ReadME

Real-Time Anomaly Detection in IoT Networks Using Hadoop & Kafka

Overview

This project focuses on real-time anomaly detection in IoT networks using the **BoT-IoT dataset**. We trained a **Random Forest model** for binary classification to detect anomalies. The system uses **Kafka** for data streaming and **Hadoop** for distributed data storage.

Pre-requisites

Ensure the following software is installed on your system:

- **Java JDK (jdk1.8.0_202)**
- **Python 9.x**
- **Apache Kafka (kafka_2.12-3.9.0)**
- **Hadoop**
- **Zookeeper**

Project Structure

- **producer.py**: Reads raw data from local files and pushes it into a Kafka topic.
- **consumer.py**: Reads data from Kafka topic, applies prediction model and writes anomaly results into HDFS.
- **app.py**: Flask API to fetch data from HDFS.
- **dashboard**: HTML/CSS/JavaScript files for the UI.
- **trained_model.pkl**: Pre-trained Random Forest model for anomaly detection.

Setup Instructions

1. **Hadoop Setup:** Ensure that hdfs and yarn are running.

```
start-dfs
start-yarn
jps
hdfs namenode
start-datanode.cmd
```

7. **Start Zookeeper and Kafka Server:** Ensure that Zookeeper and the Kafka broker are up.

```
cd C:\kafka_2.12-3.9.0
.\bin\windows\kafka-server-start.bat .\config\server.properties

cd C:\kafka_2.12-3.9.0
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

8. **Create the necessary Kafka topics**: If not already done, create a Kafka topic that you will use to stream the data.

```
        cd C:\kafka_2.12-3.9.0\bin\windows
        kafka-topics.bat --create --topic iot-stream(topic name) --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
```

List topics (to check if topic was successfully created)

```
        cd C:\kafka_2.12-3.9.0\bin\windows
        kafka-topics.bat --list --bootstrap-server localhost:9092
```

9. **Run the Data Pipeline**

Run the Kafka producer: The producer is responsible for reading data from your local source and pushing it into the Kafka topic.

```
        cd C:\kafka_2.12-3.9.0
        python producer.py
```

Run the Kafka consumer: The consumer reads data from the Kafka topic and writes it into HDFS.

```
        cd C:\kafka_2.12-3.9.0
        python consumer.py
```

Additional Commands (FOR USER HELP)

```
# Delete existing files (if needed)
hdfs dfs -rm /Bot_Iot_dataSet/*.csv

# Upload files from local to HDFS
hdfs dfs -put <"local path"> <hadoop path>

# Verify files in HDFS
hdfs dfs -ls /Bot_Iot_dataSet

# Set permissions
hdfs dfs -chmod -R 777 /Bot_Iot_dataSet

#Set up environment variables
set NO_PROXY=localhost,127.0.0.1
set HTTP_PROXY=
set HTTPS_PROXY=

#Set proxy
set HTTP_PROXY=http://<proxy>:<port>
set HTTPS_PROXY=http://<proxy>:<port>
```