IEOR Deep Learning

Final Project

Predicting Contributions and Distributions for Private Equity Funds Using LSTM

Chloe Y. Moon
Christopher D'Onofrio
Mateo Gomez

Supervised by
Prof. Ali Hirsa, Department of Industrial Engineering and Operations Research, Columbia University
&
Satyan Malhotra, Managing Partner, FlexStone Partners

December 2019

Executive Summary

As an extension of Buchner, Kaserer and Wagner's paper on stochastic modeling of private equity, this research aims to build a functioning LSTM model to predict distribution and contribution cash flows of three types of Private Equity funds, specifically Buyout Funds (all, large, mid, and small-sized), Venture Capital Funds, and Real Estate Funds. First, quarterly historical benchmark data from 2014 to 2019 are obtained from Preqin and interpolated using stochastic interpolation. Next, we built an LSTM model using Pytorch and tuned hyperparameters for each fund type and size, experimenting with different architectures and hyperparameters. This report includes both the numerical and graphical results of the model for each fund type and identifies the best performing model as well as a comparison with the original MIT model and a case study. Chloe concentrated on data collection, research in the area on previous work, and adding functionalities to the LSTM models; Christopher focused on the development of the code for data preprocessing, construction of the LSTM models and implementation of the MIT paper's model; Mateo on editing and implementing features of the code for the LSTM model and writing the vast majority of this report.

Table of Contents:

Abstract

The purpose of this research is to use LSTM (Long-Short Term Memory) Model to predict distribution and contribution cash flows of different fund types that are common in the Private Equity industry, in particular Leveraged Buyouts Funds of different sizes, Venture Capital (VC) Funds, and Real Estate Funds. The idea is to take the time series data of contributions and distributions of the benchmarks of the funds as a proxy in each quarter of the fund's life, starting on the vintage year. We use LSTM model to predict the cash flows of such variables for $t+n$, where n can be any time within a year. Historical benchmark data for each fund type is used to train and test this model, but the model will be adapted to individual funds to make these predictions, as described in the Case Study section.

Introduction

Private Equity is one industry that has been growing in the past years in the financial sector as they give alternatives investment for people who would like to take illiquidity risk with higher rewards in return. These firms invest in companies that are not listed in the public market to own a share of such company to sell it later to other investors or to the public. This model is recognized as a very profitable one on the industry. However, it has its limits when one tries to make predictions for the different factors that might affect the cash flows relevant to the returns on investment, especially due to illiquid nature of assets and lack of accessible data. This paper will introduce a novel approach in modeling two types of cash flows that investors and Private Equity firms supervise during the lifetime of every fund. The allocations of these funds not only depend on the level of commitment of the investors, but also in the timing of drawdowns and distributions (Takahashi, 2).

Institutions invest in alternative assets through “commingled limited partnership funds” , where funds are raised every few years on a blind pool basis by general partners who actively invest, manage, and harvest portfolio investments (Takahashi, 2). In the beginning, investors commit capital that gets drawn over several years by the general partner, but the certain timing of the drawdowns poses a challenge to modeling the dynamics. Moreover, unknowable changes of valuations and liquidity of the assets makes it even more difficult to accurately predict the future value of partnership interests. Because of these reasons, despite the untapped potential, only little research has been done on predicting cash flow dynamics of private equity investments (Buchner, 2010, 3).

Previous Work

Several methods have been developed in order to predict the distributions, contributions and net asset value of such funds. In 1998, an approach was made to model illiquid alternative asset funds by Takahashi and Alexander from Yale University (2001). They studied the behavior of these three variables an adapted a time series model to project future cash flows for funds in alternative asset classes, including our fund types of interest. The model uses six inputs: (1) Rate of contribution (RC), (2) Capital commitment (CC), (3) Life of the fund (years), (4) Factor describing changes in the rate of distribution over time (B), (5) Annual growth rate (G), (6) Yield (%). With these inputs, they came up with the following relations to calculate the desired cash flows and performance, which is the so-called Yale Model:

$$\text{Contribution } C_t = RC_t * (CC - PIC_t) \text{ where } PIC_t = \sum_{0}^{t-1} C_t$$
$$\text{Distribution } D_t = RD * [NAV_{t-1} * (1+G)] \text{ where } RD = Max[Y,\ (t/L)^B]$$
$$\text{Net Asset Value } NAV_t = [NAV_{t-1} * (1+G)] + C_t - D_t$$

However, the Yale model has its own limitations; the model is based on data from only 33 venture capital funds in the late 1980s, and it heavily relies on assumptions regarding growth rate, annual contribution rate, yield, and underlying net returns that, only when appropriately made, provide stable and accurate results.

Another approach by Fitch Research in the Southwest Capital Funding Report uses historical data of other funds. The idea is to use historical data to construct performance scenarios by matching specific characteristics of the portfolio funds, such as the type and age of the funds, additionally taking different economic cycles into account. However, prospective economic and investment environments do not necessarily resemble any historical period perfectly, as proven by the unprecedented rates of the capital drawdowns in the late 1990s venture capital funds (Takahashi, 3). Therefore, such an approach solely based on those factors is extremely limiting.

An alternative approach by Buchner, Kaserer, and Wagner uses continuous time stochastic model for cash flow forecasting based on observable cash flow data. It allows to model capital drawdowns and capital distributions as separate diffusion processes, unlike the Yale Model in which contributions and distributions are dependent, and additionally perform risk analysis. Although it is a more empirical and applicable approach than the Yale Model, it also relies on several assumptions; for instance, it assumes that capital distributions follow lognormal distributions and that a fund's capital drawdowns and distributions are only based on the observable economic variables they chose, disregarding the potential impact of the unobservable

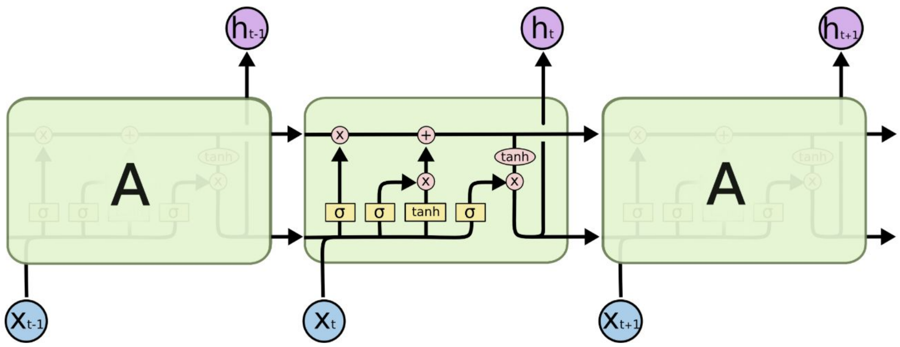
(Buchner, 2010, 15). This stochastic model is what our model is loosely based off of, as we also predict capital drawdowns and capital distributions separately as independent components over a fund's lifetime. We referenced two of Buchner's papers, *Modeling the Cash Flow Dynamics of Private Equity Funds: Theory and Empirical Evidence* and *Stochastic modeling of private equity: an equilibrium based approach to fund valuation* to build the models. Both models introduce two independent stages for capital drawdowns and contributions and apply mean-reverting square-root process to model the drawdown rate (Buchner, 2006, 2; Buchner, 2010, 5).

This paper uses pieces from the previous approaches, but deviates from the essence of both. We start by the assumption that the previous cash flows on the distributions and contributions of the Private Equity funds embed the dynamics that are needed in order to project the future cash flows. In this new approach, we are using the data of the type of funds that we want to study, train the data using a LSTM (Long Short-Term Memory) neural network with one layer and different architectures, to predict the logarithmic returns in the contribution and distribution cash flows of the fund under study.

Long Short-Term Memory Neural Networks (LSTM)

Traditional neural networks do not have the capability to efficiently store information over extended time intervals, due to "insufficient, decaying error backflow" (Hochreiter, 1). This is a key factor that we need to encompass for the purpose of predicting the contributions and distributions cash flows. We are assuming that we can predict the cash flows given that all the information is embedded in the observed values. To capture this assumption, we used Long Short-Term Memory Neural Networks. LSTM neural networks are a special kind of recurrent neural networks (RNN) which, in contrast to RNN, are able to remember short term and long term memory through memory cells. LSTM is local in space and time, and it is computationally efficient and fast as the unnecessary gradient is truncated (Hochreiter, 1). LSTMs are explicitly designed to avoid the long-term dependency problem that is caused by the RNN, and because of this property are widely used in predicting time series problems, hence we decided to utilize this model in this research.

Unlike conventional neural networks whose error signals "flowing backwards in time," resulting in either vanishing or exploding gradients, LSTM enforces constant error flow through internal states of special, self-connected units (Hochreiter, 2). LSTM utilizes memory cells, gate units, and its chain-like structure (similar to RNN) to do so; each repeating module there are four neural network layers that interact with each other.



The repeating module in an LSTM contains four interacting layers.

The upper horizontal line is called the cell state. It runs with linear interactions and thus this information is mostly unchanged. The vertical lines connecting to it are called gates, which are in charge of adding or removing information from the cell state. This gates are composed by a sigmoid neural net layer and a pointwise multiplication operation. This activation function will give the probability of the points to be retained or to be removed from the cell state. The leftmost layer is called the forget layer, which is the one that decides which information is "thrown away".

The two layers in the middle control which information will be stored in the cell state. It has two activation functions, one is sigmoid that decides which values will be updated from the new obtained values and a tanh activation function that is in charge to decide which values will we added to the cell state. Finally we have the output layer, which is based on the cell state but with an extra filter. A sigmoid layer decides the data that will be in the output and then it goes through a tanh layer to obtain better predictions (Olah).

In the case of predicting cash flows, the cell state is saving some of the cash flows that were observed in the previous quarters, the forget layer decides which of those cash flows to forget, the middle layers train the data and pass it through the cell state (adding new data or updating previous data that was stored) and the rightmost (output layer) performs the double filter to obtain the predictions.

Data

One of the bigger problems in Private Equity is the availability of data. It is challenging to obtain large volumes of high-quality data for the funds and it becomes one of the key barriers for a Deep Learning method in making accurate predictions. It is well known that Deep Learning methods are data hungry; in order to train a model, one should have a lot of data points and that have a high signal-noise ratio.

Furthermore, the fund types differ a lot from their structure, properties and cash flows, which makes it often useless to merge the data and utilize more data points. For example, a Real Estate fund has completely different behavior than a Leveraged Buyout or a Venture Capital fund. It is also important to consider the size of the fund; for example, a mid-size buyout fund will behave differently compared to a small or large buyout fund. Given this difference, we decided to experiment with Real Estates funds, Venture Capital Funds and small, mid, large and "all" size Buyout funds. The next section will explain how we tackle the lack of data and the type of data we used for this research.

Data Collection

As briefly mentioned above, data collection for Private Equity funds can be extremely expensive and challenging. We decided to stick to one of the databases that Columbia University granted us access to: Preqin. This database specializes in collecting cash flow data from different Private Equity funds and then sell it to companies or to the public. Initially, the ideal scenario was to build a model that was tuned using the cash flows of individual funds, which at the end will be what the model will be used for. Unfortunately, the obtained data was extremely inconsistent and we were not able to create a database of enough funds with sufficient cash flows to use in our model. We decided to use Preqin's historical benchmarks for each fund type (and size for buyout funds) as a proxy to estimate the cash flows for contribution and distribution in the aforementioned funds.

The data collection process was manually done, as there was no other way to gather the data more conveniently. We focused on funds with vintage year of 2014; as discussed in the meetings, funds older than this will get a very stable cash flow towards the end life of the fund, and funds younger than this will not have enough cash flow for contributions or distributions to train the model. From Preqin, we were able to obtain three different metrics for the fund's benchmark in each quarter: Called-Up (%), Distributed (%) and IRR. The Call-Up (contribution) cash flows tend to increase from the vintage year of the fund (when the first investment is made) and then stay constant after all the capital is invested. The distribution cash flows stay in low values (close to zero) during the early life of the fund and then they start increasing as the returns start to be distributed to the limited partners. The following table is an example of how the data for the cash flows of a mid-size Buyout benchmark with 2014 as vintage year looks like.

2014 Vintage-Mid Buyout						
	Median			Mean		
Time	Called Up	Distributed	IRR	Called Up-Mean	Distributed-Mean	IRR- Mean
Mar-14				3.2	0.1	
Jun-14				7.3	0.1	
Sep-14	10	0		11.1	0	
Dec-14	13	0		13	0.3	
Mar-15	15.8	0		17.1	1.5	
Jun-15	23.7	0		22.6	3.1	
Sep-15	28.1	0.2		25.9	4.5	
Dec-15	33.6	0.3		34	4.4	
Mar-16	32	0.8		34.7	7.1	
Jun-16	38.2	0.7		40.6	8.9	
Sep-16	47.6	6.7		46.9	14.4	
Dec-16	44.3	8		49.2	15.7	
Mar-17	56	8	10.4	55.1	19.5	14.7
Jun-17	59	10.6	13.3	56.3	19.1	9.2
Sep-17	67.2	12	18.6	63.6	23.3	17.5
Dec-17	71.6	17.3	18	66.6	30.2	21
Mar-18	78.7	20.1	16.3	71.5	31.5	19.5
Jun-18	81.3	16.1	20.2	75.5	33	22.5

Sep-18	83.2	22.2	15.3	80.4	31	16.8
Dec-18	83.1	36.4	15.3	81.8	47.4	18.9
Mar-19	89.6	34.9	14	84.9	44.5	16.7
Jun-19	80.8	26.9	15.4	79.7	51.5	18.3

Data Preparation

We started by performing a linear interpolation from quarter to quarter of the data points. Seeing that this interpolation did not have noise as it will be expected by the cash flows, we added a layer of complexity to increase the amount of data that we are using in the training and testing sets and have a time series that will mimic the behaviour more accurately. We added a noise term to the interpolated data (in addition to the already used linear term), sampling from zero mean normal distributions with different variances for the training set and the testing set to imitate a random walk with drift. The goal of this was to make the trained model more robust for prediction.

After the quarterly data was stochastically interpolated at a daily frequency, we created rolling windows depending on how many values we wanted to use to predict the cash flows. Our model is flexible enough to change the number of days that we are using to predict the cash flows on a given day. For purposes of this paper, we always predicted the cash flows after a year, and we used the data in the previous 2 and 3 quarters before making the prediction. For example, if today is June 30 2019, we used the data from the past 180 or 270 days to predict the contribution and distribution cash flows for June 30, 2020. This is completely flexible and it should be used for different purposes depending on the objectives of the General Partners. Finally, we split the data into 70% to the training set and 30% for the testing set. We omitted the validation set since we did not have a robust database for the cash flows that permitted us to further split the data.

Model Architecture and Parameters

Architecture

The architecture of the model is one of the most experimental parts while building the system. It finalized using a trial and error mechanism by experimenting with the architecture and the parameters in order to provide a good fit for the data. In terms of activation functions, Pytorch does not have the flexibility to change them in an LSTM model, so we decided to keep the activation functions mentioned in the LSTM section (sigmoid to control the memory and tanh to train and update the predictions, and linear functions in the state cell).

In terms of the fundamental architecture of the model we found that a single layer LSTM outperformed higher layer models, and we selected an Adam optimizer to train the models. The optimizer was training the data well (training MSE<0.0002) when we increased the number of epochs so we did not have any reason to change it. We experimented with different number of layers (2 and 3 layers), but the model running time increased significantly without achieving better results than a single layer. This is a factor that can be examined further when adding other predictor variables for the cash flows, going from a 1d LSTM to 2d or 3d LSTM, a possible expansion of the model (see below section on Further Improvements).

At first, we were experiencing periodic results and a vanishing gradient problem. This was observed in the training and testing set. For the first issue, we observed that the training set was predicted with periodicity of the data points. For example, if we used the last two quarters to predict the cash flows in a year, we would observe that the model create periodic cash flows every two quarters. To fix this, we trained the model by using batches of size 100 and also tested the data using batches of size 100 instead of passing the whole data at once (for training and testing sets).

To address the second issue, we used batch normalization, a common technique to solve vanishing gradient problems. We were observing that the model was not training the data and that the output of both training and testing sets were a constant line or a line with a minimum observable noise. After using batch normalization on the predictor cash flow data points the model started to train the data, even to create a perfect overfit with a high number of epochs. In the testing set we started observing better results and how the predictions were mimicking the testing set but with significantly more noise and having an over prediction of the log returns of the cash flows (more on this in the Results section below).

Parameters

For the parameter tuning, we run the model several times and graphed the behavior of the loss function (MSE) for the training and testing set, the value of the loss function and how the training and testing predictions fit the data visually to determine a range of learning rates and epochs for the final results. We also experimented with with different number of cash flows to make the year prediction. We started using the prior 30,60,90,120,150,180,210,240,270 days of log returns on the cash flows. The behavior of the prediction was very noisy when we used a small amount of days and it got more and more accurate as we increased the days. This is expected as the LSTM is able to remember long and short term memory so the more data with respect to time the better predictions the model will have. For the purpose of this report we decided to use 2Q and 3Q (180 and 270 days) of prior data to predict the cash flow returns as the model did not behave as accurate with less amount of days and the predictions were notoriously noisy.

We also explored the number of epochs and the value of the learning rate. To do this, we run the LSTM model keeping everything else constant to see the behavior of the testing and training set. We decided to report the results with 100 epochs and 200 epochs and learning rates of 0.001 and 0.005 since they gave persistent results, the training set was evidently shaped and not overfitted. It is interesting to notice that the more epochs we added, the higher the noise for the testing set and the more accurate the model trained the data. We tried to find a balance between visually seeing the training prediction fitting the training data and avoiding overfitting on this data that caused meaningless results in the testing set.

Summary of Section

Architecture		Parameters	
Optimizer	Adam	**Epochs**	100 and 200
Activation Functions	TANH and SIGMOID	**Learning Rate**	0.001 and 0.005
Number of Layers	1	**Sequence Length**	180 and 270 days
Batch Size	100	**Interpolation Frequency**	Daily
Other Traits	Batch Normalization		

Results

After tuning the model we went ahead and tested it for six (6) different fund types : All Buyouts, Large Buyouts, Mid-size Buyouts, Small Buyouts, Real Estate and Venture Capital. The contribution and distribution log returns of the cash flows were tested under two scenarios -- using two and three quarters of data to predict a year ahead of the log returns of the cash flows. Not surprisingly, we found better predictions for the contribution cash flows in comparison to the distribution ones in the six fund types. This is due to the number of data points we have on each cash flow and the behaviour of it. Also, the varying nature of distributions with different stages of a fund's life makes the prediction of distributions from historical benchmark data intrinsically harder than that of contributions (Takahashi, 6).

In general, we start seeing changes in the contribution cash flows at the beginning of the vintage year and then see movements through the life of the fund until they stabilized after reaching 90% of the contributions. After interpolating, we have enough data points to train the model under different changes on the cash flows which result in better predictions on the testing set. For the distribution of cash flows, we start observing significant changes after half way of the data points recorded. This leaves the model with less data points and less changes on the cash flows that will impact the training and prediction of our model. For example, the model is able to train different movements on the contribution training cash flows and, when it faces the testing set, it is able to identify some patterns and find the trend, hence providing interesting results. On the other hand, since we do not have enough data points for the distribution cash flows, the model is trained by small changes on the cash flows and not seeing many significant changes on the trend of the cash flows. As the model is just trained to predict increasing cash flows for a period of time, it is completely impossible for it to predict fluctuation on the cash flows on the testing set.

The following tables describe numerically the results of the six funds under the conditions that were mentioned previously. We observed that the Real Estate contribution cash have the best results among all the experiments.

MSE Results

Buyouts

All Buyout Funds					
1 layer, 3 quarter-Distributed			**1 layer, 3 quarter-Called Up**		
Learning Rate	***Epochs***	***Test MSE***	***Learning Rate***	***Epochs***	***Test MSE***
0.005	100	2.018555	0.005	100	0.008121
0.001	100	1.571667	0.001	100	0.008543
0.005	200	2.079997	0.005	200	0.009440
0.001	200	2.455000	0.001	200	0.007196
1 layer, 2 quarter-Distributed			**1 layer, 2 quarter-Called Up**		
Learning Rate	***Epochs***	***Test MSE***	***Learning Rate***	***Epochs***	***Test MSE***
0.005	100	0.621540	0.005	100	0.006671
0.001	100	0.632964	0.001	100	0.006810
0.005	200	0.175520	0.005	200	0.008100
0.001	200	0.372755	0.001	200	0.009110

Large Buyout Funds					
1 layer, 3 quarter-Distributed			**1 layer, 3 quarter-Called Up**		
Learning Rate	***Epochs***	***Test MSE***	***Learning Rate***	***Epochs***	***Test MSE***
0.005	100	0.075587	0.005	100	0.013211
0.001	100	0.065334	0.001	100	0.014894
0.005	200	0.023412	0.005	200	0.016357
0.001	200	0.061082	0.001	200	0.015645
1 layer, 2 quarter-Distributed			**1 layer, 2 quarter-Called Up**		
Learning Rate	***Epochs***	***Test MSE***	***Learning Rate***	***Epochs***	***Test MSE***
0.005	100	0.052140	0.005	100	0.054644
0.001	100	0.129150	0.001	100	0.045576
0.005	200	0.059990	0.005	200	0.037263
0.001	200	0.091008	0.001	200	0.032983

<table>
<tr><th colspan="6">Mid Buyout Funds</th></tr>
<tr><th colspan="3">1 layer, 3 quarter-Distributed</th><th colspan="3">1 layer, 3 quarter-Called Up</th></tr>
<tr><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th></tr>
<tr><td>0.005</td><td>100</td><td>0.105989</td><td>0.005</td><td>100</td><td>0.011944</td></tr>
<tr><td>0.001</td><td>100</td><td>0.090443</td><td>0.001</td><td>100</td><td>0.009155</td></tr>
<tr><td>0.005</td><td>200</td><td>0.118446</td><td>0.005</td><td>200</td><td>0.012495</td></tr>
<tr><td>0.001</td><td>200</td><td>0.102876</td><td>0.001</td><td>200</td><td>0.011832</td></tr>
<tr><th colspan="3">1 layer, 2 quarter-Distributed</th><th colspan="3">1 layer, 2 quarter-Called Up</th></tr>
<tr><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th></tr>
<tr><td>0.005</td><td>100</td><td>0.224291</td><td>0.005</td><td>100</td><td>0.013730</td></tr>
<tr><td>0.001</td><td>100</td><td>0.223624</td><td>0.001</td><td>100</td><td>0.011961</td></tr>
<tr><td>0.005</td><td>200</td><td>0.213327</td><td>0.005</td><td>200</td><td>0.012606</td></tr>
<tr><td>0.001</td><td>200</td><td>0.222025</td><td>0.001</td><td>200</td><td>0.008750</td></tr>
</table>

<table>
<tr><th colspan="6">Small Buyout Funds</th></tr>
<tr><th colspan="3">1 layer, 3 quarter-Distributed</th><th colspan="3">1 layer, 3 quarter-Called Up</th></tr>
<tr><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th></tr>
<tr><td>0.005</td><td>100</td><td>0.045780</td><td>0.005</td><td>100</td><td>0.006852</td></tr>
<tr><td>0.001</td><td>100</td><td>0.090560</td><td>0.001</td><td>100</td><td>0.007976</td></tr>
<tr><td>0.005</td><td>200</td><td>0.049810</td><td>0.005</td><td>200</td><td>0.005697</td></tr>
<tr><td>0.001</td><td>200</td><td>0.022330</td><td>0.001</td><td>200</td><td>0.007110</td></tr>
<tr><th colspan="3">1 layer, 2 quarter-Distributed</th><th colspan="3">1 layer, 2 quarter-Called Up</th></tr>
<tr><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th><th>Learning Rate</th><th>Epochs</th><th>Test MSE</th></tr>
<tr><td>0.005</td><td>100</td><td>0.043731</td><td>0.005</td><td>100</td><td>0.010235</td></tr>
<tr><td>0.001</td><td>100</td><td>0.026597</td><td>0.001</td><td>100</td><td>0.008406</td></tr>
<tr><td>0.005</td><td>200</td><td>0.040921</td><td>0.005</td><td>200</td><td>0.009426</td></tr>
<tr><td>0.001</td><td>200</td><td>0.052481</td><td>0.001</td><td>200</td><td>0.007947</td></tr>
</table>

Real Estate

Real Estate Funds					
1 layer, 3 quarter-Distributed			**1 layer, 3 quarter-Called Up**		
Learning Rate	*Epochs*	*Test MSE*	*Learning Rate*	*Epochs*	*Test MSE*
0.005	100	0.039960	0.005	100	0.0023928
0.001	100	0.023828	0.001	100	0.0054977
0.005	200	0.049543	0.005	200	0.0031352
0.001	200	0.032254	0.001	200	0.0044959
1 layer, 2 quarter-Distributed			**1 layer, 2 quarter-Called Up**		
Learning Rate	*Epochs*	*Test MSE*	*Learning Rate*	*Epochs*	*Test MSE*
0.005	100	0.041321	0.005	100	0.006911
0.001	100	0.028185	0.001	100	0.003819
0.005	200	0.040344	0.005	200	0.008707
0.001	200	0.040133	0.001	200	0.001706

Venture Capital

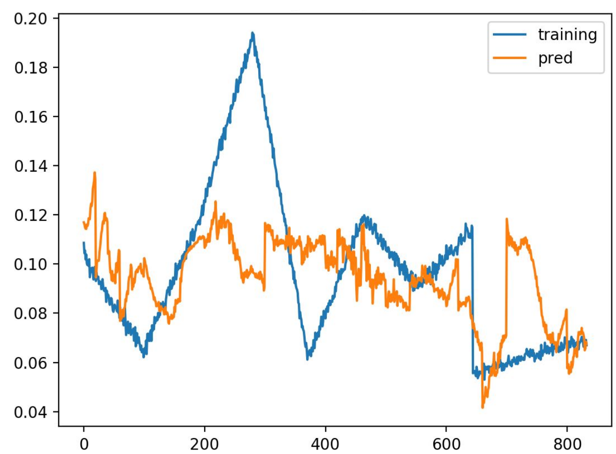
VC Funds					
1 layer, 3 quarter-Distributed			**1 layer, 3 quarter-Called Up**		
Learning Rate	*Epochs*	*Test MSE*	*Learning Rate*	*Epochs*	*Test MSE*
0.005	100	2.059170	0.005	100	0.006711
0.001	100	2.021330	0.001	100	0.006710
0.005	200	2.426530	0.005	200	0.006540
0.001	200	2.008450	0.001	200	0.009736
1 layer, 2 quarter-Distributed			**1 layer, 2 quarter-Called Up**		
Learning Rate	*Epochs*	*Test MSE*	*Learning Rate*	*Epochs*	*Test MSE*
0.005	100	1.851480	0.005	100	0.002298
0.001	100	1.934860	0.001	100	0.007330
0.005	200	1.820270	0.005	200	0.003469
0.001	200	1.981480	0.001	200	0.008957

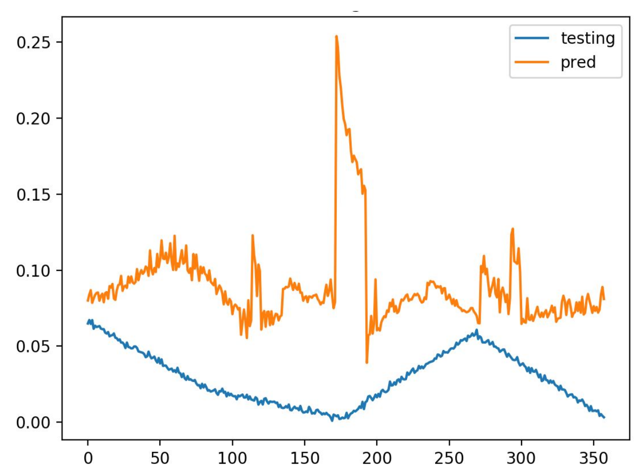
Visualizations

All Buyouts

Epochs: 100
Learning Rate: 0.005
Quarters used to predict: 2
Data points: Contributions

Train (left), Test (right) Epoch 100

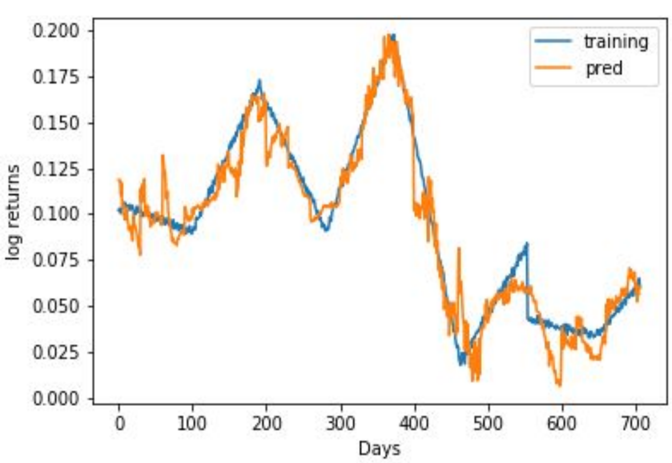


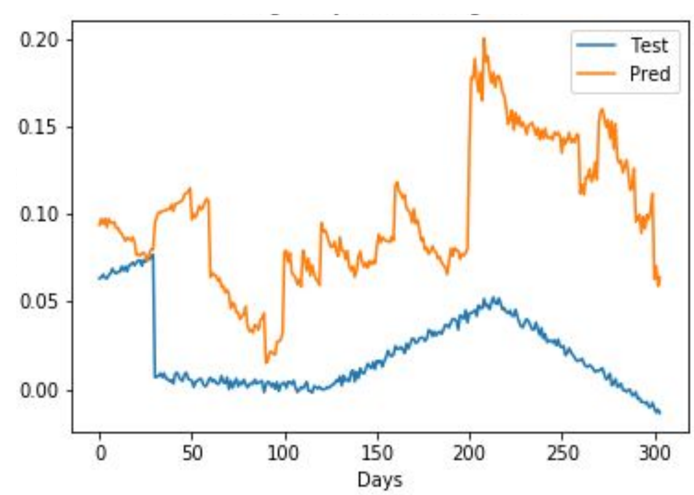


Small Buyouts

Epochs: 200
Learning Rate: 0.001
Quarters used to predict: 3
Data points: Contributions

Train (left), Test (right)



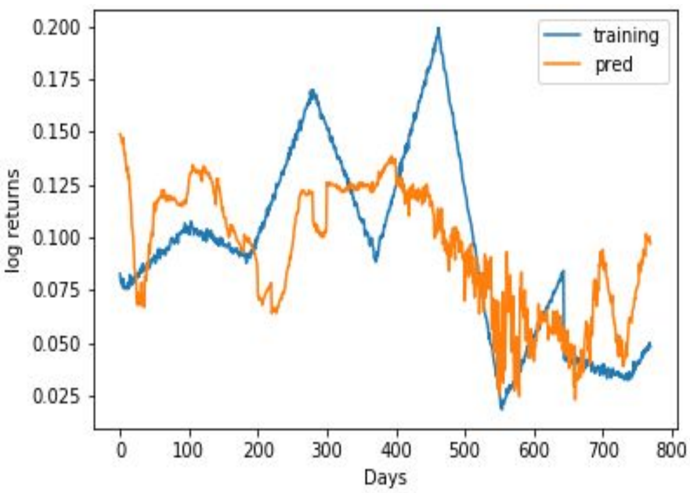


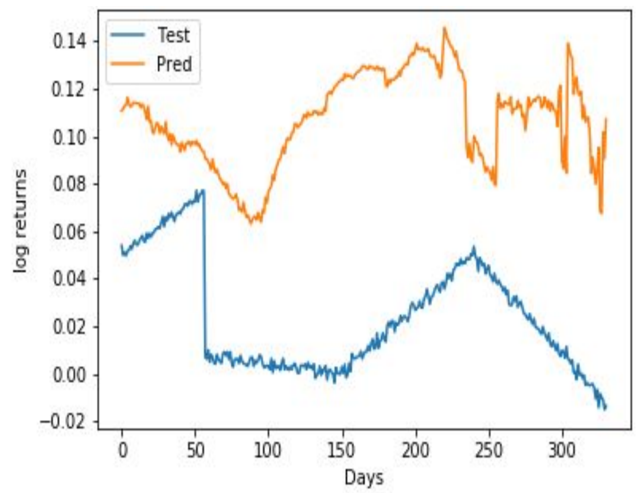
Epochs: 100

Learning Rate: 0.001

Quarters used to predict: 2

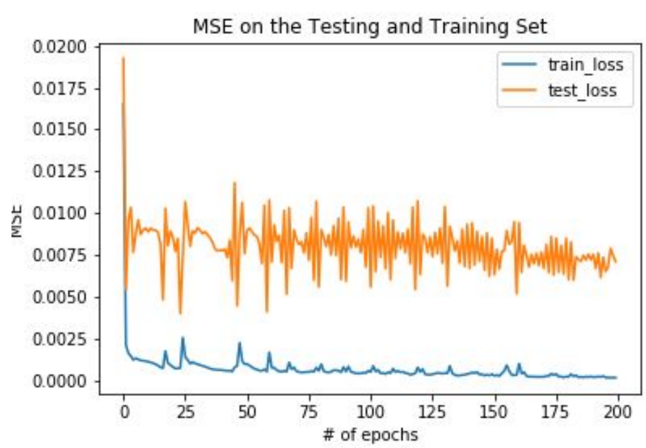
Data points: Contributions





MSE Behaviour:

100 Epochs (left), 200 Epochs (right)



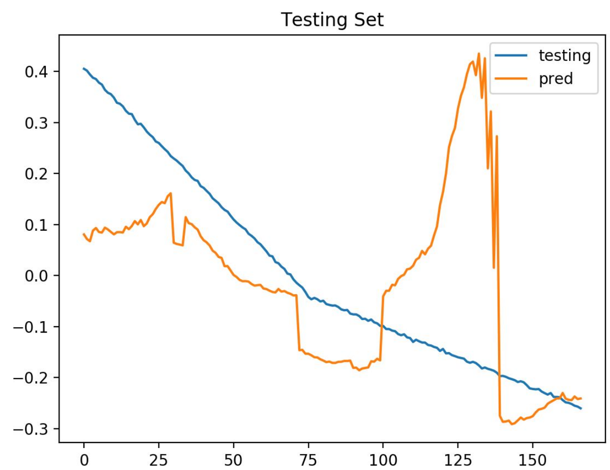
Mid Buyouts

Epochs: 100

Learning Rate: 0.005

Quarters used to predict: 3

Data points: Contributions



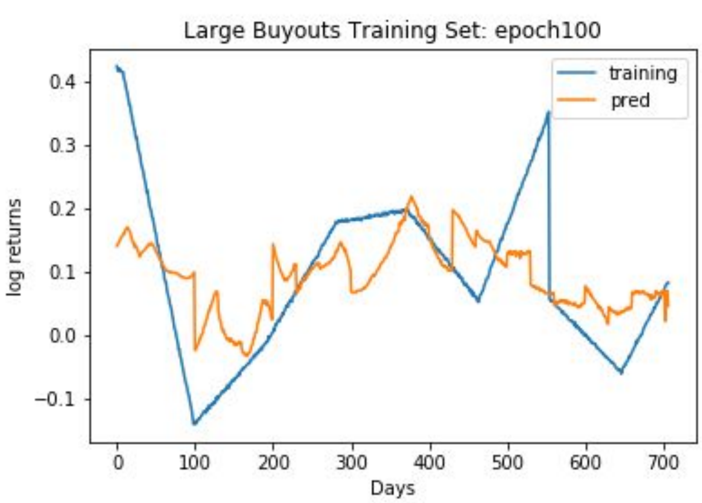
Large Buyouts

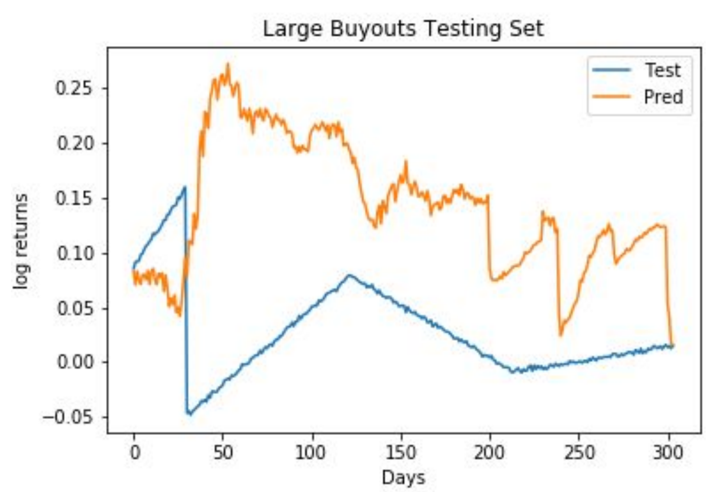
Epochs: 100

Learning Rate: 0.001

Quarters used to predict: 3

Data points: Contributions



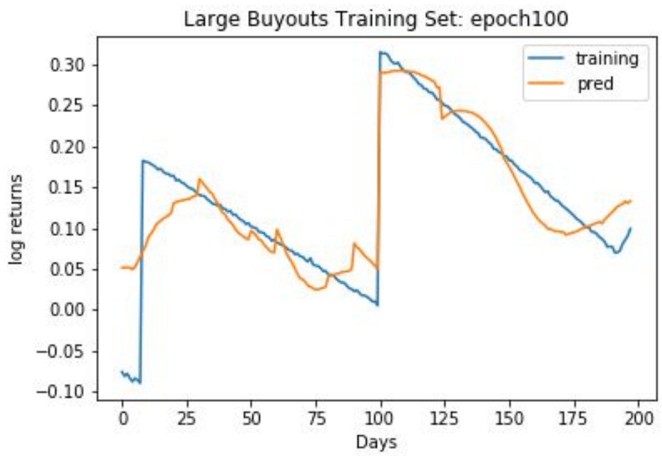


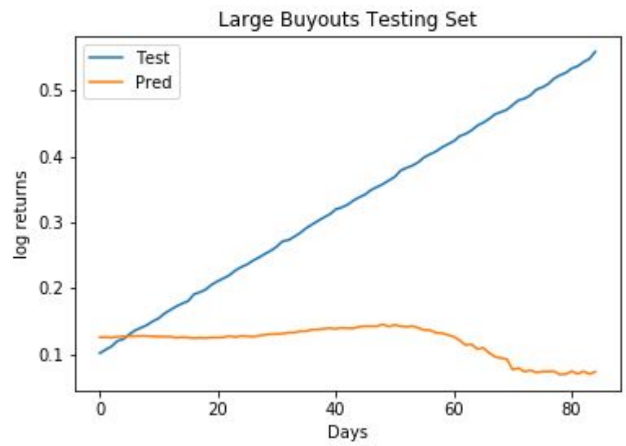
Epochs: 100

Learning Rate: 0.001

Quarters used to predict: 3

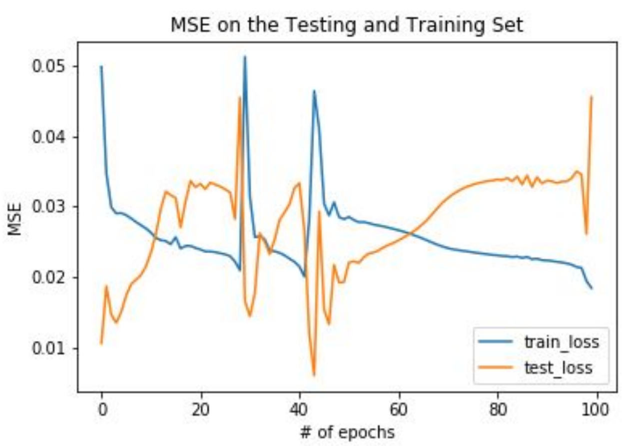
Data points: Distributions

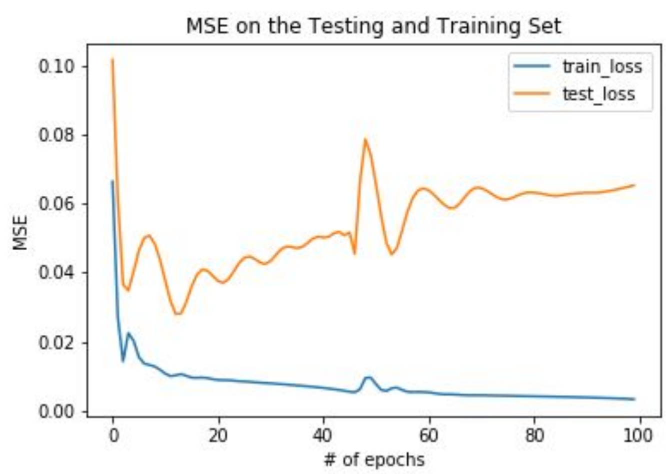




MSE Behaviour:

Contributions (left), Distributions (right)





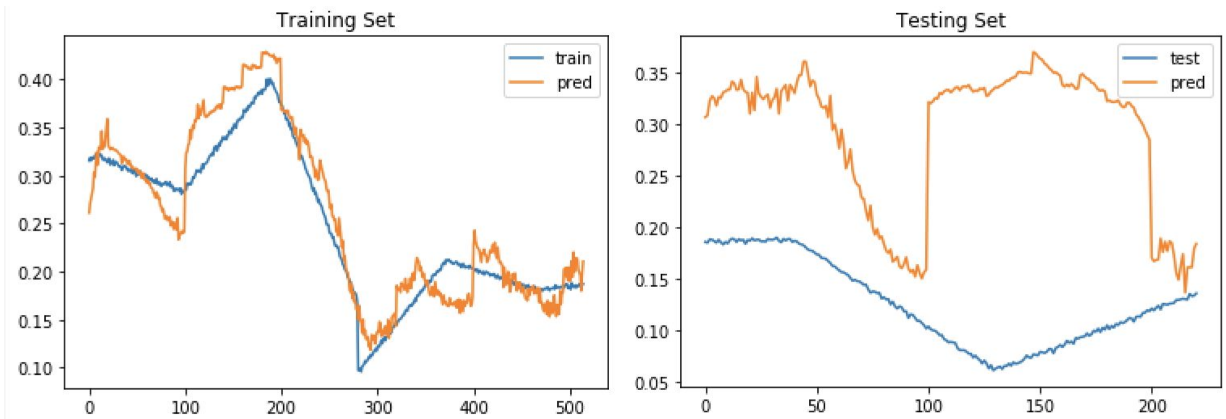
Real Estate Funds

Epochs: 100

Learning Rate: 0.005

Quarters used to predict: 2

Data points: Contributions



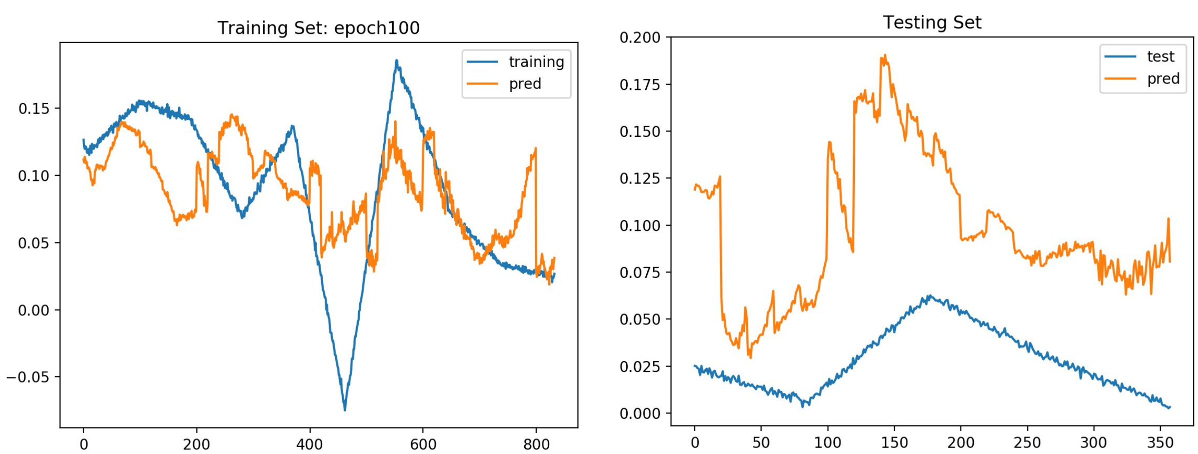
Venture Capital Funds

Epochs: 100

Learning Rate: 0.001

Quarters used to predict: 2

Data points: Contributions



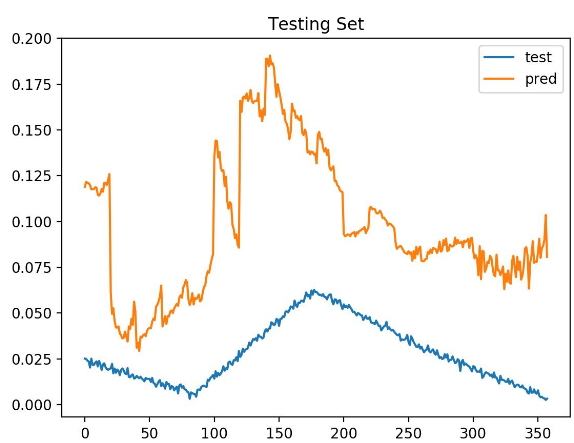
Discussion

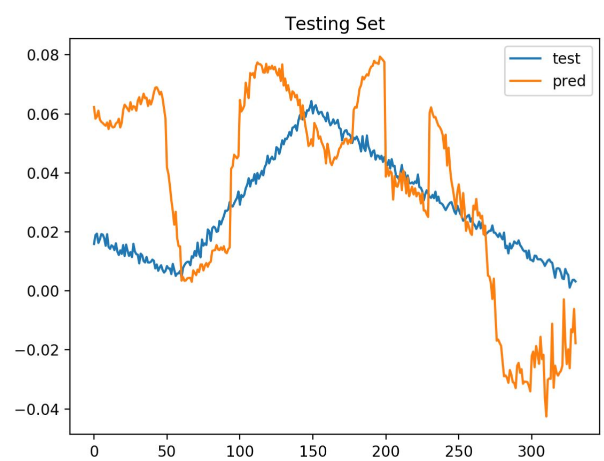
It is observed above that the contributions cash flows have a very small MSE, in each fund we reached, for some routines an error of less than 1%. However, the behaviour of the prediction is very noisy and we tend to overpredict the contribution cash flows. We realized that the overprediction is of a factor of 0.05, which is bizarre as we see that the model is training the data in an accurate way (increasing the number of epochs to 500 would achieve an almost negligible training error--perfect overfit of the data). We adjusted the testing set in the VC funds by this factor and the prediction was significantly more accurate, as shown in the graphs below. Furthermore, the loss function behaves differently for each fund, which made it very hard to find the best parameters to use and to find a balance between overfitting and accurate results.

Venture Capital Funds

Epochs: 100, Learning Rate: 0.001, Quarters: 2, Data: Contributions

No adjustment (left), Adjustment (-0.05, right)





Comparison with Diffusion Process

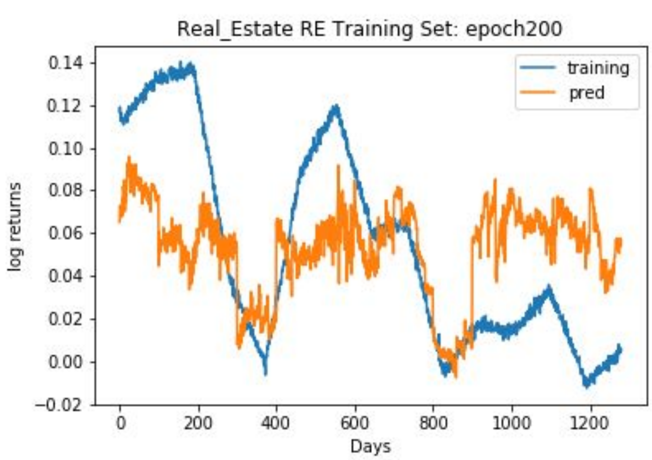
In this section, we present the results obtained by implementing a diffusion process for the prediction of cash flows based on the same data as was used in the construction of the aforementioned LSTM models, in a manner consistent with the original MIT paper. Mean reverting diffusion process were fit to each benchmark dataset, using least squares estimation for the parameters and using the same datasets for such estimations as were used for the training of the LSTM models. Note that we have used ordinary Ornstein-Uhlenbeck processes, based on Bernal's paper *Calibration of the Vasicek Model*, as opposed to mean reverting square root processes that the MIT papers by Buchner used. Such a decision was due to the negative values observed in the benchmark data, contradicting the assumption of the original MIT paper. In multiple cases, we can see that the diffusion processes outperform the LSTM models, however in other cases such as for Mid Buyouts and Real Estate funds, the LSTM models outperform. The results can be seen below.

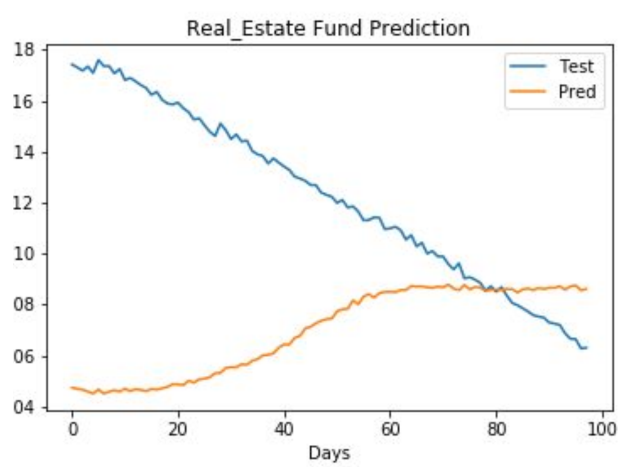
Diffusion Process			
Contribution		***Distribution***	
Fund	***Test MSE***	***Fund***	***Test MSE***
All Buyout	0.000540	All Buyout	0.012270
Small Buyout	0.002270	Small Buyout	0.422322
Mid Buyout	0.003080	Mid Buyout	0.298596
Large Buyout	0.082003	Large Buyout	0.014716
Real Estate	0.006814	Real Estate	0.009707
Venture Capital	0.001110	Venture Capital	3.877390

Case Study: Real Estate Fund

As our LSTM model performed the best with Real Estate funds benchmark data with test MSE as low as 0.0017, we decided to apply our model to a Real Estate fund to examine the model's performance. The selected fund was Prospect Ridge Advisors, a mid-sized real estate fund with AUM of ~2.3 Bn and vintage of 2015, and the cash flow data is obtained from Preqin. We trained our single layer model with 2 quarters of historical data, using learning rate of 0.001 and 200 epochs, which was the best performing Real Estate benchmark model as previously mentioned. After fitting the model, we took the rate of change that the model predicts for the fund and calculated the projected cash flows in one year's time. Here, we used similar calculation method as the MIT paper; once the rate of change of contributions were predicted for one year from now, we used the formula $C_T = C_0 e^{\int_0^T \delta dt}$ to calculate the contribution level one year from now, where C is the contribution level (in %), and δ is the rate of contribution. As stated previously, although our model fits the training data well, the test MSE is quite significant. See below for results and plots of training and test data.

Prospect Ridge Advisors Contributions	
Training MSE	0.001233
Test MSE	0.003815
Predicted Rate in 1 Year	0.047329
Predicted Contribution Level in 1 Year	54.89%
Actual Contribution Level in 1 Year	65.50%





Further Improvements

The LSTM model presented in this paper gives the starting point for further research on using Deep Learning in the Private Equity industry. As a new method that deviates from the "usual" valuation of cash flows and funds, it is important to have in mind the limitations of the model as well as where it can be improved to perform more accurate results.

As mentioned in the results section, our model tends to overpredict the testing set by a factor close to 0.05. We spent a significant amount of time to understand the root of this overprediction. We tried to change how the data was formatted, how we sent the data to the model and how we interpolate the data (mostly focusing on the noise). Any of these approaches fixed the overprediction problem. Moreover, the function used for the prediction on the testing set and on the training set is the same, and the overprediction is only visible on the testing set. If the error is not found, it is possible to use an optimization routine to calculate the adjustment factor for each type of fund to rescale the predictions and further improve the accuracy. Since this error is consistent in all the fund types, we believe that this is a possible solution.

It is observed that the predictions are very noisy, and this comes from the stochastic interpolation and the drastic changes in the data points in each quarter. This can be improved by gathering more data (actual data points during each quarter) to avoid the drastic jumps in the training set that affect the predictions. The data collection will continuously be a challenge in using Deep Learning for Private Equity since it does not provide high signal to noise ratio. Adding the validation set after gathering more data would also be really helpful in improving the models.

This model is built under the assumption that the observed cash flows have all the information to predict the cash flows for the next year. This is a very strong assumption since different funds with different sizes and from different industries can be strongly affected by external factors, and our model does not take that into consideration. The behaviour of the cash flows can be affected by macroeconomic conditions such as inflation, changes in GDP and unemployment. This can be a good starting point to make the model more robust; add more vectors as input using the aforementioned factors to achieve more accurate results. Additionally, each fund and fund type can be affected differently by different macroeconomic as well as industry-specific conditions. Therefore, perhaps categorizing all funds into more specific buckets, incorporating different fund type and industry specific factors, and experimenting with them would further make the model more robust. This will make the model more flexible and would relax a very strong assumption that was used throughout the project.

Conclusion

With the growth in allocation to illiquid alternative asset classes over the past decades, it is only getting more and more compelling to come up with better methods to project future asset values and cash flows for funds. In this project, we aimed to build a model that predicts contributions and distributions of a fund using Deep Learning. Our work differentiates from previous research in the sense that we attempted to predict future cash flows using Deep Learning model, with the assumption that the previous cash flows embed the dynamics that are significant in the future cash flows. After manually obtaining historical benchmark data from Preqin that we used as a proxy in each quarter of the fund's life, we utilized linear interpolation from quarter to quarter with noise terms to overcome the scarcity of data, which is a huge problem in the private equity space.

Our model provides a good approximation of contribution and distribution cash flows of a year later from a given day with the previous 2 and 3 quarters data, resulting in testing MSE as low as 0.0017. We experimented with three different fund types, Leveraged Buyout Funds, Venture Capital Funds, and Real Estate Funds, and found that the model tends to work the best with Real Estate Funds. Additionally, we implemented batch normalization to cope with vanishing gradients and increased flexibility as we can use a different number of days to predict the future cash flow. Such efforts differentiate our model from older tools, as older models tend to rely too heavily on certain assumptions and therefore lack flexibility and adaptability.

As this research was an unprecedented effort incorporating LSTM in predicting contributions and distributions, our model still has several limitations; the predicted results tend to be slightly higher than the actual results, and they tend to be noisy, possibly due to the interpolation. These two areas are what we hope the future teams to tackle, and with further improvements, we believe that this model will have an even stronger predicting power and significance in the future research of private equity fund modeling.

References

Bernal, V.A. (2016). Calibration of the Vasicek Model: An Step by Step Guide.

Buchner, Axel, et al. “Modeling the Cash Flow Dynamics of Private Equity Funds:Theory and Empirical Evidence.” *The Journal of Alternative Investments*, vol. 13, no. 1, 2010, pp. 41–54., doi:10.3905/jai.2010.13.1.041.

Buchner, Axel & Kaserer, Christoph & Wagner, Niklas. (2006). Stochastic modeling of private equity: an equilibrium based approach to fund valuation.

Hochreiter, Sepp, and Jürgen Schmidhuber. “Long Short-Term Memory.” *Neural Computation*, vol. 9, no. 8, 1997, pp. 1735–1780., doi:10.1162/neco.1997.9.8.1735.

Fitch Research. (2018). “SWC Funding LLC.” Aug 03, 2018. Alacra Store. Dec 23, 2019.

Takahashi, Dean, and Seth Alexander. “Illiquid Alternative Asset Fund Modeling.” *The Journal of Portfolio Management*, vol. 28, no. 2, 2002, pp. 90–100., doi:10.3905/jpm.2002.319836.

Olah, C. (2015). “Understanding LSTM Networks.” *Understanding LSTM Networks -- Colah's Blog*, colah.github.io/posts/2015-08-Understanding-LSTMs/.

Overbeck, Ludger, and Tobias Rydén. “Estimation in the Cox-Ingersoll-Ross Model.” *Econometric Theory*, vol. 13, no. 3, 1997, pp. 430–461., doi:10.1017/s0266466600005880.

Yung, J. (2018). “LSTMs for Time Series in PyTorch.” *Jessica Yung*, 11 Sept. 2018, www.jessicayung.com/lstms-for-time-series-in-pytorch/.