```
CLASS LanGaimaCyberFortSentinel:
  // Constructor to initialize all layers and their corresponding data
  FUNCTION __init__():
      Initialize all layer data (Layer-1 to Layer-7)
      Initialize the update flags for each layer


          // ****** Layer-1: GenFusion Data Repository ******
  FUNCTION load_genfusion_data_repository():
              // Load data from public and private datasets
      public_data = load_public_datasets()
      private_data = load_private_datasets()
      return public_data + private_data

  FUNCTION load_public_datasets():
              // Load public datasets
      RETURN ["Public Dataset 1", "Public Dataset 2", "Public Dataset 3"]

  FUNCTION load_private_datasets():
                      // Load private generated datasets
    RETURN ["Private Dataset 1", "Private Dataset 2", "Private Dataset 3"]


        // ****** Layer-2: SynerTrain Pipeline Augmentation ******
  FUNCTION synertrain_pipeline_function(data):
              // Perform model training with public and private datasets
      models = train_ensemble_models(data)
      augmented_results = augment_results(models)
      RETURN augmented_results

  FUNCTION train_ensemble_models(data):
              // Train different ensemble models with the data
       RETURN ["Ensemble Model 1", "Ensemble Model 2", "Ensemble Model 3"]

  FUNCTION augment_results(models):
              // Perform augmentation tasks like tokenization
      RETURN {
              "tokenization": "Tokenization results",
              "sentiment_analysis": "Sentiment Analysis results",
              "contextual_features": "Contextual Features",
              "content_recognition": "Content Recognition results",
              "feature_integration": "Integrated features"
      }


        // ****** Layer-3: LangAima IntraChain Framework ******
  FUNCTION langaima_intrachain_function(augmented_results):
              // Process the augmented results through the LangAima IntraChain Framework
       external_integration = integrate_external_components(augmented_results)
      RETURN "Processed Data from LangAima IntraChain"
```

```
FUNCTION integrate_external_components(augmented_results):
        // Simulate the integration of external systems and tools
   RETURN "External Components Integrated"
```

// ****** Layer-4: ChainTrust Content Validator ******

```
FUNCTION chaintrust_validator_function(processed_data):
                    // Validate the content's authenticity and integrity
       validated_data = validate_content(processed_data)
       RETURN validated_data

FUNCTION validate_content(processed_data):
                // Simulate content validation
       RETURN "Validated Content"
```

// ******Layer-5: LanGaima Swarm Intelligence Model ******

```
FUNCTION swarm_intelligence_model_function(validated_data):
                // Create decentralized communities for threat detection
       communities = create_communities(validated_data)
       RETURN "Threats Detected by Communities"

FUNCTION create_communities(validated_data):
                // Simulate the creation of communities based on the data
   RETURN "Communities Created"
```

// ****** Layer-6: Classification Model ******

```
FUNCTION classification_model_function(swarm_results):
                // Classify events, activities, and actions using advanced AI models
   classification_results = classify_events(swarm_results)
   RETURN classification_results

FUNCTION classify_events(swarm_results):
   // Perform classification of events, activities, and actions
       RETURN {
                "event_classification": "Event Classified",
                "activity_classification": "Activity Classified",
                 "action_classification": "Action Classified"
       }
```

// ****** Layer-7: Fine-Tuning ******

```
FUNCTION fine_tuning_model_function(classification_results):
                // Fine-tune the model based on the classification results
       fine_tuned_model = fine_tune_model(classification_results)
         RETURN fine_tuned_model
```

```
FUNCTION fine_tune_model(classification_results):
        // Simulate model fine-tuning
    RETURN "Fine-Tuned Model Ready"
```

// ****** Main Process ******

```
FUNCTION main_process():
    // Step 1: Data Curation (Layer-1)
        layer_1_data = load_genfusion_data_repository()

    // Step 2: SynerTrain Pipeline Augmentation (Layer-2)
        augmented_results = synertrain_pipeline_function(layer_1_data)

    // Step 3: LangAima IntraChain Framework (Layer-3)
        layer_3_data = langaima_intrachain_function(augmented_results)

    // Step 4: Content Validation (Layer-4)
        validated_data = chaintrust_validator_function(layer_3_data)

    // Step 5: Swarm Intelligence for Threat Detection (Layer-5)
        swarm_results = swarm_intelligence_model_function(validated_data)

    // Step 6: Classification (Layer-6)
        classification_results = classification_model_function(swarm_results)

    // Step 7: Fine-Tuning the Model (Layer-7)
        fine_tuned_model = fine_tuning_model_function(classification_results)

    // Return the fine-tuned model
        RETURN fine_tuned_model
```

// ****** Update Logic ******

```
FUNCTION update_layers(update_layer):
  IF update_layer == 3:
      // Update Layer-3 and Layer-2
      update_layer_2()
      CONTINUE process at Layer-3

  IF update_layer == 5:
      // Update Layer-5, Layer-2, and go to Layer-6
      update_layer_2()
      update_layer_5()
      CONTINUE process at Layer-6

  IF update_layer == 6:
      // Update Layer-6, Layer-5, and Layer-2, then go to Layer-7
      update_layer_2()
      update_layer_5()
      update_layer_6()
```

CONTINUE *process at Layer-7*

FUNCTION update_layer_2():
// Update the SynerTrain Pipeline (Layer-2) logic

FUNCTION update_layer_5():
// Update the Swarm Intelligence Model (Layer-5) logic

FUNCTION update_layer_6():
// Update the Classification Model (Layer-6) logic