1. Added MNIST-create-dataset jupyter notebook:
    a. Create 18 data sets (train and test hd5 files for 9 digits)
    b. For each digit, the train file only has NORMAL data. The test file has NORMAL and ABNORMAL data.
2. Added MNIST datasource under datasource/mnist.py:
    a. Use get_train_data('NORMAL') for training data.
    b. Use get_train_data('ALL') for test data. This gives both NORMAL and ABNORMAL test data. They are unshuffled meaning it is [ABNORMAL, ABNORMAL, … , ABNORMAL, NORMAL, NORMAL, …, NORMAL]. This is because if only one class at a time is given to the GanPerformanceAnalysis() class, it will not work.
3. Modified model.py file:
    a. Added sigmoid layer after output (Line 140 : `output = tf.nn.sigmoid(output))`
    b. Removed negative symbol in Line 318 and 332 : `-batch_scores = 1 * self.tf_session.run(d_output, feed_dict={x: input_batch, keep_prob: 1.})`
4. Added MNIST-train.py file:
    a. Train detective using training data
    b. Abnormal digit class is specified in **abnormal_digit**
    c. save_path and load_path is manually specified
5. Added MNIST-test.py file:
    a. Prints roc_score **(did not implement saving of this roc score)**
    b. Saves classification_report to CSV file. Here 0 is NORMAL class and 1 is ABNORMAL class. The probabilities (detection_result.data) are rounded to the closest integer