Google summer of code - student proposals
Correlations between OMNI data and Dst index

Nichita Diaconu

1 Task

For those students who want to go the extra mile, this is a challenging task and involves getting both IMF and Dst index data (read tasks 1,2). The IMF plays an important role in controlling the space weather. In particular, the “z” component of IMF (denoted by “Bz” in the data) has a significant impact on the Dst index. Students can work on projects that show insightful relations between Dst index and IMF. For those interested in machine learning you may be even develop a predictive models.

1.1 Data

The data is extracted in get_data.ipynb. More specifically, the dst data is downloaded and saved to files. To acquire the dst data I used davitpy:https://github.com/vtsuperdarn/davitpy the script, plotGMagIndices. Then the data from 2000 to 2016 is pickled to a file. Even though there was a similar solution for the OMNI dataset, it did not work as well. Therefore, we made a query in the page https://omniweb.gsfc.nasa.gov/form/dx1.html and downloaded the data. Then the data is again pickled to a more convenient format.

2 Correlation

In order to find relations between the Dst and Bz indexes we computed the covariance and between the two variables and also Pearson correlation.

Covariance:

	Bz	Dst
Bz	9.38780097	19.52207619
Dst	19.52207619	431.85933148

Pearson correlation:

	Bz	Dst
Bz	1	0.30660054
Dst	0.30660054	1

As we can see, the covariance values cannot tell us very much, so we normalize them and get the Pearson correlation. This tells us about the linear dependence between the two variables. We can see that there is significant linear dependence, 0.3. We can also see from the bare covariances that Dst varies a lot more than Bz. Now we want to see whether this dependence and variance is enough to learn some predictive model of the Dst.

3 Predictive Model

3.1 Support Vector Regressor

We first approach the Support Vector Regressor as a model because of its known capability. Because we know that the data follows some Markovian assumption, but we do not know how many of the past values are required in order to predict the next value, we do a hyperparameter search. We evaluate our algorithm using mean squared error on unseen data during training.

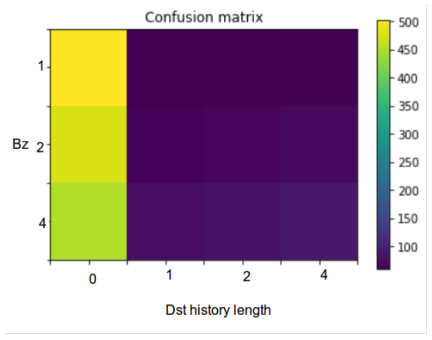

Fig.1 Confusion matrix of mean squared error values of the searched values for Bz history length and Dst history length to be used as features.

We found the best values to be 1 past Bz value and 2 past Dst values. This technique with these parameters when trained for longer and then tested again on a new set of unseen data, achieves 17 mean squared error.

3.2 LSTM

We want to see if we can do better so we trained an LSTM cell, which is a special type of Neural Network, that specializes on time series and is more prone to learning long term dependencies than a simple RNN.

We search for a good number of LSTM neurons in the values of 32, 4, 1. We find the following root mean squared errors on the test set: (This time we computed root mean squared error, as opposed to mean squared error in SVR, so in that way values are expected to be lower. We did this just to choose the best LSTM model. We will evaluate the best one in the end with the mean squared error, in order to compare it with the SVR)

Neurons	RMSE
32	9.311
4	9.880
1	13.406

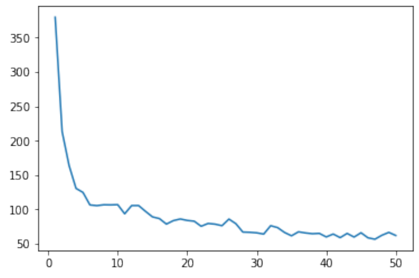

Fig.2 Loss history of the LSTM with 32 neurons

MSE of 32 neurons LSTM: 82. This is not as good as the SVR, but the SVR was trained on 50000 samples, while the LSTM was trained for 50 epochs on only 10000 samples. A longer run on the same data will achieve better performance.

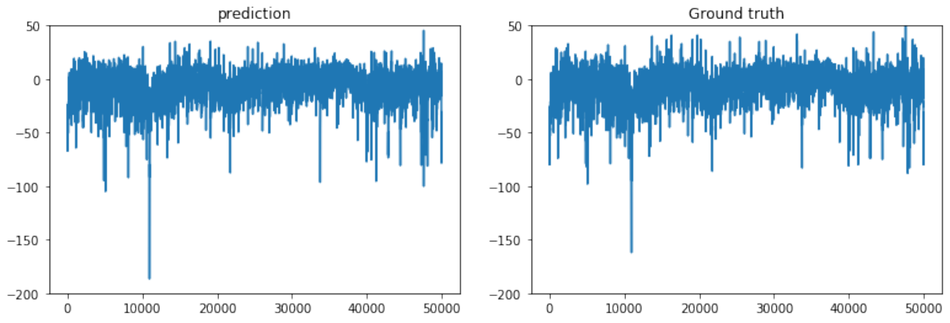

Fig.3 A sample prediction(left) vs ground truth (right)

4 Conclusion and further work

We believe this results show that the Dst can be predicted quite accurately and that with more training time, the LSTM can perform even better. Other features from the omni dataset could also be included for a better performance.