ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ "ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

"РОССИЙСКАЯ ЭКОНОМИЧЕСКАЯ ШКОЛА"(ИНСТИТУТ)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Использование рекуррентных нейронных сетей с LSTM архитектурой для предсказания доходностей погодных деривативов

Бакалаврская программа

Совместная программа по экономике НИУ ВШЭ и РЭШ

Автор:

В.Д. Фёдоров

Научный руководитель:

П. Радисевич

Москва, 2022 г.

HIGHER SCHOOL OF ECONOMICS

NEW ECONOMIC SCHOOL

THE BACHELOR'S THESIS

Forecasting Weather Derivatives Using LSTM Recurrent Neural Networks

Bachelor of Arts in Economics Joint Program of HSE and NES

Author:
V.D. Fedorov

Academic supervisor:
P. Radicevic

Moscow, 2022 г.

Аннотация

Работа посвящена прогнозированию доходностей колл опционов на HDD индекс в США с использованием методов машинного обучения. Более точное предсказание доходностей погодных деривативов способствует развитию финансового рынка. Инвесторы чувствуют себя увереннее и более активно вовлекаются в использование погодных деривативов, что повышает их ликвидность. В этой работе я предлагаю альтернативный подход прогнозирования погодных деривативов: методы машинного обучения, основанные на нейронных сетях. Используя среднюю дневную температуру за 2015-2017гг, я оцениваю NN, RNN и LSTM RNN модели, чтобы предсказать доходности колл опционов на HDD индекс на день вперёд. Согласно полученным результатам, простой NN модели достаточно для прогнозирования HDD доходностей. При этом лучшая модель не зависит от региона, так как базовой NN модели достаточно, чтобы предсказывать доходности погодных деривативов. Так же я показал, что скользящее среднее не помогает методам машинного обучения в предсказании доходностей погодных деривативов, поэтому прогноз можно делать без сглаживания погодных данных.

Abstract

This paper forecasts the returns of the HDD call options in the USA with ML methods. The weather significantly affects business activities: energy and agriculture industries, supermarket chains and the leisure industry. Thus, the company's income depends on weather and it's necessary to hedge weather risks. The paper poses an important step forward for more accurate weather derivatives forecasts. I propose an alternative forecasting approach for weather derivatives: models based on neural networks. Using daily average temperature from January 2015 until December 2017, I estimate the NN, RNN and LSTM RNN models for one day ahead HDD call options returns forecasting. My findings suggest that NN are enough to forecast HDD returns and additional complexity of the model does not increase predictive accuracy. Further, I find that sliding window does not help ML models for weather derivatives returns forecasting. Finally, I conclude that the best model does not depend on region, as basic NN structure is enough to forecast returns.

Contents

1 INTRODUCTION 3

2 LITERATURE REVIEW 5

3 DATA DESCRIPTION 7

3.1. Option price data 7

3.2. Weather data 8

4 METHODOLOGY 11

4.1. Neural Networks 11

4.2. Recurrent Neural Networks 12

4.3. Long Short-Term Memory 12

4.4. Model testing 13

5 MODEL MODIFICATIONS 14

5.1. Model specification 14

5.2. Model parameters 14

5.3. Complexity of the model 14

5.4. Sliding window 15

5.5. Geographic region 15

6 RESULTS 16

6.1. Complexity of the model and sliding window 16

6.2. Geographic region 18

7 CONCLUSION 20

REFERENCES 21

8 APPENDIX A 23

1 Introduction

The weather significantly affects business activities: energy and agriculture industries, supermarket chains and the leisure industry (Alaton et al., 2002). Abnormally high or low temperatures lead to crop losses. Hurricanes make energy companies lose equipment. According to Cogen (1998) 200 top U.S. utility companies cited that the weather is determinant for 80% of earnings performance and 50% of poorer performance. Thus, the company's income depends on weather and it’s necessary to hedge weather risks.

Better weather predictions should increase the pricing accuracy. This has two implications to the weather derivatives. First, investors become more confident. Second, more users can be attracted to trade in the market, increasing the liquidity of weather derivatives market (Cramer et al., 2017).

Until 1997 the only way to hedge weather risks was to buy insurance: insurance on property against hurricanes, floods, etc. The problem of these contracts was the need to prove the damage occurred from the weather, verification of which could be difficult. Thus, these contracts covered only extreme weather anomalies and were an incomplete instrument for hedging weather related risks.

After 1997 companies started to use weather derivatives which is a simple way to resolve the problems of financial protection against unfavorable weather conditions (Alaton et al., 2002). Weather derivatives depend solely on weather conditions. They are based on weather index, such as temperature, rainfall or wind-based indexes. Also, they differ by the type (option, futures), period and region. Thus, weather derivatives are better than insurance, as it can be difficult to prove that the weather has an impact unless it is destructive.

To forecast weather derivatives, studies usually use historical payments of derivatives, risk assessment of companies or machine learning (ML) methods, as standard derivative pricing methods are inapplicable for whether derivatives (Yoo, 2003). The underlying asset (weather) is non-tradable, thus, unlike classical derivatives, the agent cannot compile a set of securities replicating it, which means that no-arbitrage assumption is violated.

Most popular method to forecast weather derivatives is a two-step approach: first step — forecast weather, second step – calculate the price of the derivatives based on the weather forecasts. To forecast weather the authors usually use Monte Carlo simulations (Oetomo & Stevenson, 2005; Wang et al., 2015; Shah, 2017), the Burn Analysis and Alaton Model (Schiller et al., 2012) or ML methods (Cramer et al., 2017).

Recent papers show that machine learning methods outperform the benchmark models in predictive accuracy for weather forecasts. For instance, Cramer et al. (2017) compare the predictive performance of the Markov chain extended and six other popular ML algorithms, namely: Genetic Programming, Support Vector Regression, Radial Basis Neural Networks, M5 Rules, M5 Model trees, and k-Nearest Neighbours. They predict daily rainfall data from around Europe and the US and conclude that the ML systems have higher predictive accuracy for the weather data.

ML methods also can be applied to trading and prediction of historical time series price data. For instance, Liu & Zhao (2022) indicate that LSTM networks is more effective than ARIMA, SVM, NN and simple RNN in actual trading. The authors find that LSTM networks outperform the other benchmark models both in returns and return-risk ration and can help traders make profits in financial derivative market. Similarly, Yunpeg et al. (2017) find that LSTM RNN outperforms ARIMA on the multi-step ahead time series forecasting. Authors fit models on the datasets with strong periodic data, reriodic data with trend and super-long periodic data. They find that LSTM RNN can fit a wider range of data patterns compared to ARIMA and that it has higher predictive accuracy. In addition, Mhammedi et al. (2016) forecast stream flow for one day ahead prediction. The authors conclude that modification of RNN outperforms SVR and VAR on the stream flow time series data set.

Therefore, ML methods are commonly used for weather forecasts and financial data prediction. In this paper I further investigate the weather derivatives forecasting by using models based on neural networks: NN, RNN and LSTM RNN. I forecast the returns of the HDD call options in the USA using daily data from January 2015 until December 2017. The effectiveness of models based on neural networks for financial time series prediction and trading is explored (Lui Yunpeg et al., 2017; Mhammedi et al., 2016). Thus, in the paper I focus on the weather derivatives and the objective is to find an appropriate modification of the model, which gives good predictions for weather derivatives. The results indicate that NN are enough to forecast HDD returns and additional complexity of the model does not increase predictive accuracy. In addition, I find that sliding window does not help ML models for weather derivatives returns forecasting. Finally, I conclude that the best model does not depend on region as basic NN structure is enough to forecast returns.

The structure of the paper is as follows. Section 2 briefly reviews literature review. Section 3 in detail describes weather and financial data. Section 4 presents my methodology. Section 5 reviews modifications of the model. Section 6 is devoted to the estimation results and their discussion. Section 7 offers my conclusion.

2 Literature review

In this section I discuss the related literature. First, I consider characteristics of the weather data which could affect the forecasts. Second, I discuss ML methods used for derivatives forecasting. Finally, I provide the hypotheses I test to find an appropriate modification of the ML model for weather derivatives forecasting.

Weather derivatives can be based on any meteorological index such as: rain, temperature, snow, wind, while the most used variable is the temperature (Wang et al., 2015). In the paper I forecast the returns of the heating degree days (HDD) call options using daily average temperature in the USA. Thus, I consider characteristics of the temperature data: seasonality, trends, geographic location.

Many papers in this field show that there are strong seasonality and fluctuating trends in temperature data. For instance, Schiller et al. (2012) use splines to remove seasonality and trend from temperature time series. Campbell & Diebold (2003) estimate the GARCH model for daily average temperature. To model seasonality in temperature data the authors use the low-ordered Fourier series. To approximate the trend, they use the trend component in the mean equation. Then, the authors estimate out-of-sample temperature to calculate expected HDD and CDD option prices.

In the paper I use ML models to cope with seasonality and fluctuating trends in weather data. Yunpeg et al. (2017) show that LSTM RNN successfully works with strong periodic data, periodic data with trend and super-long periodic data. The authors conclude that the use of LSTM models allows to omit the preliminary analysis such as stability checking, ACF and PACF checking.

Another characteristic of the temperature data is geographic location. For instance, Schiller et al. (2012) insist that prediction errors depend on the geographic location of the weather station. They use the Burn Analysis, Alaton Model, Benth Model and Spline Model for temperature data from 35 weather stations across the USA to predict HDD and CDD contracts. The authors show that the MSREs decrease with the latitude of the station for the HDD and increase for the CDD. Oetomo & Stevenson (2005) also find that the choice of the most appropriate model depends on cities and months. In addition, they suggest that the forecasting model varies across the estimation and forecasting samples. Models that produce a better goodness-of-fit for the in-sample estimates, fail in the out-of-sample period and vice versa. Similarly, Wang et al. (2015) conclude that it is necessary to use different weather models for each geographic region for the WD in China. They model temperature by the mean-reverting Ornstein-Uhlenbeck process and use Monte Carlo simulations to forecast the HDD call options. The authors reveal that the model chosen can reasonably simulate a temperature WD contract. However, for complex regional climates of China it is necessary to develop more weather models, such as wavelet functions, B-Spline functions, and polynomial functions.

To test whether the choice of model depends on cities, I train ML models for each city and choose the model with the smallest MSE, MAPE, MSPE. If no model across cities could constantly outperform the others, then hypothesis is going to be true.

ML methods are commonly used for weather forecasts and financial data prediction. For instance, Weerasinghe et al. (2010) find that the ML methods successfully predict daily precipitation. The authors estimate the precipitation data in Sri Lanka using neural network models. In addition to the main findings, they conclude that higher prediction accuracy can be obtained if the model is trained for each season separately: the dry spell and wet spell patterns.

Complexity of the ML model does not necessary improve accuracy of prediction. Schiller et al. (2012) use models with different complexities to forecast temperature-based weather derivatives. The authors conclude that more complex mathematical models do not necessarily yield more accurate results. In contrast, Liu & Zhao (2022) insist that LSTM networks is more effective than more simple neural networks models for derivatives price forecasting. In this regard, I estimate models with different complexities where NN is the simplest one and D-DLSTM is the most complex one.

According to the academic literature sliding window could help with forecasting based on weather data. Cramer et al. (2017) test this hypothesis on the rainfall prediction using MCRP, GP, SVR, RBF, KNN and M5 models and find that sliding window helps ML methods to perform better on rainfall data. Thus, in my work I also test this hypothesis.

This work poses an important step forward for more accurate weather derivatives forecasts. I propose an alternative forecasting approach for weather derivatives: models based on neural networks, and test three hypothesis. First, I test whether complexity of the ML model improve accuracy of prediction. Second, I investigate how sliding window affects models which are based on weather dataset. Finally, I test whether the choice of model depends on cities. In particular, I train ML models for each city and choose the model with the smallest MSE, MAPE, MSPE.

3 Data description

The purpose of this Section is to describe the data used in the paper. First, I provide an overview of financial data used. Second, I discuss weather data. In the paper I forecast the returns of the heating degree days (HDD) call options using daily average temperature in the USA. The data is chosen based on two criteria. The first criterion is that the temperature is the most used weather variable for weather derivatives forecasts (Wang et al., 2015). Secondly, I use call options on the HDD index, as they are the most liquid among weather derivatives (Alaton et al., 2002).

3.1. Option price data

The sample spans from May 2015 until December 2017. HDD call options is collected from the Reuters database. In line with the academic literature, I define the forecasted variable as the first difference of the daily yields. I removed weekends, holidays and empty intervals.

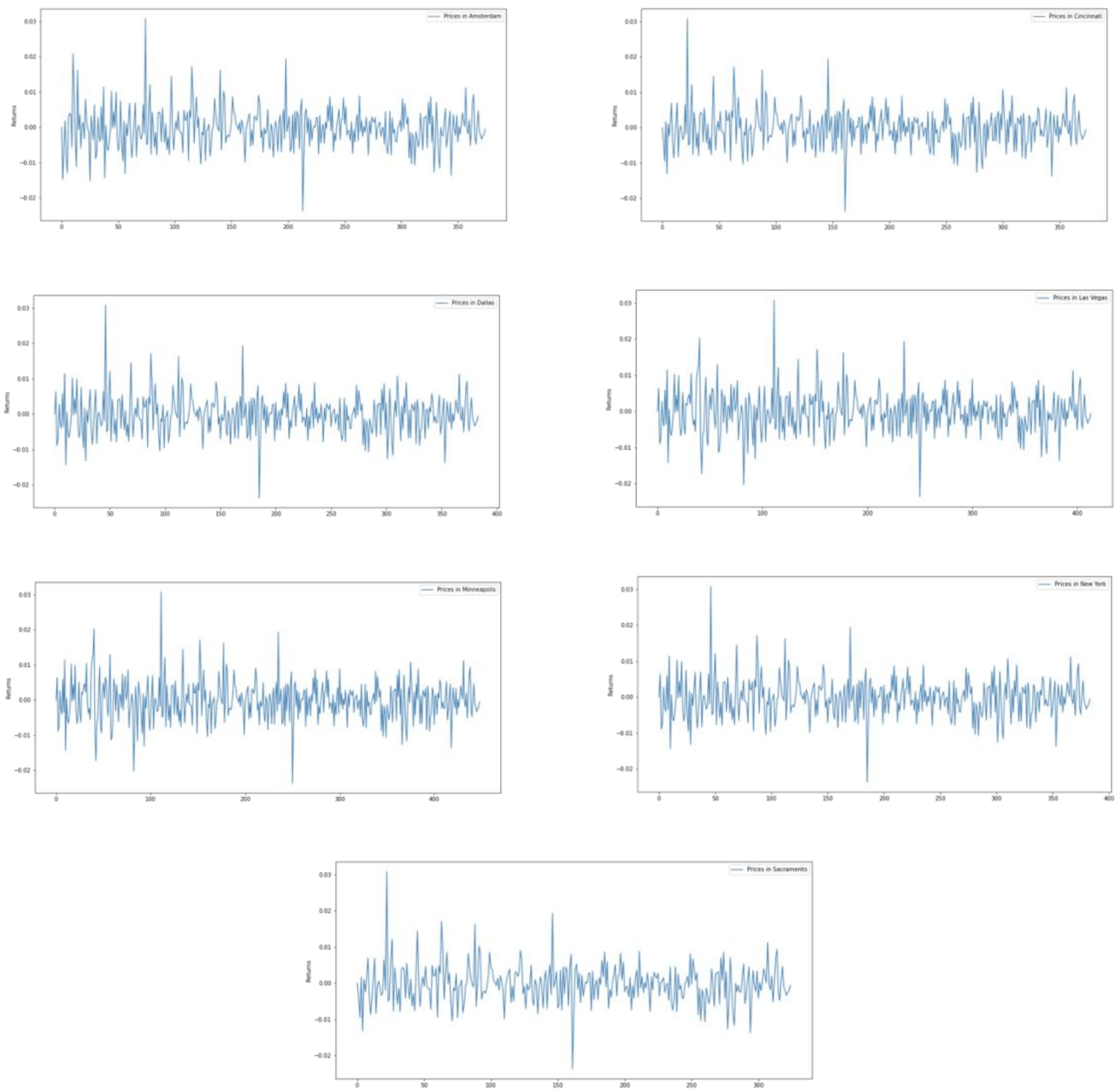

Fig. 1: Time-series plot of daily returns of HDD call option for each of seven cities

In Fig. 1 I plot the daily returns for each of seven cities. It is seen that returns have similar structure and does not have crucial outliers. From descriptive statistics it is also clear, that returns do not have significant differences. For all cities the mean is around 0, maximum and minimum values are similar. Standard deviation is around 0.55.

Tab. 1. Summary statistics of the HDD Call Option returns by Cities

Variable	Amsterdam (USA)	Cincinnati	Dallas	Las Vegas	Minneapolis	New York	Sacramento
Sample size	375	374	384	414	449	384	325
Min	-2.37	-2.37	-2.37	-2.37	-2.37	-2.37	-2.37
Max	3.08	3.08	3.08	3.08	3.08	3.08	3.08
Mean	-0.02	-0.02	-0.02	-0.02	-0.01	-0.02	-0.02
SD	0.58	0.53	0.54	0.58	0.57	0.54	0.54
Skewness	0.48	0.49	0.45	0.33	0.30	0.45	0.55
Kurtosis	3.21	3.99	3.46	2.87	2.78	3.46	4.26
JB	175.09***	262.80***	204.43***	149.43***	151.14***	204.43***	261.77***
ADF	-20.21***	-20.03***	-20.53***	-14.99***	-15.68***	-20.53***	-18.45***

Notes: JB — Jarque-Bera test (H0: the variable is normally distributed); ADF — augmented Dickey-Fuller test (H0: time series is integrated of order 1); *, ** and *** statistical significance at 10%, 5% and 1% respectively.

For all returns distribution differs from normal. First, for all cities standard deviation is less than one. Second, skewness is greater than zero, which means that distribution is skewed right. Third, Las Vegas and Minneapolis have kurtosis less than 3, which means that distribution has flatter tails than normal distribution, while all other cities have kurtosis greater than 3, which means that distribution has thicker tails than normal. Consequently, the Jarque-Bera test rejected the null hypothesis of the normality for returns in all cities at 5% significance level. According to ADF test in all cities, returns are stationary at 5% significance level.

3.2. Weather data

The sample spans from January 2015 until December 2017. The data set with temperature is collected from NOAA - National Oceanic and Atmospheric Administration. To forecast HDDs returns in each day I use previous n days. Therefore, weather data starts n days before financial data starts.

HDDs are a transformation of daily average temperature (DAT). Thus, in line with the academic literature (Campbell & Diebold, 2003; Schiller et al., 2012; Oetomo & Stevenson, 2005) I directly use DAT, measured in Celsius, for each of seven cities: Sacramento, Minneapolis, Las Vegas, New York, Amsterdam, Cincinnati and Dallas. Formally,

$$DATji = \frac{\min(Tji) + \max\,(Tji)}{2},$$

where i – day, j – station.

When there are several measurement stations in the city, the DAT is measured as an average DAT of these stations:

$$DATi = avg\left(DAT_{ij}\right),$$

where i – day, j – station.

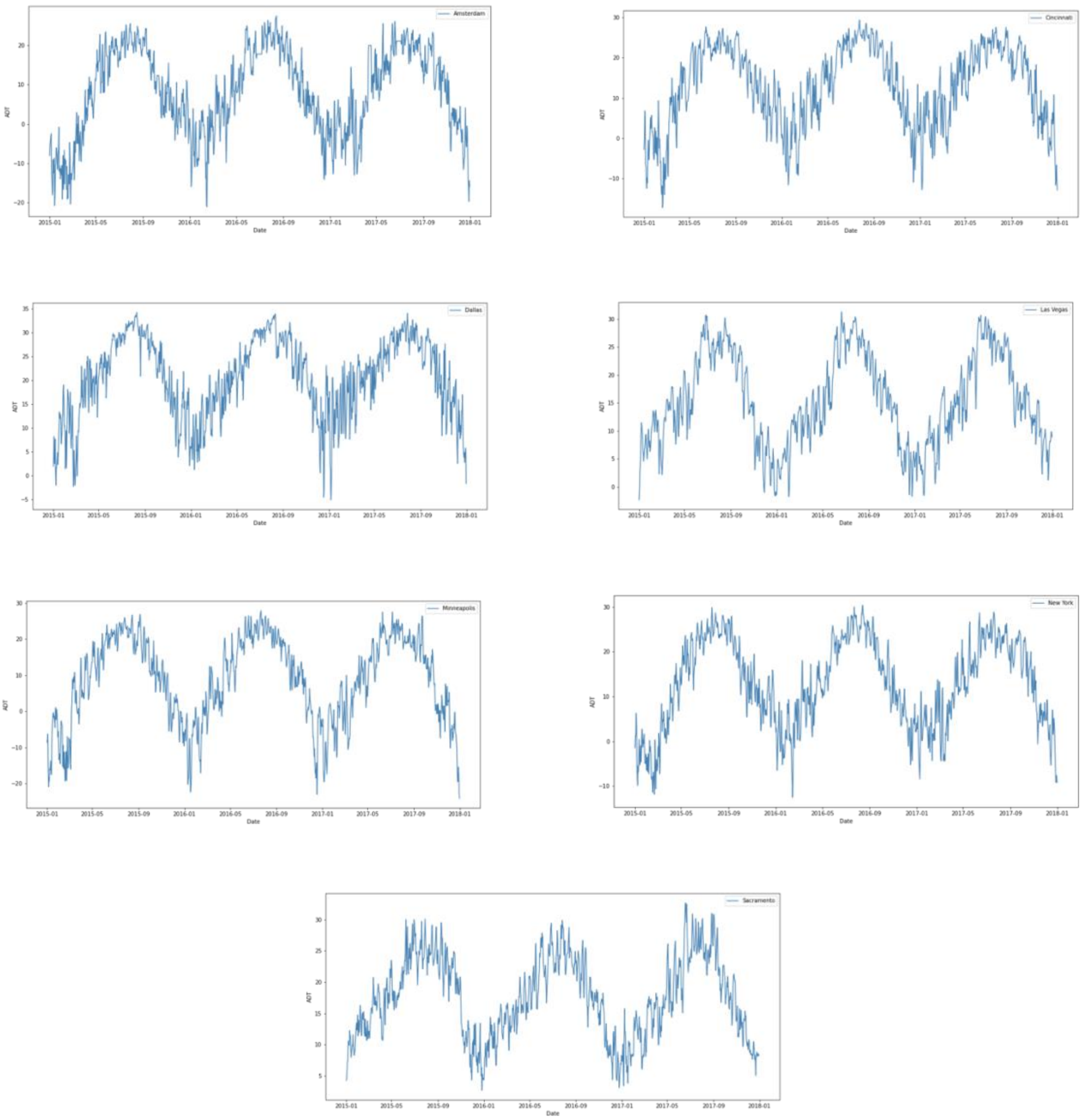

Fig. 2: Time-series plot of daily average temperature for each of seven cities, Celsius

In Fig. 2 I plot the daily average temperature for each of seven cities. It is seen that there is a strong seasonality in average temperature, that corresponds to academic literature (Campbell & Diebold, 2003; Schiller et al., 2012; Alaton et al., 2002).

Temperature data differs across cities (see Tab. 2). For example, minimal standard deviation of the temperature (6.5 in Sacramento) is almost half of the maximum one (11.8 in Minneapolis), minimal temperature differs from -24.2 in Minneapolis to 2.7 in Sacramento. Thus, I introduce 6 dummy variables, which equals one if there is the chosen city and zero otherwise, to capture possible effects of the geographic region.

Tab. 2. Summary statistics of the Daily Average Temperature (DAT) by Cities

Variable	Amsterdam (USA)	Cincinnati	Dallas	Las Vegas	Minneapolis	New York	Sacramento
Sample size	1030	1096	1096	1096	1096	1096	1096
Min	-21.1	-17.2	-5.1	-2.3	-24.2	-12.5	2.7
Max	27.5	29.3	34.2	31.3	27.9	30.4	32.7
Mean	7.7	12.9	20.4	15.5	8.8	12.9	17.5
SD	11.1	10.0	8.4	8.1	11.8	9.7	6.5
Skewness	-0.31	-0.53	-0.51	0.02	-0.51	-0.28	0.00
Kurtosis	-0.80	-0.58	-0.52	-1.00	-0.63	-0.91	-0.92
JB	55.18***	66.24***	60.49***	45.34***	65.23***	51.64***	38.95***
ADF	-2.09	-1.95	-2.12	-2.07	-1.83	-1.84	-1.81

Notes: JB — Jarque-Bera test (H0: the variable is normally distributed); ADF — augmented Dickey-Fuller test (H0: time series is integrated of order 1); *, ** and *** statistical significance at 10%, 5% and 1% respectively.

Temperature in all cities has not normal distribution. First, for all cities standard deviation is greater than one. Second, in all cities except Las Vegas and Sacramento skewness of temperature has distribution skewed left. Third, temperature in all cities have kurtosis less than 3, which means that distribution has thicker tails than normal distribution. Consequently, the Jarque-Bera test rejected the null hypothesis of the normality for temperature in all cities at 5% significance level. According to ADF test the average temperature in all cities is not stationary.

4 Methodology

In Section 4 I briefly describe the neural networks models I use: NN, RNN and LSTM RNN. To understand how models work I discuss models from the simplest one to the most complex one. Finally, I provide an overview of tests I use in the paper, to choose the best model.

4.1. Neural Networks

Neural networks (NN) consist of input layer, hidden layer and output layer (see Fig. 3). Each layer consists of neurons.

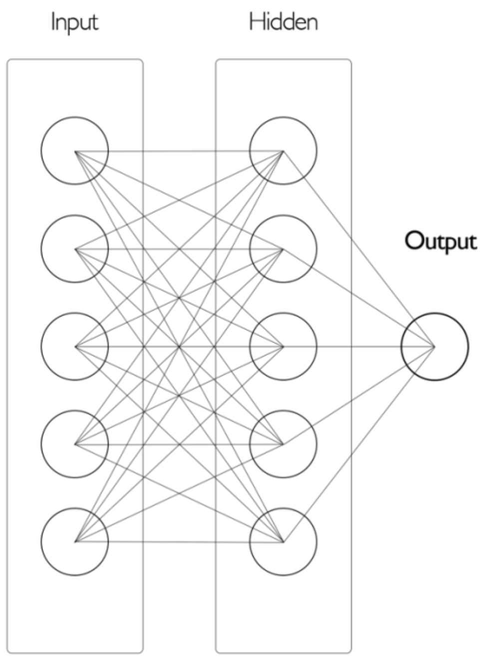

Fig. 3: Structure of the Neural Networks

Input layer takes the input vector with values between 0 and 1. Value of the neurons in the input layer equals to the corresponding value in the input vector.

Value of the neurons in hidden layer equals to the weighted sum of neurons connected to it, where each connection has a weight.

Output layer has similar structure to hidden layer. The output of the model is a weighted sum of neurons in the hidden layer. Training process returns weights that results in the smallest prediction errors.

4.2. Recurrent Neural Networks

Recurrent neural networks (RNN) differ from NN as for the input they take not only the input data X_t but also previous output h_{t-1} and returns output h_t. It allows RNN to transfer information through steps of calculations, gives it “memory”. As the information has to be recalculated on each step, the “memory” works only on short horizons and should be called short-term memory.

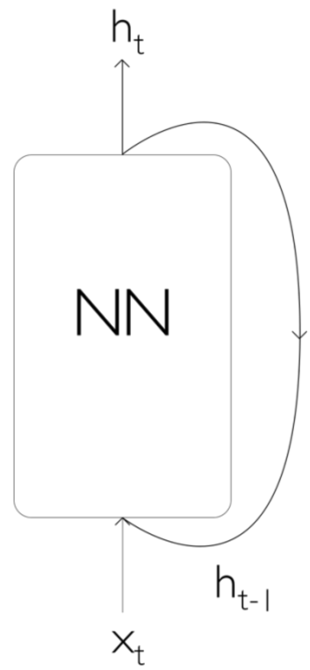

Fig. 4. Structure of the Recurrent Neural Networks

4.3. Long Short-Term Memory

Long short-term memory (LSTM) neural networks in addition to classic RNN structure have long-term memory. This memory works through the process of “forgetting” and “remembering”. On each step LSTM RNN decide whether to forget old information and whether to remember new one. It allows to transfer information through more than one step and gives the advantage in ordered data processing.

In the paper I suggest modifications of LSTM RNN, which differ by the number of layers. First basic modification has one LSTM layer – LSTM RNN. Second modification has two LSTM layers, and it is called Deep LSTM RNN (DLSTM RNN). Third modification has five LSTM layers, and it is called Deep-DLSTM RNN (D-DLSTM RNN).

Therefore, I train five models: NN, RNN, LSTM RNN, DLSTM RNN, D-DLSTM RNN, and compare their results.

4.4. Model testing

Standard econometric model testing such as: coefficient significance, model significance, testing for error correlations, are inapplicable to ML models. To evaluate predicting results studies in this field usually use error evaluation indexes. For instance, Yunpeg et al. (2017) compare LSTM and GRNN with ARIMA using mean square error (MSE) and mean square percentage error (MSPE). Sun & Huang (2020) use LSTM augmented with decomposition algorithms and compare it with BP, ELM and LSSVM algorithms. To compare results, the authors use root mean square error (RMSE) and mean absolute percentage error (MAPE). For more accurate comparison they use percentage improvement of the loss (P_{loss}) to indicate the best performing model:

$$P_{Loss} = \left(1 - \frac{Loss_i}{Loss_{basic}}\right) * 100\%,$$

where $Loss_{basic}$ – the loss of the basic model, $Loss_i$ – the loss of the model i.

In line with academic literature, I use the common error evaluation indexes to evaluate the prediction performance of models. They are mean absolute percentage error (MAPE), mean square error (MSE) and mean square percentage error (MSPE). I don't use RMSE because it is transformation of MSE, therefore MSE is enough. In addition, I evaluate percentage improvement of the loss to indicate the best performing model.

5 Model modifications

The purpose of this Section is to describe modifications of models used in the paper. First, I consider a model specification: input and output data. Second, I provide an overview of model parameters and discuss my choice. Third, I discuss the complexity of models used in the paper. Further, I describe the approach applied to weather data to improve forecasts accuracy. Finally, I consider geographic location of the weather stations.

5.1. Model specification

In the paper I forecast returns of the HDD call options using daily average temperature (DAT), thus, my inputs are DAT for the previous n days. I add dummy city because data consist of the different cities that have different data structure. Therefore, final inputs and outputs are:

$$Input_i = T_{i-1}, T_{i-2}, \dots, T_{i-n}, D$$
$$Output_i = Return_i,$$

where n – number of input days, D – is the dummy city and i – is a day.

5.2. Model parameters

Overtraining is an issue for all ML methods. To counteract it, I use drop out rate of 20%: for each iteration of training 20% of neurons are not used. Thus, model does not have the ability to "remember" the data set and overtraining is not an issue. Drop out rate is 20%, as it shows the best results.

For the input I take 10 previous days. 10 days is chosen because less days show worse results, more days show no change in results except longer training time.

I select a different number of epochs (number of training iterations) for each model modification. For example, simple models or models trained on long datasets require less epochs (5 or 10), while complex ones or models trained on short datasets (dataset with only one city) show best results with epochs up to 100.

5.3. Complexity of the model

Complexity of the ML model does not necessary improve accuracy of prediction. Schiller et al. (2012) use models with different complexities to forecast temperature-based weather derivatives. The authors conclude that more complex mathematical models do not necessarily yield more accurate results. To test this hypothesis, I choose models with different complexities where NN is the simplest one and D-DLSTM RNN is the most complex one. If complex model does not outperform others than the hypothesis is going to be true.

The more complex models are able to extract hidden structure in the complex datasets. For example, long-term memory allows model to "remember" some important information in the previous steps and use it for more accurate prediction later. Therefore, it could be useful for the returns forecasting.

5.4. Sliding window

In line with the academic literature, I use sliding window to test whether it helps with forecasting based on weather dataset (Cramer et al. 2017). To test the hypothesis, I fit ML models on raw DAT and on DAT with sliding window. If the MSE, MAPE, MSPE in the second case is lower than hypothesis is going to be true.

$$DAT_{sliding_i} = \frac{1}{n} \sum_{j=i-n}^{j=i} DAT_j ,$$

where n – number of days in sliding window.

5.5. Geographic region

Best ML model could vary for different cities (Oetomo & Stevenson, 2005). In my sample data differs across cities, and it could lead to different best models. For example, minimal standard deviation of the temperature (6.5 in Sacramento) is almost half of the maximum one (11.8 in Minneapolis). Models could work differently for datasets with different volatilities. Therefore, for different cities best model could be different. To test this hypothesis, I train ML models for each city and choose the model with the smallest MSE, MAPE, MSPE. If no model across cities could constantly outperform the others, then hypothesis is going to be true.

6 Results

This Section presents the empirical results. First, I test whether complexity of the ML model improve accuracy of prediction. Second, I investigate how sliding window affects models that are based on weather dataset. Finally, I test whether the choice of model depends on cities. In particular, I train ML models for each city and choose the model with the smallest MSE, MAPE, MSPE.

6.1. Complexity of the model and sliding window

I train 5 times for 20 epochs each modification of the model on all dataset with dummy variable on the region. Then for each modification I choose model with the smallest MSE. It is reasonable, as models can find different local minimums in one iteration. Thus, through this process, I find the model that finds the global one.

Model fitting was done in two variations – on daily average temperature (DAT) and on DAT with sliding widow of 5 days.

Tab. 3. Summary statistics of the MSE, MSPE and MAPE Losses of models fitted on Raw DAT and DAT with sliding window

Model	MSE	MSPE	MAPE	$MSE_{Sliding}$	$MSPE_{Sliding}$	$MAPE_{Sliding}$
NN	0.0071	0.0846	0.1717	0.0071	0.0816	0.1692
RNN	0.0071	0.0806	0.1703	0.0070	0.0808	0.1685
LSTM	0.0071	0.0830	0.1686	0.0071	0.0794	0.1710
DLSTM	0.0070	0.0837	0.1709	0.0071	0.0841	0.1675
D-DLSTM	0.0071	0.0816	0.1663	0.0071	0.0836	0.1709

From Tab 3. it is seen that in both cases simple models: NN, RNN, have almost the same losses as complex models such as DLSTM RNN and D-DLSTM RNN. For Raw DAT: MSE, MSPE and MAPE losses show that none of the models could be clearly defined as the best one.

Tab. 4 shows the improvement of losses compared to NN model. NN is chosen as model for comparison because it is the most basic one. If P_{Loss} is positive then the error index is smaller than in NN model. If the P_{Loss} is negative, then this model have greater error index than NN model.

From Tab. 4 it is seen that no complex model (DLSTM, D-DLSTM) can constantly outperforms the others. In particular, DLSTM RNN have the biggest improvement according to P_{MSE} of 1.4%, RNN have the biggest improvement according to P_{MSPE} of 4.7%, D-DLSTM RNN have the biggest improvement according to P_{MAPE} of 3.1%. For DAT with sliding window: for P_{MSPE}, P_{MSPE} and P_{MAPE} best models are RNN, LSTM RNN, DLSTM RNN respectively.

Tab. 4. Summary statistics of the percentage improvement of MSE, MSPE and MAPE in comparison to NN

Model	P_{MSE}	P_{MSPE}	P_{MAPE}	$P_{MSE, sliding}$	$P_{MSPE, sliding}$	$P_{MAPE, sliding}$
NN	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
RNN	0.0%	4.7%	0.8%	1.4%	1.0%	0.4%
LSTM	0.0%	1.9%	1.8%	0.0%	2.7%	-1.1%
DLSTM	1.4%	1.1%	0.5%	0.0%	-3.1%	1.0%
D-DLSTM	0.0%	3.5%	3.1%	0.0%	-2.5%	-1.0%

This result is consistent with Schiller et al. (2012). The authors conclude that more complex models do not necessarily yield more accurate results. It can be explained as basic NN structure is enough to extract hidden structure in the dataset and therefore, more complex models with additional architecture yield the same results.

Tab. 5. Summary statistics of the percentage improvement of MSE, MSPE and MAPE of DAT with sliding window in comparison to raw DAT

Model	P_{MSE}	P_{MSPE}	P_{MAPE}
NN	0.0%	3.5%	1.5%
RNN	1.4%	-0.2%	1.1%
LSTM	0.0%	4.3%	-1.4%
DLSTM	-1.4%	-0.5%	2.0%
D-DLSTM	0.0%	-2.5%	-2.8%

From Tab. 5 it is seen that sliding window does not help with temperature processing. For RNN, LSTM RNN and DLSTM RNN percentage error improvements P_{MSE}, P_{MSPE} and P_{MAPE} do not show clear results. For D-DLSTM RNN results are negative, meaning that the forecasts get slightly worse. For NN results are positive, meaning that the forecasts get slightly better.

This result contradicts the findings of Cramer et al. (2017). In the article author tests this hypothesis on the rainfall prediction using MCRP, GP, SVR, RBF, KNN and M5 models and find that sliding window helps ML methods to perform better on rainfall data. In the paper I tested NN, RNN and modifications of LSTM RNN on daily returns forecasting and found that sliding window does not help reduce errors. It can be explained as the daily returns are dependent on the difference in temperature which is reduced by sliding window.

6.2. Geographic region

To test the hypothesis about geographic region, I train models separately for each city. Similarly, to the general dataset, I train 5 times for 100 epochs each modification of the model on the dataset for each city. Therefore, I get 5 models of each modification for each city. Then for each modification I choose model with the smallest MSE. It is reasonable as models can find different local minimums in the dataset and through this process, I find the model that find the global one.

I choose 100 epochs because additional epochs (200 and 500) do not yield errors reduction. The number of epochs is greater than in previous Section because the data set is smaller – only one city instead of 7. Dropout rate is the same – 20%. Model fitting was done on raw daily average temperature (Raw DAT). Results are presented in the Tab. 6.

Tab. 6. Summary statistics of the MSE, MSPE and MAPE Losses of models fitted on Raw DAT

Model	MSE	P_{MSE}	MSPE	P_{MSPE}	MAPE	P_{MAPE}
Amsterdam (USA)						
NN	0.0087	0.0%	0.1062	0.0%	0.1952	0.0%
RNN	0.0087	0.0%	0.1042	1.9%	0.1939	0.7%
LSTM	0.0085	2.3%	0.1056	0.6%	0.1929	1.2%
DLSTM	0.0087	0.0%	0.1035	2.5%	0.1938	0.7%
D-DLSTM	0.0087	0.0%	0.1051	1.0%	0.1943	0.5%
Cincinnati						
NN	0.006	0.0%	0.0614	0.0%	0.1497	0.0%
RNN	0.006	0.0%	0.0593	3.4%	0.1481	1.1%
LSTM	0.006	0.0%	0.0596	2.9%	0.1475	1.5%
DLSTM	0.0061	-1.7%	0.0607	1.1%	0.1467	2.0%
D-DLSTM	0.006	0.0%	0.0604	1.6%	0.1489	0.5%
Dallas						
NN	0.006	0.0%	0.0611	0.0%	0.1479	0.0%
RNN	0.0061	-1.7%	0.0593	2.9%	0.148	-0.1%
LSTM	0.0061	-1.7%	0.0654	-7.0%	0.152	-2.8%
DLSTM	0.0061	-1.7%	0.0641	-4.9%	0.1527	-3.2%
D-DLSTM	0.006	0.0%	0.0595	2.6%	0.1484	-0.3%

Las Vegas						
NN	0.0081	0.0%	0.1019	0.0%	0.1881	0.0%
RNN	0.0083	-2.5%	0.109	-7.0%	0.1945	-3.4%
LSTM	0.0084	-3.7%	0.0938	7.9%	0.1859	1.2%
DLSTM	0.0083	-2.5%	0.1049	-2.9%	0.1907	-1.4%
D-DLSTM	0.0082	-1.2%	0.0977	4.1%	0.1847	1.8%
Minneapolis						
NN	0.006	0.0%	0.0593	0.0%	0.1481	0.0%
RNN	0.006	0.0%	0.0602	-1.5%	0.149	-0.6%
LSTM	0.0061	-1.7%	0.0611	-3.0%	0.1488	-0.5%
DLSTM	0.006	0.0%	0.0611	-3.0%	0.1493	-0.8%
D-DLSTM	0.006	0.0%	0.0618	-4.2%	0.1499	-1.2%
New York						
NN	0.006	0.0%	0.0594	0.0%	0.1481	0.0%
RNN	0.0061	-1.7%	0.0598	-0.7%	0.1471	0.7%
LSTM	0.0062	-3.3%	0.0566	4.7%	0.1466	1.0%
DLSTM	0.006	0.0%	0.0596	-0.3%	0.1482	-0.1%
D-DLSTM	0.006	0.0%	0.062	-4.4%	0.1503	-1.5%
Sacramento						
NN	0.0082	0.0%	0.1069	0.0%	0.1925	0.0%
RNN	0.0081	1.2%	0.1043	2.4%	0.1898	1.4%
LSTM	0.0083	-1.2%	0.1029	3.7%	0.1903	1.1%
DLSTM	0.0084	-2.4%	0.1096	-2.5%	0.1946	-1.1%
D-DLSTM	0.0086	-4.9%	0.1154	-8.0%	0.1997	-3.7%

From Tab. 6 it is seen that for each city all models have almost the same results, no model can constantly outperforms the others. In each city P_{MSE}, P_{MSPE} and P_{MAPE} either show different best models (Amsterdam, Cincinnati, Dallas, Las Vegas, Sacramento) or multiple best models (Minneapolis, New York).

This result contradicts Oetomo & Stevenson (2005). The authors find that the best model varies for different cities. It can be explained as basic NN structure is enough to forecast returns and therefore more complex models with additional architecture yield same results. Therefore, for each city all models have similar MSE.

7 Conclusion

Weather is crucial factor for the economy, companies each year suffer losses because of it. Weather derivatives allow companies to hedge this risk. This work poses an important step forward for more accurate weather derivatives forecasts. I propose an alternative forecasting approach for weather derivatives: models based on neural networks. Using daily average temperature from January 2015 until December 2017, I estimate the NN, RNN and LSTM RNN models for one day ahead HDD call options returns forecasting in the USA.

My findings suggest that a simple NN model is enough to forecast HDD returns and additional complexity of the model does not increase predictive accuracy. It can be explained as basic NN structure is enough to extract hidden structure in the dataset. Further, I find that sliding window does not help ML models for weather derivatives returns forecasting. It can be explained as the daily returns are dependent on the difference in temperature which is reduced by sliding window. Finally, I conclude that the best model does not depend on region because basic NN structure is enough to forecast returns. Therefore, for each city all models have similar errors of prediction.

For future research it could be of interest to test other classes of ML methods. It is possible that models based on K-nearest neighbors, decision trees or random forest algorithms are more suitable for the purpose of HDD return forecasting. Further, it is reasonable to test ML methods for CDD index returns forecasting. Different indexes have different data patterns and could have different results.

References

Alaton, P., Djehiche, B., & Stillberger, D. (2002). On modelling and pricing weather derivatives. *Applied mathematical finance*, 9(1), 1-20.

Cogen, J. (1998). What is weather risk. *PMA online magazine*, 5, 98.

Cramer, S., Kampouridis, M., Freitas, A. A., & Alexandridis, A. K. (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. *Expert Systems with Applications*, 85, 169-181.

Diebold, F. X., & Campbell, S. D. (2003). *Weather forecasting for weather derivatives*. National Bureau of Economic Research.

Liu, B., & Zhao, Q. (2022). Financial Derivative Price Forecasting and Trading for Multiple Time Horizons with Deep Long Short-Term Memory Networks. *Scientific Programming, 2022.*

Mhammedi, Z., Hellicar, A., Rahman, A., Kasfi, K., & Smethurst, P. (2016, December). Recurrent neural networks for one day ahead prediction of stream flow. In Proceedings of the Workshop on Time Series Analytics and Applications (pp. 25-31).

Oetomo, T., & Stevenson, M. (2005). Hot or cold? A comparison of different approaches to the pricing of weather derivatives. *Journal of Emerging Market Finance, 4(2),* 101-133.

Rajurkar, M. P., Kothyari, U. C., & Chaube, U. C. (2004). Modeling of the daily rainfall-runoff relationship with artificial neural network. *Journal of Hydrology*, 285(1-4), 96-113.

Schiller, F., Seidler, G., & Wimmer, M. (2012). Temperature models for pricing weather derivatives. Quantitative Finance, 12(3), 489-500.Oetomo & Stevenson, (2005). Hot or Cold? A Comparison of Different Approaches to the Pricing of Weather Derivatives. *Journal of emerging market Finance,* 4-2

Sun, W., & Huang, C. (2020). A novel carbon price prediction model combines the secondary decomposition algorithm and the long short-term memory network. *Energy*, *207*, 118294.

Shah, A. (2017, September). Pricing of rainfall derivatives using generalized linear models of the daily rainfall process. In *International Agricultural Risk, Finance and Insurance Conference (IARFIC) 2017 Paris meetings paper*.

Yunpeng, L., Di, H., Junpeng, B., & Yong, Q. (2017, November). Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In *2017 14th web information systems and applications conference (WISA)* (pp. 305-310). IEEE.

Yoo, S. (2003). Weather derivatives and seasonal forecast. *Department of Applied Economics and Management, Cornell University.*

Wang, Z., Li, P., Li, L., Huang, C., & Liu, M. (2015). Modeling and forecasting average temperature for weather derivative pricing. *Advances in Meteorology*, 2015.

Weerasinghe, H. D. P., Premaratne, H. L., & Sonnadara, D. U. J. (2010). Performance of neural networks in forecasting daily precipitation using multiple sources.

8 Appendix A

Tab. A1. NN model configuration

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 16, 50)            100

 dense_7 (Dense)             (None, 16, 1)             51

=================================================================
Total params: 151
Trainable params: 151
Non-trainable params: 0
_________________________________________________________________
```

Tab. A2. RNN model configuration

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 16, 50)            2600

 dense_2 (Dense)             (None, 16, 1)             51

=================================================================
Total params: 2,651
Trainable params: 2,651
Non-trainable params: 0
_________________________________________________________________
```

Tab. A3. LSTM model configuration

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50)                10400

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 10,451
Trainable params: 10,451
Non-trainable params: 0
_________________________________________________________________
```

Tab. A4. DLSTM model configuration

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_1 (LSTM)               (None, 16, 50)            10400

 dropout_1 (Dropout)         (None, 16, 50)            0

 lstm_2 (LSTM)               (None, 50)                20200

 dropout_2 (Dropout)         (None, 50)                0

 dense_4 (Dense)             (None, 1)                 51

=================================================================
Total params: 30,651
Trainable params: 30,651
Non-trainable params: 0
_________________________________________________________________
```

Tab. A5. D-DLSTM model configuration

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_11 (LSTM)              (None, 16, 50)            10400

 dropout_11 (Dropout)        (None, 16, 50)            0

 lstm_12 (LSTM)              (None, 16, 50)            20200

 dropout_12 (Dropout)        (None, 16, 50)            0

 lstm_13 (LSTM)              (None, 16, 50)            20200

 dropout_13 (Dropout)        (None, 16, 50)            0

 lstm_14 (LSTM)              (None, 16, 50)            20200

 dropout_14 (Dropout)        (None, 16, 50)            0

 lstm_15 (LSTM)              (None, 50)                20200

 dropout_15 (Dropout)        (None, 50)                0

 dense_11 (Dense)            (None, 1)                 51

=================================================================
Total params: 91,251
Trainable params: 91,251
Non-trainable params: 0
_________________________________________________________________
```

Tab. A6. Summary statistics of the MSE Loss by Cities fitted on DAT

Model	Amsterdam (USA)	Cincinnati	Dallas	Las Vegas	Minneapolis	New York	Sacramento
NN	0.0087	0.006	0.006	0.0081	0.006	0.006	0.0082
RNN	0.0087	0.006	0.0062	0.0082	0.006	0.006	0.0082
LSTM	0.0085	0.006	0.0061	0.0084	0.0061	0.0062	0.0083
DLSTM	0.0087	0.0061	0.0061	0.0083	0.006	0.006	0.0084
D-DLSTM	0.0087	0.006	0.006	0.0082	0.006	0.006	0.0086