StreamFlow: A System for Summarizing and Learning Over Industrial Big Data Streams

Mariam BARRY, Saad El JAOUHARI, Albert BIFET, Jacob MONTIEL, Eric GUERIZEC, Raja CHIKY

Contributions

StreamFlow - proposed data pipeline

Use sliding windows to summarize high-velocity data.

StreamFlow result - feature vector for machine learning tasks.

System deployed in a banking system.

Speedups in training time with good predictive performance.

Data

-network traffic data streams from BNP Paribas

-telecommunication logs

Related work

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	**StreamFlow**
Streams Aggregates					✓	✓					✔
Multiple Streams Fusion	✓	✓	✓			✓	✓		✓		✔
Categories Time-alignment					✓						
Real-world Industrial Data		✓	✓	✓	✓	✓		✓	✓	✓	✔
Industrial Big Data summurization					✓		✓	✓	✓	✓	✔
Online Machine Learning Applications		✓						✓	✓		✔

Challenges

Big Data Processing

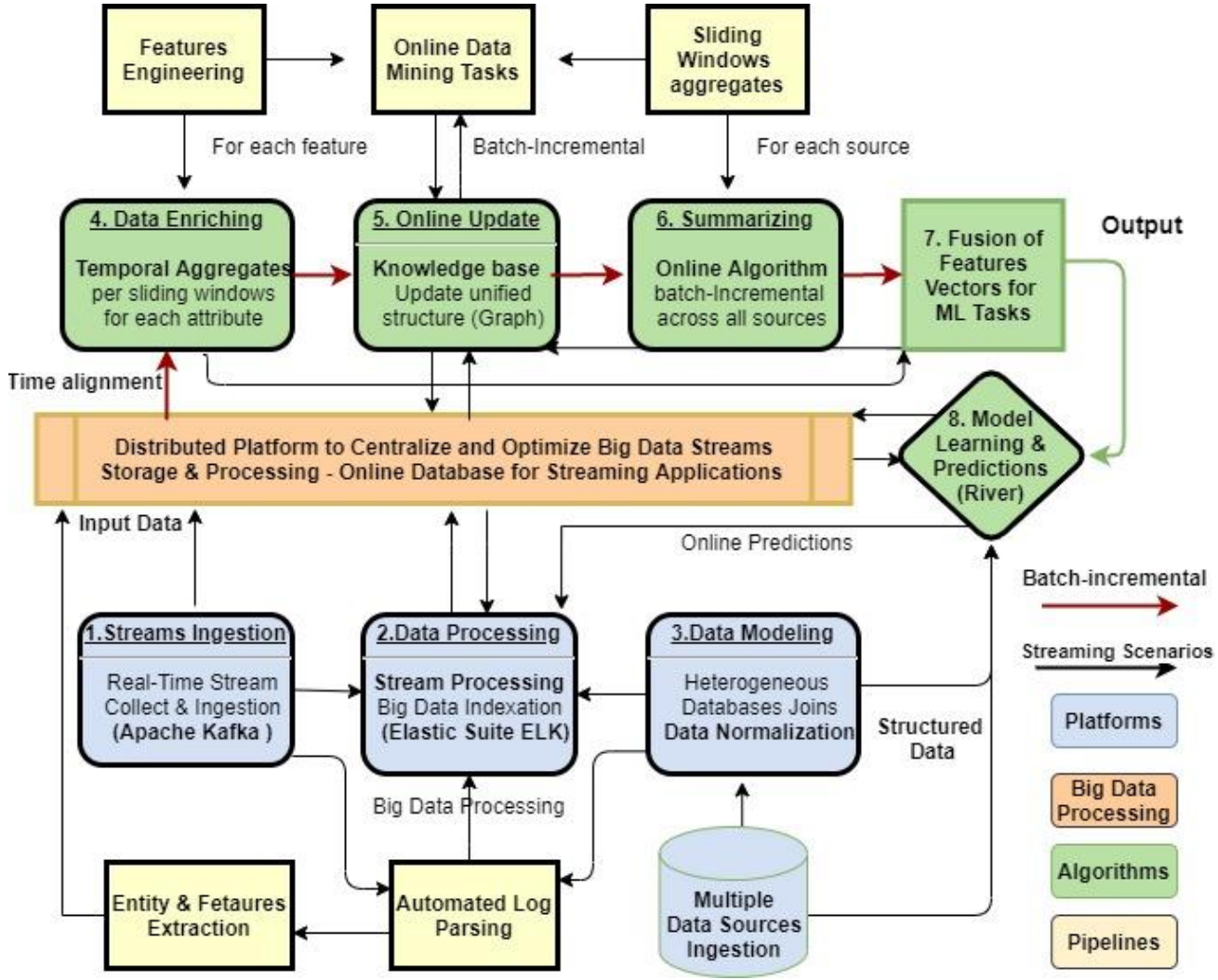
- Collect asynchronous events from multiple data sources to get a structured information
- Collecting High-velocity streams & Fine-tuning Logs pipelines, windowing strategy
- Dealing with ressources allocation across distributed platform computing partial tasks

Big Data Summarizing

- Optimizing big data clusters & partitioning ressources AIOps & IT process in Real-time
- Industrialization of a online learning model which process events summaries incrementally
- Updating a model with asynchronous batch & streams events using big data summaries

Big Data Interoperability

- Building & scaling automated Data Pipelines connecting Big Data tools (ELK, Kafka, Flink..)
- Serving (MLOps) online models to continuously learn from incremental vectors
- Set up real-time data pipelines from streams collection, storage, modeling to predictions


Features Engineering
Online Data Mining Tasks
Sliding Windows aggregates
For each feature
Batch-Incremental
For each source
4. Data Enriching
Temporal Aggregates per sliding windows for each attribute
5. Online Update
Knowledge base Update unified structure (Graph)
6. Summarizing
Online Algorithm batch-Incremental across all sources
7. Fusion of Features Vectors for ML Tasks
Output
Time alignment
Distributed Platform to Centralize and Optimize Big Data Streams Storage & Processing - Online Database for Streaming Applications
8. Model Learning & Predictions (River)
Input Data
Online Predictions
1.Streams Ingestion
Real-Time Stream Collect & Ingestion (Apache Kafka)
2.Data Processing
Stream Processing Big Data Indexation (Elastic Suite ELK)
3.Data Modeling
Heterogeneous Databases Joins Data Normalization
Structured Data
Big Data Processing
Entity & Fetaures Extraction
Automated Log Parsing
Multiple Data Sources Ingestion
Batch-incremental
Streaming Scenarios
Platforms
Big Data Processing
Algorithms
Pipelines

Elasticsearch overview

Last 15 minutes | Show dates | Refresh

Overview | Nodes | Indices | Machine learning jobs | CCR

Status	Nodes	Indices	JVM Heap	Total shards	Unassigned shards	Documents	Data
Green	21	235	46.4 GB / 84.0 GB	928	0	50,129,460,773	21.5 TB

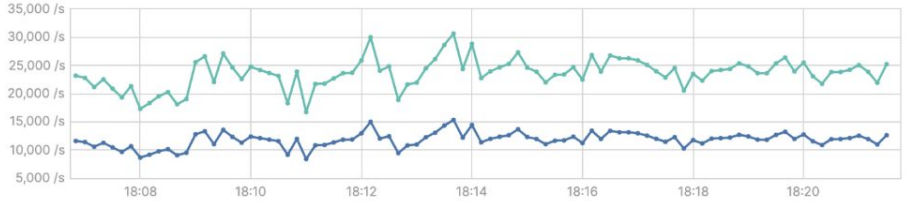
Search Rate (/s)

Total Shards 56.2 /s

Search Latency (ms)

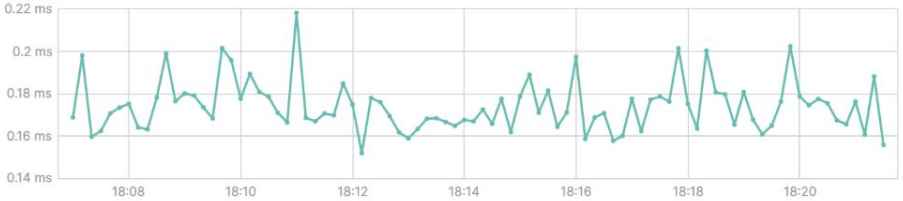
Search Latency 0.69 ms

Indexing Rate (/s)

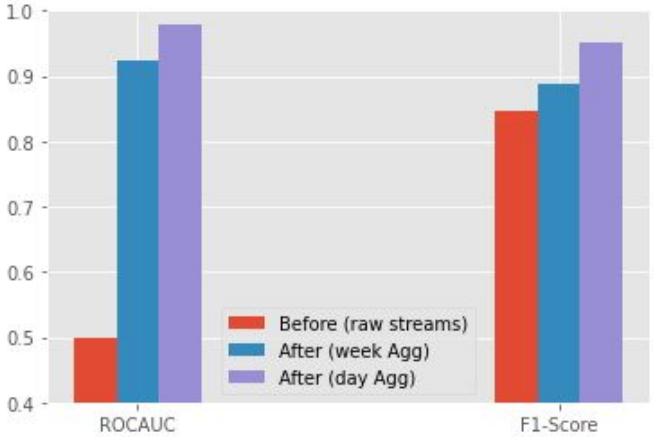


Total Shards 25,222 /s | Primary Shards 12,614.5 /s

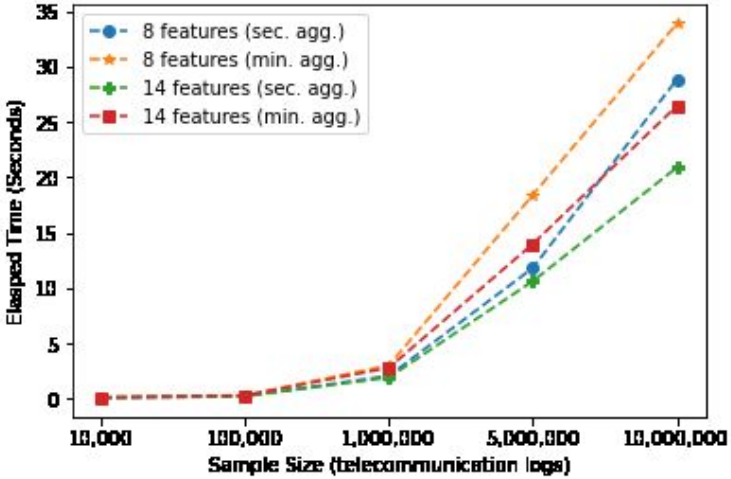
Indexing Latency (ms)



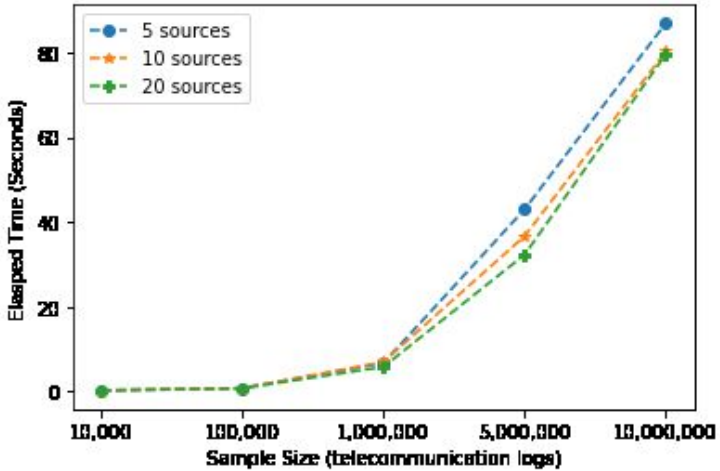
Indexing Latency 0.16 ms



Predictive performance improvement of linear regression model.



Impact of number of features on time.



Impact of number of sources on time.

	Linear Regression			CART / Decision Trees			Random Forest			ARF	HT
	ROCAUC	F1	Time (s)	ROCAUC	F1	Time (s)	ROCAUC	F1	Time (s)	F1	F1
BEFORE - Raw big data[∼1.5M]	0.500	84.53	1.70	0.987	**99.23**	4.22	0.987	**99.23**	245.61	93.84	96.33
AFTER - Minute (Week) [∼90K]	0.922	88.86	1.18	0.981	96.59	0.30	0.983	97.31	6.73	**98.23**	96.17
AFTER - Second (Day) [∼20K]	**0.978**	**95.18**	**0.54**	**0.992**	97.99	**0.06**	**0.995**	98.58	**1.57**	94.20	**97.42**