Editing a Section of an Academic Paper

Original with Edits

The sampled data was ~~been~~ split into training (70%) and testing datasets (30%) by stages; 54 stages (5508 samples) for training and 25 stages (2550 samples) for testing~~were reserved as testing dataset~~. We compared three different machine learning classifiers for this problem; random forest, bagging SVM, and artificial neural network ~~for this problem~~. The random forest method is an ensemble learning method~~,~~ which consists ~~is consist~~ of many decision tree classifiers. Random samples of data are selected to train each tree separately, and the final prediction is aggregated from all the trees (Breiman, 2001). The ~~R~~random forest can overcome the tendency of over-fitting to noise as ~~with~~ the number of trees is increased ~~increasing~~. In this case, 150 decision trees were used to leverage both efficiency and accuracy. The main advantages of the random forest are ~~is~~ the short computational time and the potential for parallel computation, since individual decision trees are independent~~. Individual decision trees are also independent in the forest~~. ~~S~~The support vector machine (SVM) classifier was also ~~is~~ chosen because of its advantage in defining binary classification problems;~~,~~ in its implementation we used the ”bagging” algorithm to aggregate 10 linear-kernal SVM classifiers (Breiman, 1994). Bagging perturbs the training dataset by re-sampling and increases accuracy. The artificial neural network we deployed consisted ~~is consist~~ of two ~~D~~ dense layers and one ~~D~~ dropout layer to ~~prevent~~ reduce over-fitting to noise. ~~In this project, we used Scikit-learn.~~ We used Scikit-learn to implement the aforementioned algorithms.

~~As~~Because the dataset is sampled to have a 1:1 ratio between fracture hit and background noise instances,~~. A~~ accuracy is used to evaluate the overall algorithm performance. ~~One other important measurement would be~~ Another important metric is the precision of fracture hit predictions,~~; it~~ which is the number of correct fracture hit predictions out of ~~all~~ the total number of fracture hit predictions. For all three algorithms, the accuracy, fracture hit precision, and run time are shown ~~showed~~ in Table 1. ~~R~~ The random forest classifier shows advantages in computational time and fracture hit precision comparing with the bagging SVM and the artificial neural network. It ~~is also predicting less~~ also returns fewer false positive predictions, ~~which would trigger less~~ and thus fewer incorrect fracture hit ”alarms” ~~in the detection~~. To test the algorithms in a more realistic scenario, we input all the streaming data of the testing stages as 400-channel~~s~~-by-10-min windows, with a 200-channel~~s~~-by-5-min overlap between two adjacent windows. Three stages of the random forest predictions are plotted in Figure 5. Only the fracture hit (positive) predictions are annotated in the figure because ~~as~~ the majority of predictions are non fracture-hits (negative). For most of the stages, the random forest performs well except for a few false-negative predictions (a failure to detect the fracture hit). We marked the first positive prediction as the time when the fracture was first ~~first being~~ detected for each stage, and compared it with the leading time~~,~~. The difference between these times tells us ~~so we can calculate how much~~ how far in advance ~~ahead~~ the algorithm can predict the fracture hit. For the 25 testing stages, the random forest achieved a median value of 108 min before the fracture reached ~~reaching~~ the DAS fiber (within the early 6% of the leading time). A summary is shown in Figure 6, where the first detection of each stage is shown as a percentage normalized by the leading time. There are three outlier stages (3, 4, 8) where the first detection does not correspond to precursor signal.