithm is capable of predicting only a cell within a mini-column – this allows that even if an input has the same mini-columns active, the algorithm can understand the temporal context of it [26]; if we take for an example the sequences 'ABCD' and 'XBCY' – when presented with the sequence 'ABC' the algorithm should predict 'D' and not 'Y', even though the last two letters of the sequence where 'BC'. The predictive state of a cell within this region is triggered by the amount of its distal connections with the other cells that are active in the moment; if the cell is active in the next timestamp, then it was correctly predicted. As the algorithm learns, it will forget sequences that are not seen in a long time; the way it forgets can be tunable, so it retains more or less information over time. In order to get the results predicted by the algorithm, a classifier region is used to decode and calculate the overlap of the predicted cells of the SDR obtained by the TM region, relatively to the actual input [38, 9]; in this way, this layer outputs a predicted distribution of all classes.

2.5.2 *Properties and how it differs from ML*

There are some obvious variations on how the HTM network presents itself when comparing it to Machine Learning methods, more precisely Deep Learning models, like the encoding of the raw input into an SDR, with no data preprocessing needed (e.g. data normalization) or the use of sparse distributions instead of dense layers of neurons; besides these variations, there are some other key differences that should be noted when comparing both networks. The HTM theory is a sequence memory learning theory and it relies on on-line learning – continuous learning - where the network is gradually and continually adapting to the new input [26]. Typically, Machine Learning algorithms rely on partitioning of the raw data, having a training dataset for learning, a validation dataset for validate the learning and a test dataset for testing the algorithm; after this process of learning, the algorithms stop learning new inputs and make predictions only based on the cases in the training dataset - it needs batches of new inputs for new training sessions, in order to keep up with new data. Another HTM feature is that the learning rules are local to each neuron, in both space and time, without the need for a global objective function – in DL all the neurons are trained to meet a global objective function, while in HTM, we are in presence of an unsupervised learning. Since data streams contain branching temporal sequences, the HTM network must be capable of predicting multiple situations at the same time; in this way, the algorithm can output a distribution of possible future outcomes. Another two key properties of the HTM networks are their robustness to failure of network elements, noise and pattern variation and its ability to use high-order (more historical data) temporal context to make predictions. Lastly, there is no hyperparameter tuning in the HTM networks, making it robust to a wide range of problems; in contrast, most ML algorithms require this optimization for each specific task [38].

2.5.3 *Current Applications of HTM*

The current applications of HTM are typically centered in the area of Anomaly Detection, useful for a vast range of applications like fraud detection, geospatial tracking, rogue human behaviour, preventative maintenance, Internet of Things (IoT) sensors, traffic patterns, natural language or network and servers monitoring.

In the last few years, there has been an increase in the availability of streaming, time-series data, bringing an opportunity to model streams in an unsupervised way in order to detect anomalous behaviours in real-time. These early anomaly detection algorithms require that the systems must process data in real-time, favoring algorithms that learn continuously, like HTM.

Although there is not an abundance of investigation in the HTM theory, when compared to DL algorithms, some of the works found in the literature include the recognition of different people in a video, automatic license plate recognition, taxi passenger count prevision and other time series tasks. In 2018, [39] compared an HTM model against Artificial Neural Network (ANN) and Support Vector Machine (SVM) networks in the recognition of people on a video, with a certain level of occlusion - after some preprocessing of the data and taking Scale-Invariant Feature Transform (SIFT) features, it claims the HTM model performed better than the ones compared to - although, the time taken by the processing of the HTM model in this task was higher than in the other models; no comparisons were made to Convolutional Neural Network (CNN)s networks, that have been getting good performances in computer vision. Another computer vision task, the automatic recognition of license plates, was investigated in [40] - after the segmentation of license plates images, the inputs were fed to an HTM model, with an Image Sensor encoder, a SP layer and a K-Nearest Neighbor (kNN) classifier, obtaining an accuracy of 95.35%, slightly better than the ones compared in the study, an ANN and a SVM network.

Relatively to time series datasets, some research was conducted, in the prediction of New York city taxi passenger count. In [41], a comparison between an HTM model and others like, Autoregressive integrated moving average (ARIMA) and Skyline was made, using real and synthetic datasets; in the paper it was demonstrated that the HTM model obtained good precision results as well as a significant decrease in processing time. Also, in 2016, [42] predicted the New York City taxi passenger count with 2.5 hours in advance, aggregating data in 30-minute intervals; after observing 10000 data records, the HTM model achieved a Mean Average Percentage Error (MAPE) of 7.8%, lower than the other DL model used in the study - a Long Short-Term Memory (LSTM) network. In response to this study, in 2020, [43] used Recurrent Neural Network (RNN)s, such as LSTM and Gated Recurrent Unit (GRU), to solve the same problem; in this approach, a more careful hyperparameter tuning and data formatting was made, leading the authors to the conclusion that both models exceeded the HTM model by 30% in lower runtime.

In [44], a wide set of anomaly detection problems were confronted in various datasets from the Numenta Anomaly Benchmark (NAB), a Numenta repository for anomaly detection with a variety of data sources, ranging from server network utilization to temperature sensors to social media chatter. An HTM model was compared against various ML models, demonstrating that it is capable of detecting spatial and temporal anomalies, both in predictable and noisy domains, having better performances in various tasks. Also, in [45] the HTM model was applied for the detection of anomalies, in an unsupervised way, in stock market datasets and in a synthetic dataset; although, there was a lack of conclusions about the efficacy of the method, partly because the ground truth was unknown. It is possible to infer that the research in HTM is still recent, with few experiments conducted, when compared to other methods.

3

METHODOLOGY AND DEVELOPMENT TOOLS

3.1 NUPIC

On the course of this dissertation, an open-source Python library named NuPIC [46] was used; it is a MI platform that implements HTM algorithms developed by Numenta, the company behind HTM theory. These algorithms are best suited for anomaly detection and prediction of streaming data sources problems. The NuPIC library is divided into three APIs: OPF, Network and Algorithms; the Algorithms API is the most low-level API, while the OPF is the higher-level one. For the Image Classification and Sound Recognition studies it was used the Network API, while the OPF API was used for the Time Series Forecasting task. This choice was purely based on how the datasets for each task were more easily adapted to each API - a Comma-separated values (CSV) file for the Time Series Forecasting and a collection of images/audio files for the other tasks. The library is compatible with Python 2.7.

3.2 IMAGE CLASSIFICATION

The first task was to investigate how well the HTM algorithms created to the date, adjust to Image Classification, a task that has achieved promising results in the Deep Learning community for the past years. It was used the widely known MNIST dataset, containing centered images with handwritten numbers,

$$k \in \{0, 1, ..., 9\}$$

, each with an input space of 28 x 28 x 1 pixels. The training set contains 60 000 images and the testing set contains 10 000 images.

This approach is similar to the one typically used in Machine Learning technics, with defined training and testing datasets. Also, this task does not involve temporal patterns, meaning that this is not a typical HTM problem, but a spatial problem.

The network used is composed of an Encoder layer, which encodes the raw input of the images into a binary vector - the encoding used is called Image Sensor, present in the NuPIC library; this encoding translates the image to a vector of 1's (black) and 0's (white); no more preprocessing was made to the input images. The next layer, Spatial Pooler, is connected to the Encoder layer, receiving the binary vectors representing the images and assigning mini-columns to the input space, that will learn to recognize the important features of the spatial input; an SDR of active columns will be the output of this layer, input to the last layer: the kNN classifier, responsible for outputting the predicted distribution of classes. As stated before, no Temporal Pooler layer is used nor needed for this task. On this task, two approaches were made: the first, using the training and testing datasets proposed by the MNIST dataset; in the second approach, the 10 000 images of the testing dataset were used as training dataset and the 60 000 images of the previous training dataset was used for testing. This second approach had the goal to investigate how well the algorithm was going to perform with less spatial patterns trained.

The following table, Table 1, illustrates the parameters necessary used in the SP layer with the NuPIC library; the parameter inputWidth is required to be 784, since its the result of the number of pixels in each image; the ratio between the number of active columns and the number of columns is 2%, the recommended sparsity; the other parameters come from the tuning in order to get better results. The network built is shown on figure 3.

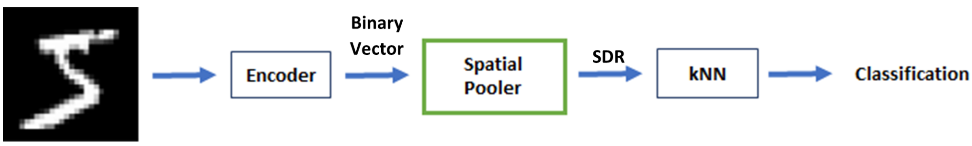

Figure 3: HTM network built for the Image Classification task.

Table 1: SP region parameters for the Image Classification task

Parameter	Value
inputWidth	784
numColumns	4096
numActiveColumnsPerInhArea	82
potentialPct	0.8
globalInhibition	1
localAreaDensity	-1
stimulusThreshold	0
synPermActiveInc	0
synPermInactiveDec	0
synPermConnected	0.2
boostStrength	0

3.3 SOUND RECOGNITION

The second task proposed was to use a Sound Recognition dataset and analyse how the HTM theory can help in the resolution of this learning exercise. The dataset used was the Spoken Digit Dataset [47], available in a GitHub repository; it comprises of english recordings of spoken digits in .wav files, with a record sampling at 8kHz, trimmed to avoid noise at the beginning and end of the recordings; 1800 files were used for training, with 45 files for each digit, from 0 to 9, for each speaker (4 speakers); for testing, 200 files were used, 5 of them for each pair digit + speaker.

Two approaches to this task could be pursued: to encode a single audio file (corresponding to a spoken digit) into one binary vector, making it a spatial problem - with no temporal extraction of sound features; or to turn this problem into a spatial-temporal one - by encoding the audio file into multiple vectors, where each one of them represents a portion of the spoken digit. This second approach, although more close to the HTM paradigm, represents a big challenge when it comes to the temporal pooling of the information learnt: how can the algorithm know on which sequence, the data presented is included? The Numenta researchers are still investigating how this ”Temporal Classification” problem, where a sequence of inputs leads to a certain class, being reset at the end, giving place to a new sequence, occurs in our brains - and similarly how it can be transposed to the NuPIC library. Another complication surges in this task: how to encode sound into an SDR while giving it the human perception of sound - some preprocessing of the data is needed to take the power spectrum of the signal and discretizing it. Further processing can be applied to the signal, taking the MFCC, a feature widely used in Speech Recognition. A Cochlea Encoder was produced in the experiments related to the NuPIC library, to encode sound as perceived by the human ear - although, the process takes too long and requires the input data to use a minimum of 100 kHz sampling frequency; for this task, around 50 000 SDRs for each spoken digit would be produced, leading to big implications on the computation time it takes to train and test the algorithm, making this option impractical.

Having these concerns in mind, for the first approach, a community encoder, not present in the NuPIC library, named Frequency Encoder, was used to take the power spectrum from the signals and discretizing it - obtaining the Fast Fourier Transform (FFT); a graphical represention of the encoding process is shown in figure 4. Some parameters need to be set to adapt the resolution and robustness to noise of the encoding process, like how many bins the signal is going to be discretized into, the size of the bin and its resolution; in the course of this task, the signals were divided into 50 bins. The tables 2, 3 and 4 show the parameters used.

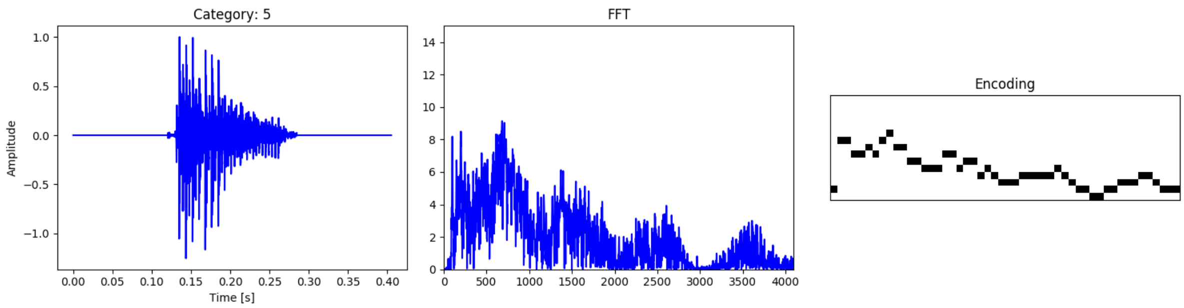

Figure 4: Encoding process of the signal representing the 'five' spoken digit, in the first approach for the Sound Recognition task.

Table 2: Frequency Encoder region parameters for the Sound Recognition task, first approach

Parameter	Value
numFrequencyBins	50
freqBinN	15
freqBinW	1
minval	0.0
maxval	15.0

Table 3: SP region parameters for the Sound Recognition task, first approach

Parameter	Value
inputWidth	750
numColumns	2048
numActiveColumnsPerInhArea	80
potentialPct	0.85
globalInhibition	1
localAreaDensity	-1
stimulusThreshold	0
synPermActiveInc	0.04
synPermInactiveDec	0.005
synPermConnected	0.15
boostStrength	3

Table 4: Classifier region parameters for the Sound Recognition task, first approach

Parameter	Value
Type	SDRClassifier
alpha	0.25
steps	0

The network layout is similar to the one used for the Image Classification task, with the encoding being fed to the SP layer and this one to a SDRClassifier. The training set was fed ten times to the network.

Figure 5: HTM network built for the Sound Recognition task, first approach.

A second approach was taken, turning this task into a temporal problem. The process of encoding created chunks of 100 ms from the signal, with a window step of 50 ms; with these chunks, 16 MFCC cepstrums were taken and encoded to SDRs by a Scalar Encoder; before the coefficients were encoded, a process of standardization was made to ensure a mean of zero and a standard deviation of 1. To surpass the issue of temporal pooling, explained previously, only the last 3/4 of the SDRs predicted in a sequence, were taken into account to the classification - meaning the first quarter of the SDRs were used as a context indication for the network to understand in which sequence it is presented with. Contrary to the first approach, a TM layer was used after the SP layer; in the following tables, the parameters used for this approach are displayed.

On both approaches, the training set was fed ten times to the network. The image 6 represents the transformation associated with the second task, transforming the raw signal into the MFCC features, used for the encoding. Similarly to the first approach where the Frequency Encoder parameters were adjusted to represent the FFT spectrum, here the Scalar Encoder parameters were tweaked to represent the range of values of the MFCC features.

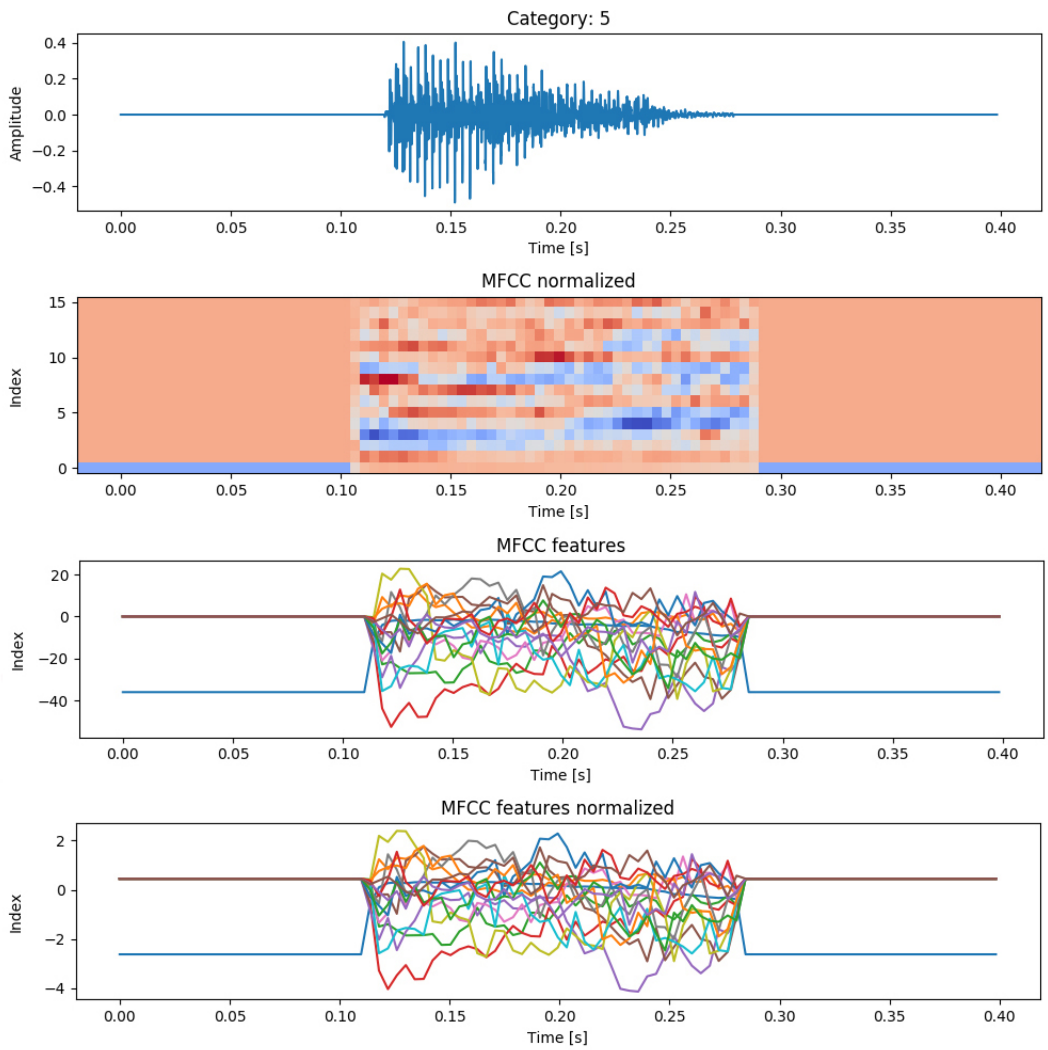

Figure 6: MFCCs features taken from the signal representing the 'five' spoken digit, for the Sound Recognition task, second approach.

Table 5: Scalar Encoder region parameters for the Sound Recognition task, second approach

Parameter	Value
minval	-5
maxval	5
w	21
resolution	0.1

Table 6: SP region parameters for the Sound Recognition task, second approach

Parameter	Value
inputWidth	1936
numColumns	4096
numActiveColumnsPerInhArea	160
potentialPct	0.85
globalInhibition	1
localAreaDensity	-1
stimulusThreshold	0
synPermActiveInc	0.04
synPermInactiveDec	0.005
synPermConnected	0.15
boostStrength	3

Table 7: TM region parameters for the Sound Recognition task, second approach

Parameter	Value
inputWidth	1936
columnCount	4096
cellsPerColumn	64
newSynapseCount	20
initialPerm	0.21
permanenceInc	0.1
permanenceDec	0.1
maxAge	0
globalDecay	0
maxSynapsesPerSegment	64
maxSegmentsPerCell	256
minThreshold	12
activationThreshold	16
outputType	normal
pamLength	1

Table 8: Classifier region parameters for the Sound Recognition task, second approach

Parameter	Value
Type	SDRClassifier
alpha	0.001
steps	1

The network built for this approach is detailed on image 7, starting at the process of encoding, feeding the multiple SDRs from the input to the network sequentially, leading to a prediction from the classifier.

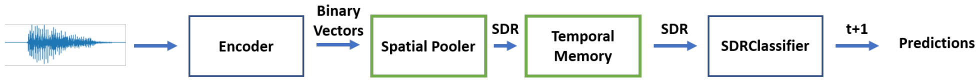

Figure 7: HTM network built for the Sound Recognition task, second approach.

3.4 TIME SERIES FORECASTING

In this last task, the data used was obtained by a script, to get stock fluctuations for various companies, pulling data from Yahoo Finance, ranging from 2006-01-03 until 2020-09-18. Seven datasets were created, each related to a S&P 500 company: Amazon, Google, HCA Healthcare, Disney, McDonald's, Johnson & Johnson and Visa; the HCA Healthcare dataset only has data from 2011-03-10, and the Visa dataset from 2008-03-19. These companies were chosen due to their familiar popularity and represent a wide range of business areas. Another particularity taken into account, is the inclusion of data after the declaration of the Covid-19 pandemic by the World Health Organization (WHO) - declaration on march 11 2020. All seven datasets have the same fields: Date, Open, High, Low, Close, Volume and Name. The Name field corresponds to the stock's ticker name, not of use for the forecasting.

In the following figures, 8 to 12, is shown the range of values that are present in the seven different datasets. It is possible to observe that the values have high ranges and are very different from dataset to dataset. Amazon demonstrates the widest ranges for almost every attribute, except for the volume - Visa had a higher volume of transactions, although at lower prices.

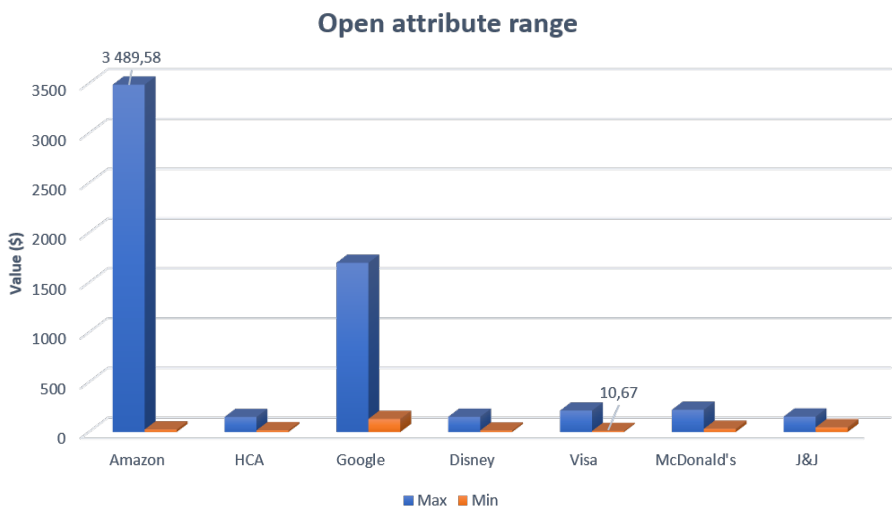

Figure 8: Open values range in the seven datasets.

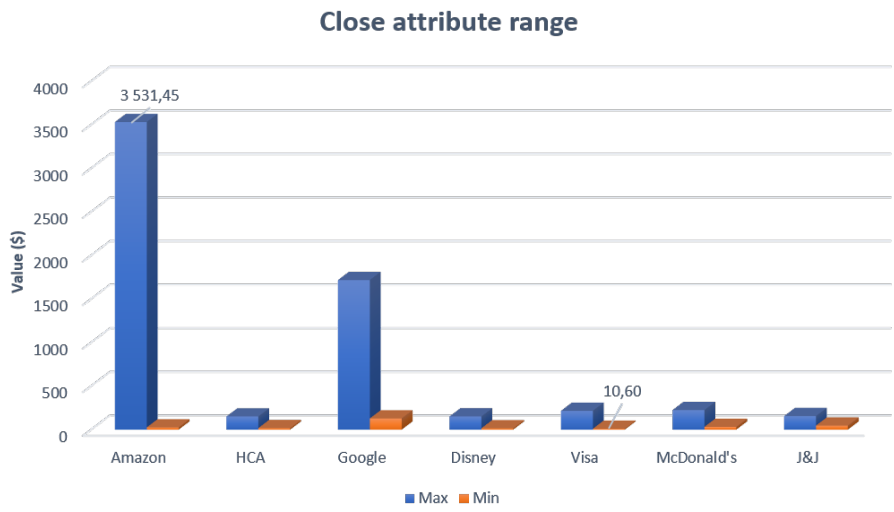

Figure 9: Close values range in the seven datasets.

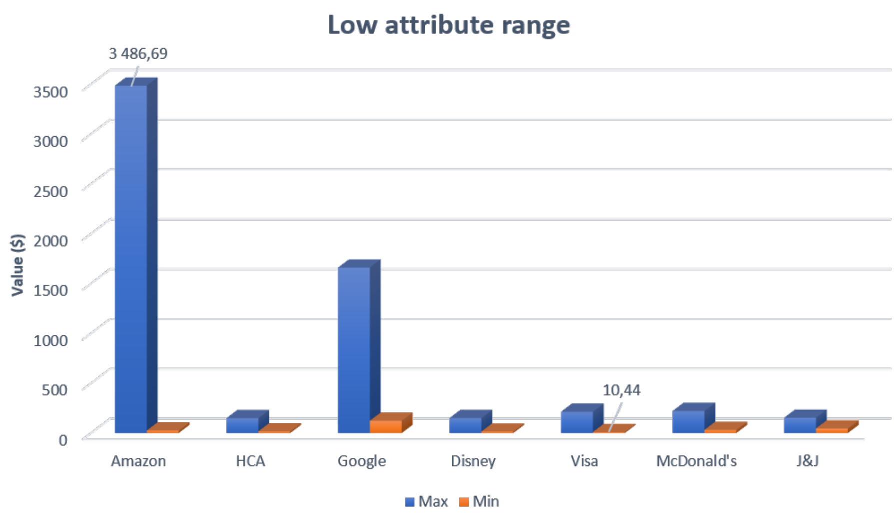

Figure 10: Low values range in the seven datasets.

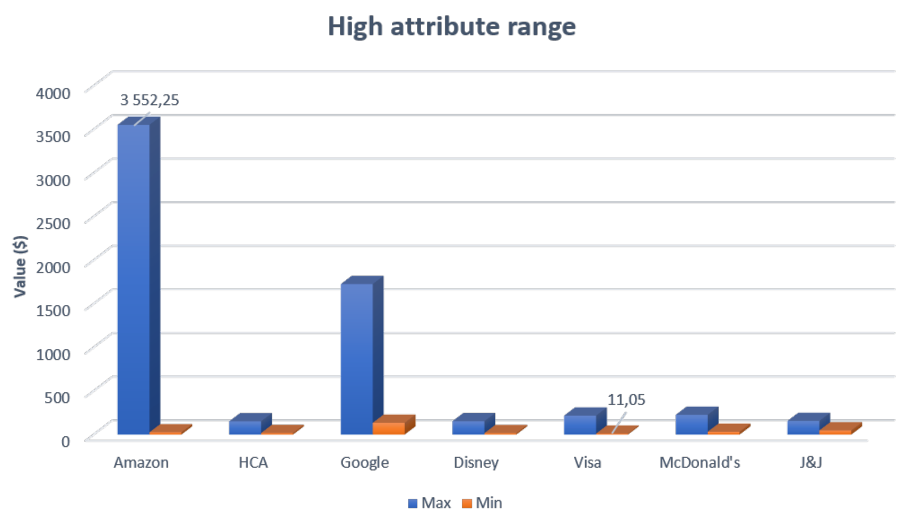

Figure 11: High values range in the seven datasets.

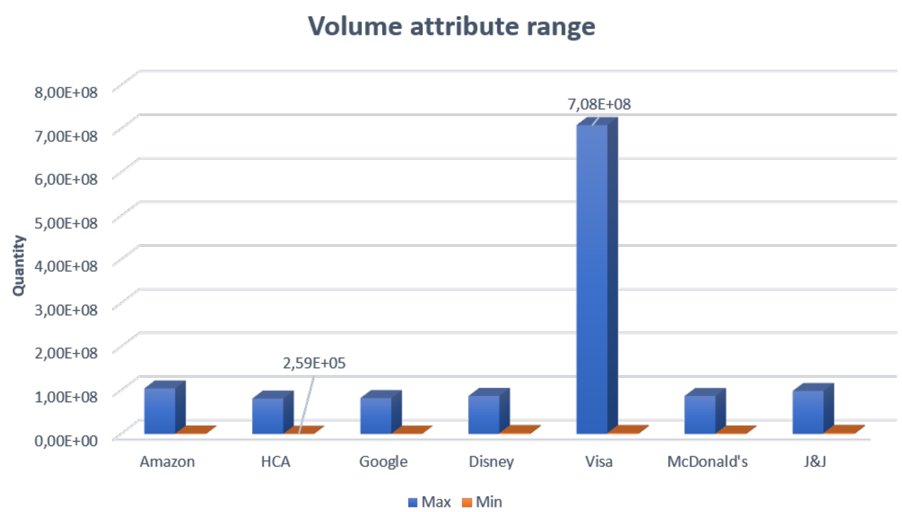

Figure 12: Volume values range in the seven datasets.

The goal is to predict the Close value of the companies' stocks for the next day, only using the historical values presented before. This forecasting represents a more HTM focused approach, with a stream of data being fed to the network, with the possibility of on-line learning, and with temporal and spatial patterns to be discovered.

By plotting the close values for the Amazon and Visa datasets, figures 13 and 14, it is possible to observe that for both there has been an increase throughout the years. Although, the Covid-19 pandemic appeared to have a bigger impact to the Visa stocks, than it did for the Amazon stocks - possibly explained by its business area.

Figure 13: Amazon close value progression in the dataset.

For this task, a more traditional HTM network was used, with an Encoder layer, SP layer, TM layer and a Classifier. The TM layer is justified by the fact that there is a temporal axis in the data and temporal patterns must be gathered for a better forecasting, contrary to the previous tasks. The following tables show the parameters used in the various regions and in figure 15 the network built is presented.

Like for the previous tasks, some values set to the parameters are required, taken into account the way the encoding process is made, like the "inputWidth" of the regions. Other values are usually standard, with little modifications, like the inhibtion and the increase and decrease of the permanence of synaptic connections, that try to represent the learning process in the neocortex; parameters like "columnCount" and number of active columns per inhibition area were tested to give more or less complexity and memory to the network. For the encoding of the data in this task, most values are

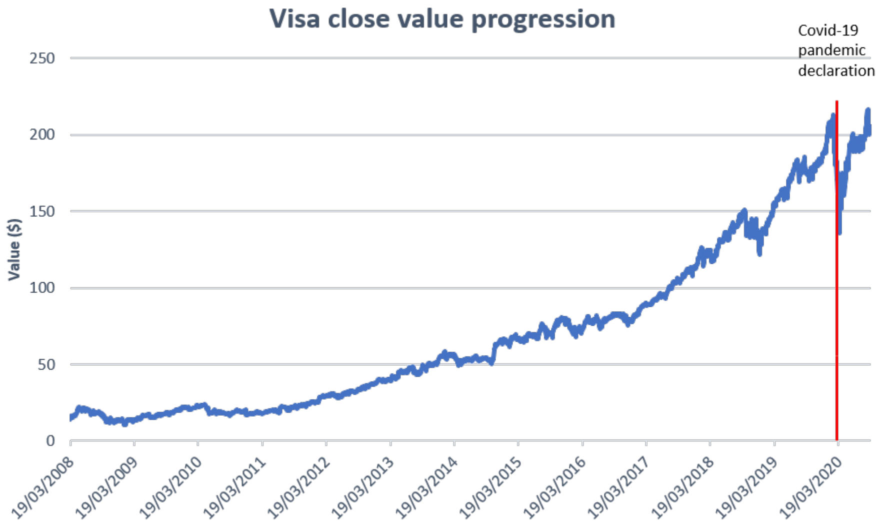

Figure 14: Visa close value progression in the dataset.

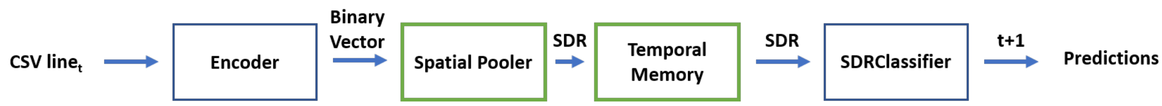

Figure 15: HTM network built for Time Series Forecasting task.

scalar, so a Random Distributed Scalar Encoder was used, with a resolution that reflects the range of values of the attributes - the volume has a larger range, meaning a larger resolution was used; for the date, a Date Encoder represents the days taken into account if its a day of the week and the season it is included.

Table 9: Encoder region parameters for the Time Series Forecasting task

Input	**Type of Encoder**	**Parameters**
date	DateEncoder	season = dayOfWeek = 3
open	RandomDistributedScalarEncoder	Resolution = 0.5
high	RandomDistributedScalarEncoder	Resolution = 0.5
low	RandomDistributedScalarEncoder	Resolution = 0.5
close	RandomDistributedScalarEncoder	Resolution = 0.5
volume	RandomDistributedScalarEncoder	Resolution = 200

Since the HTM is a continuous learning theory, there is no training/validation/test sets; the data is learnt and predicted in a continuous way. To access the learning, the metrics Root Mean Square Error (RMSE), MAPE and Absolute Average Error (AAE),

Table 10: SP region parameters for the Time Series Forecasting task

Parameter	Value
inputWidth	2033
columnCount	4096
globalInhibition	1
localAreaDensity	-1
numActiveColumnsPerInhArea	160
potentialPct	0.85
synPermConnected	0.1
synPermActiveInc	0.04
synPermInactiveDec	0.005
boostStrength	3

Table 11: TM region parameters for the Time Series Forecasting task

Parameter	Value
inputWidth	2033
columnCount	4096
cellsPerColumn	64
newSynapseCount	20
initialPerm	0.21
permanenceInc	0.1
permanenceDec	0.1
maxAge	0
globalDecay	0
maxSynapsesPerSegment	64
maxSegmentsPerCell	256
minThreshold	12
activationThreshold	16
outputType	normal
pamLength	1

Table 12: Classifier region parameters for the Time Series Forecasting task

Parameter	Value
Type	SDRClassifier
alpha	0.25
steps	1,5

were taken in three moments: to the entire dataset, 365 days before the declaration of the Covid-19 pandemic by the WHO and after the declaration. With these three moments, it is possible to get a better understanding of how quick (in terms of input data needed) the algorithm is to achieve good previsions, while infer how it adapts to dramatic changes in the input data (in this case, a consequence of the pandemic). No preprocessing of the data is made, meaning that some metrics may have higher values than the ones usually seen in other studies, where the data values are standardized into smaller intervals.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\widehat{x_i} - x_i}{x_i})^2} \tag{1}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{(\widehat{x_i} - x_i)}{x_i}\right| \times 100 \tag{2}$$

$$AAE = \frac{\frac{1}{n}(\sum_{i=1}^{n}|\widehat{x_i} - x_i|)}{(\frac{1}{n}\sum_{i=1}^{n}x_i)} \tag{3}$$

4

RESULTS AND DISCUSSION

4.1 IMAGE CLASSIFICATION

The HTM model used, with only a SP layer, classified 96.04% of the 10 000 images correctly, which is not a great result comparing to state-of-the-art DL models that can reach an accuracy greater than 99%. On the second approach, with less training data (10 000 images for training and 60 000 for testing), the model achieved 93.20% of accuracy, demonstrating that the theory can get satisfying results with less data. It was possible to observe that the library is not prepared for Computer Vision problems, when it is taken into account the large amount of time required for the training - as the network was being fed with new input data, it constantly performed significantly slower.

Although the results are not the best, they were expected since only a layer of the HTM theory was used; this task does not provide a temporal problem, one of the features of the theory; also, the NuPIC library does not provide efficient handling of image encoding, leading to accuracy and power computing/time efficiency faults - more research is being conducted for this matter, turning the encoding of images into a sequence of images, as perceived by the human brain, with the help of the saccadic eye movement; with this sequence of images, a perception of the environment can be collected, turning the task into a temporal problem. Another issue with the encoding of images in the library, is that the image is only read as a sequenced collection of pixels, with no feature extraction; more work is underway in order to investigate grid cells and how they can represent space, hopefully helping in the way images can be encoded to an HTM network.

4.2 SOUND RECOGNITION

The task of sound recognition was divided into two approaches: at first, the problem was handled as simply a spatial problem; in the second approach, the problem was adjusted into a spatial-temporal problem; the HTM model achieved an accuracy of 73% for the testing dataset, in the first approach, and an accuracy of 82% for the second one. A slight improvement in accuracy is observed when the network used is more close to the HTM paradigm, with not only a spatial, but also a temporal layer. However, this temporal approach led to a more time consuming training - approximately 10 hours on an Intel Core i7-8550U CPU, 4 cores, with a Processor Base Frequency of 1.80 GHz. This long training time is fruit of the use of the Network API used in the library, for problems that are not solved using on-line training.

Similarly to the first task, the results were not satisfying in terms of accuracy, when compared to state-of-the-art ML networks, that can achieve more than 90% accuracy for this Spoken Digit dataset; it should be taken into account that the training data was only fed to the network ten times, not the usual high number of epochs conducted by other networks. The results can be justified by the reasons mentioned on the 3.3 section: the NuPIC library does not support an official encoding mechanism for audio files, meaning that more experiments must be conducted to achieve a good encoding of the raw data, while maintaining its meaning; furthermore, the second approach, turning the task into a Temporal Classification problem, requires a process of temporal pooling that is not supported by the library.

4.3 TIME SERIES FORECASTING

The results obtained for all seven datasets, related to the S&P 500 stock index, are shown in the following table, including the MAPE, RMSE and AAE metrics for three periods of time: metrics taken during the entire dataset, 365 days before the declaration of Covid-19 pandemic by the WHO and after the declaration - until the 18th September 2020. The values were obtained by comparing the predicted close value of the next day (one day in advance), to the actual value recorded. Comparisons with other methods are hard and not reliable, since most use a partition of data for training and testing, unlike the on-line learning process developed in this project.

Table 13: Time Series Forecasting metrics results

	Total			1 Year pre-pandemic			Post-pandemic		
	MAPE	RMSE	AAE	MAPE	RMSE	AAE	MAPE	RMSE	AAE
Amazon	1.61	18.82	8.40	**1.44**	36.43	25.19	2.00	66.31	51.61
Google	1.25	11.99	6.73	**1.24**	21.82	14.70	1.91	35.36	25.38
HCA	1.45	1.75	1.03	**1.39**	2.63	1.81	3.10	4.53	3.19
Disney	1.13	1.22	0.72	**1.13**	2.19	1.42	2.31	3.58	2.52
McDonald's	**0.86**	1.56	0.87	0.88	2.64	1.71	1.81	5.33	3.22
J&J	**0.73**	1.12	0.68	0.86	1.88	1.17	1.42	3.01	1.95
Visa	1.28	1.59	0.81	**1.16**	2.80	1.90	2.10	5.32	3.70

In the following graphics, the predicted and actual values are displayed along the time axis, as well their difference. The algorithm kept a good performance, following the trends of market close value through the time, for all datasets. As expected, the algorithm suffered in its previsions around the time of the declared pandemic; although, it was able to get some stability afterwards, in line with the possible stability that the stock market can offer in such unstable period.

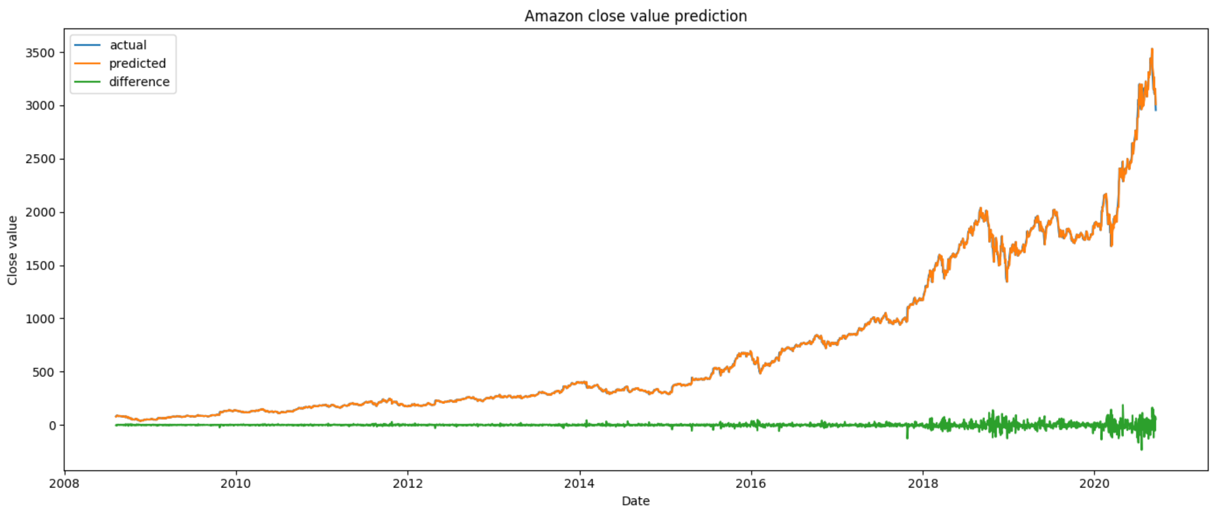

Figure 16: Amazon close value prediction through time.

It becomes possible to infer that the HTM algorithm run in this experiment learnt the temporal and spatial patterns quickly, making valid predictions, close to the actual values, with just few records fed to the network. The MAPE values were lower for every dataset in the more stable period before the pandemic, with the exception of the McDonald's and J&J datasets, that got better results in the total period. All MAPE

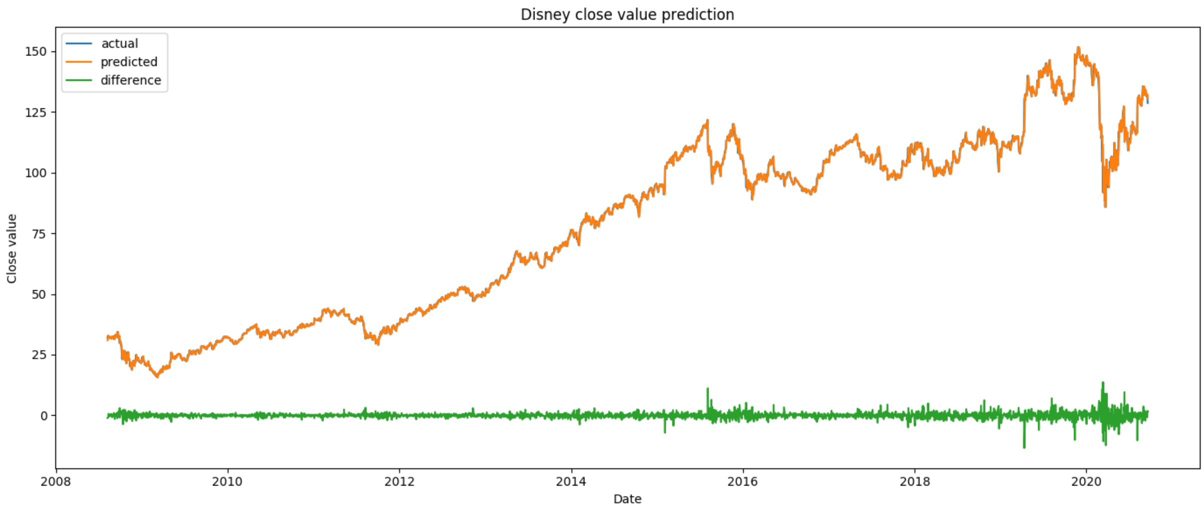

Figure 17: Disney close value prediction through time.

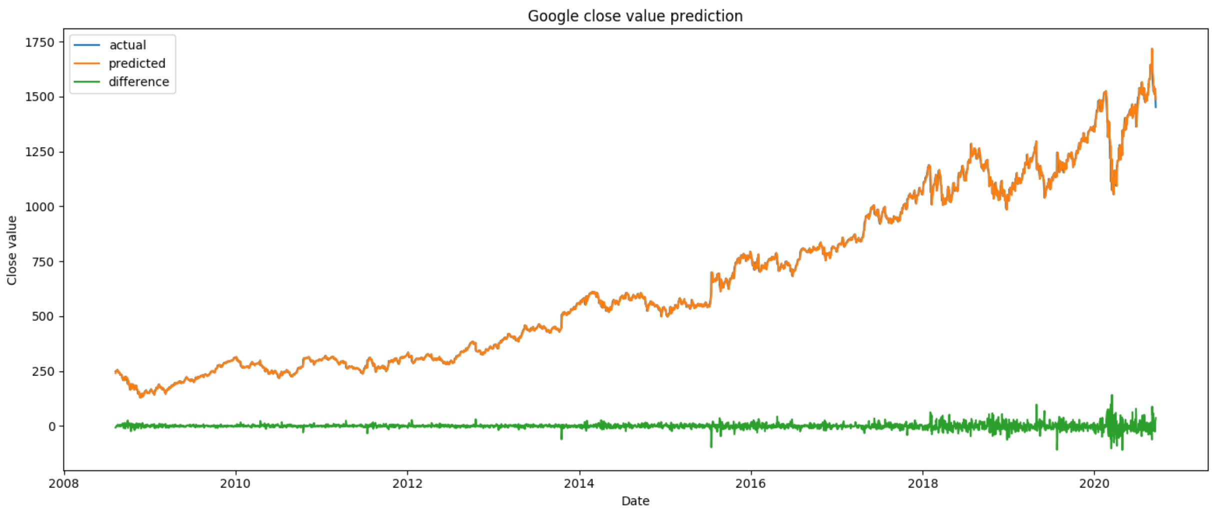

Figure 18: Google close value prediction through time.

values increased for the post-pandemic period, although, not as much for the Amazon dataset - this can be explained by the more stable stock pricing in this company, fruit of its business area. In general, the RMSE and AAE values increased through time; since these are not percentage metrics, and the data is not normalized, this increase can be explained by the higher close values in the stock market in the last years across all datasets. However, the lower MAPE values means that the algorithm kept learning and did not suffer from the inclusion of data through time.

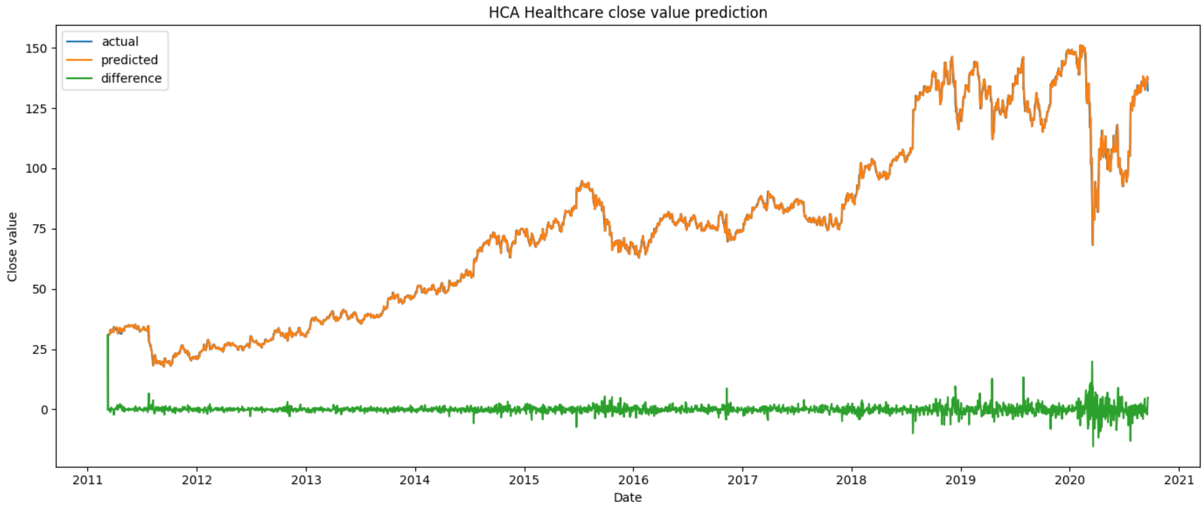

Figure 19: HCA close value prediction through time.

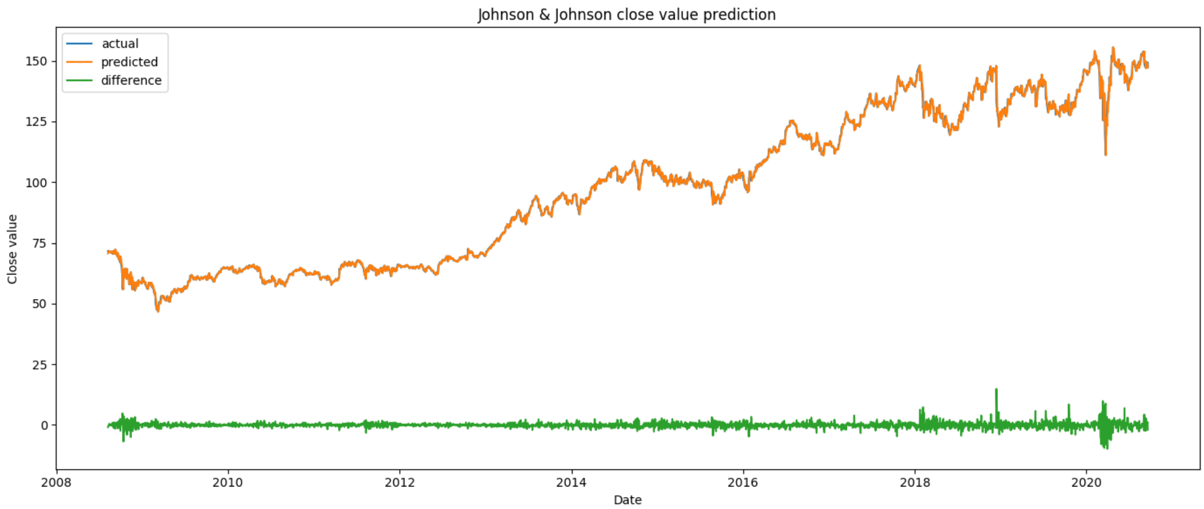

Figure 20: J&J close value prediction through time.

The results obtained on this experiment were very promising showing that the HTM theory provides a solid framework for time series forecasting, achieving good predictions with few data. Furthermore, the algorithm maintained a good performance for all seven datasets, through time, being robust to spatial and temporal noise and bigger complexity of data and anomalies in the input data caused by the pandemic; another important fact taken from this task was the ability of the algorithm to have good and similar performance for seven different datasets, with no need for hyperparameter tuning.

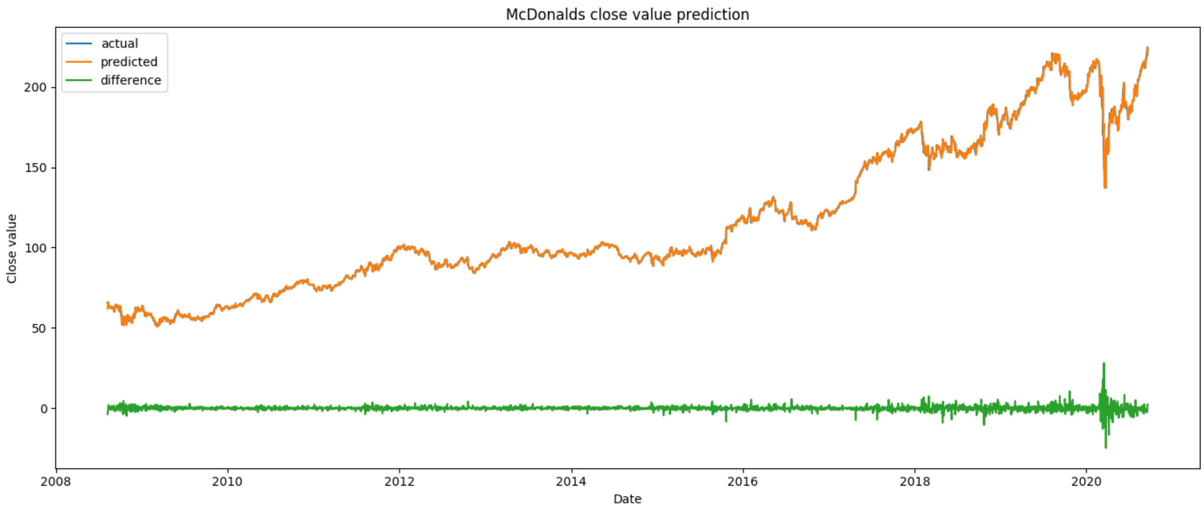

Figure 21: McDonald's close value prediction through time.

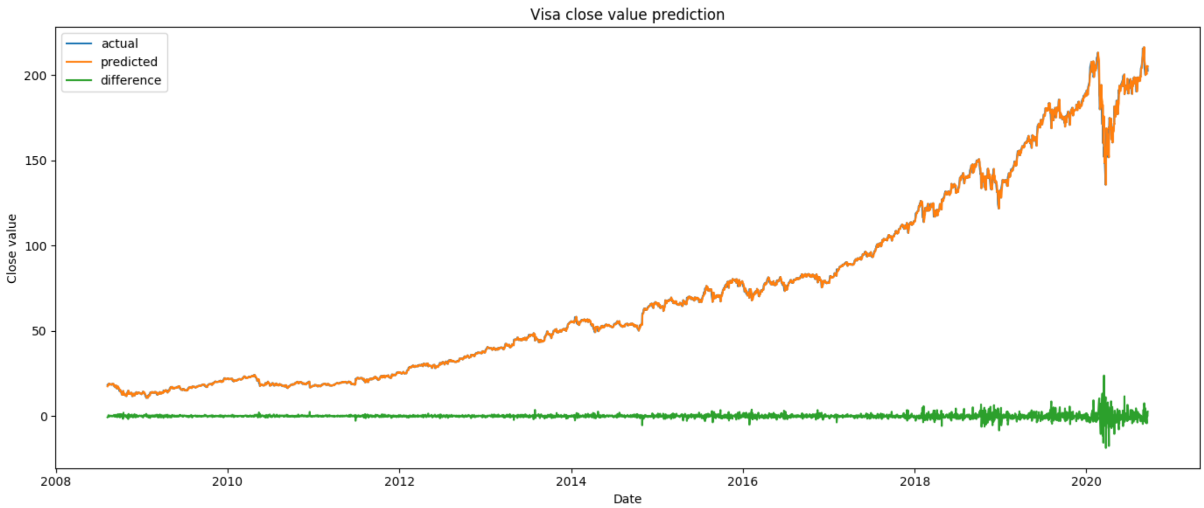

Figure 22: Visa close value prediction through time.

5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

The objectives proposed at the beginning of this dissertation were achieved; in a first phase, it was conducted an investigation about how the HTM theory works, with its explanation in the section 2.5; in summary, the theory relies on our knowledge of the learning process in the neocortex - regions organized in an hierarchical structure, with a high sparsity of active neurons, capable of storing and discerning temporal patterns. HTM networks are usually built as continuous learning networks - they gradually and continually adapt to the new input.

After this investigation, the theory was tested in three different scenarios: for an image classification task, using the MNIST dataset; for a sound recognition task, using the Spoken Digit dataset; and finally, for a time series forecasting task, using a stock market dataset. The results obtained were compared in terms of accuracy, compute efficiency, amount and type of data required and noise robustness; on one side, the HTM model performed well on the time series task, a continuous learning approach, with no need for a specific training dataset nor huge amount of data nor long training processes; also, the model adjusts itself very well through time, being robust to spatial and temporal noise, greatly because of the use of SDRs and their properties; on a more negative side, the model failed in the image classification and sound recognition tasks, mainly due to the encoding process and how it is not well supported in the Python libraries up to dates. Another positive takeaway, is that the HTM theory provides a solid understanding of the learning process, unlike the black box algorithms, characteristic in DL methods; it is possible to infer the decisions taken by the algorithm and to check the current state of the network.

Although the theory presents itself with good visions of how the neocortex works and how this knowledge can be translated into an algorithm, it still needs a lot of research not only in the theory domain, in pair with our current view of the learning process in the neocortex, but also in the development of libraries that can help the data science community to take advantage of it, for the resolution of complex problems, usually confronted with the use of ML models.

5.2 PROSPECTS FOR FUTURE WORK

The HTM theory presents itself as an evolving theory, with a lot of ground still to be discovered. In a first phase, the knowledge of how our neocortex works is still expanding, with new discoveries of the learning process; with this building knowledge, there should be an equal investment on the creation of libraries capable of translating it into useful algorithms.

Some areas of research inside HTM include: a better understanding of how the apical connections can be applied in the TM region; the use of grid cells modules, that should be capable of representing objects with less ambiguity; the representation of images using saccadic eye movement, consisting in multiple inferences on the same pattern, while it is moved for a few pixels at a time, emulating fast eye movements that focus attention in different parts of an object. These areas of interest are currently being investigated, with some research code available in the ”htmresearch” repository [48]. Besides the development of the theory itself, other future work may include the combination of HTM with other methodologies; for instance, for a Computer Vision task, a previous step of preprocessing and feature extraction can precede the representation of the image into an SDR. This approach will result in a network that moves away from the biological HTM theory, however it can bring good results to these tasks.

BIBLIOGRAPHY

[1] P.; Russel, S.; Norvig. *Artificial Intelligence : A Modern Approach.* Prentice Hall Series in Artificial Intelligence, Boston: Pearson, third edition, 2010.

[2] J. G. Carbonell, R. S. Michalski and T. M. Mitchell. An Overview of Machine Learning. *Springer*, (Machine Learning), 1983.

[3] Kimihiro; Fadlullah, Zubair Md.; Tang, Fengxiao; Mao, Bomin; Kato, Nei; Akashi, Osamu; Inoue, Takeru; Mizutani. State-of-the-Art Deep Learning : Evolving Machine Intelligence Toward Tomorrow ' s Intelligent Network Traffic Control Systems. 19(4):2432–2455, 2017.

[4] M I Jordan and T M Mitchell. Machine learning: Trends, perspectives, and prospects. 349(6245), 2015.

[5] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. 2015.

[6] João Fontes. *Intelligent medical image analysis : a Deep Learning approach to breast cancer diagnosis*. PhD thesis, 2018.

[7] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin. Biological and machine intelligence (bami). Initial online release 0.4, 2016.

[8] Sandra Hawkins, Jeff; Blakeslee. *On Intelligence*. 2014.

[9] Davide Maltoni. Pattern Recognition by Hierarchical Temporal Memory. Technical report, 2011.

[10] Dogan Ibrahim. An overview of soft computing. *Procedia Computer Science*, 102:34–38, 2016.

[11] Z. Ghahramani. Probabilistic machine learning and articial intelligence. volume 521, pages 452–459, 2015.

[12] Cristiana Neto, Fábio Senra, Jaime Leite, Nuno Rei, Rui Rodrigues, Diana Ferreira, and José Machado. Different scenarios for the prediction of hospital readmission for diabetic patients. *Journal of Medical Systems*, 45(1), jan 2021.

[13] Barbara Martins, Diana Ferreira, Cristiana Neto, António Abelha, and José Machado. Data mining for cardiovascular disease prediction. *Journal of Medical Systems*, 45(1), jan 2021.

[14] Cristiana Neto, Maria Brito, Vítor Lopes, Hugo Peixoto, António Abelha, and José Machado. Application of data mining for the prediction of mortality and occurrence of complications for gastric cancer patients. *Entropy*, 21(12), 2019.

[15] Iyad Aqra, Norjihan Abdul Ghani, Carsten Maple, José Machado, and Nader Sohrabi Safa. Incremental algorithm for association rule mining under dynamic threshold. *Applied Sciences*, 9(24), 2019.

[16] Rita Reis, Hugo Peixoto, José Machado, and António Abelha. Machine learning in nutritional follow-up research. *Open Computer Science*, 7(1):41–45, dec 2017.

[17] Ana Pinto, Diana Ferreira, Cristiana Neto, António Abelha, and José Machado. Data mining to predict early stage chronic kidney disease. *Procedia Computer Science*, 177:562 – 567, 2020. The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops.

[18] I.; Hinton G. Krizhevsky, A.: Sutskever. Imagenet classification with deep convolutional neural networks. *Proc. Advances in Neural Information Processing Systems*, 25:1090–1098, 2012.

[19] C.; Najman L.; LeCun Y. Farabet, C.; Couprie. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell*, 35:1915–1929, 2013.

[20] A.; Povey D.; Burget L.; Cernocky J. Mikolov, T.; Deoras. Strategies for training large scale neural network language models. *Proc. Automatic Speech Recognition and Understanding*, page 96–201, 2011.

[21] L. Cardoso, F. Marins, R. Magalhães, N. Marins, T. Oliveira, H. Vicente, A. Abelha, J. Machado, and J. Neves. Abstract computation in schizophrenia detection through artificial neural network based systems. *The Scientific World Journal*, 45, 2015.

[22] J.; Bottou L.; Karlen M.; Kavukcuoglu K.; Kuksa P. Collobert, R.; Weston. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[23] Vernon B. Mountcastle. The columnar organization of the neocortex. volume 120, pages 701–722, April 1997.

[24] Kathleen S. Rockland and Noritaka Ichinohe. Some thoughts on cortical minicolumns. *Experimental Brain Research*, 158(3), July 2004.

[25] Kevan A.C. Douglas, Rodney J.; Martin. Neuronal circuits of the neocortex, 2004.

[26] Jeff Hawkins and Subutai Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, 10, mar 2016.

[27] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A theory of how columns in the neocortex enable learning the structure of the world. *Frontiers in Neural Circuits*, 11, oct 2017.

[28] Marcus Lewis. Locations in the neocortex: A theory of sensorimotor object recognition using cortical grid cells, April 2019.

[29] Roddy M. Grieves and Kate J. Jeffery. The representation of space in the brain. *Behavioural Processes*, 135:113–131, feb 2017.

[30] Yi Gu, Sam Lewallen, Amina A. Kinkhabwala, Cristina Domnisoru, Kijung Yoon, Jeffrey L. Gauthier, Ila R. Fiete, and David W. Tank. A map-like micro-organization of grid cells in the medial entorhinal cortex. *Cell*, 175(3):736–750.e30, oct 2018.

[31] Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. The htm spatial pooler - a neocortical algorithm for online sparse distributed coding. *Frontiers in Computational Neuroscience*, 11, nov 2017.

[32] P. Földiák. Forming sparse representations by local anti-hebbian learning. *Biological Cybernetics*, 64(2):165–170, dec 1990.

[33] Wulfram Gerstner and Werner M. Kistler. Mathematical formulations of hebbian learning. *Biological Cybernetics*, 87(5-6):404–415, dec 2002.

[34] Sen Song, Kenneth D. Miller, and L. F. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, sep 2000.

[35] Richard Kempter, Wulfram Gerstner, and J. Leo van Hemmen. Hebbian learning and spiking neurons. *Physical Review E*, 59(4):4498–4514, apr 1999.

[36] Liming Xie, Kai Yang, and Xiaorong Gao. Multi-object recognition by optimized hierarchical temporal memory network. *Optik*, 127(19):7594–7601, oct 2016.

[37] Scott Purdy. Encoding data for htm systems, 2016.

[38] Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. Continuous online sequence learning with an unsupervised neural network model. *Neural Computation*, 28(11):2474–2504, nov 2016.

[39] Fabián Fallas-Moya and Francisco Torres-Rojas. Object recognition using hierarchical temporal memory. In *Intelligent Computing Systems*, pages 1–14. Springer International Publishing, 2018.

[40] Afeefa PP and Pillai Praveen Thulasidharan. Automatic license plate recognition(alpr) using htm cortical learning algorithm. In *2017 International Conference on Intelligent Computing and Control*, 2017.

[41] Jia Wu, Weiru Zeng, and Fei Yan. Hierarchical temporal memory method for time-series-based anomaly detection. *Neurocomputing*, 273:535–546, jan 2018.

[42] Yuwei Cui, Chetan Surpur, Subutai Ahmad, and Jeff Hawkins. A comparative study of htm and other neural network models for online sequence learning with streaming data. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, jul 2016.

[43] Jakob Struye and Steven Latré. Hierarchical temporal memory and recurrent neural networks for time series prediction: An empirical validation and reduction to multilayer perceptrons. *Neurocomputing*, 396:291–301, jul 2020.

[44] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, nov 2017.

[45] Anandharaj A. and P. Bagavathi Sivakumar. Anomaly detection in time series data using hierarchical temporal memory model. In *Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019]*. IEEE, 2019.

[46] Matthew Taylor, Scott Purdy, Breznak, Chetan Surpur, Austin Marshall, David Ragazzi, Subutai Ahmad, Numenta-Ci, Andrew Malta, Pascal C. Weinberger, , Akhila, Marcus Lewis, Richard Crowder, Marion Le Borgne, Yuwei, Christopher Simons, Ryan J. McCall, Luiz Scheinkman, Mihail Eric, Utensil Song, Keithcom, Nathanael Romano, Sagan Bolliger, Vitaly-Krugl, James Bridgewater, Ian Danforth, Jared Weiss, Tom Silver, David Ray, and Zuhaagha. Numenta/nupic: 1.0.5, 2018.

[47] Zohar Jackson, César Souza, Jason Flaks, Yuxin Pan, Hereman Nicolas, and Adhish Thite. Jakobovski/free-spoken-digit-dataset: V1.0.8, 2018.

[48] Numenta. *HTM Research code repository*, 2020 (accessed October 25, 2020). https://github.com/numenta/htmresearch.

A

PUBLICATION

HIERARCHICAL TEMPORAL MEMORY THEORY APPROACH TO STOCK MARKET TIME SERIES FORECASTING

Authors: Regina Sousa, Tiago Lima, António Abelha and José Machado

Title: Hierarchical Temporal Memory theory approach to stock market time series forecasting

Year: 2020

Abstract: Over the years and with the technological innovations that have appeared, the relevance of automatic learning methods has increased exponentially, playing a determining role in society. More specifically, Deep Learning with the ability to recognize audio, image and time series prediction, has helped to solve various types of problems. In this article, a new theory is presented, called HTM. HTM is based on biological functions of the brain as well as its learning mechanism. The theory that will be later described has been applied to a time series forecast, with close values in the stock market, for seven of the S&P500 index companies, Amazon, Google, HCA Healthcare, Disney, McDonald"s, Johnson Johnson and Visa. The results are of significant relevance, showing a low percentage of error in forecasts made over time. It can be stated that the learning curve of the algorithm is fast identifying trends in the stock market for the seven data universes. Briefly, HTM presents itself as a good continuous learning method for forecasting time series data sets, being robust for the tuning of hyper-parameters between different data sets in the same problematic sphere.

Keywords: Time Series Forecasting; Hierarchical Temporal Memory; Stock Market; Regression; Machine Intelligence, Deep Learning

State: Submitted for publication