Forecasting Plasma Instability in Fusion Simulations

DATA 322

Tamu Suttarwala

Introduction

California is home to a number of fusion research institutions, such as the LLNL in Livermore and the D3-D National Fusion Facility in San Diego. It is in San Diego that a Fusion Data Science and Digital Engineering Centre was established, in order to use machine learning to solve research problems.

Using the 1-dimensional fusion reaction model created by Dr. Ken Owens, our task is to predict instability in the system, preferably enough time before the instability occurs in order to take appropriate preventative action. The goal therefore is to detect instability, detect it as early as possible, and be able to rely on a model that demonstrates the ability to improve considering the limited volume of data available currently.

Data Preparation and Exploration

The dataset contains 14 subsets, or folders, of simulations of 10,000 particles for 501 time-steps. The position and velocity of each of these particles is given in the dataset. Since individual particles are of little concern, and what matters is a metric that explains the status of the entire folder at each time step, the folders were summarised using the feature:

$$KE^{ft} = \frac{1}{N}\sum_{n=1}^{10,000} V_n^2$$

Which can be described as the kinetic energy (KE) metric for a folder f at a time step t. V is the velocity of the particle n; f has 14 distinct values, and t has 501 values. I removed folders v1_8, and v2_2 because of the rampant discontinuities present, and trimmed the folders v1_4 and v2_0 down to eliminate their discontinuities. In Fig. 1 below, we can clearly see a difference in the KE metric between the stable folders (v0_0 to v1_4) and the unstable ones. A simple logistic regression should be enough for prediction, and it appears that the prediction could be made very early with access to merely a few time-steps of velocity data.

Additionally, I prepared another metric called the sign change S, which is defined as the total number of sign changes in velocity of all particles of a folder at a particular time period. So, for instance, at $t = 0$, $S_0 = 0$, since no sign changes can happen at the first time-step; then at $t = 1$, if 53 out of 10,000 particles experience a sign change in velocity (and hence a change in direction of motion), $S_1 = 53$ for that particular folder. After cleaning the data as mentioned before, we observe count of change in sign in Fig. 2 below. Here as well, there is a clear difference between the stable and unstable folders' behaviours.

Methods

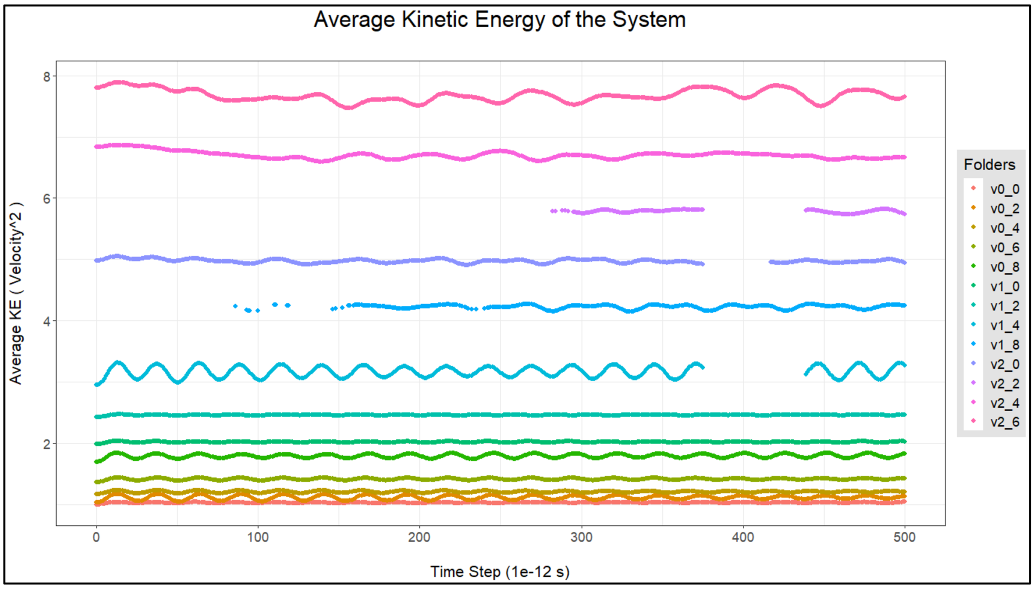

Fig. 1

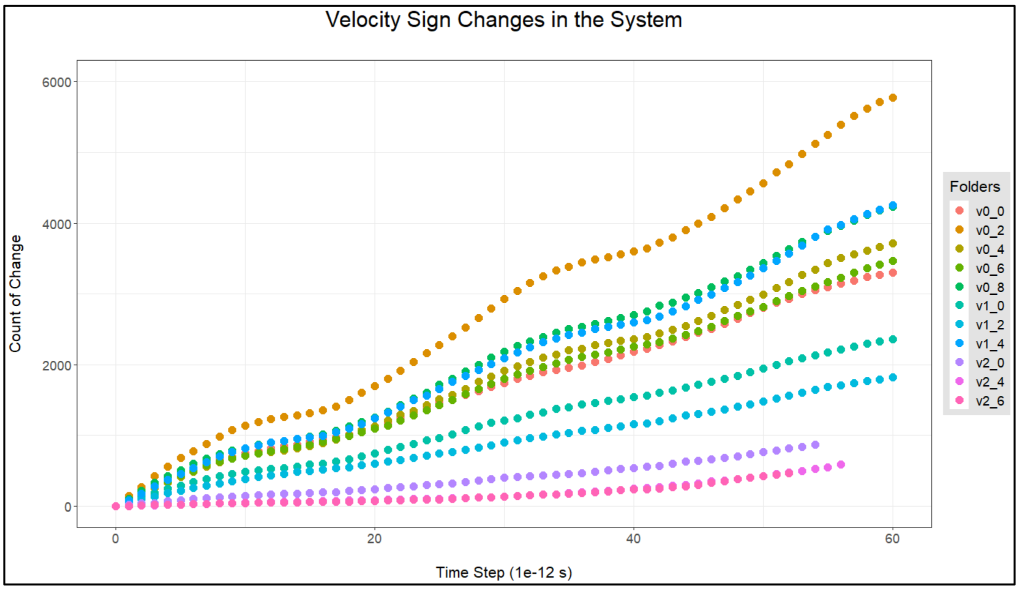

Fig. 2

For the KE prediction $[p = 1, n = 4048]$ I assume the model to have access to all data before a user-defined *timestep* variable for training, and the testing is to be done at the *timestep* value. For e.g., if $timestep = 10$ the training data contains all observations with time-steps less than 10, and the test is to be run on all values at time-step of 10. Higher the *timestep*, more time and data it takes to determine stability of a fusion simulation. Logistic regression was deemed to be a sufficient method, since a linear decision boundary appears to satisfy differentiation of stable and unstable folders, as seen in Fig. 1.

New data can easily be integrated by updating the dataset, using the equation described above, for additional folders. In theory, the minimum number of test observations required for a 1% error rate at 95% confidence interval, $n \geq 9604$. We have less than half that in the entire dataset.

$$0.01 = 1.96\sqrt{\frac{0.5^2}{n}}$$

For the sign-change prediction $[p = 2, n = 4048]$, a similar reasoning as mentioned above was used for the test-train split, using a *timewait* variable. Support vector classifier (SVC) was deemed to be sufficient, since there is a possibility of some porosity requirement in the early time-steps, as seen in Fig. 2. New data can easily be integrated by updating the dataset in this case too.

Results

For the KE prediction method, the ROC curve and the confusion table are given below for $timestep = 1$:

Truth	0	1
Predicted		
0	8	0
1	0	3

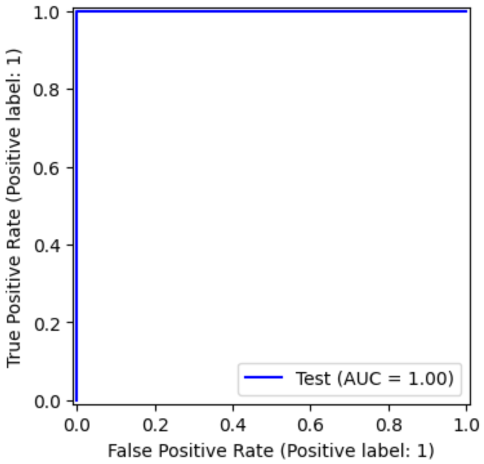

Fig. 3

As expected, the model needed just one time-step worth of data to accurately separate the unstable folders. With more data, the model can be refined, and it is expected that *timestep* may need to increase, so that more training data is being used.

For the sign prediction method, the results at $timewait = 1$ are:

Truth	0	1
Predicted		
0	8	3
1	0	0

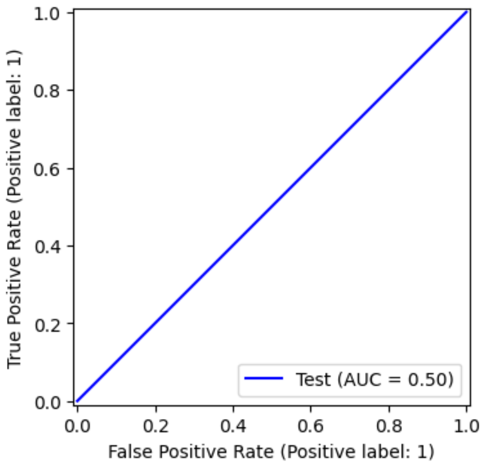

Fig. 4

This is clearly a poor show, as the model assumes they are all stable when access to only the first time-step is made available. However, with just one more time-step of data, the results improve:

Truth	0	1
Predicted		
0	8	0
1	0	3

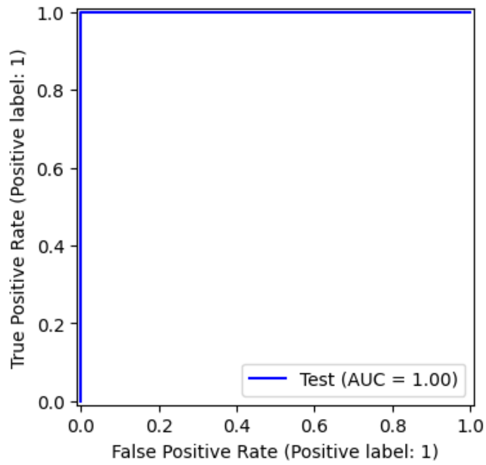

Fig. 5

The ROC curves here are trivial, since the misclassification rate is zero. But in theory, a false positive (predicted unstable, but truly stable) is much more preferable over false negative (predicted stable, but truly unstable, potentially causing significant time and financial loss).

In both cases, the lead-time is extremely small – just one to two time-steps. Considering the systems become unstable after over 50 time-steps, this is more than enough to compute and implement preventative actions in order to delay and/or avoid instability.

Discussion & Conclusion

In conclusion, both of these models, with the limited data available to use, are highly effective. I believe that is because of the real physics involved. In the KE prediction, an unstable system can be expected to have generally higher levels of kinetic energy than the stable systems, as particles are in motion at much higher and uncontrolled speeds.

The sign prediction method is a little complicated. Initially I expected the unstable systems to exhibit higher sign changes. But talking to Dr. Peter Overholser made me realise that particles in the stable systems have velocities closer to zero, and therefore are more prone to switch signs, than the unstable systems where the velocity magnitudes are larger, and therefore more resistant to sign changes.

As more data is introduced, and as the simulation moves towards higher dimensions, the core physics of the simulation still holds. Which is why I have more confidence in the high levels of feature engineering performed in both of my methods to produce effective results in real-world situations as well.

In the process of feature engineering, I explored the particle accelerations, I went so far as to Fourier transform the acceleration vs time-step data to extract the fundamental frequencies, in the hopes of finding another model. Those features are a good avenue to explore in the future.

Some AI assistance was used for data processing. Help was also found in workplace discussions with Dr Peter Overholser.

Appendix: Code

Link to feature engineering R code for KE prediction

Link to feature engineering R code for sign count

Link to Python code for KE model

Link to Python code for sign count model