

Predicció del preu de criptomonedes utilitzant tècniques de Deep Learning

Abel Cervantes Álvarez

Resum— L'objectiu d'aquest projecte consisteix en aplicar de manera comparativa tres models diferents de Deep Learning que serveixin per a predir quin serà el preu d'una criptomoneda en el minut següent i el desenvolupament d'una aplicació que posi els models en producció per ajudar en la presa de decisions a l'hora de fer inversions a curt termini. Consisteix en resoldre un problema de predicció de tendències utilitzant diferents tipus de xarxes neurals artificials. Les dades a utilitzar provenen de diferents fonts, es fan servir dades històriques dels moviments de la pròpia criptomoneda, indicadors financers i informació de la Blockchain. Els models emprats són: el perceptró multicapa, xarxes Long short-term memory (LSTM) i la combinació de capes ConvNet i LSTM. Per a la construcció de cada model s'aplica la mateixa metodologia amb l'objectiu d'aconseguir els millors resultats possibles tenint en compte les limitacions de temps i hardware. Es fa un anàlisi comparatiu i es detallen els punts forts i febles de cada model aplicat en la predicció de tendències. Finalment, els models es posen en producció en una aplicació que permet observar el rendiment en temps real.

Paraules clau— Criptomoneda, Bitcoin, Deep Learning, Machine Learning, Xarxes Neurals, Xarxes Neurals Recurrents, Xarxes Neurals Convolucionals, LSTM, Predicció de tendències, Algorisme de trading, Aprenentatge supervisat, Keras, Python.

Abstract— The aim of this project consists of a comparative application of three different Deep Learning models which are capable to predict the price of a cryptocurrency in the next minute and the development of an application which puts the models in production and helps with short term invest decision making. It consists of the resolution of a time series forecasting problem using different kinds of artificial neural networks. The data is collected from different sources, historical price and volume, technical analysis indicators and Blockchain data. The models used are: Multilayer perceptron, Long short-term memory (LSTM) and the combination of convolutional networks (ConvNet) and LSTM. For each model the same methodology is applied with the aim of achieve the best performance considering the limitations of time and hardware. A comparative analysis is made that shows the strengths and weakness of each model applied to time series forecasting. Finally, models are put in production into an application that shows the performance with real live data.

Index Terms— Cryptocurrency, Bitcoin, Deep Learning, Machine Learning, Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, LSTM, Time Series Forecasting, Trading algorithm, Supervised Learning, Keras, Python.



1 INTRODUCCIÓ

EL 16 de Desembre de 2017, el Bitcoin va assolir el seu preu més alt, 19.343,04 \$. En un any, el seu preu es va multiplicar per 30. En l'actualitat (2018), el seu preu fluctua entre els 6.000 i els 11.000 \$.[1]

El 31 d'Octubre de 2008, Satoshi Nakamoto (un pseudònim del creador o creadors) va compartir un *paper* anomenat *Bitcoin: A Peer-to-Peer Electronic Cash System*[2] en una *mailing list* de criptografia. En aquest *paper* es detalla un sistema per realitzar transaccions de forma electrònica sense la necessitat d'una tercera part mitjançant el protocol P2P i una tecnologia que més endavant rebia el nom de *Blockchain*. No va ser fins al Gener de 2009 quan es va posar en marxa la xarxa Bitcoin i es va fer públic el primer client *open source*.

L'autor d'aquest sistema va agafar idees d'altres projectes

molt semblants que ja s'havien intentat desenvolupar en la dècada dels 90 com el DigiCash[3] i el BitGold[4] a la dècada dels 2000, tot just després de la gran crisi econòmica. Tots aquests projectes estaven influenciats per un moviment anomenat *Chyperpunk*[5], un grup format per gent que creu en les possibilitats que ofereix la criptografia per construir sistemes que respectin la privadesa i que el control de les accions i la confiança resideixi en les persones que utilitzen el servei i no en institucions o empreses privades que fan d'intermediari.

De la mateixa manera que el diner fiduciari[6], el Bitcoin és un sistema de comptabilitat. És un mètode per registrar quants diners tenim, a qui hem prestat diners, qui ens ha prestat a nosaltres, quines operacions de compra i venda hem fet, etc. La principal diferència amb el sistema tradicional és que no es necessita una tercera part que s'encarregui de controlar aquesta comptabilitat. En el sistema tradicional, aquesta part la ocupen els bancs, que s'encarreguen de realitzar les transaccions i comprovar que els diners són reals. En el sistema que proposa Nakamoto, aquest control el realitzen els mateixos usua-

- E-mail de contacte: abel.cervantes@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Gemma Sanchez Albadalejo (Computer Vision Center)
- Curs 2017/18

ris que formen part del sistema. Totes les operacions es registren en un gran llibre de comptabilitat i són els usuaris els que d'una manera col·lectiva i consensuada, controlen i afegeixen registres. D'aquesta manera s'evita l'ús d'un sistema centralitzat, ineficiències, comissions de tercers, corrupció, etc.

Aquest gran llibre de comptabilitat és obert, està disponible de manera online i replicat en cada computador que decideixi participar en el manteniment del sistema. Les operacions es registren de forma anònima amb l'ús d'adreces *hash* per ocultar als usuaris però la informació de les transaccions és totalment pública i es poden consultar i rastrejar des de un navegador[7]. A diferència d'un sistema tradicional, no existeix un registre centralitzat, i per tant, les possibilitats de hackejar el sistema són realment nul·les, ja que s'hauria d'alterar el llibre en una gran part dels computadores que formen part del sistema a la vegada[8].

El llibre està format per una cadena de blocs (Figura 1) de mida il·limitada. En cada bloc trobem un número variable de transaccions (*Tx*) que s'ajusta a les limitacions de mida del bloc, més tota la informació i les metadades necessàries per tal de que el protocol Bitcoin funcioni.

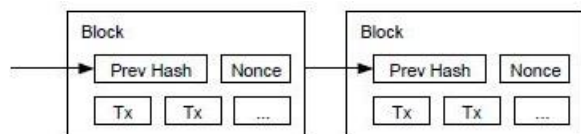


Figura 1. Cadena de blocs segons Nakamoto.

Una vegada s'afegeix un bloc a la cadena, la operació és irreversible, ningú pot modificar ni eliminar contingut dels blocs, només afegir-ne. Els encarregats de recollir les transaccions, construir els blocs i afegir-los a la cadena s'anomenen *miners*. L'operació de minar consisteix en aconseguir un determinat número de zeros a l'inici d'un codi *hash* després de realitzar dos funcions *hash* SHA-256 consecutives a partir de la informació del bloc. Per tenir diferents resultats *hash* per a un mateix bloc, s'utilitza un atribut anomenat *Nonce* (*mentrestant*) que permet la seva modificació i per consegüent, la possibilitat de generar diferents *hash*. Aquest procés de modificar el *Nonce* i computar funcions *hash* s'anomena *hashcash proof-of-work* (*PoW*) i és una competició entre tots els miners on la recompensa consisteix en una quantitat de Bitcoins frescos més la suma de les comissions que afegeixen els usuaris en les seves transaccions.

Amb el pas del temps el Bitcoin ha anat guanyant popularitat, són molts els usuaris que participen i de diferent manera, per una banda hi ha els miners, motivats per l'al·licient de generar diners a canvi de potència computacional, n'hi ha que aprofitant l'anonimat i la no regulació de la moneda, aprofiten per realitzar accions il·lícites

que han provocat moments on s'ha qüestionat el projecte, apareixen alguns comerços físics i digitals que accepten criptomonedes com a sistema de pagament, caixers automàtics als carrers i usuaris que aprofitant la volatilitat dels preus aprofiten per fer inversions.

En l'actualitat, inspirant-se en el projecte Bitcoin, s'han creat més de 1000 projectes de criptomoneda amb diferents particularitats, anomenades *alt coins*[9]. S'han desenvolupat plataformes d'intercanvi de divises (*exchanges*) on els usuaris poden comprar monedes amb divises normals o altres criptomonedes, fet que ha provocat la creació d'un nou producte d'inversió global[10].

2 OBJECTIUS

L'objectiu d'aquest TFG consisteix en crear una aplicació que pugui predir preus futurs de criptomonedes a curt termini utilitzant tècniques de Deep Learning i que serveixi per donar suport a l'hora de prendre decisions en el trading de criptomonedes.

El problema consisteix en predir el que passarà analitzant les dades d'una tendència, el que es coneix com a *Time Series Forecasting* (Predicció de tendències).

La criptomoneda sobre la qual es construiran els models serà el Bitcoin, degut a que fins al moment, és la moneda amb més informació històrica i també la més accessible per a la obtenció de dades.

S'utilitzaran dades de diferents fonts, algunes pròpies de la utilització de la moneda com el *OHLC* (*Open High Low Close*) i el volum, altres relacionades amb la tecnologia *Blockchain*, *hash rate*, dificultat, número de transaccions, comissions i també indicadors que es fan servir en anàlisi tècnic tradicional com el *momentum* i les mitjanes mòbils.

Les prediccions es realitzaran minut a minut, és a dir, el sistema ens indicarà el preu que tindrà la moneda en el minut següent.

L'aplicació permetrà la selecció de 3 models diferents de Deep Learning per fer les prediccions. El primer model consisteix en el perceptró multicapa (xarxa neural artificial composta per varies capes ocultes que estan completament connectades entre elles), el segon model consisteix en les capes Long short-term memory (xarxa neural artificial recurrent, on les neurones tenen memòria) i per últim, el tercer model consisteix en l'aplicació de capes convolucionals i Long short-term memory (Xarxes convolucionals que treballen amb una dimensió i són capaces d'extreure característiques de tendències).

Per últim, es construirà una interfície gràfica (Figura 2) que recollirà dades en temps real de la criptomoneda i

que permetrà a l'usuari seleccionar el model amb el qual vol fer les prediccions.

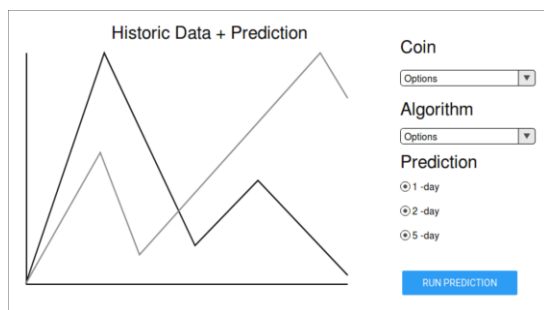


Figura 2. Prototip de la interfície gràfica.

3 ESTAT DE L'ART

3.1 Trading algorísmic

L'aplicació d'algorismes d'intel·ligència artificial en el món de les finances és una realitat des de fa més de 30 anys. A l'edat de 40 anys, Jim Simons, un matemàtic nord-americà, després de realitzar algunes operacions de *trading* amb les quals va obtenir certs beneficis per pura sort, va observar que existien estructures que permetien predir els moviments dels mercats. A partir d'aquesta observació, Jim va conformar un equip de científics i van començar a construir models predictius basant-se en grans quantitat de dades (en aquell temps, recollides a mà), essent un dels pioners en l'aplicació del Machine Learning en aquest àmbit.[11]

Des de la dècada dels 70, el trading algorísmic acompanya als *traders* humans, on diferents estratègies han sigut implementades, des de estratègies que segueixen regles predefinides basades en models estadístics fins a models que es basen en l'aprenentatge de la repetició de patrons a partir d'indicadors, informació històrica, xarxes socials, notícies, resultats periòdics de les companyies, etc.

En l'actualitat, s'estima que aproximadament el 90% de les operacions estan produïdes per un algorisme, mentre que el 10% restant són produïdes per un humà[12]. Tot i això, els rendiments que aconseguixen els *hedge funds* que utilitzen intel·ligència artificial encara estan per sota dels tradicionals. Hi ha estudis, però, que afirmen haver aconseguit estratègies basades en algorismes de Deep Learning que presenten millors retorns que els models tradicionals[13].

3.2 Trading algorísmic de criptomonedes

En l'àmbit de les criptomonedes, han aparegut nous *hedge funds*[14] que s'han especialitzat únicament en la inversió d'aquestes, mentre que altres ja existents han afegit les criptomonedes com un producte nou.

És sapigut que el Machine Learning comença a tenir un paper important en el món de les finances, tant és així, que existeix un índex[15] per avaluar el rendiment que tenen els *hedge funds* que prenen decisions basades en aquest tipus d'algorismes. En el cas de les criptomonedes no està tan clar. Existeixen algunes plataformes que asse-

guren la utilització d'aquests algorismes[16] però no està tan estandarditzat com en els altres productes.

Actualment, l'aplicació del Deep Learning en les criptomonedes és més aviat un camp d'estudi. En la comunitat investigadora existeixen multitud de *papers*[17] on els autors intenten construir models basant-se en anàlisi de sentiment o en informació històrica.

3.3 Hipòtesi dels mercats eficients

L'aplicació del Machine Learning per a predir el comportament dels mercats és un tema que genera controvèrsia i existeix un debat entre economistes, matemàtics i especuladors. Existeix una hipòtesi desenvolupada per Eugene Fama[18], que assegura que els mercats són informalment eficients, el preu dels productes financers s'ajusta ràpidament a la nova informació, de manera que no hi ha cap producte infravalorat o sobrevalorat. Aquesta hipòtesi implica que no és possible "guanyar al mercat" sinó és a partir d'informació privilegiada. Per tant, la informació històrica no és rellevant per a determinar el comportament futur.

4 METODOLOGIA

4.1 Tecnologia

Per resoldre problemes de Time Series Forecasting existeixen diverses tecnologies, algunes basades en models estadístics com ARIMA[19] i altres relacionades amb el Machine Learning. En aquest cas, s'utilitzaran tècniques de Machine Learning, i més concretament, Deep Learning. El Deep Learning[20] consisteix en l'aplicació de models de xarxes neurals artificials amb múltiples capes ocultes en la seva arquitectura (Figura 3).

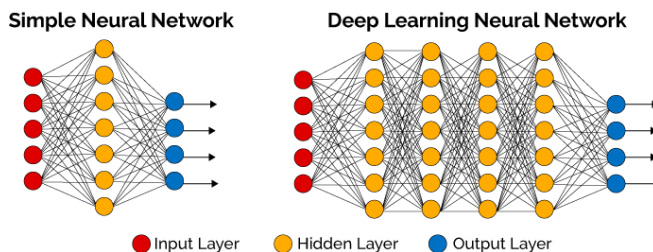


Figura 3. Perceptró vs perceptró multicapa

Una xarxa neural artificial és un sistema computacional que simula un procés d'aprenentatge i que s'inspira en el funcionament biològic de les xarxes neurals que componen el cervell dels éssers vius.

4.1.1 Breu Història de les Xarxes Neurals Artificials

A l'any 1943, McCulloch i Walter Pitts van crear un model computacional per imitar el funcionament d'una neurona, basat en la lògica matemàtica[21]. El seu treball va obrir la via d'estudi de les xarxes neurals en el camp de la intel·ligència artificial. A partir d'aquest model, Rosenblatt va crear el perceptró[22], un algorisme per al reconeixement

ment de patrons, considerat com la xarxa neural *feed-forward* més simple, ja que només està format per una capa. Els primers models basats en xarxes neurals tenien dos problemes: No podien processar la operació OR exclusiva i els computadors de l'època no eren prou potents[23]. A l'any 1975, Werbo, va crear l'algorisme *Backpropagation*[24] que solucionava el primer problema, accelerava el procés d'aprenentatge i tornava a d'espertar l'interès en l'estudi de les xarxes neurals.

4.1.2 Procés d'aprenentatge

El funcionament d'aquests algorismes consisteix en el mapeig d'unes entrades (els exemples) a una sortida (el target o sortida objectiu). Aquest procediment es realitza observant una gran quantitat d'exemples amb la seva corresponent sortida associada. El mapeig entre entrada y sortida es realitza a partir d'una seqüència de transformacions simples presents en les neurones de cada capa. El resultat de les transformacions varia segons la funció utilitzada i els pesos associats. El procés d'aprenentatge consisteix en ajustar aquests pesos per aconseguir que el mapeig entre entrada i sortida sigui el més semblant al que indiquen els exemples i la seva sortida associada. Quan parlem de Deep Learning, ens referim a l'existència d'un model d'aquestes característiques, amb múltiples capes (Deep) i amb desenes o milions de pesos (Learning, procés d'aprendre el valor d'aquests pesos).

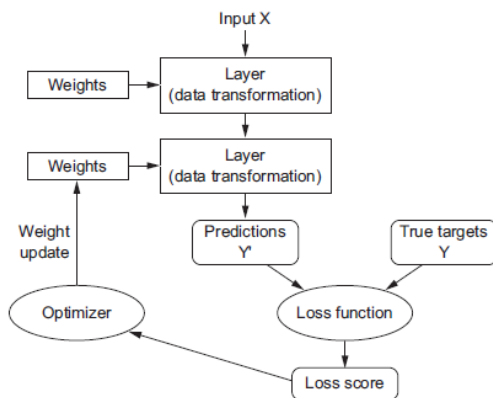


Figura 4. Neural Network Workflow

Per ajustar els pesos és necessari mesurar quina és la distància entre la sortida que genera el model y la sortida real. Aquesta tasca la realitza la funció de pèrdua (*Loss function*) o també funció objectiu, que bàsicament s'encarrega de calcular la diferència (*Loss score*) entre les dues sortides. L'actualització dels pesos a la xarxa la realitza l'optimitzador, aquest fa un petit ajust dels pesos en la direcció que indica la *Loss score* i d'aquesta manera ajuda a que l'exemple que esta avaluant en aquell moment s'acosti més a la sortida esperada. L'algorisme per ajustar els pesos s'anomena *Backpropagation*.

A l'inici, els pesos tenen un valor aleatori (o qualsevol altre mètode d'inicialització de pesos), les primeres transformacions i càlculs de la *Loss function* deriven en *Loss scores* molt altes, però a mesura que la xarxa va analitzant més i més exemples, aquesta *Loss score* va disminuint i el

resultat és una xarxa amb la funció de pèrdua minimitzada, la sortida associada a cada exemple està el més a prop possible a la sortida real per a aquell conjunt d'exemples.

Fins al moment, aquests models han demostrat resultats prometedors en algunes tasques com la classificació d'imatges, generació de text, traducció, generació d'imatges, conducció autònoma, sistemes recomanadors, motors de cerca, superjugador de Go, entre d'altres[25].

Per a la realització d'aquest treball s'analitzen tres models diferents de Deep Learning que s'expliquen a continuació.

4.1.3 Perceptró Multicapa (MLP - ANN)

El perceptró multicapa és un tipus de xarxa neural artificial *feedforward*, sense cicles, composta per almenys tres capes ocultes exceptuant la capa d'entrada. Les funcions d'activació de les neurones són no lineals i el funcionament de la xarxa coincideix exactament amb l'explicat anteriorment.

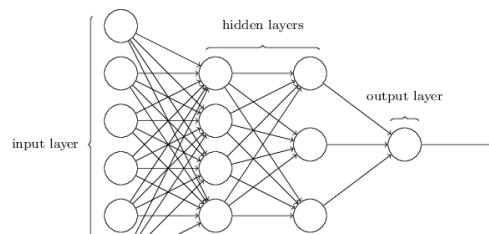


Figura 5. Perceptró Multicapa

L'objectiu d'aquesta xarxa consisteix en mapejar la informació de cada minut amb el preu del minut següent. No es tindrà en compte l'ordre de les entrades, cada input és independent de l'anterior, per tant les entrades podran ser introduïdes de forma aleatòria. El resultat d'aquest model s'espera que no sigui gaire eficient, ja que no al no tenir-se en compte el factor del temps, la informació que conté el minut no hauria de ser prou rellevant per a saber amb exactitud el que passarà al minut següent.

4.1.4 Xarxa Neural Recurrent (LSTM - RNN)

A diferència del perceptró, les xarxes recurrents tenen memòria. Aquestes xarxes permeten mantenir l'estat del que han vist prèviament, de manera que aquest estat té influència alhora de realitzar les transformacions sobre les noves entrades.

Concretament es faran servir les capes Long Short-Term Memory (LSTM), desenvolupades per Hochreiter and Schmidhuber a l'any 1997. Aquesta variant de capa recurrent és una solució al que s'anomena *Vanishing Gradient Descent Problem*[26], i es produeix quan es necessita retenir informació per a ser recordada des de molts passos enrere. Aquest fet també succeeix en el perceptró multicapa quan el número de capes és excessivament gran, el model no convergeix.

Com podem veure a la figura 6, la memòria (*carry track*) es guarda per a ser accessible quan es necessiti, de manera que la informació més llunyana segueix intacte i pot ser utilitzada en cada cel·la quan es necessiti.

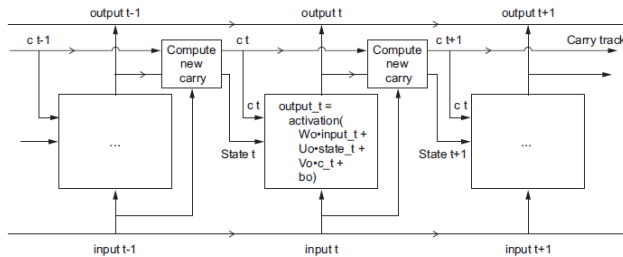


Figura 6. Anatomia cel·la LSTM

Aquestes capes són computacionalment molt més intenses, requereixen més memòria i realitzen més càlculs, degut al superior nombre de paràmetres respecte al perceptró. Tot i així, teòricament és un dels models candidats a resoldre millor el problema.

4.1.5 ConvNet + LSTM (CVN + RNN)

Les xarxes convolucionals[27], no sols tenen aplicació en l'aprenentatge basat en imatges, sinó que també es poden aplicar per extreure característiques de tendències.

El següent model que es proposa (Figura 7), consisteix en l'aplicació d'una capa convolucional d'una dimensió que generi una nova seqüència de característiques més reduïda per més endavant servir com a entrada a una o unes capes LSTM. L'intenció d'aquesta proposta es reduir la complexitat de les capes LSTM, ja que suposen un gran temps d'entrenament i tenint en compte els recursos dels que es disposa i el temps, aquesta proposta pot generar resultats semblants en menys temps.

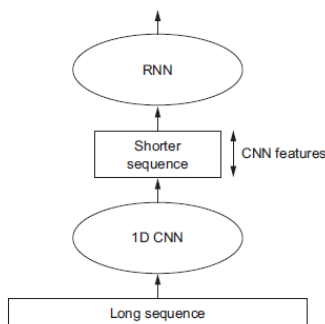


Figura 7. CNN + LSTM Model

4.2 Recursos

L'objectiu d'aquest treball no consisteix en la implementació d'aquests algorismes, si no en la seva utilització i la parametrització d'hyperparametres per aconseguir el millor percentatge d'efectivitat en la resolució del problema.

Per construir models de forma ràpida i senzilla, s'ha optat per la utilització de Keras[28], un framework gratuït desenvolupat per François Chollet que permet el prototipatge ràpid de models Deep Learning utilitzant Tensor-

Flow com a *backend*. TensorFlow[29] és el framework de Machine Learning més utilitzat en tot el món[30], i Keras l'utilitza millorant la seva sintaxi i millorant la productivitat. Conjuntament amb Keras, s'ha optat per fer servir un wrapper anomenat Hyperas[31] que permet aplicar tècniques per optimitzar els hyperparametres d'un model construït amb Keras. Per a la manipulació de dades i la construcció dels datasets, s'ha optat per la utilització de Numpy[32] i Pandas[33]. El primer per realitzar operacions amb matrius i tensors i el segon per gestionar i fer transformacions sobre datasets. Per a la visualització de dades s'ha optat per matplotlib[34]. La interfície gràfica està realitzada amb TKinter[35], el Toolkit per defecte per a construir GUI's en Python. Els datasets que contenen la informació històrica minut a minut estan disponibles a Kaggle[36] i la informació històrica de la Blockchain està disponible a blockchain.info[37]. El hardware utilitzat consisteix en una CPU Intel i7 2700K i una GPU Nvidia 1070.

4.4 Deep Learning workflow

La metodologia emprada per a la construcció de models ha consistit en el seguiment d'un *workflow* universal [38] utilitzat en aquest tipus de models. A continuació s'expliquen les etapes del workflow i els continguts treballats en cadascuna.

4.4.1 Definició del problema

Es vol predir el preu que tindrà el Bitcoin al minut següent.

Hipòtesi: La informació històrica que genera el Bitcoin, la Blockchain, els indicadors calculats d'anàlisi tècnic i la informació de les xarxes socials poden reflexar el comportament dels inversors alhora d'invertir en segons quines condicions i les regles d'altres mètodes automàtics d'inversió.

El problema a resoldre consisteix en l'aprenentatge de patrons que apareixen en les tendències de preu i volum, la seva correlació amb les dades que genera la Blockchain, els indicadors d'anàlisi tècnic i el sentiment a les xarxes socials.

És un problema d'aprenentatge supervisat de tipus Scalar Regression (predicció d'un únic valor continu, el preu). La proposta consisteix en un model multi-entrada[39], una entrada serà el model de la informació numèrica històrica i l'altre correspondrà a l'anàlisi de sentiment a les xarxes socials.

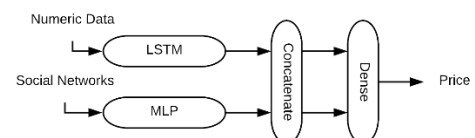


Figura 8. Proposta multi input model

Aquest treball es centrarà en la realització del model que treballa amb la informació numèrica.

4.4.2 Construcció del data set

La part més important per resoldre un problema fent servir Deep Learning és la d'aconseguir dades que representin bé el problema a resoldre i tenir-ne una quantitat suficient. D'aquesta manera, amb l'entrenament d'un model adequat i ben parametrizat, serà possible arribar a una adequada generalització del problema per poder predir amb un cert nivell d'efectivitat sobre dades mai vistes.

Les dades provenen de diferents fonts, per una banda tenim la informació històrica OHLC (minut a minut, Open High Low Close Volum) des de Gener del 2012 fins al Març de 2018, disponible a Kaggle (Figura 9).

Timestamp	Open	High	Low	Close	Volume_(BTC)	Volume_(Currency)	Weighted_Price
1499159640	294000	294000	293950	293950	30.434878	8946524.2701	293956.30468
1499159700	293950	294000	293801	294000	5.49290504	1614717.9041	293964.28526
1499159760	294000	294000	293993	293999	3.30529932	971745.60594	293996.25022

Figura 9. Kaggle minute by minute Bitcoin dataset

Per altre banda, des de Blockchain.info, podem obtenir la informació històrica de:

- Dificultat[40]: Indicador de la dificultat actual per minar la criptomoneda.
- Hash rate[41]: Potència de càlcul actual que hi ha al sistema Bitcoin.
- Preu de les transaccions: Quantitat que es paga de comissions.
- Número de transaccions diàries.

El target o sortida objectiu és el valor Close, que representa el preu de "tancament" de la moneda per a cada minut. Sobre aquest preu es calcularà un indicador anomenat Momentum[42] que es fa servir per fer anàlisi tècnic i que representa la diferència entre preus en dos punts diferents en el temps. És un indicador que s'utilitza per treure senyals de compra o de venda.

Totes les features s'agrupen i es crea un únic dataset general. A partir d'aquest dataset es realitzen les següents operacions:

- Gap fill: Neteja de NaN values. L'estratègia que es segueix per a omplir els forats consisteix en aplicar el valor anterior fins a que el forat desapareix[43].
- Histograma de cada feature. Les dades segueixen distribucions normals? La resposta és No totes.
- Visualització de dades. S'observa que tenen escales molt diferents, cal tenir aquest factor en compte per aplicar normalització.
- Normalització de dades, Estandardització. Com que les dades tenen escales diferents i no segueixen una distribució normal, les dades s'estandarditzen en un rang 0-1. Les dades de validació i test s'estandarditzen amb valors del conjunt d'entrenament per tal de no filtrar informació d'una partició a l'altre[44].

Finalment, s'escurça el dataset eliminant la informació dels orígens de la moneda, on el seu nivell d'utilització era molt baix i poc representatiu.

4.4.3 Mètode de validació i mètrica de rendiment

Per avaluar el rendiment dels models s'ha triat la mètrica *Mean Squared Error*, ja que és una mètrica molt popular i que funciona molt bé en problemes de regressió d'un escalar. Un altre opció hagues sigut la *Mean Absolute Error*, però aquesta mètrica no dóna els mateixos resultats que l'anterior[45]. En canvi, per avaluar el model i com a mètrica de rendiment, s'ha triat el MAE, ja que ens dóna un valor directe del rendiment del model sense haver de realitzar cap transformació.

El mètode de validació triat ha sigut el hold-out, particions del dataset en conjunt d'entrenament, validació i test. S'ha triat aquest mètode ja que es disposa de dades suficients i altres mètodes com el k-fold, nested k-fold o leave-one-out, suposarien temps d'entrenament no raonables amb el hardware que es disposa.

El mètode hold-out consisteix en assignar una proporció del dataset, per exemple un 80% i la resta per test. Després, en el conjunt d'entrenament es torna a fer una partició per al conjunt de validació, per exemple 80% entrenament i 20% validació. Amb el conjunt de validació ajustem la configuració del model per reduir overfitting o underfitting i amb el conjunt de test avaluem el rendiment real del model sobre dades mai vistes.

4.4.4 Construcció i entrenament dels models

La metodologia emprada per a la construcció i configuració dels tres models ha sigut la mateixa, amb la única diferència dels hyperparametres que hi ha disponibles a cada arquitectura.

Al principi s'estableixen de forma més o menys manual les configuracions i algunes parts més automatitzades, com per exemple crear els mateixos models variant número de neurones dins d'un rang o un valor concret d'un altre hyperparametre. Per automatitzar aquesta part, no haver de programar molts bucles aniuats i de forma que afecti a més paràmetres, s'utilitza Hyperas. Amb aquesta eina podem definir rangs en la definició dels valors dels paràmetres i automàticament es queda amb la versió més òptima. Com que els models tarden bastant a entrenar, els rangs no són gaire extensos.

Els hyperparametres que s'han tingut en compte i els rangs per a les configuracions són:

- Número de neurones per capa (1..1024)
- Número de capes (1..10)
- Funcions d'activació (tahn, sigmoid, relu)
- Epochs (5-150)
- Batch Size (2-128)
- Dropout (0-25)
- Recurrent Dropout (0-25)

- L1 L2 Regularization (0.001)
- Optimitzadors (SGD, RMSprop, Adam)
- Learning Rate (0.01-0.0001)
- Loss (MAE, MSE)
- Mètriques (MAE, MSE)
- Proporcions hold-out (80-20, 90-10)
- Momentum Window (5-10)
- Sample Window (1-32)
- LSTM (Stateful, Stateless)
- Last-Layer Activation (Linear, Sigmoid)

4.4.4 Overfitting

Un dels principals reptes del Machine Learning consisteix en ajustar l'optimització del model amb la generalització. L'optimització del conjunt d'entrenament és una tasca que les xarxes neurals saben realitzar molt bé. La generalització és el rendiment que té el model sobre dades mai vistes.

A la figura 10 podem observar l'evolució de la funció de pèrdua durant la fase d'entrenament del model LSTM. La línia de cercles correspon a la puntuació obtinguda en el conjunt d'entrenament mentre que la línia solida correspon a la puntuació del conjunt de validació.

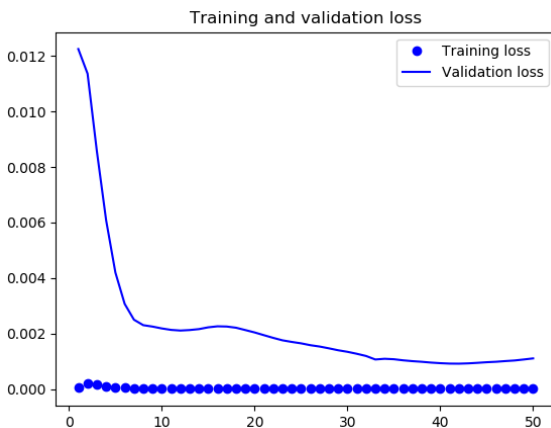


Figura 10. Exemple d'overfitting a la xarxa LSTM

El rendiment que s'obté sobre el conjunt d'entrenament en els tres models és sempre superior al de validació i test, per tant el model presenta overfitting, el model minimitza molt bé sobre el conjunt d'entrenament però quan enseyem dades mai vistes, el rendiment disminueix, el model no està aprenent de forma generalitzada el problema. Per resoldre o minimitzar l'overfitting s'han aplicat diverses tècniques amb resultat negatiu, ja que afectava massa al rendiment final dels models[46].

- Diferents mides de l'arquitectura del model.
- Aplicar Dropout a cada capa.
- L1 i L2 regularization.
- Més dades.

5 RESULTATS

A continuació es presenten les arquitectures finals, les configuracions i els resultats per a cada model.

5.1 Perceptró multicapa (ANN)

Hyperparametre	Valor	Hyperparametre	Valor
Neurones	8-32-32-128-32	Optimitzador	RMSprop
Capas	7	Learning rate	0,001
Activació	ReLU	Funció Loss	MSE
Epochs	50	Mètrica	MAE
Batch size	32	Hold-out	80-20
Dropout	0	Momentum window	10
Recurrent dropout	0	Sample window	1
Regularitzador L1 L2	Cap	Activació ultima capa	Lineal
MAE : 0,0027 (~52\$)			

Taula 1. Configuració xarxa MLP

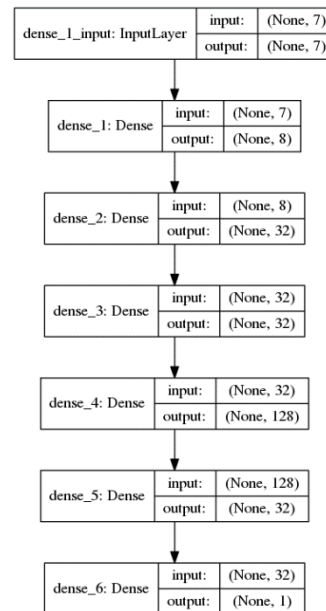


Figura 11. Arquitectura perceptró multicapa

Per al perceptró multicapa s'han realitzat més de 50 configuracions d'hyperparametres diferents amb un promig de temps d'entrenament de 3 hores per cada model. Finalment, l'arquitectura que ha donat millors resultats (Figura 11) està composta per 7 capes contant la capa d'entrada i la última d'activació.

La mètrica de rendiment ens indica que l'error absolut mitjà en el conjunt de test és de 0,0027, que correspon a una mitja de 52\$. Com podem observar a les figura 12, hi ha zones on el rendiment del model no és gaire bo, no segueix la tendència, però també n'hi ha d'altres on la tendència es segueix perfectament i amb poc error (Figura 13).

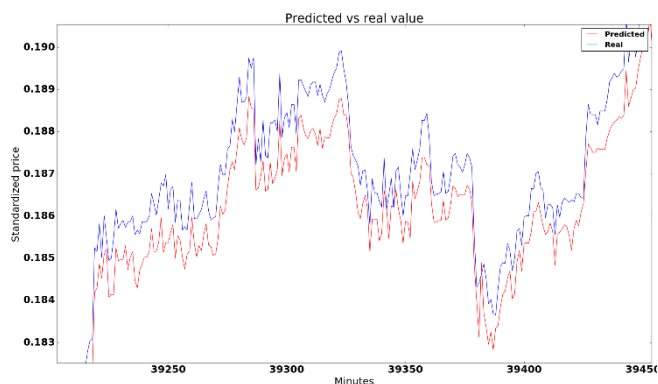


Figura 13. Valor predit vs valor real - MLP

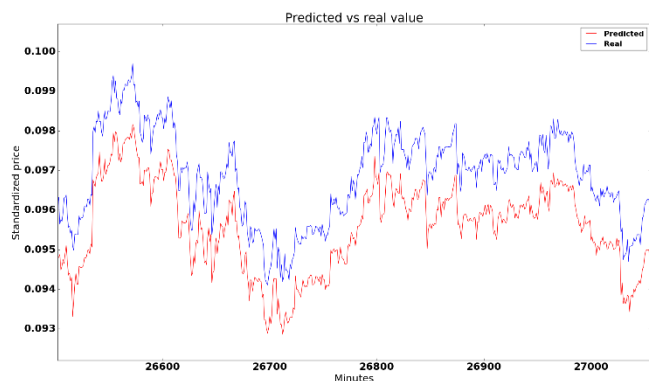


Figura 12. Valor predit vs valor real - MLP

5.2 Long short-term memory (LSTM)

Hyperparametre	Valor	Hyperparametre	Valor
Neurones	32-32	Optimitzador	SGD
Capes	4	Learning rate	0,001
Activació	Tahn	Funció Loss	MSE
Epochs	150	Mètrica	MAE
Batch size	32	Hold-out	80-20
Dropout	0	Momentum window	10
Recurrent dropout	0	Sample window	6
Regularitzador L1 L2	Cap	Activació ultima capa	Lineal
MAE : 0,0022 (~42\$)			

Taula 2. Configuració xarxa LSTM

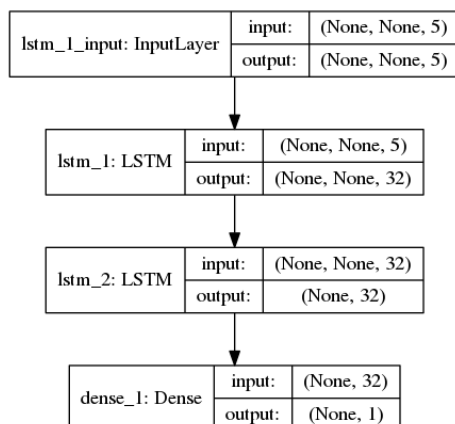


Figura 14. Arquitectura xarxa LSTM

Per a la xarxa LSTM s'han realitzat més de 30 configuracions diferents amb un promig de temps d'entrenament de 10 hores per cada model. L'arquitectura que ha assolit millors resultats consisteix en dues capes LSTM de 32 unitats amb els hyperparàmetres de la taula 2.

La mètrica de rendiment indica que l'error mitjà per a dades mai vistes és de 0,0022, que correspon a 42\$.

Com en el cas del perceptró, hi ha zones on el model té un error superior (Figura 15) i d'altres on l'error és inferior i segueix molt millor la tendència (Figura 16).

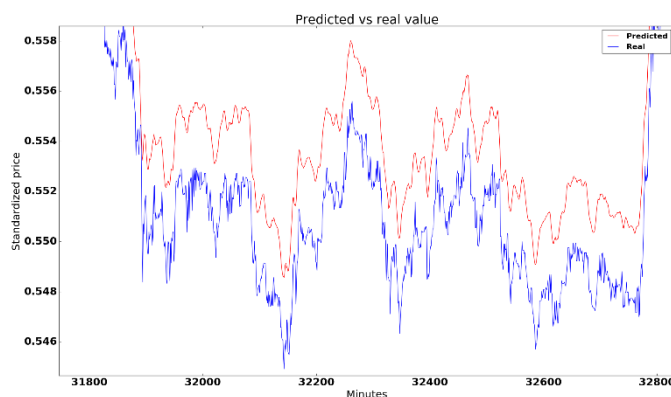


Figura 15. Valor predit vs valor real - LSTM

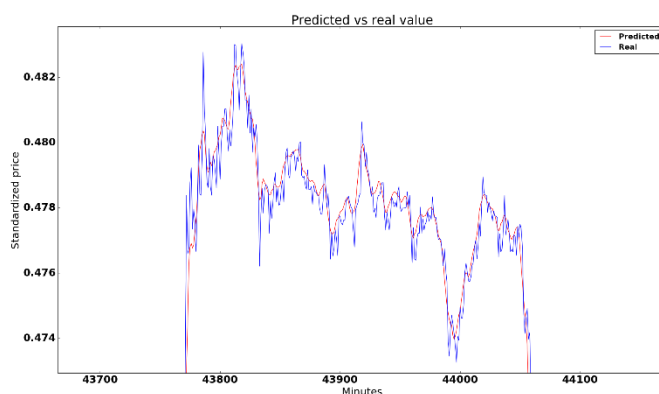


Figura 16. Valor predit vs valor real - LSTM

5.3 ConvNet i LSTM (CNN + LSTM)

Hyperparametre	Valor	Hyperparametre	Valor
Neurones	-	Optimitzador	RMSprop
Capes	6	Learning rate	0,001
Activació	Conv1D	Funció Loss	MSE
Epochs	50	Mètrica	MAE
Batch size	32	Hold-out	80-20
Dropout	0	Momentum window	10
Recurrent dropout	0	Sample window	6
Regularitzador L1 L2	0	Activació ultima capa	Lineal
MAE : 0,01 (~193\$)			

Taula 3. Configuració ConvNet + LSTM

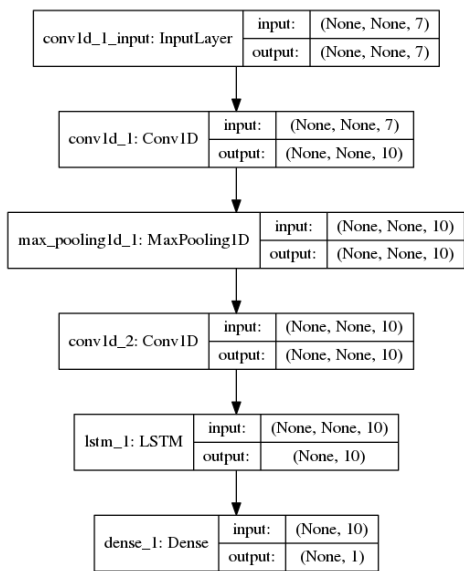


Figura 17. Arquitectura ConvNet + LSTM

Sobre aquest model no s’han pogut realitzar tantes configuracions com en els casos anteriors degut al temps. Concretament s’han pogut entrenar 5 models i la arquitectura que ha donat millors resultats (Figura 17) està composta per 6 capes amb la configuració de la taula 3.

La mètrica de rendiment indica que l’error absolut mitjà és de 0,01 que correspon a 193\$ aproximadament.

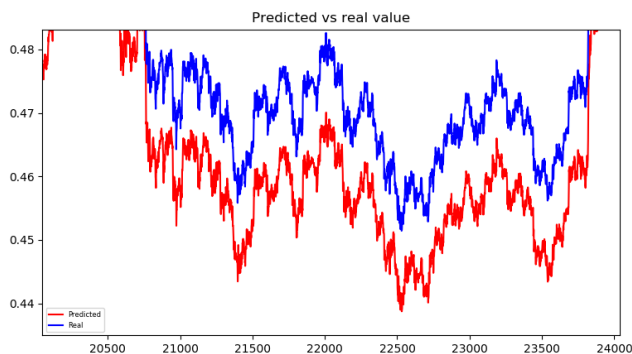


Figura 18. Valor predit vs valor real - ConvNet + LSTM

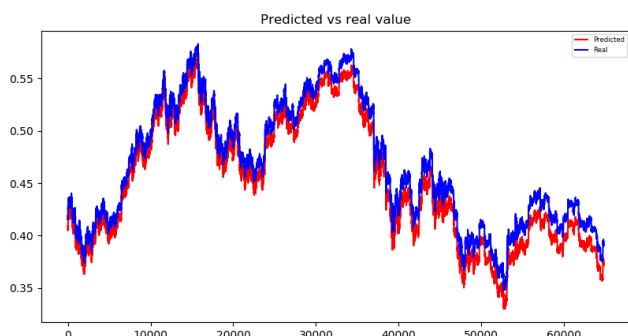


Figura 19. Valor predit vs valor real - ConvNet + LSTM

5.4 Aplicació

Per posar els models en producció he construït una aplicació que segueix en temps real l’evolució dels preus minut a minut. L’aplicació permet seleccionar amb quin tipus d’algorisme es vol fer la predicció, mostra el preu actual i el predit i permet ajustar la finestra en intervals de 10, 20, 30 o 60 minuts.

A la figura següent es mostra l’arquitectura de l’aplicació:

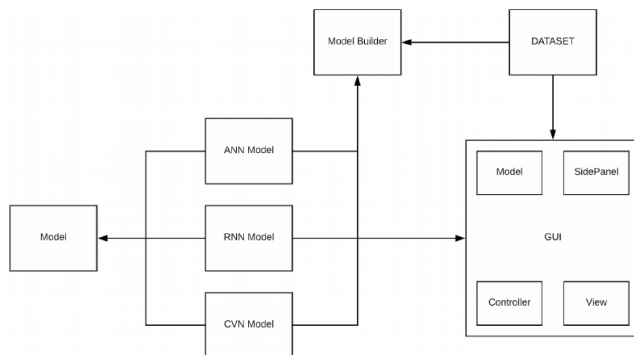


Figura 20. Arquitectura de l’aplicació.

Dataset: Conté tots els mètodes per fer operacions amb el dataset.

Model: Conté tots els mètodes comuns entre els diferents models de xarxes neuronals, tots els models necessiten un mètode per entrenar, predir, carregar les dades, etc.

ANN, RNN i CVN Model: Contenen l’arquitectura de la xarxa.

Model Builder: Conté els scripts per entrenar models, carregar models ja entrenats, visualitzar resultats intermedis, etc.

GUI: Conté la lògica del frontal de l’aplicació, separada en model, vista i controlador. La vista està composta per la part gràfica i un sidepanel que conté tots els widgets.

Resultat del frontal:

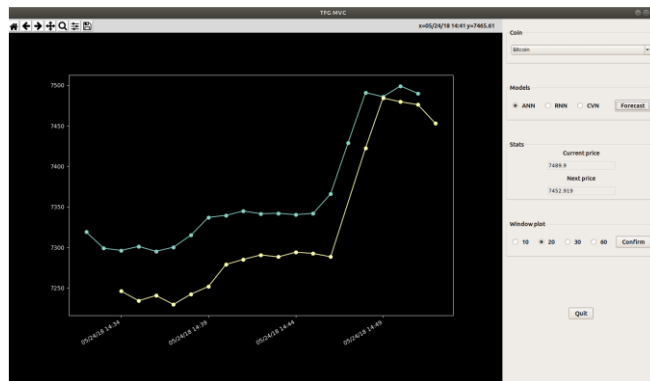


Figura 21. Interfície gràfica

6 Conclusió

A priori, el model recurrent amb capes LSTM hauria de donar els millors resultats per resoldre aquest tipus de problemes, seguit del convolucional i el perceptró. Els resultats obtinguts mostren que el model LSTM dóna millors resultats sobre dades mai vistes i el resultat ha sigut obtingut amb menys esforç comparat amb el perceptró si ens basem en la quantitat de models explorats. Per altre banda, el model convolucional hauria de servir com una alternativa a un model amb moltes capes LSTM, aconseguint resultats similars (a priori, pitjors) en menys temps d'entrenament. En aquest cas, degut a limitacions de hardware, no s'han pogut explorar gaires hiperparàmetres i els resultats per a aquest model no són gaire representatius.

A més dels hiperparàmetres de configuració de cada xarxa, aquest tipus de problema n'afegeix d'altres, com la mida de les finestres amb les quals es representa cada mostra i es calculen els indicadors. Durant la fase d'entrenament, d'alguna manera o altre, l'objectiu consisteix en aprendre la variació en el preu del minut següent en base a la informació que hi ha en els minuts anteriors i aquesta quantitat de minuts es defineix amb una finestra. El perceptró intenta aprendre quin és el comportament del minut següent quan es troba amb finestres similars, el model recurrent fa exactament el mateix però no s'oblida del que ha passat abans de veure aquella finestra i el model convolucional fa el mateix que l'anterior però reduint el número de característiques.

Crec que no és just donar aquests resultats com a definitius per valorar el tipus de model aplicat a aquest problema i el seu rendiment final. El handicap del temps limitat i els recursos hardware per entrenar els models ha sigut clau en els resultats obtinguts. L'espai de cerca continua essent molt gran i per consegüent l'espai de millora dels models. Des de l'elecció de les features candidates i finestres fins al número d'epochs per entrenar el model, existeix un gran número de possibilitats.

També crec que un factor important és la qualitat del data set. El Bitcoin existeix des de fa una dècada però la seva utilització a gran escala només és de 2 anys. Això provoca que a les dades no es veuen reflexades certes situacions que si podria tenir un producte de 10 anys. Aquest fet queda demostrat per l'overfitting existent en cada model, de fet aquest problema es prou conegut i fins i tot han aparegut projectes de criptomoneda per col·laborar a resoldre'l realitzant competicions amb dades anònimes i obtenint recompenses per als millors models[47].

Com a treball futur seria interessant seguir explorant més models i completar l'altre part que realitza un anàlisi de sentiment a partir de dades provinents de xarxes socials, fòrums i repositoris de codi. Els inversors d'aquest producte tenen un perfil 100% digital i moltes decisions es

prenen a partir d'opinions, notícies, activitat dels participants d'un projecte i el full de ruta d'un projecte.

AGRAÏMENTS

Per últim m'agradaria agrair a tot l'equip docent de l'Escola d'Enginyeria per tota la transferència de coneixement durant la realització del grau, a la tutora d'aquest treball per els consells en cada entrega i a la meua família per recolzar-me a l'hora de matricular-me en aquest grau en una certa etapa de la meua vida.

BIBLIOGRAFIA

- [1] Bitcoin price evolution, Blockchain, <https://blockchain.info/es/charts/market-price?timespan=all>. [Consulta Febrer de 2018].
- [2] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, <https://bitcoin.org/bitcoin.pdf>. [Consulta Febrer de 2018].
- [3] David Chaum, DigiCash, <https://en.wikipedia.org/wiki/DigiCash>. [Consulta Febrer de 2018].
- [4] Nick Szabo, BitGold, https://en.wikipedia.org/wiki/Nick_Szabo. [Consulta Febrer de 2018].
- [5] Petri B, The untold history of Bitcoin, <https://medium.com/swlh/the-untold-history-of-bitcoin-enter-the-cypherpunks-f764dee962a1>. [Consulta Febrer de 2018].
- [6] Diario Expansion, Dinero fiduciario, <http://www.expansion.com/diccionario-economico/dinero-fiduciario.html>. [Consulta Febrer de 2018].
- [7] Blockchain, Block explorer, <https://blockchain.info/es>. [Consulta Febrer de 2018].
- [8] Investopedia, 51% attack, <https://www.investopedia.com/terms/1/51-attack.asp>. [Consulta Febrer de 2018].
- [9] CoinMarketCap, List of all cryptocurrencies, <https://coinmarketcap.com/all/views/all/>. [Consulta Febrer de 2018].
- [10] Bitcoin, Bitcoin exchanges, <https://bitcoin.org/en/exchanges#international>. [Consulta Febrer de 2018].
- [11] Ted Talks, Jim Simons: The mathematician who cracked Wall Street, https://www.ted.com/talks/jim_simons_a_rare_interview_with_the_mathematician_who_cracked_wall_street/up-next. [Consulta Maig de 2018].
- [12] Financial Times, The quickening evolution of trading, <https://www.ft.com/content/77827a4c-1dfc-11e7-a454-ab04428977f9>. [Consulta Maig de 2018].
- [13] John Alberg, Zachary C. Lipton, Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals, <https://arxiv.org/abs/1711.04837>. [Consulta Maig de 2018].
- [14] Diario Expansion, Hedge Fund, <http://www.expansion.com/diccionario-economico/hedge-fund.html>. [Consulta Febrer de 2018].
- [15] EurekaHedge, Machine Learning Hedge Fund Index, <http://www.eurekahedge.com/Research/News/1614/Artificial-Intelligence-AI-Hedge-Fund-Index-Strategy-Profile>. [Consulta Maig de 2018].
- [16] Crypto-ML, Buy and Sell triggers generated by Machine Learning algorithms, <https://crypto-ml.com>. [Consulta Maig de 2018].
- [17] Laura Alessandretti, Machine Learning the Cryptocurrency Market, <https://arxiv.org/abs/1805.08550>. [Consulta Maig de 2018].

- 2018].
- [18] Investopedia, Efficient Market Hypothesis, <https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp>. [Consulta Febrer de 2018].
- [19] Wikipedia, ARIMA, https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average. 2018. [Consulta Febrer de 2018].
- [20] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, Deep Learning. <https://www.nature.com/articles/nature14539>. [Consulta Maig de 2018].
- [21] McCulloch, Warren; Walter Pitts, A Logical Calculus of Ideas Immanent in Nervous Activity. <https://link.springer.com/article/10.1007%2FBF02478259>. [Consulta Maig de 2018].
- [22] Rosenblatt, The perceptron: A probabilistic Model for Information Storage And Organization in the brain. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775>. [Consulta Maig de 2018].
- [23] Minsky, Marvin; Papert, Seymour. Perceptrons: An Introduction to Computational Geometry. https://books.google.es/books?id=Ow1OAQAIAAAJ&redir_esc=y. [Consulta Maig de 2018].
- [24] Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. https://books.google.es/books?id=z81XmgEACAAJ&redir_esc=y. [Consulta Maig de 2018].
- [25] Machine Learning Timeline milestones. https://en.wikipedia.org/wiki/Timeline_of_machine_learning. [Consulta Maig de 2018].
- [26] Sepp Hochreiter, Long Short-Term Memory, <http://www.bioinf.jku.at/publications/older/2604.pdf>. [Consulta Maig de 2018].
- [27] François Chollet, Deep Learning with Python. Sequence processing with ConvNets. Pp. 225-232. [Consulta Maig de 2018].
- [28] Keras, The Python Deep Learning library, <https://keras.io/>. [Consulta Maig de 2018].
- [29] TensorFlow, Open source Machine Learning frameworks, <https://www.tensorflow.org/>. [Consulta Maig de 2018].
- [30] DZone, Top 8 Deep Learning Frameworks, <https://dzone.com/articles/8-best-deep-learning-frameworks>. [Consulta Maig de 2018].
- [31] Hyperas, Hyperparameter optimization, <https://github.com/maxpumperla/hyperas>. [Consulta Maig de 2018].
- [32] Numpy, Scientific computing Package, <http://www.numpy.org/>. [Consulta Maig de 2018].
- [33] Pandas, Python Data Analysis Library, <https://pandas.pydata.org/>. [Consulta Maig de 2018].
- [34] Matplotlib, Python 2D plotting library, <https://matplotlib.org/>. [Consulta Maig de 2018].
- [35] Tkinter, Tkinter for Python 2 - GUI Toolkit, <https://docs.python.org/2/library/tkinter.html>. [Consulta Maig de 2018].
- [36] Kaggle, Bitcoin Historical Data, <https://www.kaggle.com/mczelinski/bitcoin-historical-data>. [Consulta Maig de 2018].
- [37] Blockchain, Blockchain stats, <https://blockchain.info/stats>. [Consulta Maig de 2018].
- [38] François Chollet, Deep Learning with Python, "The universal workflow of Machine Learning", pp. 111-116. [Consulta Maig de 2018].
- [39] Keras funcional API, multi input models, <https://keras.io/getting-started/functional-api-guide/>. [Consulta Maig de 2018].
- [40] Bitcoin difficulty, <https://www.quora.com/Who-adjusts-Bitcoin-mining-difficulty>. [Consulta Maig de 2018].
- [41] Bitcoin Hash Rate, <https://www.quora.com/What-does-Bitcoin-hash-rate-mean>. [Consulta Maig de 2018].
- [42] Investopedia, Momentum, <https://www.investopedia.com/terms/m/momentum.asp>. [Consulta Maig de 2018].
- [43] Udacity, Machine Learning for trading, "Incomplete Data", <https://eu.udacity.com/course/machine-learning-for-trading--ud501>. [Consulta Maig de 2018].
- [44] François Chollet, Deep Learning with python, "Preparing the data", pp. 86. [Consulta Maig de 2018].
- [45] Medium, MAE and RMSE - Which Metric is Better?, <https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d>. [Consulta Maig de 2018].
- [46] François Chollet, Deep Learning with python, "Fighting overfitting", pp. 104-110. [Consulta Maig de 2018].
- [47] Numerai, Encrypted Data for Efficient Markets, <https://numer.ai/homepage>. [Consulta Maig de 2018].