Lab4-Task1: Guided solution for Identify 300% Average Delay Anomalies.

Objective:
Identify flights for each airline (carrier) where the arrival delay of a specific flight is more than 300% of the average delay of all prior flights for that carrier up to the current flight's date.

Guided Solution:
Spark Environment Setup: Import Libraries:

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql import Window
```

Initialize Spark Session: Establish a local Spark session utilizing all available cores with 4GB memory allocation.

```
spark = SparkSession \
    .builder \
    .master("local") \
    .config("spark.driver.memory", "4g") \
    .appName('ex4_anomalies_detection') \
    .getOrCreate()
```

Define Window Specification: The window partitions data by Carrier and orders flights chronologically by flight_date.

```
all_history_window = Window.partitionBy(F.col('Carrier')).orderBy(F.col('flight_date'))
```

Load Data: Fetch the flight data from the specified S3 path and cache it for enhanced performance during subsequent operations.

```
flights_df = spark.read.parquet('s3a://spark/data/transformed/flights/')

flights_df.cache()
```

Calculate Historical Average Delay: For each flight of a specific carrier, compute the average delay of all prior flights including the current one. The result is stored in a new column named avg_till_now.

```
avg_delay_df = flights_df\
    .withColumn('avg_till_now', F.avg(F.col('arr_delay')).over(all_history_window))
```

Determine Delay Deviation:

For each flight, calculate the percentage difference between its arrival delay and the historical average delay until its date.

```
deviation_df = avg_delay_df\
    .withColumn('avg_diff_percent', F.abs(F.col('arr_delay') / F.col('avg_till_now')))
```

Filter Outliers:

Retain only the flights where the delay deviation is more than 300% (or 3.0 as a decimal).

```
outliers_df = deviation_df.where(F.col('avg_diff_percent') > F.lit(3.0))
```

Display Results:

Showcase the records that exhibit a substantial deviation from their historical average delay. Release the cached data and shut down the Spark session.

```
outliers_df.show()
flights_df.unpersist()
spark.stop()
```

This solution identifies flights with exceptional delays compared to the historical average of their respective carriers. Recognizing such anomalies can aid airlines in identifying and addressing operational issues, ensuring better future performance.