```
## Fonction de la détection d'anomalie Isolation Forest
def detectionAnomaly(donnee, taille_File):

    ##on récupère le n-1 premiers éléments de la File
    liste_de_donnee = donnee[0:taille_File]
    df = convert_list_database(liste_de_donnee) # on défnit la base de donnée de ref

    # on ajoute le nouveau élément qu'on veut tester, donc df_motif est composé des n-1 derniers élément
s de la File
    df_modif = convert_list_database(donnee[1:taille_File +1])

    contamination_un_element = 1*1/taille_File # on veut qu'un seul élément contaminer parmi la taille de l
a file

    ## Partie de la détection d'anomalies
    model = IsolationForest(n_estimators=100, max_samples='auto', contamination=contamination_un_ele
ment,max_features=1.0)

    model.fit(df[['c1']].values)

    df_modif['anomaly']=model.predict(df[['c1']].values)

    ## on récupère le dernier élémment: celui que l'on souhaite regarder
    dernier_elt = df_modif.tail(1).values
    dernier_elt = dernier_elt.tolist()

    if dernier_elt[0][len(dernier_elt)] == -1: # sinon anomalie détectée alors (à déterminer selon votre choix)

        print("Erreur détecter")

    ## On peut aussi renvoyer la valeur de l'état du dernier élément qui pourra être réutiliser papr la suite
    return dernier_elt[0][len(dernier_elt)]
```