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

I noticed that the covariance values cannot tell us absolute things, so we normalize them and get the Pearson correlation. This measures the linear dependence between the two variables. I can see that there is significant linear dependence, 0.3. I can also see from the raw 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

I first approach the Support Vector Regressor as a model because of its 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. I evaluate my algorithm using mean squared error on unseen data during training.

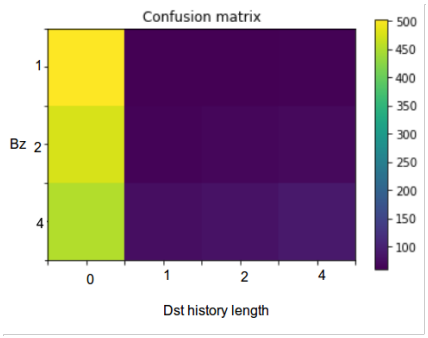


Fig.1 Confusion matrix of mean squared error values of the searched values for Bz history length and Dst history length to be used as features.

I 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.

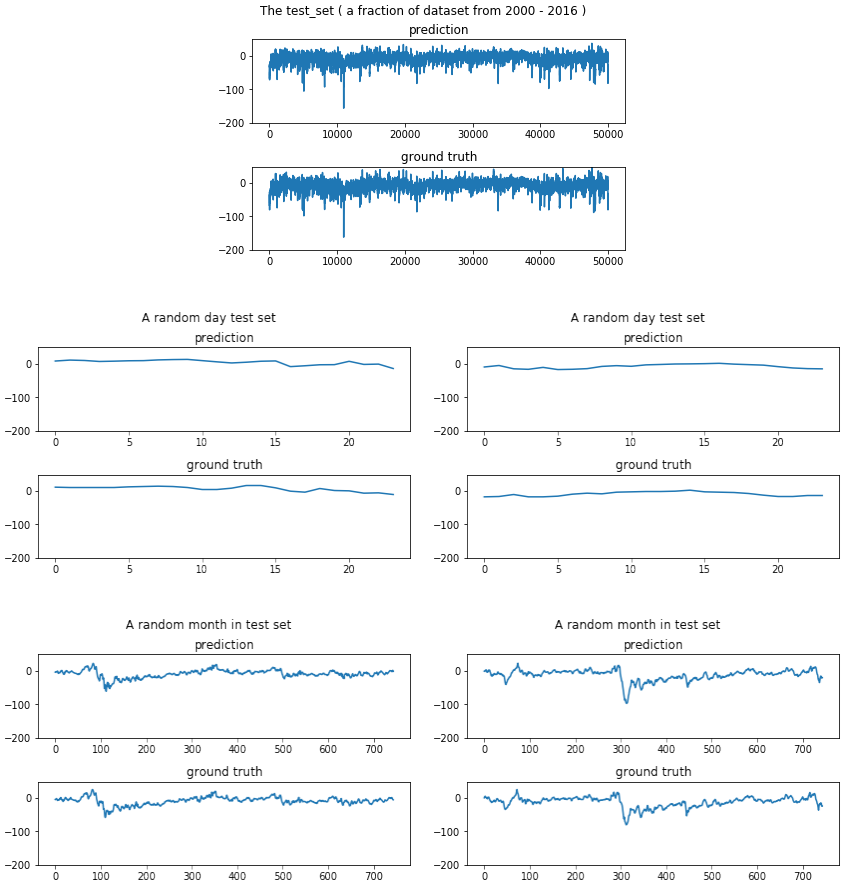


Fig. 2 Examples from the test set.

I realize now that due to the small variance from one hour to another, by feeding the dst of the past hour as input feature, the algorithm would perform fairly well, just by predicting the same dst value.

3.2 SVR only with Bz feature

I further look into how well we can predict DST values only with Bz. I search for the number of past Bz values that SVR requires for gaining a small mean squared error. The results are shown if Fig. 3

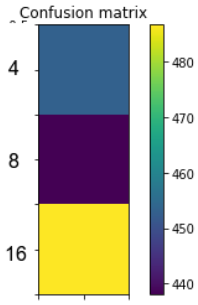
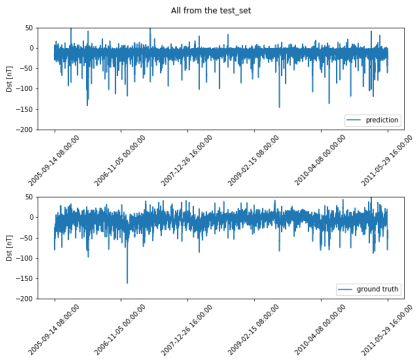


Fig. 3 Confusion matrix Bz history len. The value is Mean squared Error. We searched for history of length 4, 8 and 16.

I found length 8 to be the best length of past Bz values to be fed as input features to the SVR.

I then trained the SVR with this parameters on a bigger subset of the data and achieved 129 MSE. In Fig. 4 we present plots of predictions on the whole test set, random months and random days.



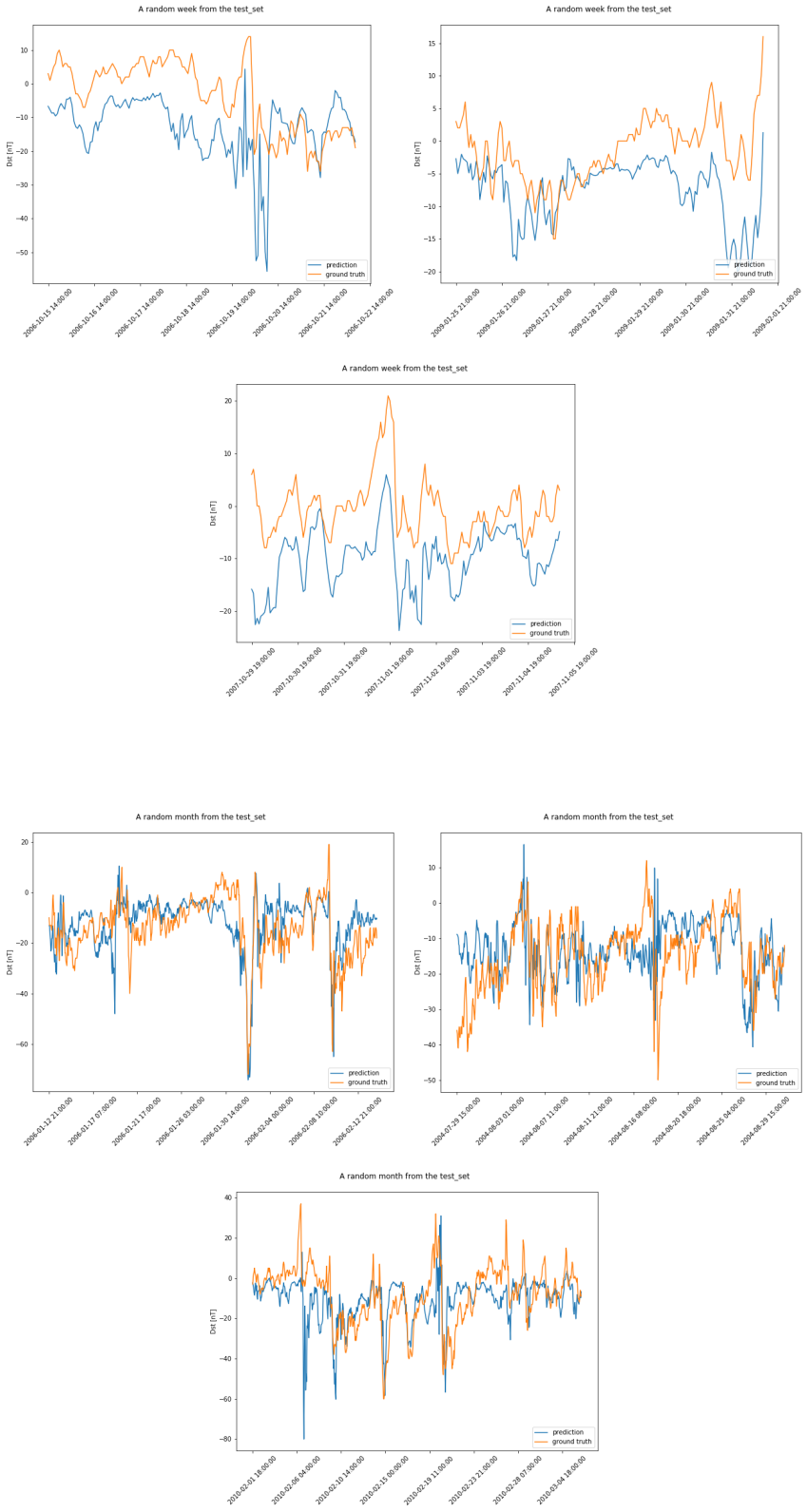


Fig. 4 Examples from the test set.

3.3 LSTM

I 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.

I search for a good number of LSTM neurons in the values of 8, 4, 16. I find the following mean squared errors on the test set:

Neurons	MSE
16	718
8	612
4	586

I then let the LSTM with 4 neurons train for longer on a bigger subset of the dataset. This process resulted in the following learning curve, Fig. 5 and a MSE of 290.

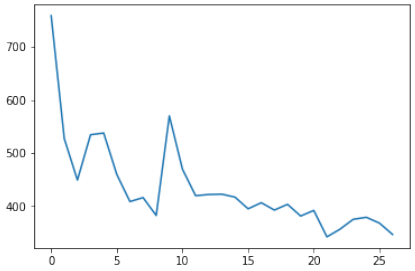
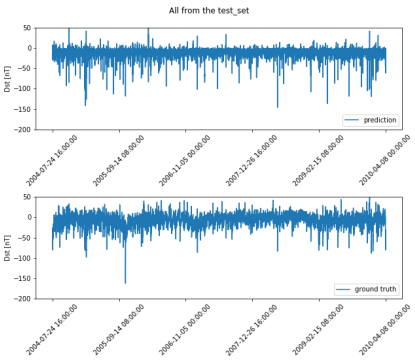


Fig.5 Loss history of the LSTM with 4 neurons

MSE of 4 neurons LSTM: 290. This is not as good as the SVR. A longer run on more data could result in the LSTM performing better than the SVM.



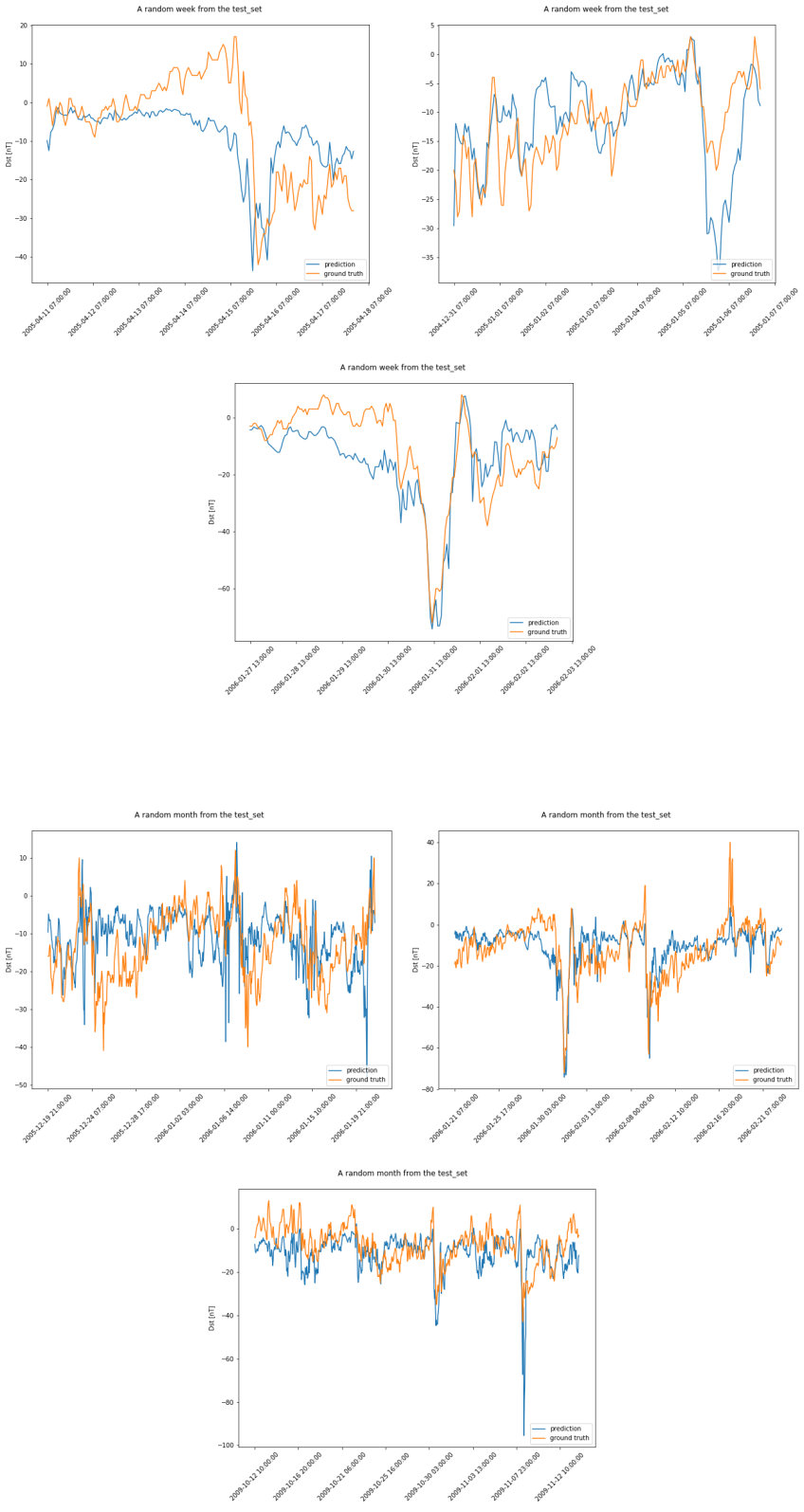


Fig. 6 LSTM 4 neurons. Examples from the test set.

4 Results

In this work, we train algorithms to predict a continuous variable. Therefore we measure mean squared error and accuracy (with a difference of at most 1) of the models on the test data:

	MSE	Accuracy
SVR with Bz features	129	7.642%
LSTM with Bz fetures	290	4.34%

5 Conclusion and further work

In this work I analyzed the relations between the DST and the Bz indexes. Moreover, we tried to create a predictive model for DST. By far the best method is to use the past DST values in order to predict the current DST values due to low variance from one hour to the next of the DST value. When trying to make a predictive model of the DST based on the Bz only, we achieved 129 MSE with SVR and 290 MSE with LSTM. The results could be further improved by training the LSTM for longer, making better use of the data, adding more than just the Bz feature in order to predict the DST.