This package consists of the source code of gC3S_E, which was developed by Dr. Yunlong Mi for predictive data stream mining. One can use it freely (for academic purposes only) at your own risk. For other purposes, please contact Dr. Yunlong Mi directly.

Quickstart for gC3S **or** gC3S _E **(work on Eclipse workstation):**
JDK environment: jdk 1.8 or above.
Data format: The Data of the CSV format file looks like the following:

```
29.7037,21.3278,1.8359,0
30.4719,5.5551,36.8715,0
33.2494,-3.937,52.1075,0
...
```

Step 1: set the file path in ParametersUtil.java

```
public static String train_path = "./data/train[1].csv";
public static String test_path = "./data/test[1].csv";
```

Step 2: set some related parameters in ParametersUtil.java, such as the parameters for TrafficStream1 as follows:

```
/** gC3S or gC3S_E: error corrects rate for concept drift detection. */
public static String methodType = "gC3S";
/** Show the results by bachSize or overall accuracies.*/
public static String showResult="overall";// bachSize or overall

/** Three required parameters */
/** Lambda(i): concept falling space, the $\lambda $ value and P */
public static int lambda  = 8; // it represents lambda = 8/10
public static double P = 1;// concept falling, P=1 or P=1/10
/** MaxSize: The size of concept spaces for each class. */
public static int conceptSZ =300;
/** Chunk size: The size of each data chunk. */
```

```
public static int C =50             ;
/** Delta_{d}: For gC3S_E, when isConceptDrift < conceptDriftTheta, concept drift occurs. */
public static double conceptDriftTheta=3;//[0,3.15], 3.15 > PI
```

Step 3: run gC3S or gC3S_E in runMethod.java

```
/** Load datasets */
long s1 = System.currentTimeMillis();
Vector<Object> train_vec = LoadDataUtil.loadData(ParametersUtil.train_path.replace("indexNum", String.valueOf(index)));
Vector<Object> test_vec = LoadDataUtil.load2Data(ParametersUtil.train_path.replace("indexNum", String.valueOf(index)),
        ParametersUtil.test_path.replace("indexNum", String.valueOf(index)));

long e1 = System.currentTimeMillis();
System.err.println("Load dataset：  " + (e1 - s1) + "(ms)");

/** Instantiation system, 实例化系统 */
GC3S gC3S = new GC3S(train_vec);
/** Initial system, 系统初始化 */
long s2 = System.currentTimeMillis();
gC3S.initialS();
long e2 = System.currentTimeMillis();
System.err.println("Initial system：  " + (e2 - s2) + "(ms)");

/** Evaluating and updating system, 系统动态更新与评估 */
long s3 = System.currentTimeMillis();
gC3S.evaluateS(test_vec);
long e3 = System.currentTimeMillis();
System.err.println("Evaluating system：  " + (e3 - s3) + "(ms)");
```

NOTE: Please cite our work if you use these source condes in any way.