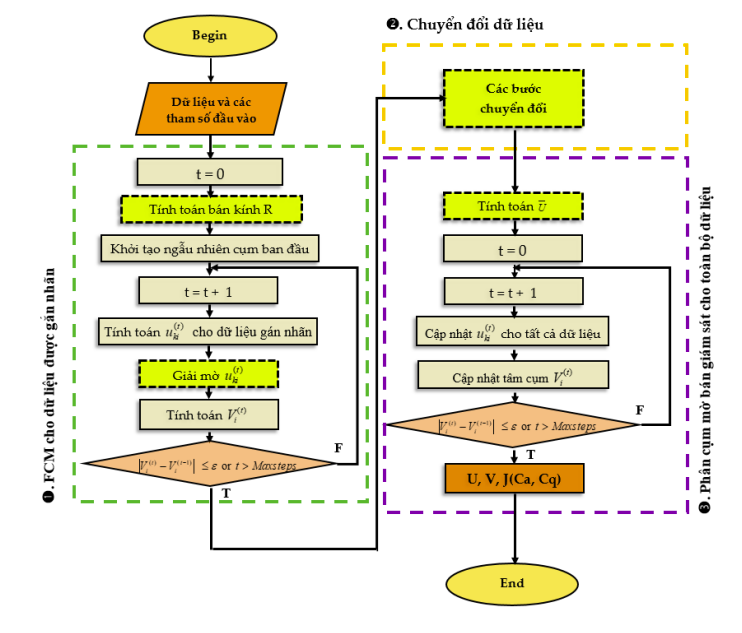
# III. CODE GIẢI THUẬT BÁN GIÁM SÁT MỜ PYTHON

## **C. Thuật toán TS3FCM**

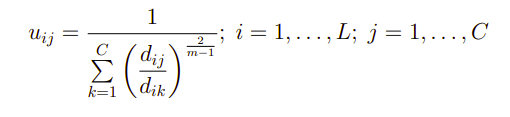
### 3.1. Các bước giải thuật toán TS3FCM

- Chi tiết thuật toán TS3FCM:

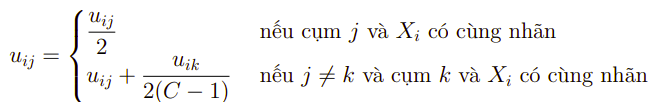


\* Bước 1:Phân cụm mờ cho dữ liệu đã được gán nhãn

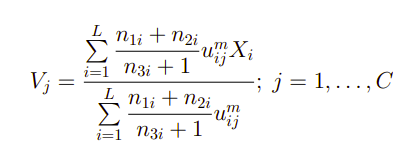
* Thuật toán FCM cải tiến:
  + Đầu vào: Bộ dữ liệu X = {X1, X2, . . . , Xk, . . . , XN } với số điểm dữ liệu N; số điểm dữ liệu được gán nhãn trong X: L < N; số cụm C; ngưỡng ε; số mờ m; số mũ α và số lần lặp tối đa MaxStep > 0. Đầu ra: Ma trận độ thuộc u và tâm cụm V
  + Đầu ra: Ma trận độ thuộc u và tâm cụm V .
* Begin
  + Bước 1: t = 0
  + Bước 2: Khởi tạo ngẫu nhiên các tâm cụm: Vi(t) <- random, j = 1,...,C
  + Bước 3: Thực hiện lặp các bước sau đây (4-8):
  + Bước 4: t = t +1
  + Bước 5: Tính toán uij(t)cho dữ liệu được gán nhãn ( i =1 ,...,L ; j=1,...,C ) bởi công thức



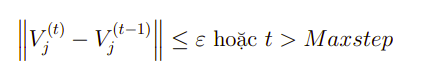
* + Bước 6: Giảm độ thuộc uij(t) theo công thức



* + Bước 7: Tính toán Vj(t) ( j = 1, C) theo công thức



* + Bước 8: Kiểm tra điều kiện dừng của thuật toán theo công thức



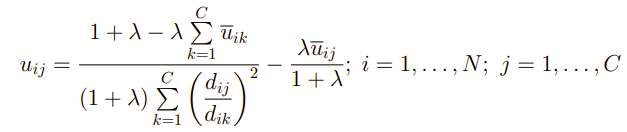
nếu thoả mãn thì dừng thuật toán, nếu không thỏa mãn thì quay lại Bước 3.

\* Bước 2: Chuyển đổi dữ liệu

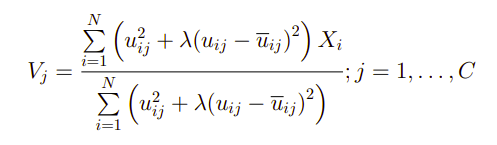
* Từ kết quả đầu ra của khối thứ nhất sẽ thu được các tâm cụm của dữ liệu được gán nhãn và sử dụng chúng làm tâm cụm ban đầu cho bộ dữ liệu của các điểm dữ liệu chưa được gán nhãn. Ở đây, sử dụng thuật toán FCM cho dữ liệu chưa được gán nhãn để thu được các giá trị độ thuộc của dữ liệu chưa được gán nhãn. Sự kết hợp các giá trị độ thuộc của cả dữ liệu được gán nhãn và chưa được gán nhãn tạo nên các giá trị độ thuộc đầu vào (U) cho tất cả các điểm dữ liệu của bước tiếp theo.

\* Bước 3: Phân cụm bán giám sát mờ cho toàn bộ dữ liệu

* Thuật toán phân cụm bán giám sát mờ mới
  + Đầu vào: Bộ dữ liệu X = {X1, X2, . . . , Xk, . . . , XN } với số điểm dữ liệu N; số điểm dữ liệu được gán nhãn trong X: L < N; số cụm C; ngưỡng ε; số mờ m; số mũ α và số lần lặp tối đa MaxStep > 0. Đầu ra: Ma trận độ thuộc u và tâm cụm V
  + Đầu ra: Ma trận độ thuộc u và tâm cụm V .
* Begin
  + Bước 1: t = 0
  + Bước 2: Thực hiện lặp các bước sau đây (3-6)
  + Bước 3: t = t + 1
  + Bước 4: Tính toán uij(t) (i = 1, . . . , N, j = 1, . . . , C) bởi công thức



* + Bước 5: Tính toán Vj(t) (j = 1, . . . , C) theo công thức

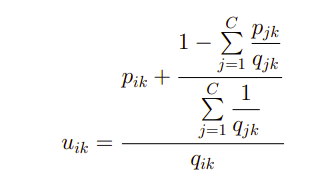


* + Kiểm tra điều kiện dừng: || Vj(t) - Vj(t-1)|| ≤ ε hoặc t > MaxStep.. Nếu thoả mãn điều kiện thì dừng thuật toán, nếu không thỏa mãn thì quay lại Bước 2

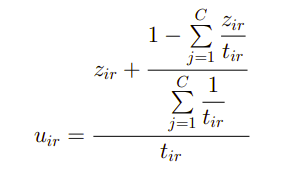
## **D. Thuật toán CS3FCM**

### 4.1. Các bước giải thuật toán CS3FCM

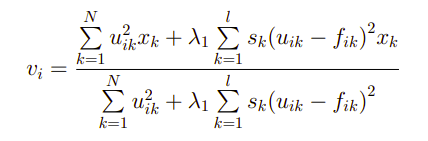
* Thuật toán CS3FCM:
  + Đầu vào: Bộ dữ liệu X = {X1, X2, . . . , Xk, . . . , XN } với số điểm dữ liệu N; số cụm C; độ thuộc của điểm dữ liệu i đối với cụm j : uij , tham số mờ m; ngưỡng ε và số lần lặp tối đa MaxStep > 0.
  + Đầu ra: Ma trận độ thuộc u và tâm cụm V .
* Begin
  + Bước 1: Thực hiện phân cụm FCM trên bộ dữ liệu Xl ∪ Xu để thu được kết quả ước lượng nhãn đầu ra Y˜ , ma trận phân hoạch U˜ và ma trận NC
  + Bước 2: Tính toán giá trị trọng số sk
  + Bước 3: Xây dựng biểu đồ cục bộ W
  + Bước 4: t = 0
  + Bước 5: Khởi tạo ngẫu nhiên các tâm cụm: Vi(t) <- random, j = 1,...,C
  + Bước 6: Thực hiện lặp các bước từ 7-12:
  + Bước 7: t = t + 1
  + Bước 8: Tính toán uik(t) bởi công thức



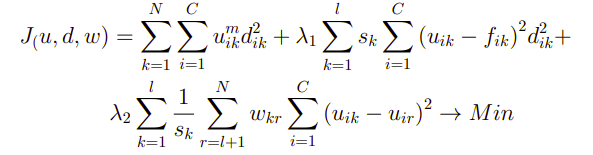
* + Bước 9: Tính toán uir(t) bởi công thức



* + Bước 10: Tính toán vi(t) bởi công thức



* + Bước 11: Tính toán giá trị của J(t) bởi công thức



* + Bước 12: Kiểm tra điều kiện dừng: | J (t) − J (t−1)| ≤ η hoặc t > M axStep. Nếu thoả mãn điều kiện thì dừng thuật toán, nếu không thoả mãn thì quay lại Bước 6.