Stream Data mining: A survey

Mining data streams introduces new complications to the data mining industry, such as determining how to mine continuous high-speed data items that can only be inspected once. In this work, the author provides an overview of the constantly developing field of Data Streams. They go through the theoretical underpinnings of data stream analysis. There is a discussion of the many ways for mining data streams. The first methodology entails summarizing the entire data set or analyzing a subset of the incoming stream. Some of the tactics employed are sampling, load shedding, drawing, summary data structures, and aggregation. This post will go through these strategies briefly. The data set's size is uncertain, which poses issues with the sampling procedure while reviewing the data stream. This sort of methodology includes the approximation algorithm, the sliding window method, and the output granularity algorithm. Each of these approaches is investigated in the context of data stream analysis. The need to swiftly comprehend the huge volumes of data created every day has given rise to data stream processing, a novel data processing technology. Data is supplied in the form of continuous, high-volume, fast, and time-varying streams in this new paradigm, which must be analyzed in near real-time. The process of designing algorithms based on sketching techniques starts with computing drawings across an arbitrarily specified time window and constructing a sketch pool. This pool of drawings was used to determine relaxed periods and average trends, which were rapid and accurate. Data stream mining research has raised a host of issues and research challenges.

Why Because many actual data streams come at irregular rates, showing burstiness and fluctuating data arrival rates over time, optimize memory space and CPU resources when processing large data sets. Mining desired task variations in data sources and integrating them Transferring data mining discoveries across a low-bandwidth wireless network When dealing with continuous streams of data, high accuracy in outcomes is required. Real-world needs, including mobile device requirements. Another significant difficulty is determining how to efficiently store stream data with a timeline and retrieve it in response to user requests at particular time intervals. In a nutshell, the bulk of existing mining techniques use one-pass mining algorithms, with only a few addressing drifting. Because of limited memory, present approaches provide approximations.