Previsione dei prezzi del titolo azionario JPM

Salvatore Latora

Luglio 2022

Abstract

La relazione mira a mostrare i risultati ottenuti da alcuni modelli implementati allo scopo di ottenere delle previsioni del prezzo di apertura del titolo azionario JPM, relativo alla banca d'affari americana JPMorgan Chase & Co. In particolare sono stati implementati due modelli di rete neurale, uno basato su CNN, mentre il secondo basato su un ecoder decoder CNN-LSTM. Si sono ottenute e confrontate le performance di entrambi i modelli, anche in riferimento alle performance ottenute da un modello ARIMA, utilizzando i dati dal 29-12-2014 al 31-12-2021 presi da Yahoo Finance.

1 Introduzione

I mercati finanziari rappresentano il luogo nel quale si realizzano le operazioni di contrattazione e scambio di strumenti finanziari di varia natura, a medio o lungo termine. Tra gli strumenti finanziari vi è l'*Azione*, o *Stock*: è un titolo finanziario rappresentativo di una quota della proprietà di una società per azioni. Prevedere l'andamento futuro di un titolo finanziario rappresenta una sfida. Coloro che supportano la teoria economica e *l'ipotesi di efficienza di mercato* sostengono nell'impossibilità di prevedere con accuratezza l'andamento futuro dei prezzi. Tuttavia altri studiosi hanno dimostrato che è possibile fare ciò con un alto grado di accuratezza utilizzando modelli di machine learning o deep learning.

2 Analisi Esplorativa

L'analisi esplorativa condotta è volta ad analizzare il comportamento della serie ed l'eventuale allontanamento dell'ipotesi che sta alla base del *modello classico* che si è utilizzato come benchmark, ovvero il modello ARIMA.

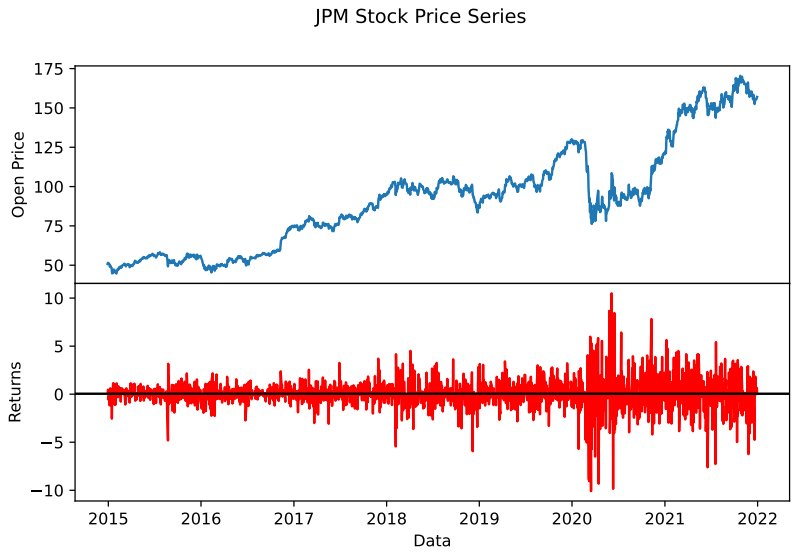

Figure 1: Serie dei prezzi di apertura del titolo JPM e relativi rendimenti

La figura 1 mostra come il trend del prezzo di apertura del titolo in esame, dall'inizio del periodo considerato sino alla fine, è fortemente crescente passando da un minimo di 44.29, toccato il 2 Febbraio 2015, a un massimo di 168.92 toccato il 25 Ottobbre 2021. Inoltre la serie

si muove attorno a una media di prezzo pari a 91.34. Da notare che intorno al primo trimestre del 2020 si registra un forte calo del prezzo, passando da un valore di 125 a un valore di circa 70. La serie si muove sempre attorno a questo valore fino alla fine del 2021, in cui inizia di nuovo un forte trend in crescita. Tale crollo del prezzo si verifica in concomitanza con l'evento pandemico di Covid-19, un momento di grande incertezza sia per gli investitori che per i mercati finanziari: all'inizio del periodo pandemico si registra un forte aumento della volatilità del titolo; come mostra la figura 1 si registra un forte cluster di volatilità e un'asimmetria positiva nei rendimenti proprio all'inizio dell'evento pandemico. L'incertezza sembra diminuire, dal 2021 in poi, in cui la volatilità del titolo diminuisce rispetto al periodo immediatamente precedente. E' utile notare che, in generale, la volatilità, che è un concetto attinente alla varianza, ha una tendenza generale ad aumentare durante tutto il periodo considerato, ciò significa che la serie potrebbe non essere stazionaria in varianza.
Al fine di verificare l'ipotesi di stazionarietà in media della serie, ipotesi che sta alla base del modello ARIMA, si è condotto l'ADF test. L'ipotesi nulla è la presenza di radice unitaria, contro l'ipotesi alternativa di assenza di radice unitaria. Il test statistico utilizza 5 ritardi sulle differenze prime, oltre a un drift e un trend; il valore della statistica test è pari a -0.4315 con un pvalue di 0.904, pertanto si accetta l'ipotesi nulla: la serie risulta essere non stazionaria in media e si comporta come un Random Walk. E' necessario lavorare sulle differenze prime al fine di ricondurci alla stazionarietà in media e poter applicare il modello ARIMA per fare previsione.
Per studiare la struttura di autocorrelazione della serie, si sono analizzati ACF e PACF della serie dei prezzi.

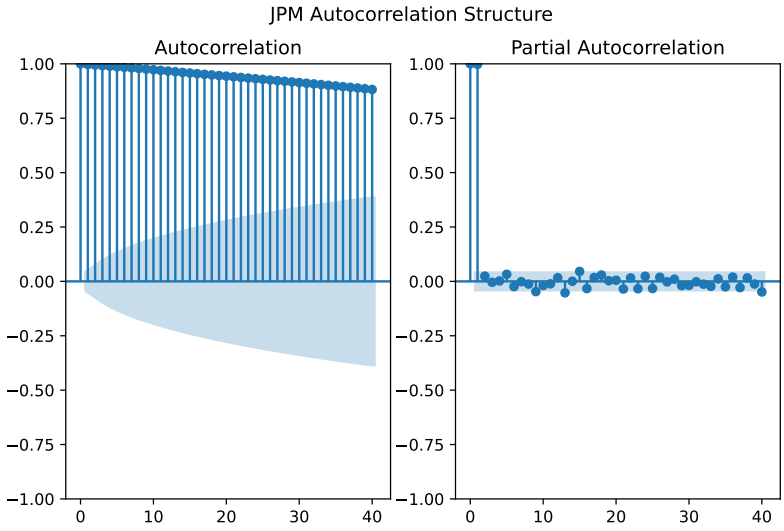

Figure 2: Funzione di autocorrelazione e autocorrelazione parziale della serie in esame

Dalla figura 2 possiamo notare come l'ACF mostra una forte persistenza di autocorrelazioni significative che decrescono molto lentamente, mentre il PACF mostra due forti autocorrelazioni parziali significative e vicino all'unità. Ciò conferma l'esistenza di una forte componente AR e tale comportamento ci porta a pensare che l'ordine del modello AR potrebbe essere pari a 2, tuttavia occorre utilizzare anche un ordine di integrazione per rendere la serie stazionaria sui livelli.

3 Modelli

In questa sezione vengono riportati i principali risultati del modelli implementati. In particolare sono stati messi a confronto tre modelli:

- ARIMA
- *CNN.*
- *encoder decoder CNN-LSTM.*

L'intera serie è stata suddivisa in due macro finestre temporali, utilizzate come training e test set per i modelli implementati. Come training set si utilizzano i prezzi di apertura che vanno dal 29-12-2014 al 04-11-2014, e corrisponde a circa il 72% delle osservazioni disponibili; Per la validazione del modello ARIMA si è utilizzata una procedura di *walk-forward validation* a 1 giorno, mentre per i modelli di rete si è utilizzata una procedura di *multi step walk-forward validation* a 5 o 10 giorni.

3.1 ARIMA

In particolare si sono confrontati due modelli, ARIMA(1,1,0) e ARIMA(2,1,0), scegliendo il primo in quanto tutte le misure di bontà di adattamento vanno in suo favore. Il modello fornisce una stima della componente autoregressiva pari a -0.0411 con un pvalue pari a 0.003, pertanto risulta leggermente significativa a un livello di significatività del 5%, anche se comunque risulta essere vicino a 0.05. Il test Ljung-Box condotto sui residui è non significativo e ciò significa che il modello è stato in grado di cogliere la dipendenza seriale presente nella serie, tuttavia il test Jarque-Bera per la verifica dell'ipotesi di normalità dei residui viene non significativo pertanto i residui del modello non sono White Noise gaussiani. In termini previsivi, l'errore del modello, misurato in termini di *Mean Squared Error* (MSE), risulta essere abbastanza contenuto e pari a 6.550. La figura 3 mostra la serie e le previsioni del prezzo fatte dal modello ARIMA(1,1,0).

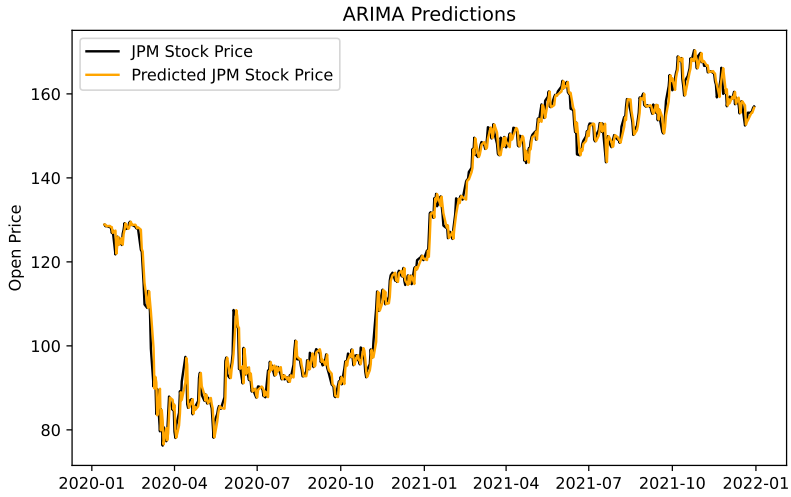

Figure 3: Prezzo di apertura nel test set e relativa previsione del modello ARIMA

3.2 CNN

Il primo modello di rete neurale utilizzato è basato sul primo modello descritto in [1]. L'architettura del modello è la seguente:

- Layer di input di dimensione (5,1)
- Layer convoluzionale con kernel di dimensione 3 e funzione di attivazione *relu*
- *Max Pooling* di dimensione 2
- Layer Flatten
- Layer Denso con funzione di attivazione *relu*
- Layer di output denso con funzione di attivazione *relu* e dimensione di uscita (5,1)

L'autore suggerisce l'utilizzo della funzione di attivazione *relu* in ogni layer, l'algoritmo *ADAM* come algoritmo di ottimizzazione, una dimensione del mini batch pari a 4 e 20 epoche. Come metrica di loss anche in questo caso si è utilizzato il MSE. Per la validazione del modello si è utilizzata una procedura di *multi step walk-forward validation*, usando una finestra temporale di cinque giorni e i successivi cinque giorni vengono predetti dalla rete e usati per la valutazione del valore di loss implementato.

Il modello inizialmente si è presentato abbastanza instabile sia sul training set che sul test set: l'inizializzazione casuale dei pesi del modello rendeva molto variabile le performance del

modello. Per evitare ciò, dopo vari tentativi e diverse possibili soluzioni, l'inizializzazione dei pesi della rete da un'uniforme di parametri compresi tra 0 e 0.02 ha reso la rete molto più stabile sul training set e con una variabilità nelle performance sul test set abbastanza contenuta. Anche a causa dell'instabilità del modello, si è provato a cambiare diversi parametri della rete allo scopo di capire l'influenza di essi sulle performance e cercare di trovare un modello che sia il più stabile possibile. In particolare, si sono messi a confronto due algoritmi di ottimizzazione (*ADAM* ed *RMSprop*), l'utilizzo di un learning rate inferiore e l'introduzione di un ulteriore livello convolutivo. Al fine di evitare l'overfitting si è implementato l'*early stopping*.
I tempi computazionali per tutti i modelli implementati sono al di sotto dei 10 secondi, abbondantemente trascurabili e pertanto non si riportano approfondimenti in tal senso.

Come mostra la figura 4, il modello con algoritmo ADAM raggiunge un valore molto piccolo dell'MSE sul training set già dalla seconda epoca; non vi è molta differenza tra l'MSE valutato sul training set e l'MSE valutato sul validation set pertanto non è presente overfitting. L'MSE valutato sul test set risulta essere pari a 30.83, mostrando poca variabilità fra un'esperimento e l'altro. Dalla figura 4 si nota come il prezzo di apertura venga leggermente sovrastimato soprattutto a partire da Luglio 2021.

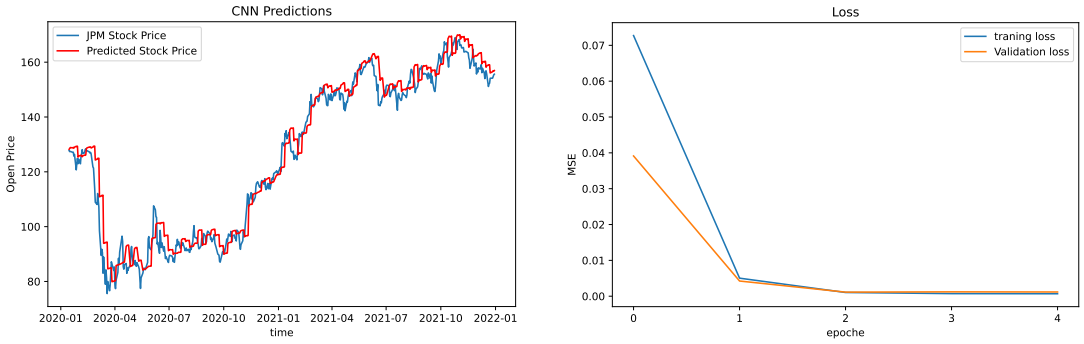

Figure 4: Previsioni e Loss sul training del modello iniziale

Utilizzando come algoritmo di ottimizzazione *RMSprop* le performance del modello su training, come anche mostra la figura 5, sono peggiori rispetto a quelle ottenute utilizzando come algoritmo ADAM: in questo caso, anche se l'MSE sul training è abbastanza contenuto, vi è evidenza di leggero overfitting in quanto l'MSE sul validation set inizia ad aumentare dalla seconda epoca. Anche l'MSE valutato sul test set risulta essere maggiore di quello ottenuto nel caso precedete, pari a 56.03. Dalla figura 5 si nota come il modello sottostima il prezzo di apertura, in misura maggiore a partire da Aprile 2021. L'algoritmo ADAM risulta quindi più adeguato per la rete.

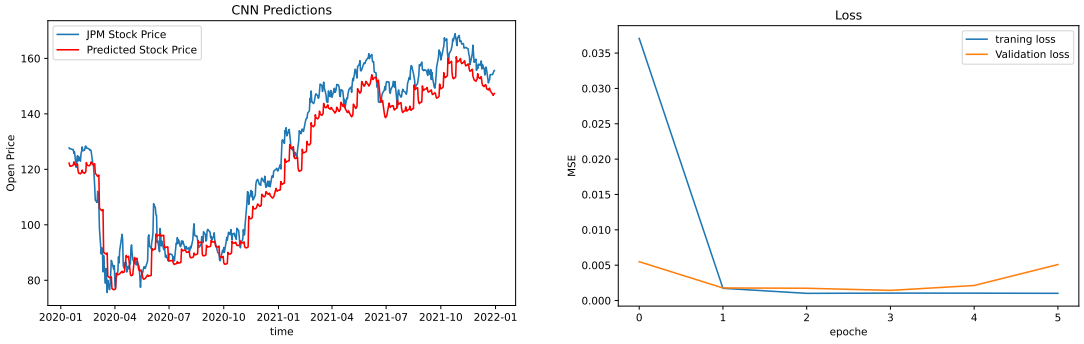

Figure 5: Previsioni e Loss sul training del modello iniziale con RMSprop

Portando il learning rate dell'algoritmo *Adam* a 0.0001, la figura 6 mostra che, il comportamento dell'MSE durante la fase di apprendimento è simile sia sul training che sul validation set a partire dalla 14 epoca; vengono utilizzate un totale di 20 epoche. Non vi è evidenza di overfitting. L'MSE sul test set è pari a 37.25, leggermente più alto al caso in cui si utilizza un learning rate di 0.001, per cui questo parametro non sembra influenzare molto il modello.

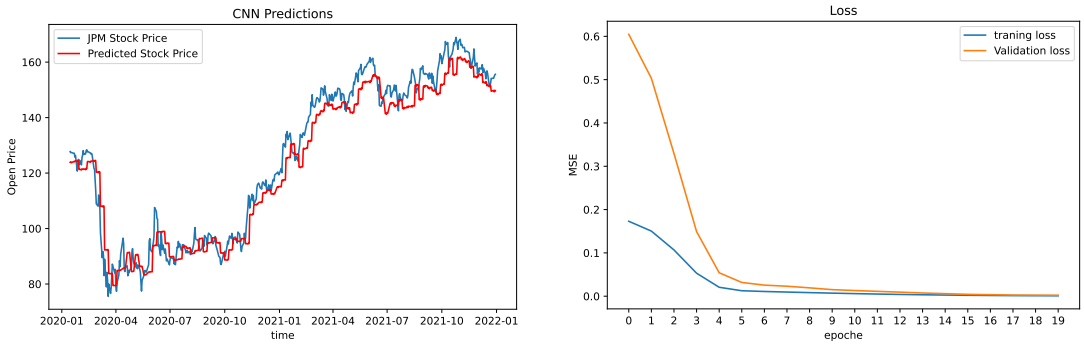

Figure 6: Previsioni e Loss sul training e validation set del modello iniziale con learning rate pari a 0.0001

Con l'introduzione di un ulteriore layer convolutivo, le performance nella fase di addestramento, come mostrato dlla figura 7, sono molto simili sia sul training set che sul validation set in quanto l'MSE risulta molto simile soprattutto a partire dalla seconda epoca, pertanto non è presente overfitting. L'MSE valutato sul test set risulta pari a 24.53, pertanto il modello presenta le migliori performance fra i modelli basati su CNN.

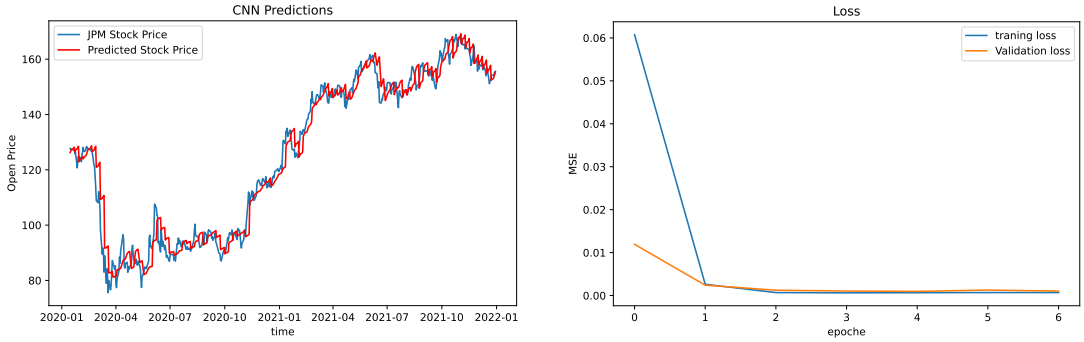

Figure 7: Previsioni e Loss sul training e validation set sul modello con un secondo layer convolutivo

Nonostante si siano inizializzati i pesi da una distribuzione uniforme in (0; 0.002), addestrando più volte le reti si nota ancora una variabilità nelle performance dei modelli, che comunque non è eccessiva.

3.3 Encoder Decoder CNN-LSTM

Questo modello è basato sul secondo modello basato su LSTM descritto in [1]. L'architettura del modello e di tipo *ecoder-decoder*. Anche in questo caso si è utilizzata la multi-step walk forward validation, utilizzando una finestra temporale di dieci giorni in cui addestrare il modello per fare previsione e valutare l'errore di previsione sui successivi cinque giorni. Anche in questo caso i tempi di calcolo sono al di sotto dei 10 secondi per tutti i modelli implementati, pertanto trascurabili. In particolare, il modulo di encoder è costituito da:

- Layer di input di dimensione (10,1);
- Layer convoluzionale unidimensionale con kernel di dimensione 3 e funzione di attivazione *relu*;
- Un secondo layer convoluzionale unidimensionale con kernel di dimensione 3 e funzione di attivazione *relu*;
- *Max Pooling* di dimensione 2
- Layer Flatten come livello di output.

Il modulo di decoder prende in input l'output del modulo di encoder, e l'architettura è la seguente:

- Layer *RepeatVector*, che funge da ponte tra il modulo di encoder e il modulo di decoder, con lo scopo di ripetere 5 volte l'input del modulo di encoder;
- Layer LSTM con funzione di attivazione *relu*
- *Max Pooling* di dimensione 2
- Layer Flatten
- Layer *TimeDistributed* di tipo denso con funzione di attivazione *relu*
- Layer di output *TimeDistributed* con funzione di attivazione *relu* e dimensione di uscita (5,1)

Anche in questo caso i tempi computazionali sono molto bassi, al di sotto dei 10 secondi, e trascurabili, pertanto non verranno approfonditi.
Inizialmente il modello è stato addestrato usando un numero massimo di epoche pari a 20, una dimensione del mini batch pari a 16, l'algoritmo di ottimizzazione *Adam* e funzione di attivazione *relu* per ogni layer, come suggerito in [1]. Anche in questo caso, per evitare l'overfitting, si è implementato l'early stopping. Come nei casi precedenti, dopo aver più volte addestrato le reti, si è notato una certa variabilità nelle performance sia sul training e validation set sia sul test set. Per tale motivo, l'inizializzazione dei pesi in ogni layer viene fatta da una distribuzione uniforme in (0,0.02).

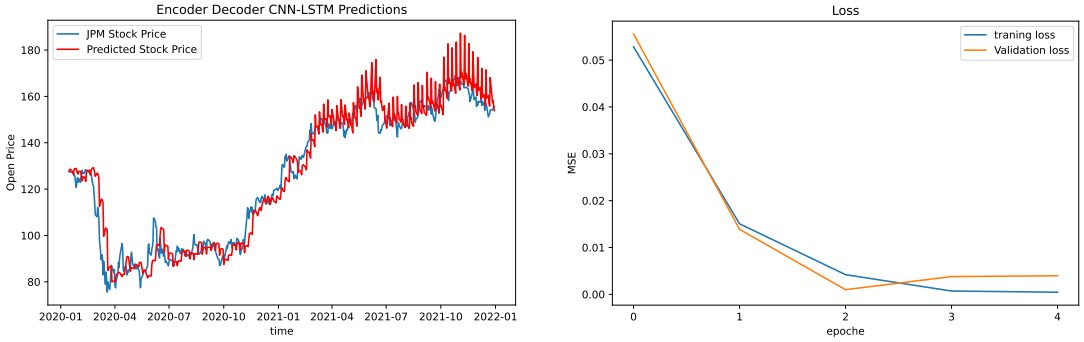

Figure 8: Previsioni e Loss sul training e validation set sul modello LSTM con Adam

Utilizzando come algoritmo di ottimizzazione *ADAM*, il modello performa bene sul training durante la fase di addestramento, raggiungendo un MSE prossimo a zero, ma leggermente peggio sul validation set, come mostrato dalla figura 8, indice di un possibile overfitting del modello. Sul test set L'MSE risulta essere pari a 54.65, con una leggera variabilità tra un esperimento e l'altro; Le previsioni fornite dal modello a partire da Aprile 2021 hanno una grossa variabilità iniziando ad oscillare molto e sovrastimano il prezzo di apertura.

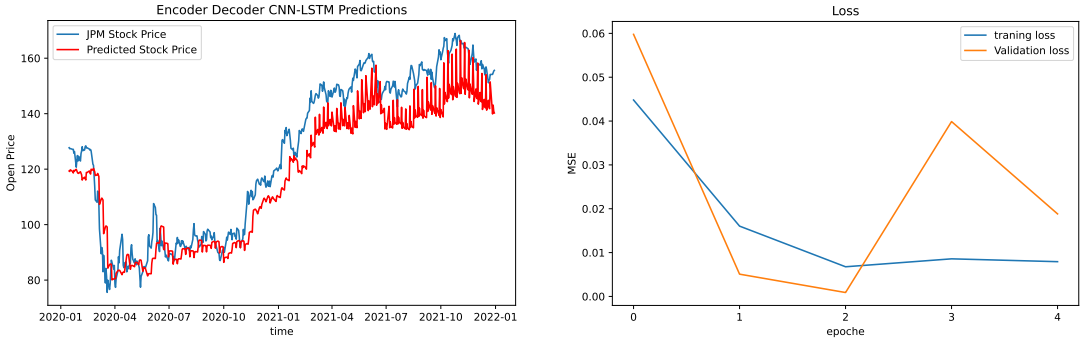

Figure 9: Previsioni e Loss sul training e validation set sul modello LSTM con RMSprop

Utilizzando come algoritmo di ottimizzazione *RMSprop* Le performance del modello in fase di addestramento peggiorano: la figura 9 mostra che vi è una grande differenza tra l'MSE sul training e l'MSE sul validation set soprattutto nelle ultime due epoche, mostrando performance peggiori sul validation set rispetto al training e quindi overfitting. L'MSE sul test set è pari

a 122.20, molto più grande di quello ottenuto con l'algoritmo *ADAM*, ed inoltre, le previsioni assumono una certa variabilità a partire da Aprile 2021 e sottostimano il prezzo di apertura soprattutto nell'ultimo periodo. L'utilizzo di questo algoritmo non porta nessun guadagno in termini di performance.

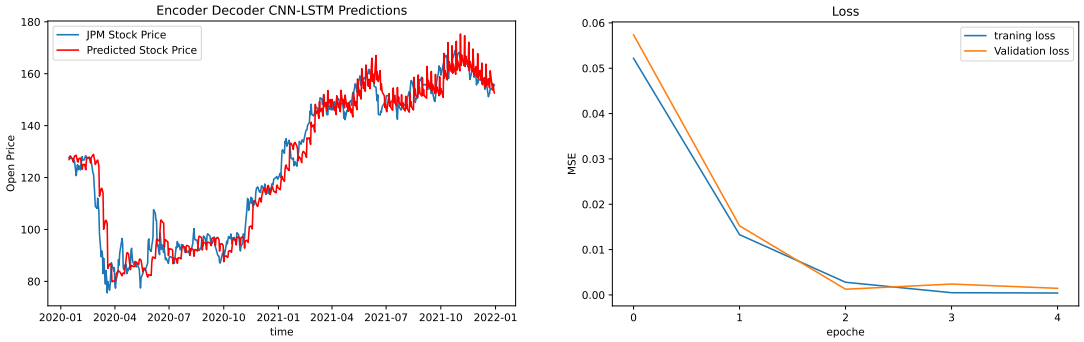

Figure 10: Previsioni e Loss sul training e validation set sul modello con funzione tanh per il layer LSTM

L'utilizzo di una funzione di attivazione *tanh* per il layer LSTM, non migliora in maniera determinante le performance sul test set del modello: l'MSE è pari a 43.02, e la figura 10 mostra come le previsioni assumono una certa variabilità a partire da Aprile 2021, che comunque sembrerebbe essere inferiore rispetto ai casi precedenti, per cui sarebbe opportuno utilizzare la funzione di attivazione *tanh*. L'MSE sia sul training che sul validation risultano essere molto simili durante la fase di apprendimento, pertanto non si ha overfitting.

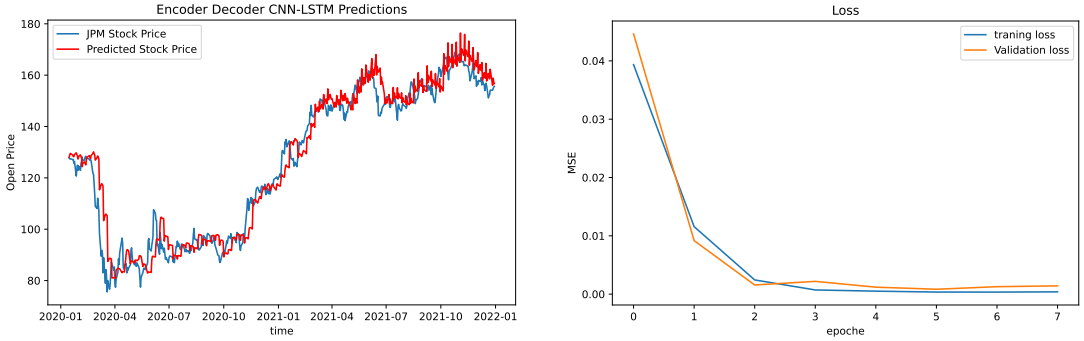

Figure 11: Previsioni e Loss sul training e validation set sul modello con un ulteriore layer LSTM e TimeDistributed

Si è provato ad aggiungere un secondo layer LSTM con funzione di attivazione *tanh* e un terzo layer *TimeDistributed* di tipo denso con funzione di attivazione *relu*, facendoli lavorare in parallelo a quelli già presenti nel modello, con lo scopo di aumentare la profondità della rete per migliorare le performance sul training. La figura 11 mostra che l'MSE sul training risulta essere simile a quello valutato sul validation set durante tutta la fase di addestramento, pertanto non vi è overfitting. La variabilità delle previsioni risulta inferiore soprattutto nell'ultimo periodo, rispetto ai casi precedenti. L'MSE sul test set risulta pari a 47.55, più alto rispetto alla rete meno profonda.

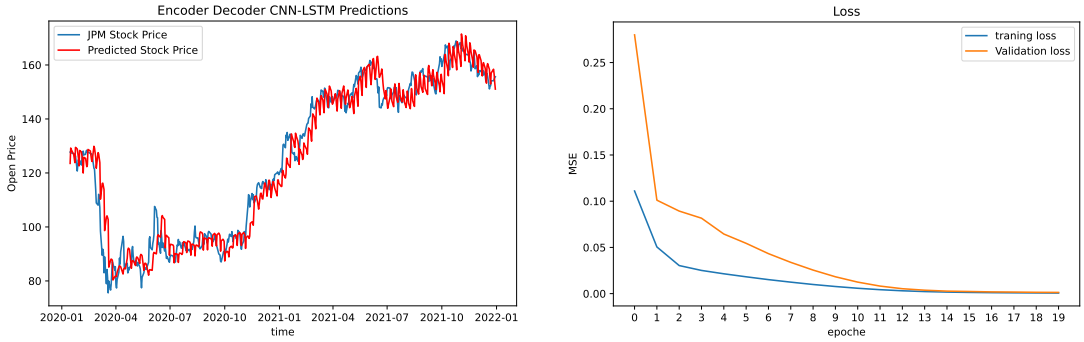

Figure 12: Previsioni e Loss sul training e validation set sul modello con un learning rate di 0.0001

Infine, abbassando il learning rate dell'algoritmo *Adam* a 0.0001, si nota, dalla figura 12 che l'MSE sul validation set diminuisce gradualmente sino ad arrivare ad un valore simile a quello calcolato sul training nelle ultime epoche, pertanto non c'è overfitting; Le previsioni sono abbastanza simili a quelle ottenute precedentemente, utilizzando un learning rate di 0.001. L'MSE calcolato sul test set risulta pari a 42.66, simile al caso in cui si utilizza un learning rate di 0.001.

Anche in questo caso occorre precisare che, dopo aver effettuato l'addestramento delle reti più volte, nonostante si siano inizializzati i pesi da una distribuzione uniforme in (0; 0.02) è ancora presente una minima variabilità nelle performance dei modelli.

4 Conclusioni

In conclusione, i modelli di rete si sono dimostrati poco adeguati a fornire delle previsioni accurate dei prezzi di apertura del titolo JPM, dimostrandosi poco stabili e con un errore di previsione abbastanza alto, cosa che in realtà fa il modello ARIMA con un errore di previsioni abbastanza contenuto. Il miglior modello di rete risulta essere il modello CNN con due livelli convolutivi, in quanto ha fatto registrare l'MSE sul test set più basso rispetto alle altre reti, in più esperimenti, ma in ogni caso più grande rispetto a quello ottenuto con l'ARIMA con una differenza di circa 18 punti. Inoltre le previsioni fornite sono meno variabili di quelle ottenute con la seconda tipologia di rete implementata, soprattutto nell'ultimo periodo considerato. In realtà, come ha messo anche in evidenza la letteratura, fare previsioni su serie finanziarie risulta particolarmente ostico, sia per la natura stessa della serie che ha il tipico comportamento di un Random Walk, sia per il fatto che vi sono tanti altri fattori da tenere in considerazione. Alcuni di essi, come ad esempio l'estrapolazione di notizie da fonti di informazione quali Twitter o Bloomberg, e la presa in considerazione di altre serie cointegrate alla serie in esame, che aiutano a prevedere il comportamento di essa, possono essere implementate nello sviluppo futuro di modelli di rete per ottenere dei modelli più performanti da un punto di vista previsivo. Inoltre sarebbe anche interessante provare ad utilizzare i modelli di rete per la previsione della volatilità del titolo finanziario.

References

[1] Zihao Gao. Stock price prediction with arima and deep learning models. In *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pages 61–68, 2021.

[2] Sidra Mehtab and Jaydip Sen. Stock price prediction using cnn and lstm-based deep learning models. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pages 447–453, 2020.