CSE5523 Final Project Report

Hui-Jun Chen

Contents

1 Introduction

Economics has a long tradition in processing data that holds a different view compared with Computer science. Applied Economists dig into data to find the correlation and causality between different factors, while Computer scientists have developed Machine Learning to predict out-of-sample data points. In other words, Economists try to interpret data in the past to understand the causality, while Computer scientists try to make future predictions based on the pattern existing in the data. This report aims to investigate the effectiveness of the recurrent neural network to replicate the interpretation power that linear regression can provide, and also the preciseness of the linear regression in out-of-sample prediction. My result shows that (1) linear interpretation is good within-sample interpretation but performs devastatingly in out-of-sample prediction, (2) trained with long enough epochs, the recurrent neural network can reach the interpretation power of linear regression, but maintaining good enough out-of-sample predictions power requires a high number of layers, and (3) Both `GRU` and `LSTM` performs generally better than `SimpleRNN`, and the design of reset gate in `GRU` can prevent noise from outliers, while the outcome generated by `LSTM` exacerbates with outliers.

The dataset I am using is yearly-based time-series data from Lanteri (2018). The dataset is from *Aircraft Values*, a UK-based consulting company that specialized in aircraft trading, and contains transactions of used aircraft between different companies. The sample size of this dataset is 8597, and contains variables such as:

- the log of the used aircraft price,
- the log of the used aircraft price deflated using Hodrick–Prescott filter,
- year dummy for when the transaction happens,

- the model type dummy for transacted aircraft, totally 38 types.
- the age of transacted aircraft, and
- interaction term between aircraft age and model for each transaction.

One big drawback about this dataset is the lack of labeling. All the variable labels I've written above are gained from a one-line comment in the replication code for Lanteri (2018), downloaded from the website of American Economics Review. Therefore, it is pretty hard to interpret the correlation or causality between variables. For instance, I can only label the model type dummies as type 1 to 38, rather than counterparts that can be found in the real world. As a result, it is difficult to do traditional interpretation in Economics and understand the correlation between variables. Another disadvantage is that the small sample size and limited variables. Based on econometric theory, the sample size of 8597 is a big enough number for the central limit theorem to hold and to make an interpretation, but not large enough to make meaningful training for the Machine learning algorithm. Therefore, combined with two drawbacks, instead of taking this project as the empirical result for my third-year paper, I've downsized the scope of this project to just compare the interpretation and prediction power of linear regression and recurrent neural network.

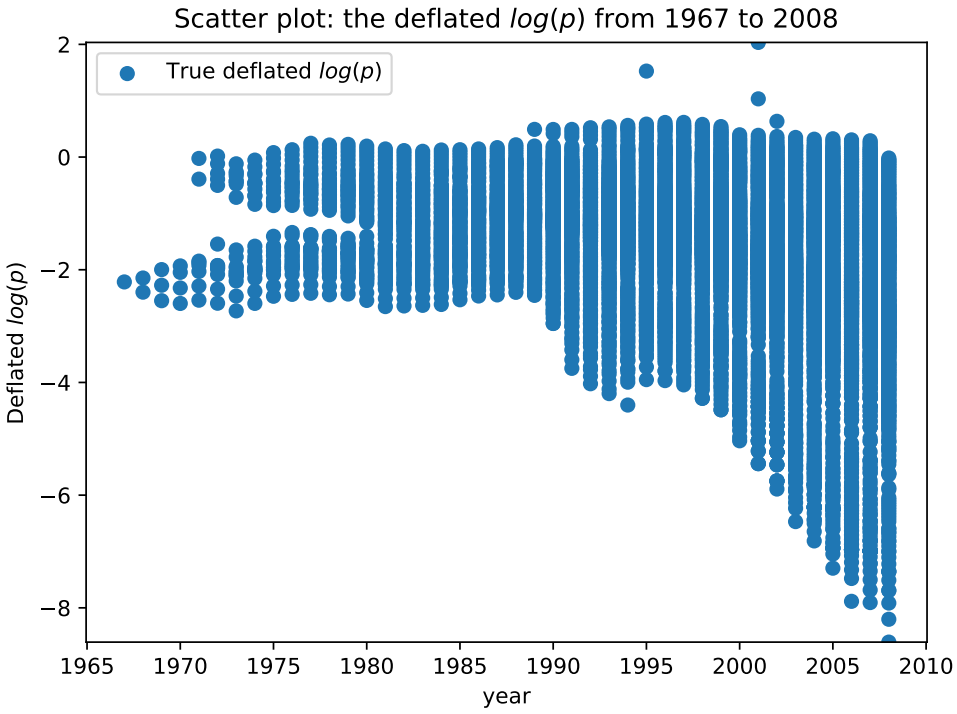
2 Methodology

For linear regression, I am using the `linear_model` module in `sklearn` package. For the recurrent neural network, I am using `keras` API in `tensorflow` package as the main tool for implementing the recurrent neural network. To investigate the interpretation power of both linear regression and recurrent neural networks, I use all of the samples. Since this is a time-series dataset, randomly splitting the data into the training set and test set would possibly destroy the time series structure of this data, I've separated the first 80% of the sample as the training set and the last 20% of the sample as testing set after sorting the whole dataset by year. I set the batch/window size to 10. To systematically test out the effect of epochs, hidden layer number, and model choice, I've trained 12 models: 25 and 100 epochs, 1 and 4 hidden layers, and three model choices as `SimpleRNN`, `LSTM` and `GRU`. Each hidden layer is activated by `ReLu`, and all models have only 1 dense layer, activated by `linear`. For models with only 1 hidden layer, the hidden unit is 32; for models with 4 hidden layers, the hidden units for each layer are $[128, 64, 32, 16]$. Since `linear_model` naturally use `mean_squared_error` as loss function, I applied the same loss function when implementing all three recurrent neural network models. For evaluation metrics, I am using R^2 for in-sample interpretation and `mean_absolute_error` for out-of-sample prediction.

3 Result

Figure 1 shows the distribution of used aircraft transactions in each year. As shown in the figure,

Figure 1: Distribution for deflated $\log(p)$ of used aircraft transactions

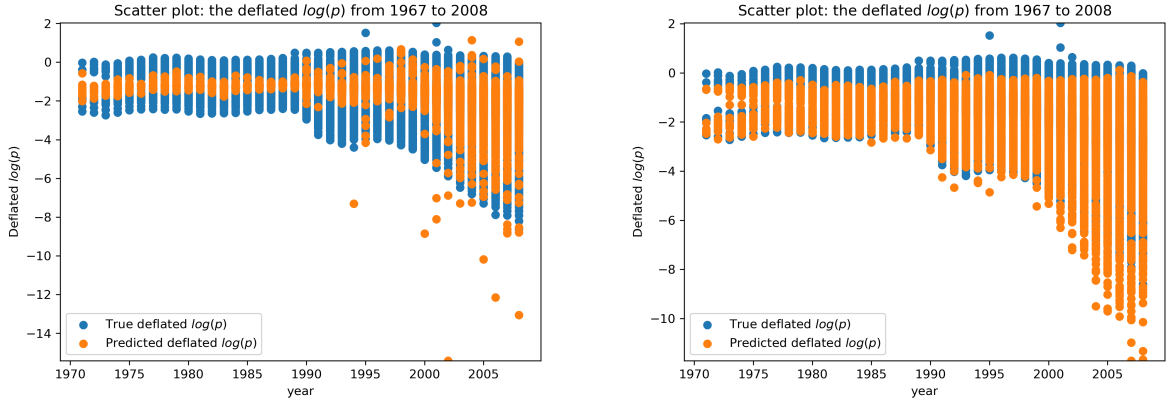


we have uneven numbers of data between the first 10 years (1967 to 1976) and the last 10 years (1999 to 2008). This phenomenon is pretty common in time-series data in Economics. Not only the quality of data collection in the early year is problematic, but the sample size is going to grow with the expansion of the airline industry. However, I conjecture that such uneven nature might affect the quality of training if I am only using the first 80% of data points for training. Whether my models in the recurrent neural network also learns the prosperity of the airline industry requires further investigation that I cannot finish due to time constraint.

3.1 Result on Interpretation Power

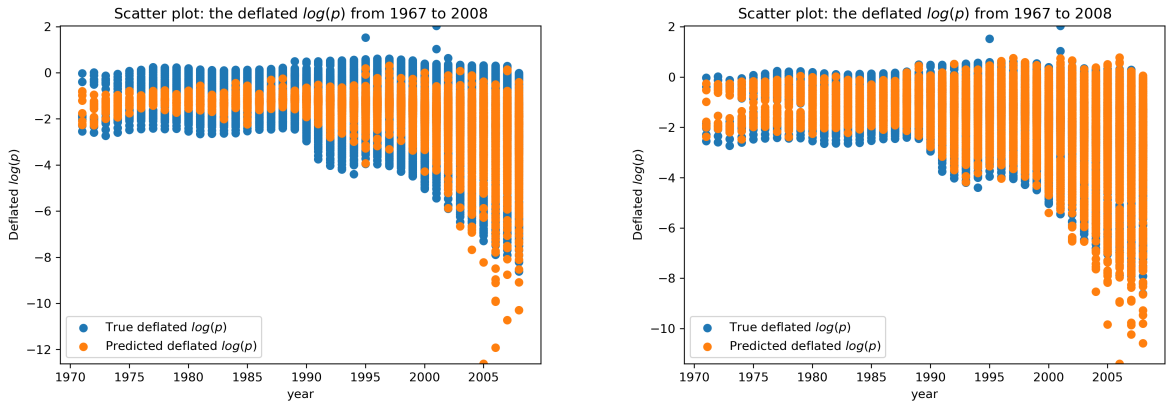
Figure 2, 3, 5 and table 1 summarized the result in the interpretation power. From figure 2 and table 1, higher layer number leads to higher R^2 and Adjusted R^2. The R^2 increased from 0.2751 in 1 layer `SimpleRNN` to 0.4512 in 4 layer `SimpleRNN`. The `LSTM` model in figure 3 combined with the 4 layer `LSTM` rows in table 1 shows that the longer the training, better the prediction power. However, counterintuitive result appears when we compare 4 and 1 layer `LSTM` row in table 1. For 1 layer `LSTM`, the longer the training epochs, the lower its interpretation power, i.e., lower R^2 and adjusted R^2. From figure 4, it is pretty clear that the combination of long-term and short-term memory of patterns make more extreme predictions due to some outliers in the true deflated $\log(p)$. I conjecture that the combination of long-term and short-term memory can be easily affected by

Figure 2: Effect of layers: `SimpleRNN` as example



(a) 1 layer `SimpleRNN` **(b)** 4 layer `SimpleRNN`

Figure 3: Effect of epochs: 4 layers `LSTM` as example



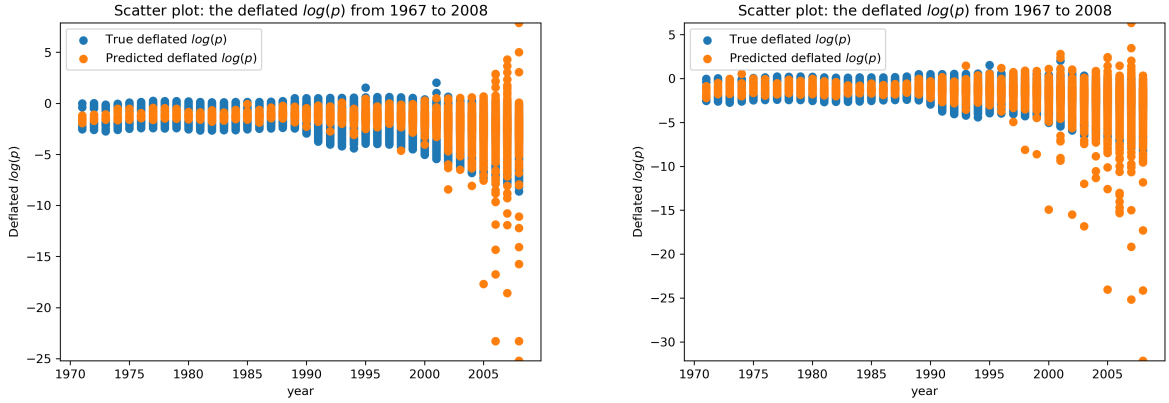
(a) epoch 25 **(b)** epoch 100

the outliers in the data, causing the loss of interpretation power. Such drawbacks do not appear in `GRU`, I believe that the addition of reset gate in `GRU` is effective in preventing overinterpretation of the data, as shown in figure 5.

3.2 Result on Out-of-sample Prediction Power

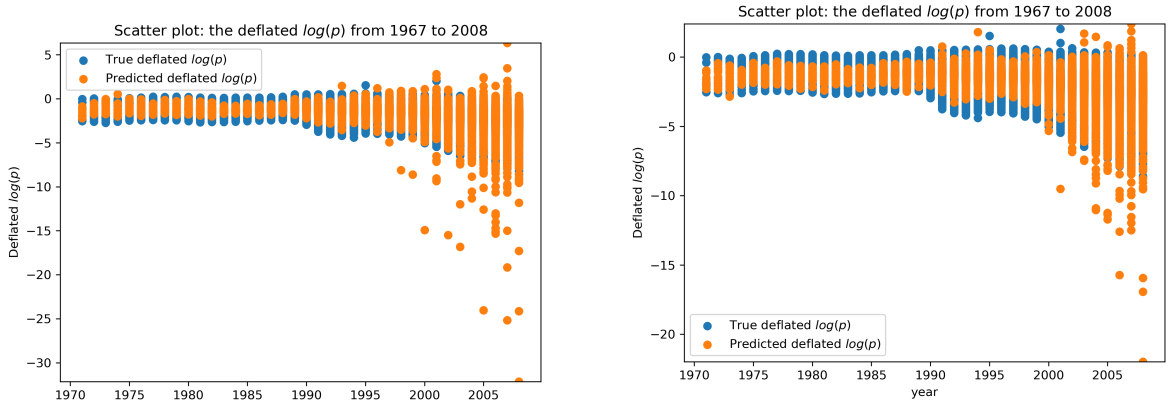
Since my metrics is `mean_absolute_error`, the lower the better. For the out-of-sample prediction, the first thing I observed from table 2 is how terrible the `linear_model` performs. It misses the true value by over 10^{10} times. It also makes plotting the figure for both true value and predicted value become meaningless, as shown in appendix figure 16. As usual, increases in the number of layers decrease the `mae`, and thus raise the preciseness of my prediction. What's counter-intuitive is that the increase in the number of epochs from 25 to 100 raise the error in most cases of recurrent neural network. One might blindly think that 100 epochs might be the time when the error is temporarily rising, and falsely guess that this phenomenon will disappear when training with long enough epochs. However, the combination of small numbers of layers and large epochs is dan-

Figure 4: Effect of epochs: counterexample with 1 layer `LSTM`



(a) epoch 25 **(b)** epoch 100

Figure 5: Model Comparison: `LSTM` v.s. `GRU` with epoch 100, layer 1

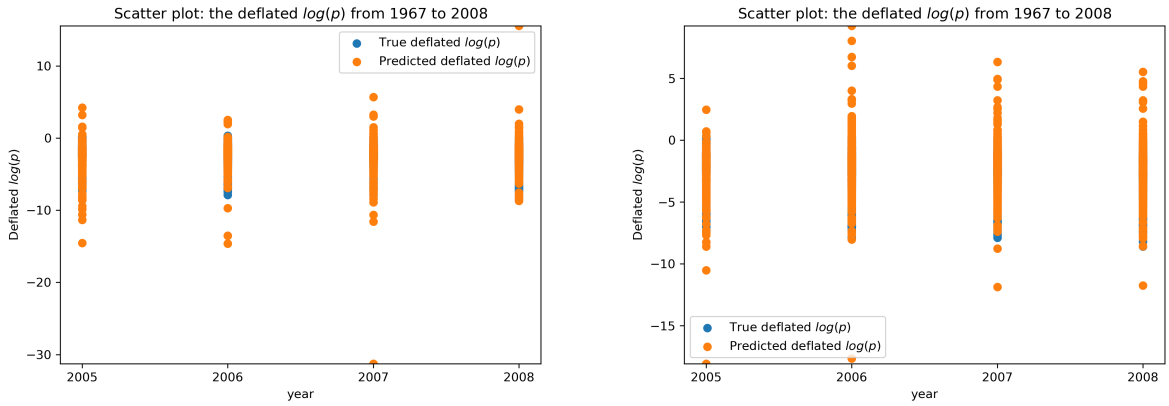


(a) LSTM **(b)** GRU

Table 1: Interpretation Power Comparison

	Epoch			
	25		100	
	R^2	Adjusted R^2	R^2	Adjusted R^2
`linear_model`	0.9632	0.9627	0.9632	0.9627
1 layer `SimpleRNN`	0.2751	0.2648	0.5068	0.4998
4 layer `SimpleRNN`	0.4512	0.4434	0.8887	0.8871
1 layer `LSTM`	0.4586	0.4510	0.3755	0.3667
4 layer `LSTM`	0.5334	0.5268	0.9338	0.9329
1 layer `GRU`	0.4357	0.4277	0.6040	0.5983
4 layer `GRU`	0.6063	0.6007	0.9335	0.9325

Figure 6: Prediction comparison with low layer number and high training epochs: LSTM



(a) epoch 25, 1 layer

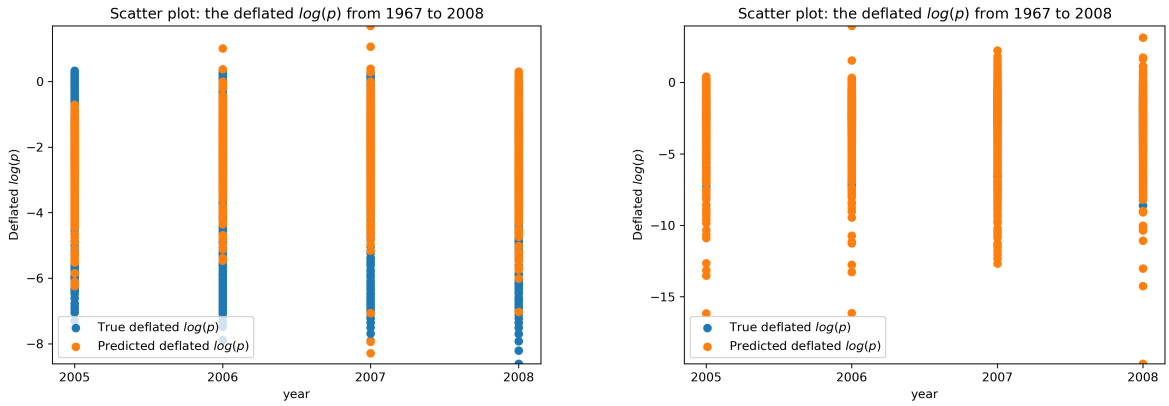
(b) epoch 100, 1 layer

gerous. As shown in figure 6, long training epochs in LSTM can generate over -35 of prediction on $\log(p)$ when the lowest true value is only -8, and can generate over -25 of prediction in GRU (figure 7). Furthermore, in figure 8 and 9 downloaded from tensorboard, the loss function and the mae metrics of 1 layer GRU fluctuates consistently after epoch 30, and the scale and volatility are larger than the 4 layers counterpart. Therefore, it is the numbers of layers, or the design of the neural network itself decides the performance of the model, which cannot be improved by longer training epochs.

Table 2: Out-of-sample Prediction Power Comparison

	Metrics: mae	
	Epoch	
	25	100
linear_model	12023313799.8250	12023313799.8250
1 layer SimpleRNN	1.4788	1.6571
4 layer SimpleRNN	1.8899	1.5920
1 layer LSTM	1.6038	2.0776
4 layer LSTM	1.4790	1.5889
1 layer GRU	1.5390	2.1386
4 layer GRU	1.5272	1.5931

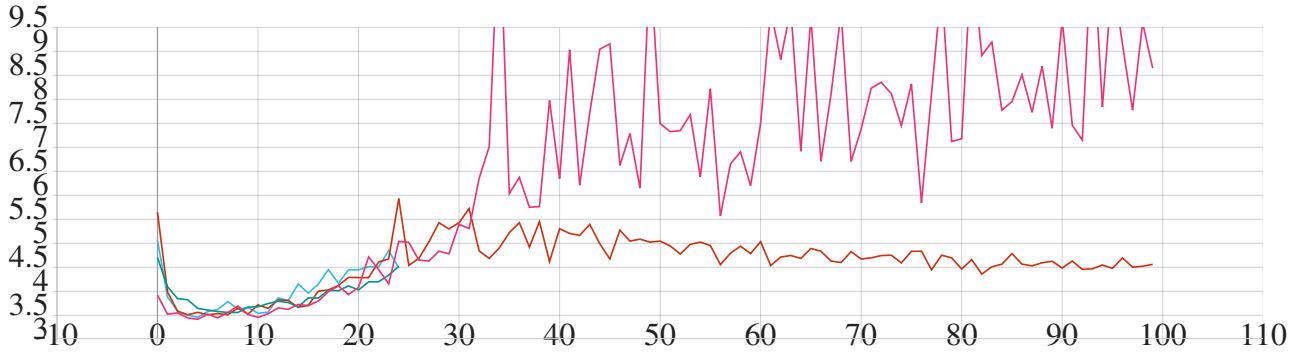
Figure 7: Prediction comparison with low layer number and high training epochs: GRU



(a) epoch 25, 1 layer

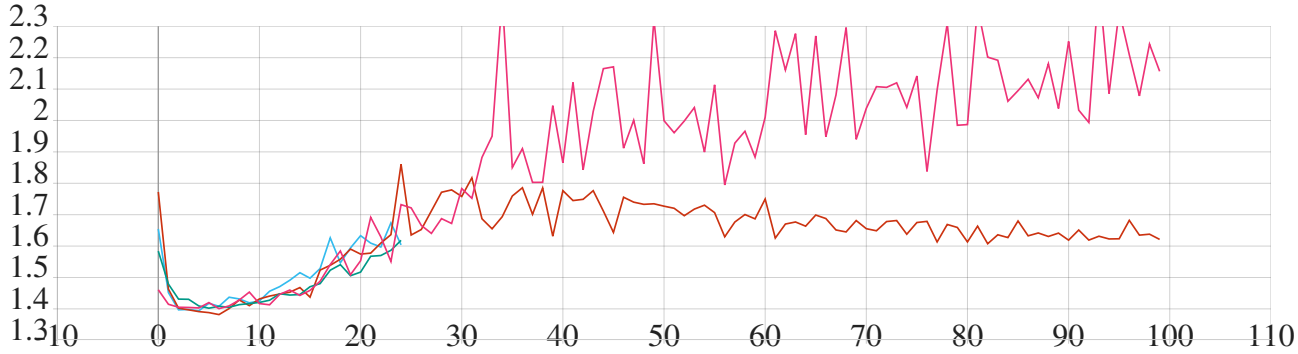
(b) epoch 100, 1 layer

Figure 8: Loss function per epoch: GRU



pink: 1 layer GRU with 100 epochs; brown: 4 layers GRU with 100 epochs; green: 1 layer of GRU with 25 epochs; cyan: 4 layers of GRU with 25 epochs

Figure 9: mae metrics per epoch: GRU



pink: 1 layer GRU with 100 epochs; brown: 4 layers GRU with 100 epochs; green: 1 layer of GRU with 25 epochs; cyan: 4 layers of GRU with 25 epochs

4 Conclusion

In this report, I've compared the linear regression and three recurrent neural network models and see their performance on in-sample interpretation power and out-of-sample prediction power. I've found that the power of the recurrent neural network can replicate the interpretation power that the linear regression has, and also maintain relatively well in the out-of-sample prediction. However, all details in training a neural network are critical. The combination of lower layers of network and high training epochs can generate a large deviation from the true value since the model has learned the pattern of outliers. The architecture and design of the network directly determines the performance, regardless of the training epochs. `GRU` performs generally better than `LSTM`, and I conjecture that the design of reset gate decreases the effect of long-term memory that `LSTM` tries to capture but unfortunately amplified.

The limitation of this report mainly lies in the limitation on this dataset. The relatively small sample size makes training and testing faster, but also left the question that whether the above observations hold in big data. Also, the number of data points available in the beginning period is very different compared with the end period, creating difficulties for the recurrent neural network to learn based on the pattern in the beginning period. In the future, I hope I can use big and balanced data to check whether the above observations hold.

A Additional figures for different models

Figure 10: GRU Prediction Total layer1 epoch25 window10 loss mse

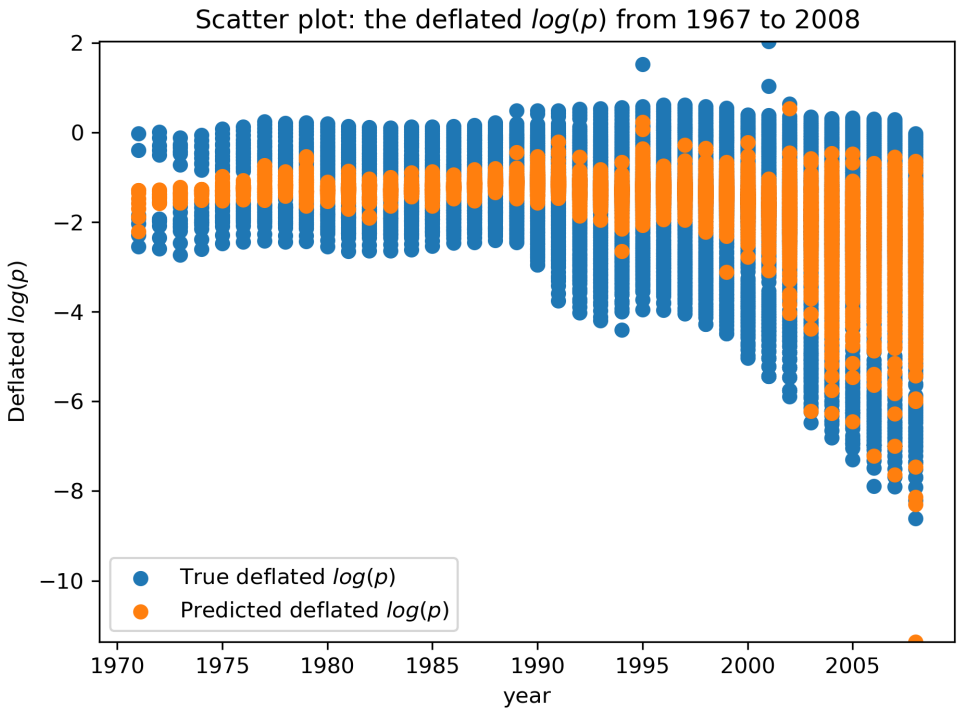


Figure 11: GRU Prediction Total layer4 epoch100 window10 loss mse

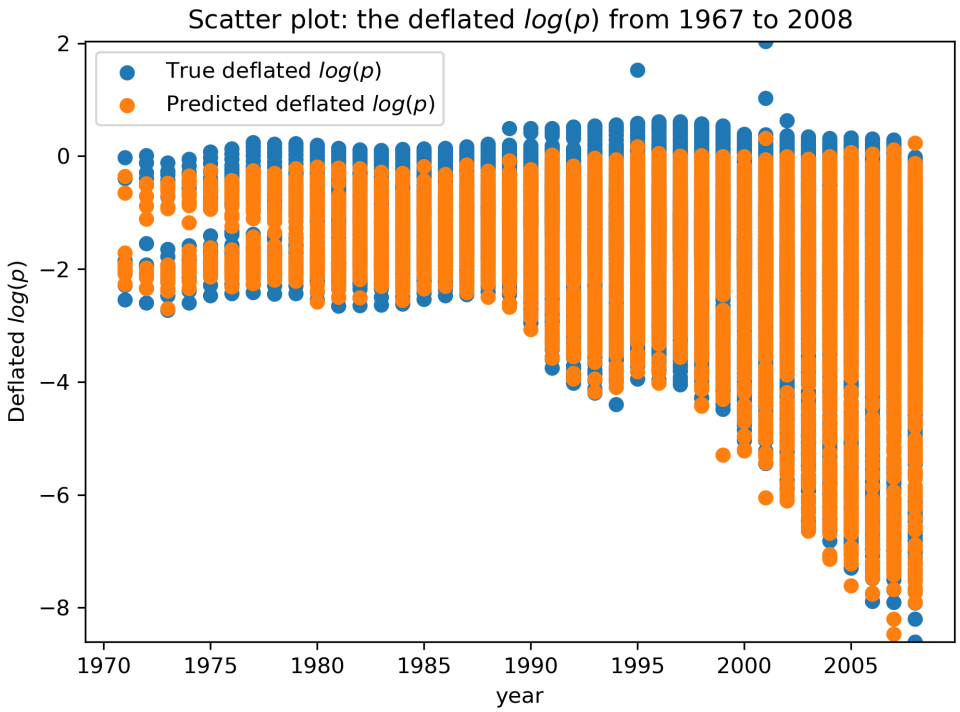


Figure 12: GRU Prediction Total layer4 epoch25 window10 loss mse

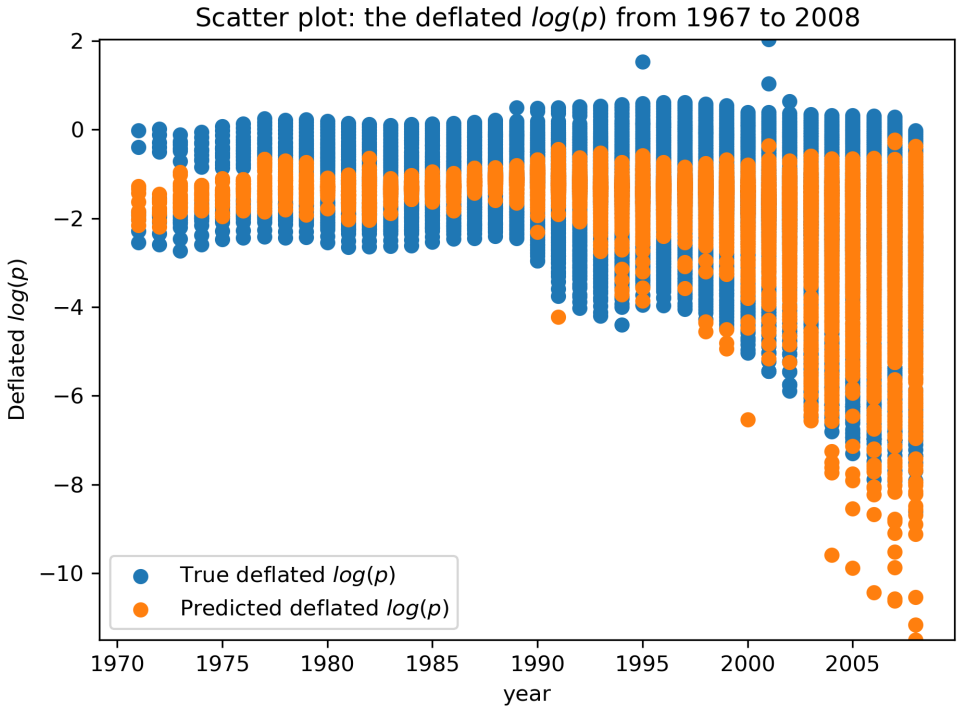


Figure 13: GRU Prediction Train Test layer4 epoch100 window10 loss mse

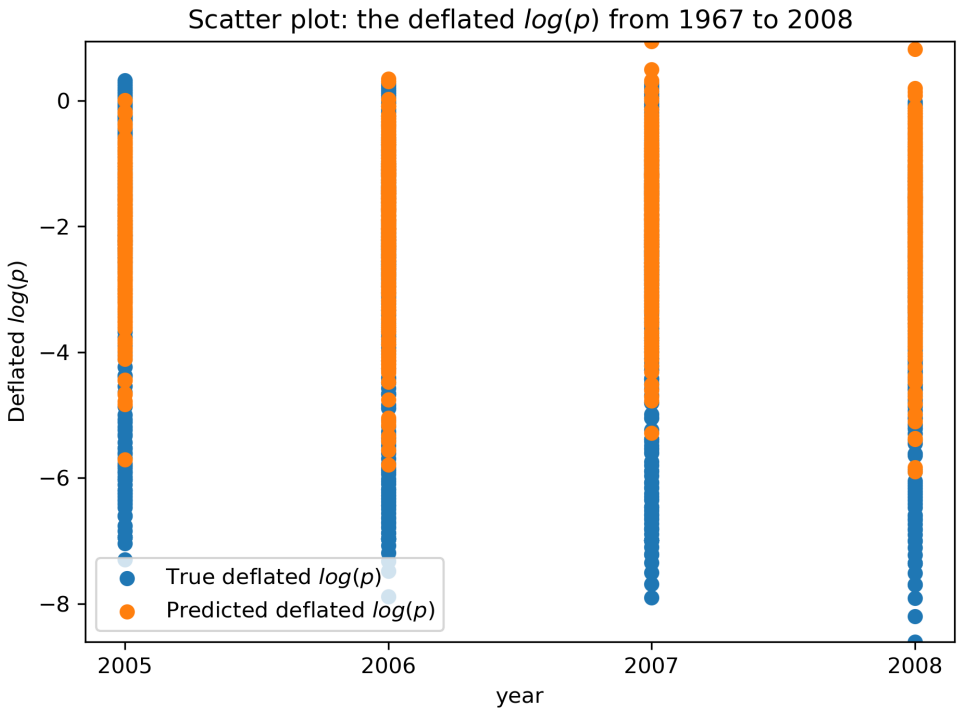


Figure 14: GRU Prediction Train Test layer4 epoch25 window10 loss mse

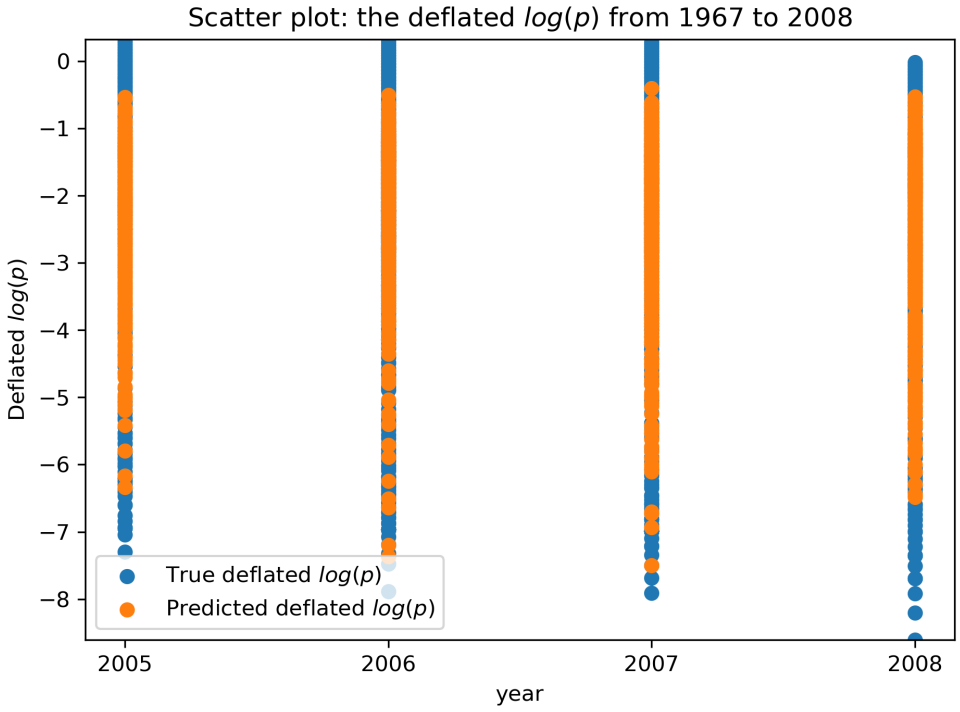


Figure 15: LinearRegression Prediction Total

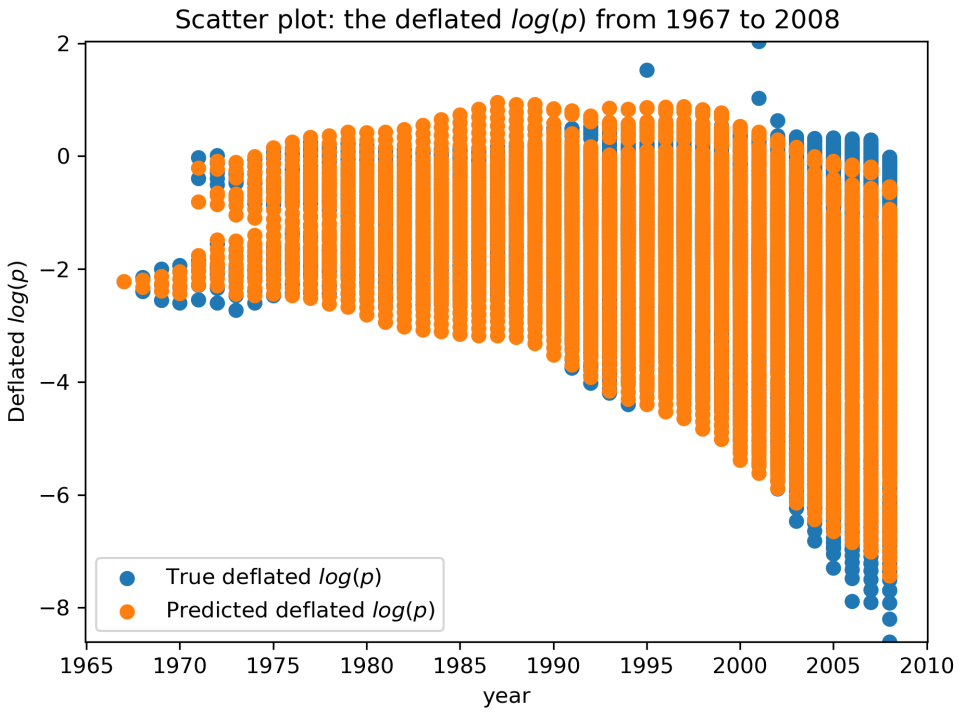


Figure 16: LinearRegression Prediction Train Test

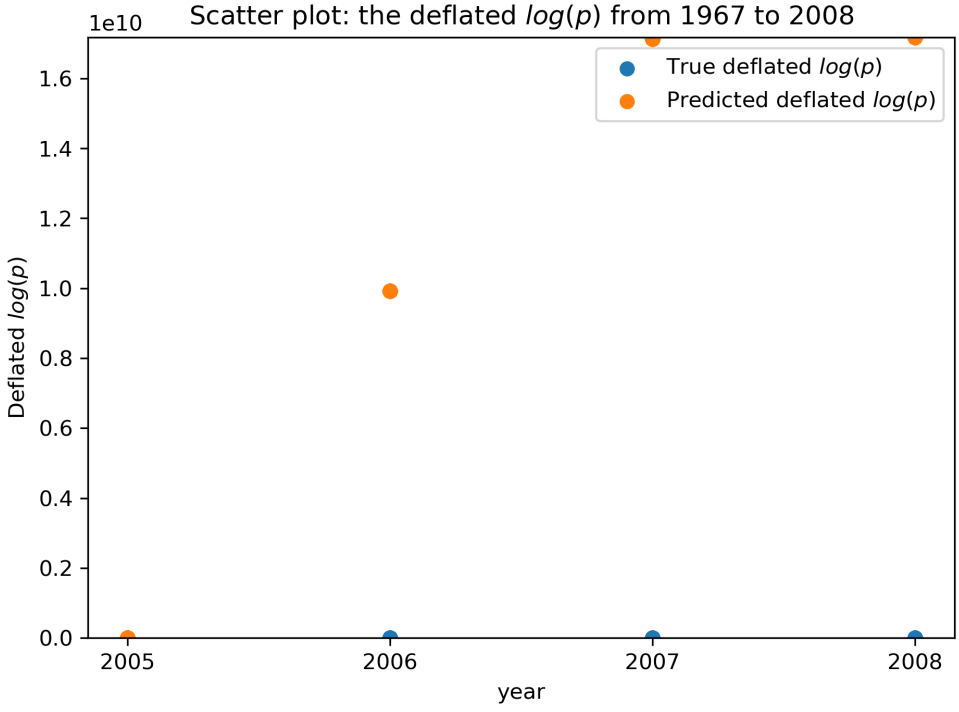


Figure 17: LSTM Prediction Train Test layer4 epoch100 window10 loss mse

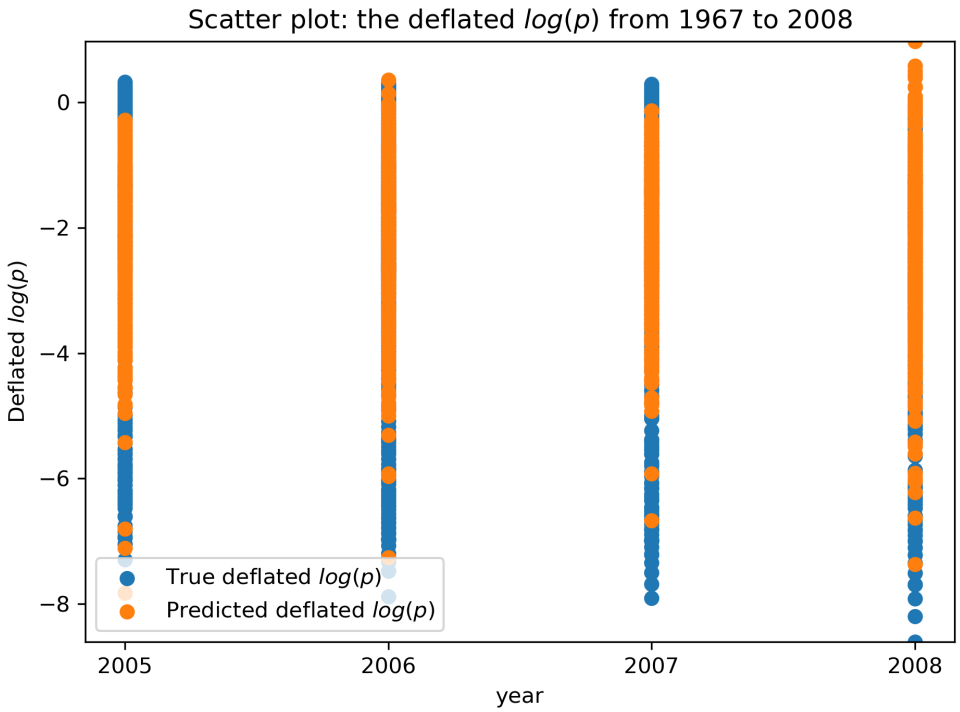


Figure 18: LSTM Prediction Train Test layer4 epoch25 window10 loss mse

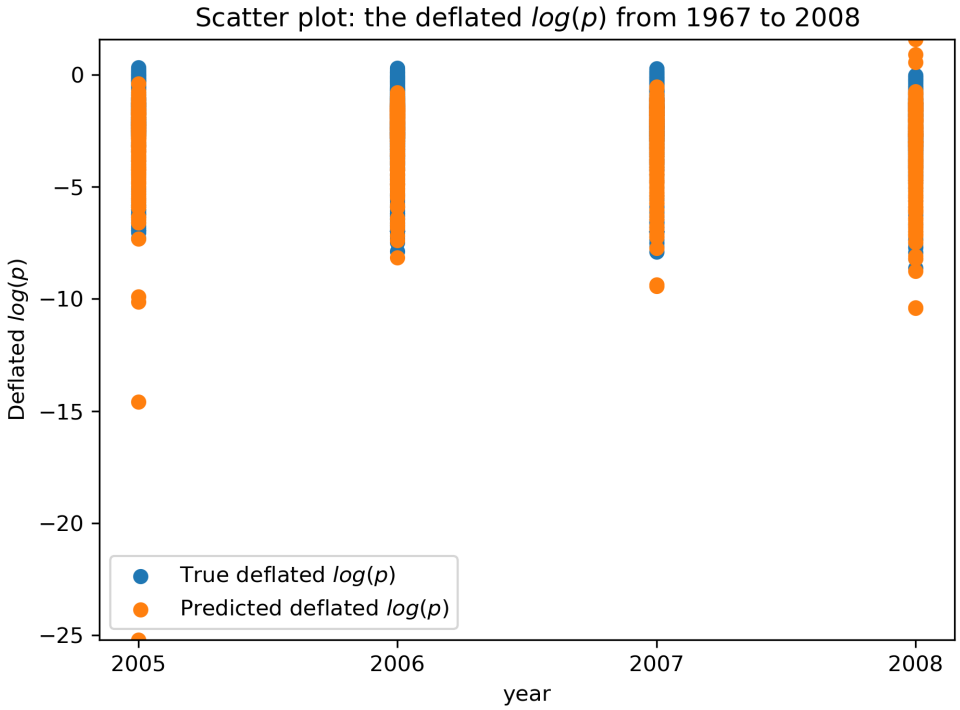


Figure 19: SimpleRNN Prediction Total layer1 epoch25 window10 loss mse

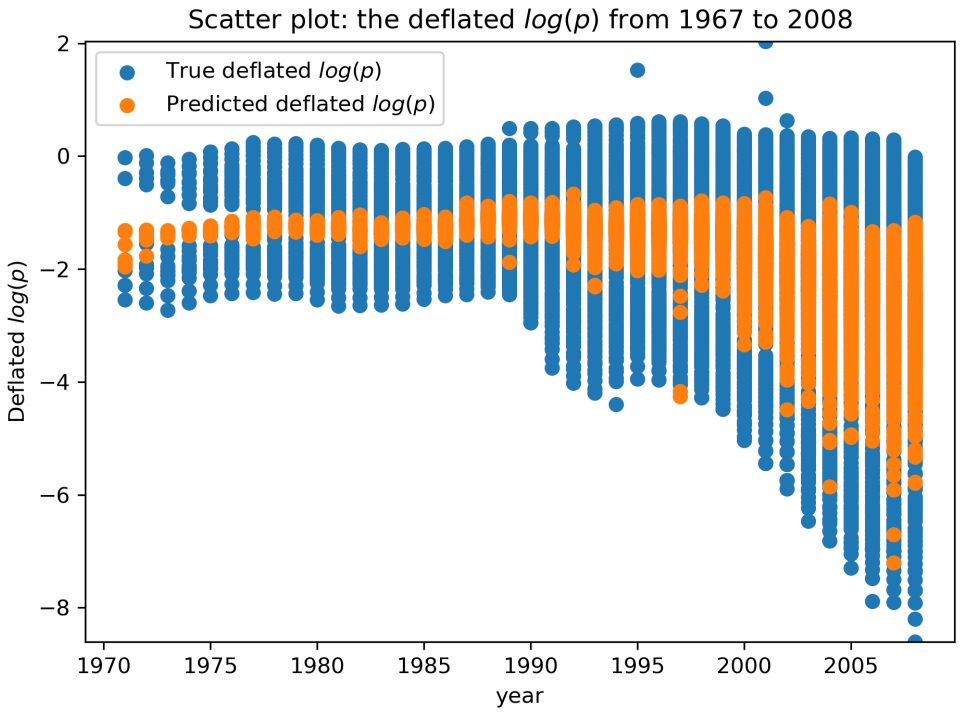


Figure 20: SimpleRNN Prediction Total layer4 epoch25 window10 loss mse

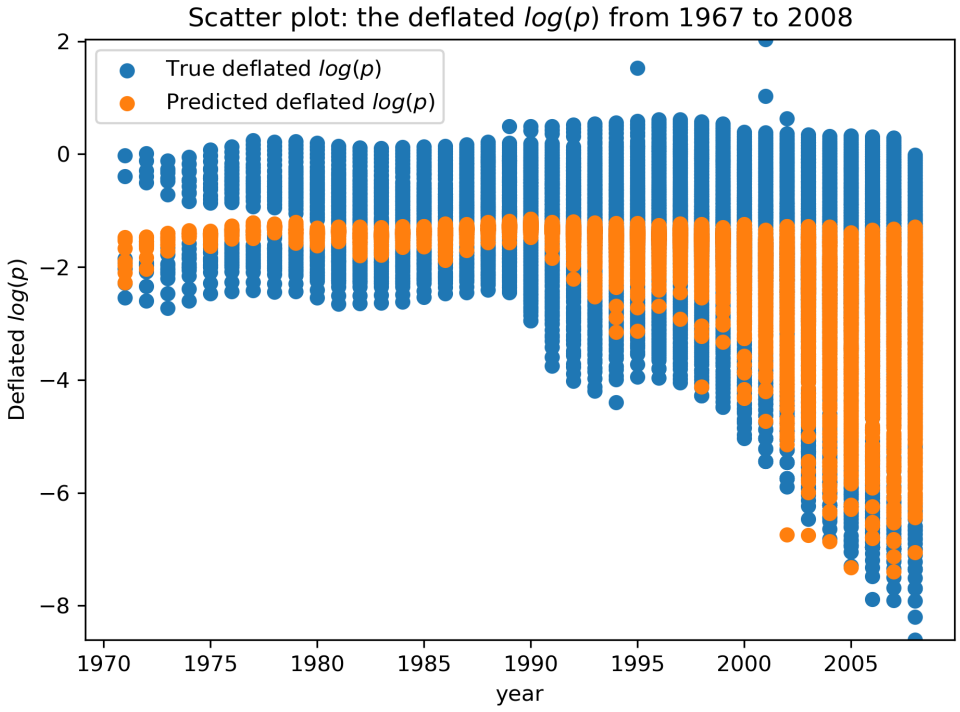


Figure 21: SimpleRNN Prediction Train Test layer1 epoch100 window10 loss mse

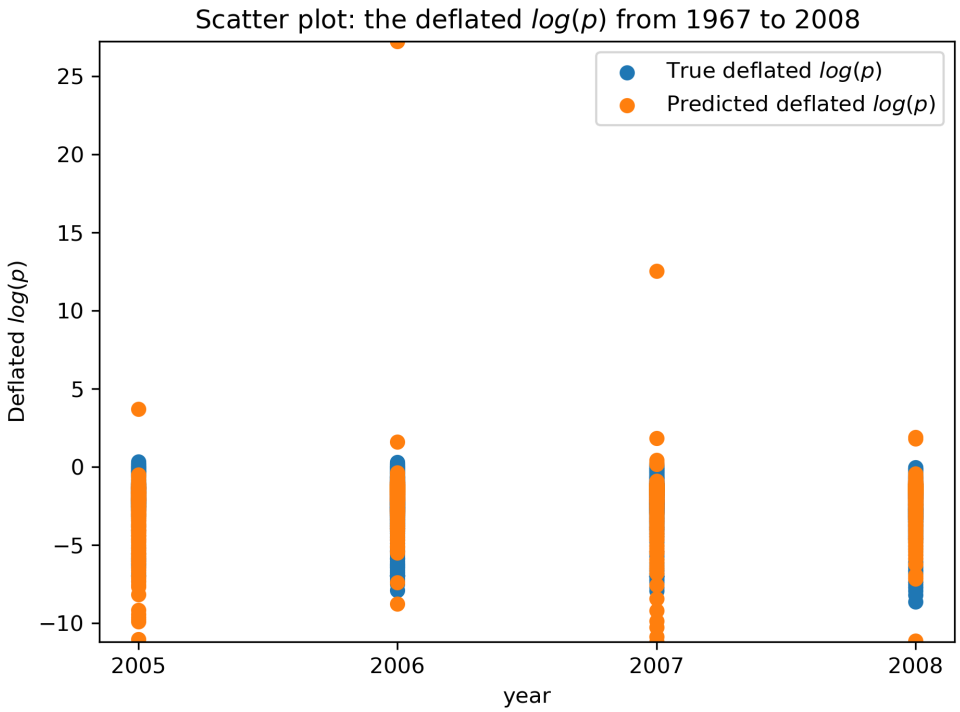


Figure 22: SimpleRNN Prediction Train Test layer1 epoch25 window10 loss mse

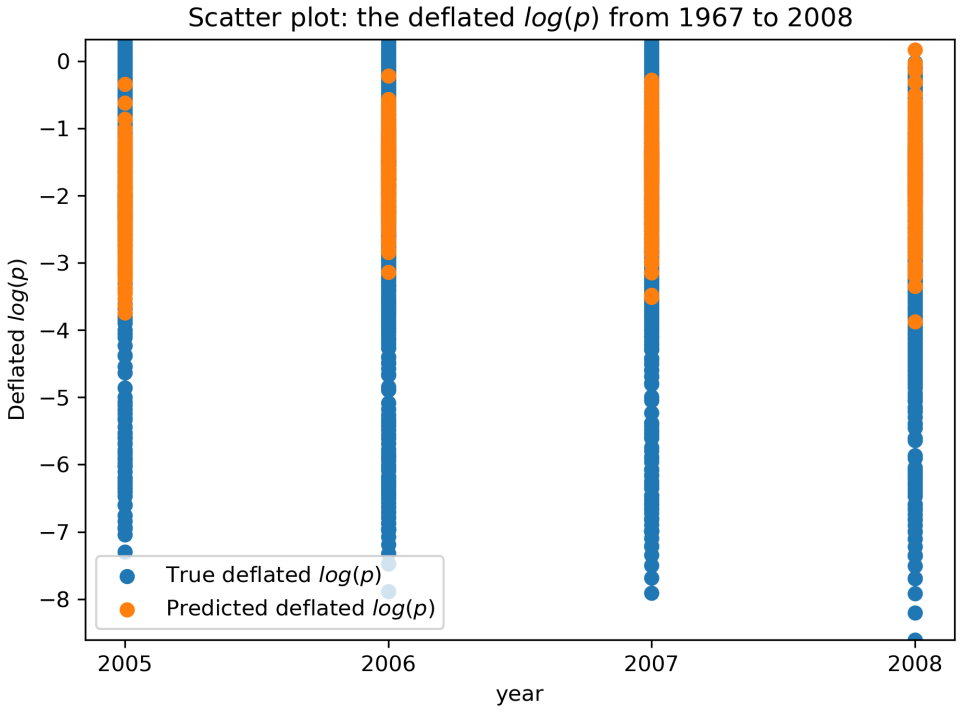


Figure 23: SimpleRNN Prediction Train Test layer4 epoch100 window10 loss mse

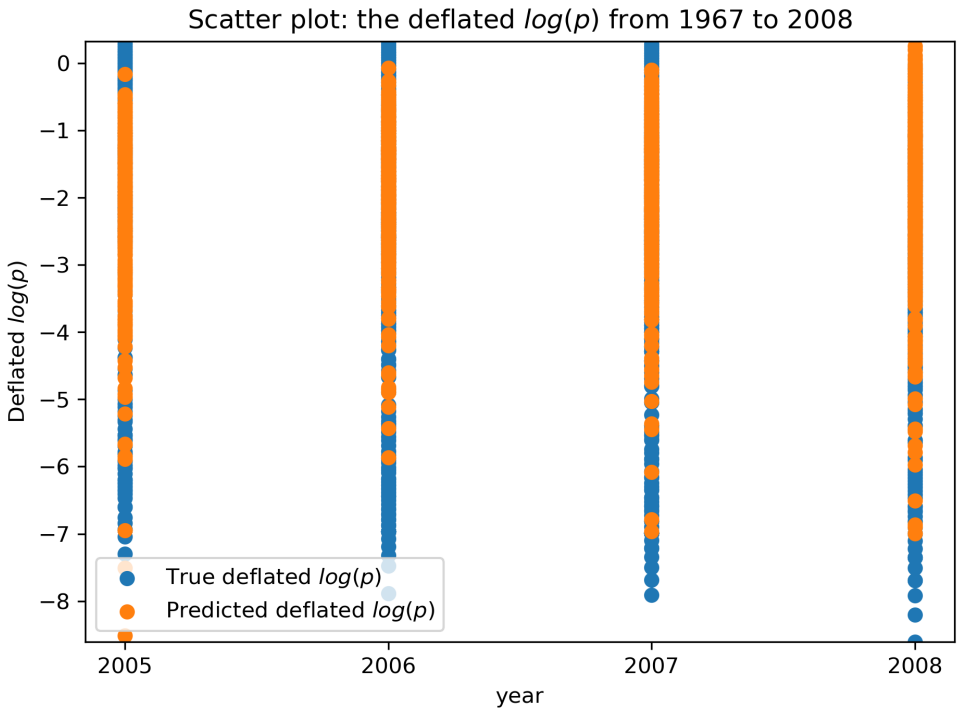
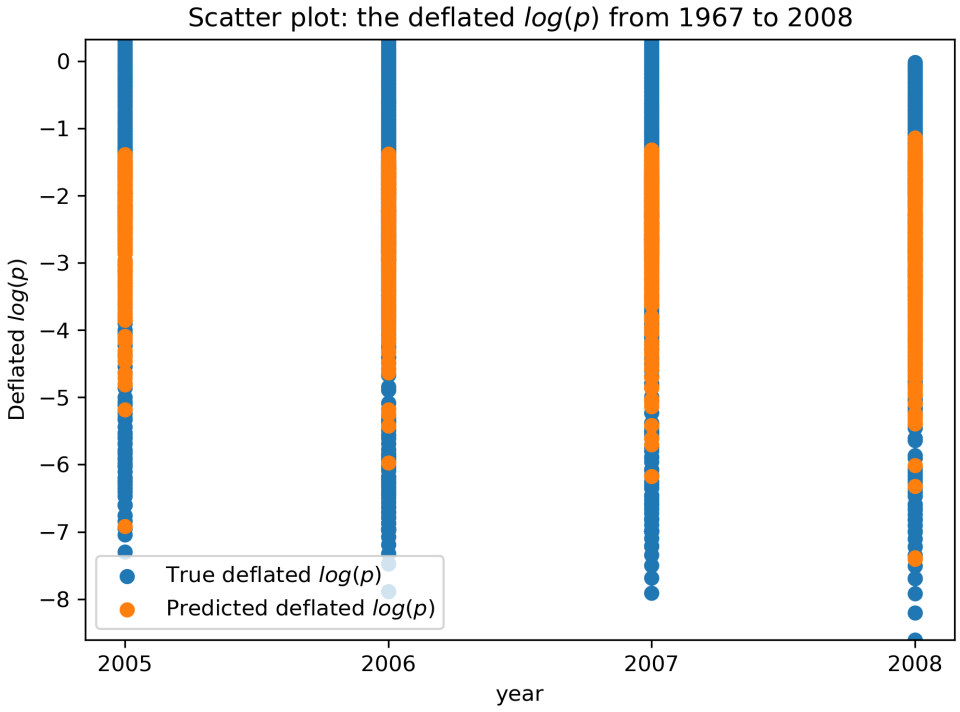


Figure 24: SimpleRNN Prediction Train Test layer4 epoch25 window10 loss mse



References

Hodrick, Robert J. and Edward C. Prescott (Feb. 1997). "Postwar U.S. Business Cycles: An Empirical Investigation". In: *Journal of Money, Credit and Banking* 29.1, p. 1. ISSN: 0022-2879.

Lanteri, Andrea (Sept. 2018). "The Market for Used Capital: Endogenous Irreversibility and Reallocation over the Business Cycle". In: *American Economic Review* 108.9, pp. 2383–2419. ISSN: 0002-8282.