École Polytechnique Fédérale de Lausanne

QCE Plasma regime investigation

CS-433: Machine Learning

Team: CaptchaML

Aymeric de Chillaz
Hugo Majerczyk
Eric Saikali

Abstract

This study, in collaboration with the Swiss Plasma Center, reveals crucial insights into plasma behavior in nuclear fusion. More precisely, it exhibits the analysis of the underneath structures of plasma using different machine inputs and the construction of a promising LSTM model to predict plasma state over time based on programmed tokamak machine inputs.

1 Introduction

1.1 Project description

The proposed project, initiated by the Swiss Plasma Center lab (SPC), aims to explore the diverse plasma containment regimes generated by the tokamak engine and seeks to better understand nuclear fusion. These regimes are labeled as L-mode, QCE-H mode, and ELMy mode, with a particular focus on achieving the QCE-H regime. The SPC lab has provided us with a dataset comprising 60 experiments, referred to as "shots," each meticulously documenting machine and physical states of the tokamak and plasma. Each shot is a time series consisting of around 1s of analyses sampled at 10kHz, with some shorter than 1s.

Our initial project focused on highlighting patterns in the data using unsupervised learning, concentrating only on the 19 machine inputs. However, after a thorough investigation, we concluded with our assistants that the data was not suited for such analysis. Avid to make something of this great opportunity with the SPC, we didn't stop there and shifted towards a supervised learning approach aimed at predicting plasma states, still based on the same 19 machine inputs but also using the regime state label.

1.2 What problem are we solving?

Our project tackles two challenges within the realm of nuclear fusion research. First, through advanced analytics, we seek a better understanding of patterns governing plasma states within the tokamak. This analytical phase aims not only to aid the researchers at the SPC in validating existing hypotheses but also to uncover novel factors influencing experimental outcomes. Second, the prediction component of our project aims at improving initial baselines, setting higher standards for state prediction. Utilizing time series analysis, our project aims at creating a practical tool for the SPC that predicts how their shots will behave over time given a time series of tokamak machine inputs. Beyond all, we seek to provide a roadmap for future research in nuclear fusion.

1.3 Why is it worth solving?

By gaining insights into the complexities of plasma behavior, we contribute directly to advancements in fusion research, potentially getting closer to providing the world with a clean energy solution. Our project aligns with the urgency of addressing global energy challenges, making it scientifically exciting and environmentally impactful.

2 Unsupervised Exploration

2.1 Data Exploration

In our initial analysis of the provided data, we inspected the characteristics of our 19 machine inputs. Notably, we identified that only the "isbaffled" feature has binary categorical values, while the remaining features are numerical. Examining the distribution of our three-classed label we found that half the samples fall into the ELMy H-mode, with the remaining quarters distributed between the L and QCE H modes.

Cross-validating the ranges of our features and the supposed normal ranges provided by our assistant revealed that none of our samples were outliers. Furthermore, a glance at the temporal aspect of the shots indicated a degree of continuity between subsequent time steps, which was later validated with visualizations and clustering.

2.2 Visualization of the data

2.2.1 PCA

To gain a visual understanding of our data, we employed Principal Component Analysis (PCA) to project our shots into a lower-dimensional space. PCA proved to be instrumental because it maintains the structure of the data and just finds the dimensions with max relative variance, the principal components. Analyzing all the data, the first three principal components captured 54% of the variance, ensuring confidence in our visualizations.

Our plots revealed that shots followed continuous paths, forming "snake-like" structures. When plotting multiple shots, we discovered some of them overlapped at certain points. In addition, recognizing the data's sparse nature and density along these "snakes", we deduced that a clustering algorithm favoring spherical clusters, like K-means, was not appropriate. DBSCAN on the other hand, would be able to follow the complex shapes of our machine inputs.

2.2.2 PCA Density

PCA provides a way to visualize high-dimensional data; however, even then discerning the spatial density of specific plasma regimes is a tricky endeavor. In essence, we want to understand what points in space, are "hot spots" or agglomerations of a certain regime. Thus, we mapped all our data into their two principal components, ensuring a simplified yet informative representation. Then we employed heatmaps to convey the density and distribution of plasma regimes across the spatial landscape. (Please refer to Figure 1 in the Appendix). From these spatial patterns, we managed to reverse the PCA mapping and gained a better understanding of the machine inputs that were correlated to these plasma states. However, these patterns are very limited as they take every timestep as "independent" without considering the previous steps in the shot. LSTMs will be considered to solve this issue.

2.2.3 UMAP

In our attempt to visualize our complex data, we gave Uniform Manifold Approximation and Projection (UMAP) a shot. Like TSNE, it serves as a dimensionality reduction while keeping cluster structures intact, but offers scalability advantages. Tuning our UMAP model was essential for the task at hand: we focused on n_neighbors and min_dist. Firstly, higher n_neighbors, the number of neighboring points used in local approximations, results in more global structure being preserved. Additionally, min_dist which controls how tightly the embedding is allowed to compress points together, is favorable to clustering for lower values. With this knowledge in mind, we optimized UMAP for clustering with n_neighbors and min_dist respectively equal to 40 and 0.1. Despite this, the intricacies of our data presented challenges in achieving valuable findings: the extracted clusters didn't show any signs of inter-shot or intra-shot trends. Yet as we continue our exploration, the insights gained from both PCA and UMAP enrich our understanding of the data's structure, which will be essential for later stages. Most importantly, we will keep using PCA for the visualization of our clusters and predictions.

2.3 Traditional Clustering

2.3.1 Baseline - Kmeans

Due to the sparse nature of the data we did not expect K-mean to perform well and we were right. Hyperparameter tuning was performed using the elbow method and yielded `k=4`. Figure 2 (Please refer to the Appendix) is a plot of the 2-dimensional PCA of the obtained results and reflects the absence of intra-shot trends or even inter-shot trends. As such, we tried a slightly controversial technique, yet one that can yield great results: we used UMAP to reduce the dimensionality by half and expand the distances between the clusters and then we performed K-means. Even when creating "artificially enhanced" clusters with UMAP, the results were subpar, so we finally rejected this clustering algorithm.

2.3.2 DBSCAN - L-fold Hyperparameter tuning

Given the unique characteristics of our data, we determined that DBSCAN was a more suitable choice for clustering. Hyperparameter tuning becomes pivotal for effective analysis, and it involved tweaking two key parameters: MinPts (minimum number of points per cluster) and ϵ (radius for neighborhood computation). In practice, a common rule of thumb suggests setting the min cluster size to 2 times the number of features, so we were only tuning ϵ. The ideal ϵ value is best discovered through a silhouette score plot. Our approach involved identifying an epsilon that gave a large score, while not overfitting or underfitting.

However, silhouette score computation time is too important, since it runs in $O(N^2)$. Instead, we propose L-fold Hyper-parameter tuning which involves dividing our data into L groups, computing silhouette scores for each fold, and then plotting the means or medians across the L folds (Please view Figure 3 in the Appendix). Remarkably, with `L = 10`, our computation time went from 10 hours to 2 hours 30 min.

To find the best ϵ from the plot, we considered the parameters that yielded top scores. Upon closer inspection, we noted that the silhouette score exhibits a slight rise towards 0.966 before descending around 2.7. Then on, since we favored the high clustering associated with lower ϵ values, we wanted one of the lowest of these hyperparameters. Theoretical indications favored $\epsilon = 0.966$. Running DBSCAN with that value yielded 29 intriguing clusters.

2.3.3 DBSCAN - Visualization

Visualizing the whole data did not look like much; however, when looking at each cluster individually that's when we can potentially make some interesting findings.

In eight distinct clusters generated by our algorithm, a noteworthy achievement emerged: the clear identification of similar shots. This discovery opens up possibilities for investigating groups of comparable shots, presenting an opportunity to delve into subtle variations. The potential lies in understanding the nuanced shifts in labels among these closely related shots, thereby gaining insights on the machine inputs that influence plasma state changes. To analyse the different clusters, we created per feature violin plots showing the machine input distributions. This led to some findings, but nothing amazing as the derived insights were limited by the number of shots. Indeed, deriving conclusive patterns from a handful of similar shots is far from accurate when there are 19 potential parameters and even more associations between them.

Another noteworthy observation arises from the identification of 15 single shots as distinct clusters. This behavior can be attributed to the spatial trajectories of these shots, which do not intersect. It's essential to emphasize that the presence of these single shots doesn't necessarily imply distinct characteristics. For instance, two perfectly parallel shots wouldn't cross and thus wouldn't be clustered together.

Additionally, our exploration revealed an imbalance in mode representation within specific clusters, offering intriguing insights into correlated paths with distinct plasma states. These imbalances add a temporal dimension to our understanding, contrasting with the heatmaps generated on the PCA-reduced space which only looked at specific points in space and not how shots got there.

2.3.4 Findings

In summary, DBSCAN allowed us to cluster somewhat similar shots and shed light on the intricate relationships between machine inputs and plasma labels. Notably, it unveiled the nuanced reality that very similar shots may not necessarily fall within the same cluster, emphasizing a limit of this clustering algorithm on our data. However, these analyses face limitations imposed by the restricted number of shots in our dataset. Most importantly, it is essential to note that the continuous nature of our time series led to clustering groups of shots, thus exploring inter shot patterns, while leaving intra shot trends unexplored.

2.4 Novel Clustering

Introducing a novel approach to clustering shots in our scarce and continuous time series, we sought an alternative to traditional algorithms like K-means and DBSCAN, which demonstrated suboptimal performance in our context. Notably, the DBSCAN analysis revealed that two parallel shots were not clustered together. On the contrary, two completely perpendicular shots were clustered together if they crossed each other. We devised a method centered around relative distances and shot orientation. Taking two shots, we define their distance as the sum of euclidean distances between matching time steps. Of course, the data must be normalized to consider each machine input equally in the metric. Note that this technique is especially powerful since even two parallel shots, but going in opposite directions are given a high distance. After computing the distance between each pair of shot, one can group them by defining a threshold under which shots are clustered together. This method proved to be quite useful in gaining a deeper understanding of our data and providing clusters that we had higher confidence in.

3 Prediction

Having performed thorough visual analysis and clustering, we arrive at an important realization: our data, fundamentally rooted in human inputs to a machine, doesn't align with the nuances of unsupervised learning. This leads us to pivot towards supervised learning, with a primary focus on predicting labels. Crucially, our previous analyses are not in vain, because they have granted us valuable insights on our data.

Notably, the identification of clusters of similar shots underscores the importance of ensuring that our training data comprehensively represents these groups, thus avoiding potential oversights. This is especially necessary due to the scarcity of our data.

Moreover, our past experiences highlight the temporal intricacies of our data. If we want to predict a label at a certain point in space, we must consider how it got there! Notably, because the previous machine inputs define the current state of the plasma.

In the end, our models were trained on 50 shots that were selected as explained above and tested on the 10 remaining shots.

3.1 Baseline - Linear Model

While setting our simple baseline, we intentionally sidestepped temporal dependencies, opting for a straightforward approach that only considers the current time step to predict labels. For this task, we employed a multinomial logistic regression model, expecting modest results. And yet, we were quite surprised with the results. In fact, the F1 scores achieved for the L-mode, QCE-H mode and ELMy mode were respectively 0.9591, 0.4676, and 0.7967.

3.2 Exploring Temporal Dynamics

In our quest to consider time dependencies in our predictive model, Recurrent Neural Networks (RNNs) emerged as a natural choice. RNNs are a type of neural network architecture specifically designed to handle time series thanks to a hidden state and loops within the network which allow information to persist and influence future predictions. But we also know of one of RNN's shortcomings: the "vanishing gradient" problem. This occurs during the training when the gradients become extremely small causing the model to struggle with extended time dependencies.

To address the previous problem, Long Short-Term Memory Networks (LSTMs) were introduced to our solution as they were specifically designed to deal with this issue. This characteristic makes them a more robust choice for modeling our complex data and lengthy sequences.

3.3 Pre-learning Methods

Before delving into model implementations, we confronted several challenges to ensure that our data was adapted for our models and that we would obtain optimized results.

3.3.1 Padding

Firstly, the variability in shot lengths presented a hurdle, as RNNs and LSTMs demand uniformly sized time series when they are train in batch, which is the standard way of training them for performance. Our chosen strategy consisted in padding to the left with zeros, thus ensuring that the model could train on all our shots in batch. We also introduced a new label, the null state 0, to denote the padded regions. Remarkably, results showed that the model had a smooth transition from the null state to our actual labels once the padding ended. Most importantly, this approach guarantees that the model doesn't retain valuable information during zero-padding and then learns from the genuine machine inputs. This step is especially important since we don't want to reduce the size of our data by discarding shorter shots.

3.3.2 Data Augmentation

Additionally, we were confronted with the realization that our complex models would be prone to overfitting on our very limited data. Initially, we were inclined towards data augmentation by the introduction of random noise. Similarly to machine learning on images, we also wanted to expand our data through shifts and slight rotations. However, the domain experts at the SPC lab informed us that when working with the machine inputs of tokamak experiences, the inputs are exact. Thus, we made the informed decision to discard this method.

3.3.3 Reducing Shot Length

As we continued our search for a solution to the scarcity of our data, we found the solution when overcoming another hurdle. We discovered that our time series were much too long for RNNs and LSTMs, thus our models were unlikely to find dependencies on such a long term. We deemed that splitting our shots was the most beneficial solution as it also meant that we would increase our number of time series, thus hitting two birds with one stone. One envisioned possibility consisted in dividing each shot into contiguous blocks of a predefined size such as 500. However, this raised concerns about information continuity across blocks, potentially limiting the depth of our analysis. Ideally, we would want a method that samples data from the whole time series, thus keeping track of the machine input changes throughout the experiment.

A nuanced solution emerged through steps-based shot splitting. For instance, if we want around 500 time steps after processing, then we will go for steps of $11000/500 = 22$, (where 11000 is the approximate number of steps per time series). This leads to the creation of 22 shots where each time step corresponds to a jump of 22 from the previous one. This approach not only increases our number of shots by 22, but also selects points from throughout the shot, thus ensuring that the whole trajectory of the shot is encapsulated. Note that since our machine inputs are quasi-linear and continuous, dividing shots in such a fashion should not be problematic as we can realistically assume that the samples stepped over are near the direct line between the two points.

3.4 Recursive Neural Network

As we ventured into the implementation of both Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTMs), our goal was to gain insights into the intricate dynamics of time-dependent plasma behavior. Anticipating the gradient vanish of RNNs we kept it in our analysis to have an additional benchmark while also serving as a stepping stone towards the LSTM. Our RNN therefore did not benefit from the shot length reduction.

3.5 Long Short-term Memory

On the LSTM side though, the training was not without its surprises as the initial runs yielded worse results than the simple linear baseline. How could that be? Intuitively, a model considering time dependencies within the time series had to perform better, so we delved deeper into the training dynamics. It became apparent that overfitting was the culprit; the training error was decreasing while the testing error was following the opposite trend. To address this issue, we performed data augmentation as explained previously: performing step-based splitting of shots. Quite instantly the results were much more promising. In fact, we finally reached better results than the baseline! Our initial step size of 22 proved effective, but hyperparameter tuning revealed that reducing the step size further enhanced performance. Hyperparameter tuning led to the optimal step size, 11.

3.6 Hyperparameters fine-tuning

To find the best model for our task, we fine-tuned our hyperparameters using the grid-search method. Each model was optimized using the Adam optimizer and using the CrossEntropy loss function weighted by the distribution of labels in our training data. The hyperparameters are the learning rate used in the optimizer, the number of stacked RNN or LSTM layers one after the other, the size of the hidden cells in each RNN or LSTM layers (*hidden_size*) and the dropout probability.

Using a dropout probability both in each RNN or LSTM layer and in a specific final layer allowed to reduce the overfitting, by randomly dropping layers during the training process, it serves as a regularization method.

3.7 Findings

In the end, as *Table 1* in the Appendix shows, the RNN and LSTM models tend to perform better than the baseline. More importantly, RNN turns out to perform better on the test shots than LSTM. The reason for that is the large overfitting that occurs, which is revealed during the training of our LSTM model as shows in *Figure 4* in the Appendix, despite all the precautions we took to limit its effects.

3.8 Prediction Tool

In our quest to provide tangible insights to the Swiss Plasma Center (SPC), we've crafted a prediction tool. This tool, tailored for domain experts, offers a visualization of machine inputs and unveils predictions generated by our LSTM model.

Given a time series of programmed machine inputs, we utilize our LSTM model to make predictions about the states of the plasma at each timestamp. Then, we map the high-dimensional machine inputs into a lower-dimensional space using PCA to have a final visualization. We intend to leave a practical tool, which the domain experts will use to get an idea of how their experiment would perform in the tokamak.

3.9 Approach limitations

While we did beat our baseline model, we believe that the LSTM solution explained in this report has yet to reach its full potential. Indeed, our model would have much to gain from a more abundant and diverse data set. This would reduce overfitting, enhance model generalization and potentially lead to better predictions.

The analyses performed in this project are anchored in the characteristics of the data provided by the Swiss Plasma Center. Thus, the trained LSTM may not generalize well for other tokamaks. Yet we believe that the methods used could be reused in other settings.

The time-dependent models that were trained consisted in RNNs and LSTMs; however, there exist other types of architectures. GRU for one, is a more recent alternative to LSTMs that appears to yield similar results but with higher efficiency. There could be much to learn from implementing other solutions!

4 Conclusion

This journey led us through the intricacies of plasma behavior, unveiling pivotal insights throughout the way. Much was learned about the nuanced path of data investigation, model exploration and refinement.

Initially, unsupervised learning techniques proved unsuitable for the sparse and continuous nature of our data. And yet those analyses were not in vain, as they allowed us to gain some valuable insights about our data. Notably, its continuous nature enabled certain data augmentation techniques which we refer to as "step-based splitting". Moreover, it led to the discovery of clusters of similar shots, emphasizing the need for representative training data.

Then, we moved to prediction, notably with time-based models such as RNN and LSTM. These enabled us to model the intricacies of our data and obtain reasonable results. However, it was the deep understanding of the data that enabled the strategic implementation of data augmentation and padding to further propel our models and beat the baseline.

In essence, our journey was not just a jump from one model to another; it was a dynamic process of learning from the data, adapting to its intricacies, and refining our strategies accordingly.

5 Ethics consideration

Several ethics considerations can be raised regarding this project. First of all, its inherent nature as a physics simulation project using data from physics experiments prevents any privacy issue as it doesn't consist of sensitive or personal data and does not contain troublesome bias. Moreover, the project itself is taking part in an international research effort around the ITER project and the development of a future energy producing method using nuclear fusion, so it aims at benefiting the largest amount of people possible, and its results should be publicly accessible. As an open research and given its form the end users should be able to understand the solution and its limits so it gives them empowerment over it. However, it consists of state of the art research, so such technology may actually not be available in regions that do not have the technological abilities to leverage it, which could reduce its fairness. Moreover, although predictions errors in the solution should not have negative impacts given their intent, this model could hypothetically be misused for research with diferent purpose such as weapon development. Finally, its long term goal is a sustainable objective of producing energy with lower carbon footpring, but the simulations needed to obtain the data and research the system need non-negligible amounts of energy (a plasma at milions of degrees is formed in a closed structure).

Appendix

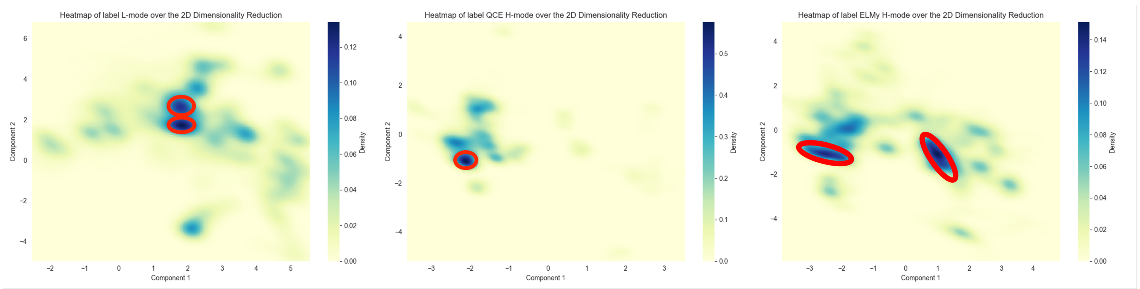

Figure 1: Graph showing the "hot spots", circled in red, of Plasma modes in 2D PCA.

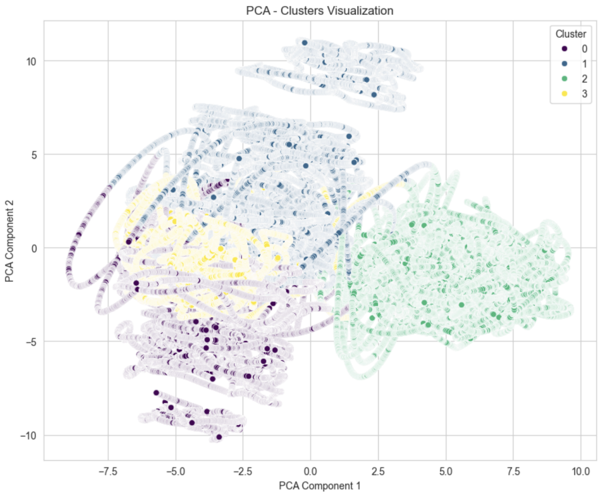

Figure 2: 2D PCA plot of k-mean clusters (k = 4).

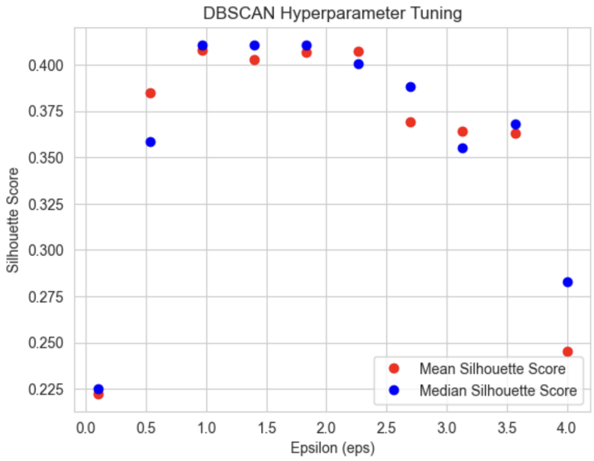

Figure 3: Plot of DBSCAN's median and mean silhouette scores as a function of epsilon (L-Fold Tuning).

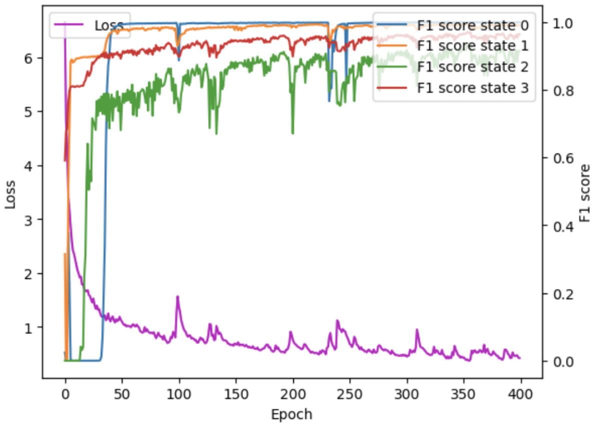

Figure 4: Plot of LSTM training performance evolution over time, using Adam optimiser, Cross Entropy criterion, hidden_size=128, num_layer=4, learning_rate=0.001 and drop_prob=0.2).

	Baseline	RNN	LSTM
f1-scores state 0 (dummy)	-	**0.9999**	0.9957
f1-scores state 1 (L-mode)	0.9591	**0.9592**	0.9408
f1-scores state 2 (QCE-H mode)	0.4676	**0.6399**	0.4838
f1-scores state 3 (ELMy mode)	0.7967	0.7696	**0.8250**

Table 1: Comparison of several classifiers' predictions' performances on test shots