

DELFT UNIVERSITY OF TECHNOLOGY

MASTERS THESIS

---

# The State of Data Streaming Practices at ING

---

Author:

Kanya Paramita KOESOEMO

Supervisor:

Dr. Asterios KATSIFODIMOS

*A thesis submitted in fulfillment of the requirements  
for the degree of Master of Science  
in the*

Web Information Systems Group  
Software Technology

November 19, 2021

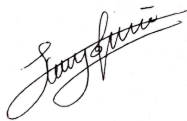


## Declaration of Authorship

I, Kanya Paramita KOESOEMO, declare that this thesis titled, "The State of Data Streaming Practices at ING" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: \_\_\_\_\_



Date: 19-11-2021  
\_\_\_\_\_



*“Strive not to be a success, but rather to be of value.”*

Albert Einstein



DELFT UNIVERSITY OF TECHNOLOGY

# *Abstract*

Electrical Engineering, Mathematics and Computer Science  
Software Technology

Master of Science

## **The State of Data Streaming Practices at ING**

by Kanya Paramita KOESOEMO

The development of data stream processing has become one of the key themes in the database and distributed system community throughout the world as data has grown on a large scale and in a range of industries over the last several years. Because data stream processing is a relatively new breakthrough in data-driven approaches, several teams at ING are investigating its possibilities. Thus, this thesis aims to provide insight on data stream processing practices at ING using research survey methodology. We conducted an extensive study that included a review of data streaming academic publications, online questionnaire distributed to 45 practitioners at ING, and in-depth interviews with 5 streaming practitioners. Our survey research aimed at understanding: (i) the use cases of data streaming; (ii) the types of streamed data users have; (iii) the streaming tasks and computation users run on their stream; (iv) the machine learning task users performed in their streams; and (v) the streaming software and tools used to process their streams. Results from academic review became the basis of designing the questionnaire. We discussed the answers of the participants to our questionnaire by highlighting common trends and challenges they faced. Through our interviews, we were able to get detailed answers on some of our questions. Our research discovered several interesting observations regarding data stream processing in practice. Particularly, real-time monitoring and event categorization are the popular use case for data streaming, data contained in streams represent a diverse range of entities and is homogeneous in format, type and category, machine learning implementation in streaming environment is prevalence, Apache Kafka is a commonly used stream processing engine and complexity of data streaming implementation is the challenge most expressed by our participants.

### *Thesis Committee:*

<i>Chair:</i>	Full Prof. Dr. Arie van Deursen, TU Delft
<i>University Supervisor:</i>	Assistant Prof. Dr. Asterios Katsifodimos, TU Delft
<i>University Supervisor:</i>	Georgios Siachamis, TU Delft
<i>Committee Member:</i>	Dr. Marios Fragakoulis, TU Delft
<i>Committee Member:</i>	Jerry Brons, ING





## *Acknowledgements*

This thesis is a great accomplishment for me and it has been quite an experience to do this research especially since this topic is new for me. Nevertheless, this research has made me learned many new things and acquired better endurance in facing uncertainties. I hope this work can be useful for other researcher and to anyone who read it.

I would like to express my appreciation to my supervisor, Asterios Katsifodimos, for the encouragement, advice, and trust that he put on me to do this thesis research. I would also like to hugely thank my PhD mentor, Georgios Siachamis, for his endless guidance, feedback and discussion during the process of this thesis. I would not be able to go this far without his support throughout my thesis journey. Both of them have been such a great listener to me whenever I have anything to share.

I would like to give my special thanks to Jerry Brons and Elvan Kula for giving me the opportunity to do my thesis at ING and of course all the support and help they have given me to conduct my research within the company. I am grateful I had the chance to do research on how data streaming being practiced in the industrial environment — doing industrial research is something that I always wanted to do. I would also like to thank my fellow colleague and students in AI For Fintech Laboratory for making my internship time at ING fun, inspirative and collaborative during our (remote) working time together.

I would like to thank my mother, Putri, and sister, Mutia, for unconditionally giving their endless support from afar. They were always ready to be reached out whenever I needed them and believed in me whenever I had doubts on myself. If it was not for them, I would not have made it this far.

Last but not least, I want to give my special appreciation to Manisha, Sukhleen, and Cor-Jan as my fellow thesis warriors who had been such a great company through my struggle on staying positive during the pandemic. I would also like to thank my friends Ginta, Asmita, Dira, Farizky, Diego, Sulton, Sho, Fauza and Azza — our friendships are deeply meaningful to me throughout my university days.

Kanya Paramita Koesoemo  
Delft, The Netherlands  
November 19, 2021



# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Questions . . . . .	2
1.2 Contributions . . . . .	2
1.3 Outline . . . . .	3
<b>2 Background and Related Work</b>	<b>5</b>
2.1 Data Streaming & Stream Processing . . . . .	5
2.2 Survey Studies on Stream . . . . .	8
2.2.1 Search Strings . . . . .	8
2.2.2 Retrieval Steps . . . . .	9
2.2.3 Findings . . . . .	9
<b>3 Survey Framework</b>	<b>13</b>
3.1 Academic Publications Review . . . . .	14
3.2 Online Questionnaire . . . . .	14
3.3 Individual Interviews . . . . .	15
<b>4 Academic Publication Review</b>	<b>17</b>
4.1 Scope Definition . . . . .	17
4.2 Identification of Studies . . . . .	18
4.3 Keywording . . . . .	22
4.4 Data Extraction . . . . .	22
4.5 Result Mapping & Discussion . . . . .	24
<b>5 Questionnaire &amp; Interview Approach</b>	<b>33</b>
5.1 Questionnaire Methodology . . . . .	33
5.2 Interview Methodology . . . . .	37
<b>6 Results</b>	<b>43</b>
6.1 Stream versus Non-Stream Practices . . . . .	43
6.2 Demographics of Survey Participants . . . . .	46
6.3 RQ1: Streaming Use Cases . . . . .	49
6.4 RQ2: Streamed Data Characteristics . . . . .	51
6.5 RQ3 & RQ4: Streaming Task & Machine Learning Computation . . . . .	55
6.6 RQ5: Streaming Software & Tools . . . . .	57
6.7 Challenges & Expectation in Streaming Practices . . . . .	58
6.8 Observations . . . . .	60
6.9 Reflection on Survey . . . . .	61

<b>7 Conclusion</b>	<b>65</b>
7.1 Summary . . . . .	65
7.2 Future Work . . . . .	67
<b>A Data Stream Processing Questionnaire</b>	<b>69</b>
<b>Bibliography</b>	<b>75</b>

# List of Figures

2.1	Stream processing topology as a directed acyclic graph . . . . .	6
2.2	A venn diagram that illustrates the correlation between Data Stream Processing and other concepts that are commonly misinterpreted as stream processing . . . . .	7
2.3	Growing numbers of survey research papers on stream from 1997 - June 2021 . . . . .	10
2.4	Number of survey research works done for each survey themes . . . . .	10
3.1	Diagram of our survey process main steps where the arrow shows what kind of information are being passed between the steps. . . . .	13
4.1	Systematic literature review process of academic publications in data stream processing. . . . .	17
4.2	Results of each steps in the papers selection process. . . . .	20
4.3	Taxonomy of identified data operator streaming tasks and its technique	28
4.4	Taxonomy of identified general streaming tasks and its technique . . . . .	28
6.1	Distribution of the number of respondent who practiced data stream processing and who did not . . . . .	44
6.2	Distribution of the reason stated by the respondent on why they used data stream in their work . . . . .	45
6.3	Distribution of the reason stated by the respondent on why they did not use data stream in their work . . . . .	46
6.4	Distribution of the department at ING where the participants worked at	47
6.5	Distribution of the size of our participants' working team . . . . .	48
6.6	Distribution of the data streaming knowledge level of the participants who used streams . . . . .	48
6.7	Distribution of the streaming use cases at ING that our questionnaire participants worked on . . . . .	49
6.8	Distribution of the questionnaire response for data format in streams .	53
6.9	Distribution of the questionnaire response for type of data in streams data points . . . . .	54
6.10	Distribution of the questionnaire response for their streams data source	55
6.11	Distribution of the respondents' answer for the streaming tasks they performed . . . . .	56
6.12	Distribution of the respondents' answer for stream processing engines that they used . . . . .	58



# List of Tables

2.1	Number of papers resulted from each retrieval steps . . . . .	9
4.1	Research questions for the systematic literature review of academic publication. . . . .	18
4.2	Retrieved papers from the study search process in 4 top conferences in database management field. . . . .	19
4.3	Properties of the study as keyword references . . . . .	23
4.4	Data streaming applications and example use of stream in various fields identified from academic papers . . . . .	25
4.5	Comparison of stream processing engines . . . . .	31
5.1	Mapping our the questionnaire's sections to research questions and properties resulted from academic publication review . . . . .	34
5.2	Overview of our questionnaire's questions . . . . .	36
5.3	Interview questions guideline with estimated time per section and related research questions . . . . .	38
5.4	Overview of the work area of the interviewees . . . . .	38
5.5	Overview of the duration of recorded interviews written in hh:mm:ss format . . . . .	39
5.6	A dummy example of quotation with the assigned codes and commented content inferred from the sentence . . . . .	40
5.7	Overview of number of codes processed from each interviews . . . . .	40
5.8	Overview of the process of the code grouping for each themes . . . . .	41
6.1	Distribution of the job role of our participants . . . . .	47
6.2	Distribution of the answer of number of streams users . . . . .	51
6.3	Overview of the answer for entities represented in data within streams from both questionnaire and interview . . . . .	51
6.4	Distribution of the answer of average streams throughput . . . . .	52
6.5	Distribution of the answer of maximum streams throughput . . . . .	53
6.6	Distribution of the answers of streamed data category . . . . .	54
6.7	Distribution of the participants response on implementing machine learning computation within their streams . . . . .	57





*Dedicated to my grandmother, Resi, and my mother, Putri,  
who inspire me to pursue my academic goals...*



## Chapter 1

# Introduction

Breakthroughs in information technologies have enabled massive-volumes of high-speed data and the capacity to continually retain data. Computing large amount of information is a new trend in future computing due to its nature in volume, velocity, and diversity. This phenomenon has been leading to various computing challenges such as the increasing needs to process data in real-time. Hence, data streaming practices have recently been prominent since a variety of applications produce a significant volume of data at high speed.

Despite the huge gains of conventional databases throughout the last few decades, a modern application paradigm, fast-growing amount of data and data diversity presented a powerful challenge to it. As stated by Fragkoulis et al., 2020 that data has developed on a massive scale and in a variety of fields during the last 20 years, the development of data stream processing became one of the leading subjects in the database and distributed system community around the world. Data stream processing is the real-time processing of large amounts of data delivered at high velocity from multiple sources with low latency. With its promising benefits, streaming data has become a common application model in a variety of fields, including financial services, entertainment and media platforms, communication data management, network monitoring, and so on.

In financial services industry especially, innovation and implementation of streaming technologies has been prominently desired. Since the presence of data in the modern bank has grown dramatically, a number of financial services companies were quick to recognize the value of being event driven. In response to market fluctuations, customer behaviour, and a change in regulatory standards, banks and other financial institutions are undergoing a comprehensive digital transformations. One of the primary reasons for this shift is that data stream processing enables financial services organisations to respond to information in a whole new way by using real-time insight and data-driven technologies that enable companies to react to changes and notifications in real time.

As a large bank in the Netherlands, ING is one of the early adapters of data streaming practices in financial service industry. By the second quarter of 2021, ING served around 38.5 million customers, corporate clients and financial institutions across more than 40 countries<sup>1</sup>. As an organization, ING is highly digital with 90 percent of the interaction with its primary customer were digital<sup>2</sup> and its ability to process 209 million mobile payment transactions<sup>3</sup>. Thus, data streaming processing is recognized as an important capability for ING's data driven ambition.

---

<sup>1</sup><https://www.ing.com/MediaEditPage/ING-profile-2Q2021.htm>

<sup>2</sup><https://www.ing.com/MediaEditPage/ING-profile-1Q2021.htm>

<sup>3</sup><https://www.ing.com/MediaEditPage/Factsheet-2Q2021.htm>

Data stream processing is a relatively recent innovation in data driven approaches and therefore many teams at ING are exploring its potential. Within the organization, teams are able to investigate and innovate technology used to solve use cases and interest on data stream practices were growing. As the early agent of data stream technologies, the journey started within ING when a solution was needed to improve one of their use case for online banking. They were able to improve the system to be less expensive, resilient and easier to scale up in order to process the ever growing amount of data needed for producing real-time result. Processing data in streaming fashion was a perfect solution to solve these issues and it became an essential component in the organization's data pipeline.

Despite the ubiquity of appeal for data streaming practices, there has not been a research on how data stream processing is actually used in practice by people who work in the industry. Thus, this thesis aims to provide insight on data stream processing practices in the industry using survey methodology. We intended for these remarks to be a useful guideline on possible future research in data stream processing area and a beneficial overview of data streaming practices for ING to make further decision on stream development within the organization.

## 1.1 Research Questions

To facilitate the understanding of data streaming practices in the industry area, we address in this work five research questions that focus on common aspect in engineering practices such as use case, data, computational tasks, and software used. Thus, the research questions of our research are:

- *RQ1: What use cases do users implement their streaming pipeline for?*
- *RQ2: What types of streamed data do users have?*
- *RQ3: What kind of streaming task & computations do users run on their streams?*
- *RQ4: Which machine learning task users perform in their streaming pipeline?*
- *RQ5: What software & tools do users use to perform their streaming processes?*

## 1.2 Contributions

To the best of our knowledge, our research was the first to focus on doing survey research of data streaming practices in the industry. As a first step towards a better understanding of data streaming practices, this thesis make contributions of the followings:

- Insights and observations on the practices and challenges on data streaming implementation in the industry that can be used as guidance for future researches
- Re-usable questionnaire framework on data streaming practices that can be extended by future possible researcher
- Systematic academic publication review on the topic of data streaming practices within the researchers
- Comparison between observations obtained on data streaming practices within researchers and practitioners

## 1.3 Outline

The remainder of this thesis is structured as follows. In Chapter 2 we discuss background information and related work on streaming survey research. In Chapter 3 we describe the survey framework where we explain the process of each survey steps. In Chapter 4 we present the academic publication review where we explain the process of papers selection, analysis of the selected paper and the result of the review. Chapter 5 presents the approach we did in carrying out the questionnaire and interview for our research. The results and findings of both questionnaire and interview are discussed in Chapter 6. Here we also explain our reflection on the questionnaire and interview methodology and its results. In Chapter 7 we conclude our work and present directions for future work.



## Chapter 2

# Background and Related Work

In this section, background on data streaming and stream processing is explained. Related work on survey researches in data streaming fields and studies that carried out extended survey work are also being presented.

### 2.1 Data Streaming & Stream Processing

*Data streaming* is an environment where data is being generated continuously from one or more data sources which are implicitly ordered by arrival time or by timestamps, with the purpose of being processed in near real-time. [Golab and Özsu, 2003]

The sources of data itself are typically massive and boundless such as device usage statistics, system networks, internet user activity logs, and online transactions. The data fed by these sources is contained in tuple units, where each tuple is a definite collection of key-value pairs consisting of a key (the unique identifier being used to access the value) and a value (the content of to be retrieved) that may be adjusted to represent complicated data structures [Querzoni and Rivetti, 2017]. **Streams** in this environment, or data streams, can be defined as sequences of unbounded tuples that are generated continuously and have some notion of timely order [Botan et al., 2010].

*Stream processing* refers to the processing of large volumes of data in real-time manner with minimal latency, immediately after the data is generated at high velocity from many resources. [Kolajo, Daramola, and Adebisi, 2019].

Unlike traditional data processing, which applies the process after the data is saved, stream processing fundamentally reverses the order of the entire method by allowing for data to be processed as it flows with faster processing time. Gomes et al., 2019 explained three top advantages of data stream processing. First, stream processing provides significantly faster insight to its users as it offers continuous data pipeline between all moving parts of a company and the individuals who make decisions. Secondly, stream processing improves the efficiency of business operational because the real-time nature of stream processing enable its users to react and respond to crisis events much quicker than any other data processing methods. It gives continuous data that keeps the operation running. Lastly, it gives the possibility to process enormous amount of data with less IT infrastructure cost because streaming processing utilizes ingestion cloud services that can help your organization manage workloads efficiently by auto-scaling the clusters to optimize costs.

These benefits of implementing stream processing are made possible by some of its important properties. Turaga et al., 2010 elaborated four important properties that characterize stream processing applications. These properties are:

1. Continuous Data Sources — Data sources of stream processing applications should generate data that flows constantly that possibly has no end and contains either organized or unstructured data.
2. Continuous and Long-Running Analysis — To generate continuous stream of output results, data that is being injected needs to be processed on-the-go. Thus, analytic processes need to be done in real-time and in an incremental manner to handle the streamed data items.
3. Time-To-Respond Performance Requirements — Stream processing must be able to match the data input rates to offer results as fast as possible in a quality that is as high as possible while responding to dynamic changes in the system and data.
4. Failure Tolerance Requirements — Stream processing applications have to be able to manage internal state of long-running processes and cope with data issues such as data loss, corrupted data and out-of-order data.

In their work, Querzoni and Rivetti, 2017 and Kolajo, Daramola, and Adebisi, 2019 also explained that applications of stream processing can be identified by its topology that can be represented as directed acyclic graph (DAG) as shown in Figure 2.1. Data *streams* are represented as the graph's edges and stream *operators* are represented as nodes. Streams transfer data from one operator to another. Operators are executing the processing tasks such as filtering, parsing, data normalization, feature extraction, duplicates removal and so on. There are 2 special types of operators that any stream processing topology must contain which are the *source*, the operator that connects to data source, and the *sink*, the operator that connects to an external system that consume the stream processing results.

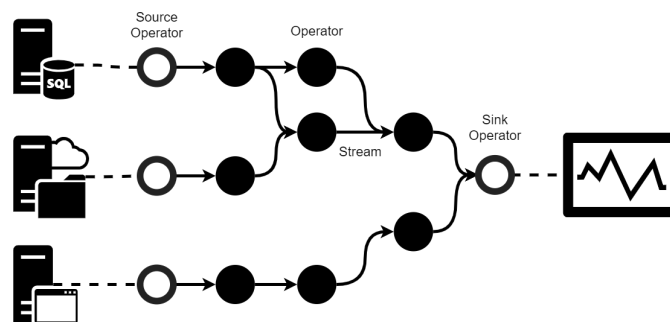


FIGURE 2.1: Stream processing topology as a directed acyclic graph

However, there are several other processing concepts that commonly misunderstood as data stream processing which are Complex Event Processing, Real-Time Operating System, Batch Processing and Event Monitoring. In the following, the description of these concepts and their distinction with data stream processing is explained. Figure 2.2 shows how Data Stream Processing and these other seemingly similar concepts can partly intertwined with each other but not completely the same.

*Complex Event Processing*, according to Luckham, 2008, is a defined set of tools and techniques for analyzing and controlling the complex series of interrelated events that drive modern distributed information systems. It is the method of processing



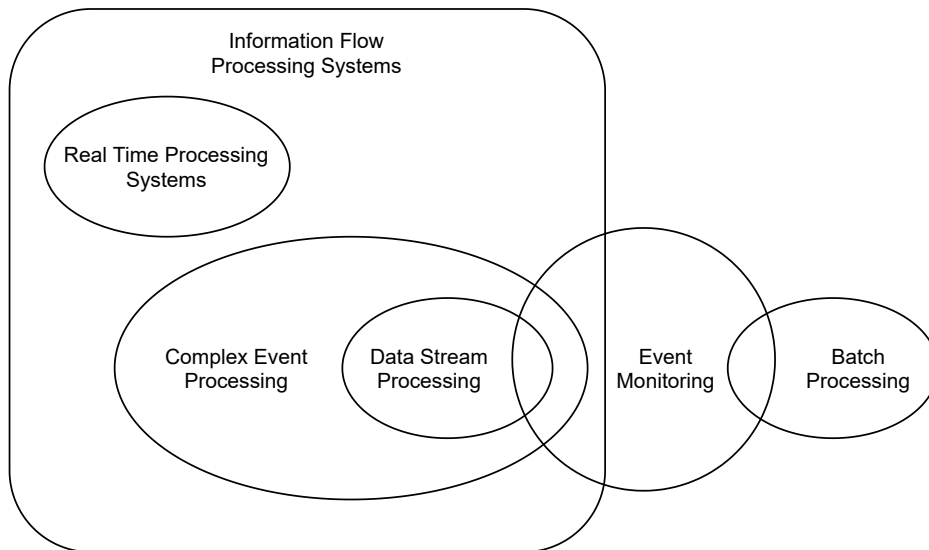


FIGURE 2.2: A venn diagram that illustrates the correlation between Data Stream Processing and other concepts that are commonly misinterpreted as stream processing

multiple streams of events and correlating seemingly unrelated events to produce new insights for the business domain. Although the terms Data Stream Processing and Complex Event Processing are frequently used interchangeably, they are not entirely synonymous. While both are dealing with streams of data, Data Stream Processing focuses on providing a solution for real-time data query processing that handles only generic data without identifying related events and Complex Event Processing focuses more on associating semantics with the data so that the system is able to detect and understand correlating events [Zhao et al., 2017]. Complex Event Processing systems put great emphasis on the ability to detect complex patterns of incoming events that involve sequencing and ordering relationships between the events [Cugola and Margara, 2012].

*Real-Time Operating System* is a type of operating system designed to support real-time applications that process events data that must comply with strict time limits in order for the system under control to function properly [Stankovic and Rajkumar, 2004]. Because all processing task must be done within the defined constraints or the system will fail, as explained by Stankovic and Ramamritham, 1995, Real-Time Operating System focuses on monitoring the relevant priority of competing tasks dynamically and able to make changes to the tasks priority accordingly. Some examples of the real-time operating systems are airline traffic control systems, airlines reservation system, and stock market real-time systems. These Real-Time Operating Systems can be misinterpreted with Data Stream Processing practices because of its real-time response ability. The difference of Data Stream Processing to Real-Time Operating Systems is that Data Stream Processing perform continuous computation to data as it flows through the systems without critical time deadline on when the output should be produced. The only constraint on Data Stream Processing is that its stream output rate should be faster or equal to the data input rate [Shahrivari, 2014].

*Batch Processing* is the processing of a large volume of data by performing batches of jobs in a non-stop sequential order within a specific time span [Martin et al., 2015]. There is actually a huge difference between Batch Processing and Data Stream Processing in the way they work. Batch Processing refers to processing data that is

collected first in batch fashion while Data Stream Processing processes continuous stream of data immediately as it is being produced [Shahrivari, 2014]. While Data Stream Processing analyze the data in continuous fashion and provide the response immediately after the data processed, Batch Processing analyze data in snapshots and provide response after the completion of the batch job [Chang, Damodaran, and Melouk, 2004]. Despite these notable differences, Batch Processing practices can be confused with Data Stream Processing when the time span used to run batches of job are really small that it seems like the data is being processed in real-time manner.

*Event Monitoring* in IT, based on the definition from Klar, 1992, is the process of detecting, collecting and signaling occurrences of specified events to operating system processes, active database rules, and human operators where the event occurrences may stem from software or hardware systems. Event Monitoring practices are often misinterpreted as Data Stream Processing practices because both of them hold the same purpose which are to present event-based data in real-time manner so that immediate action can be taken. Even though Event Monitoring could be the use case of Data Stream Processing, it might not always utilize data streaming concept to serve it purpose. It focuses more on the event detection technique thus various method are allowed in its event collection and processing steps such as batch processing [Demers et al., 2007], polling machine [Mansouri-Samani and Sloman, 1997], message broker [Moser, Rosenberg, and Dustdar, 2010], etc.

## 2.2 Survey Studies on Stream

Before trying to address our research questions through our user study, we performed a literature study. Our goal was to collect previous surveys on data streaming technologies, identify possible research gaps and acquire an overview of the current research state in the field. With this overview, we got an insight of which topics or problems that has and has not been surveyed.

To find and collect previously performed surveys on streaming technologies, we opted to use Google Scholar<sup>1</sup>. After experimenting with other options, using Google Scholar proved to be the most efficient way for us. Doing independent searches in different publisher databases required more time in execution while produced less exhaustive search result compared to using Google Scholar. We used a software that obtains and evaluates academic citations and its metadata from Google Scholar. We collected information of the publication such as raw citations and provides academic paper information, including the title, authorship, year published, number of citations, abstract, etc.

### 2.2.1 Search Strings

We first needed to figure out which keyword combination would retrieve the best result of papers about stream survey research. A test set of papers were created from papers referenced by the most cited stream survey research paper. A set of potential combination of keywords were collected from this set such as "stream", "survey", "data stream", "stream processing", "data flow", etc. We then ran the combinations of keywords on the test set to see which combinations produce the most accurate list of papers. These test run resulted in 4 combinations of keywords. Finally, we

---

<sup>1</sup><https://scholar.google.com/>

TABLE 2.1: Number of papers resulted from each retrieval steps

Keyword Combination	Search	Clean	Filter
Combination 1	193	83	55
Combination 2	78	52	47
Combination 3	41	16	14
Combination 4	12	6	4
Total	324	157	<b>120</b>

used these 4 combinations of keywords as the search strings and ran an independent search process for each combinations. The keywords combinations are:

- Combination 1: ("data" OR "stream" OR "survey"),
- Combination 2: ("data" OR "stream" OR "review"),
- Combination 3: ("stream" OR "processing" OR "survey"), and
- Combination 4: ("stream" OR "processing" OR "review").

### 2.2.2 Retrieval Steps

The retrieval process was divided into 3 steps which are publication searching, cleaning, and filtering.

*Search* — In this step, we first did a pilot search to see which paper properties should we use to retrieve papers accurately and effectively. For the pilot, we used the same test set for the search string to test each combination of paper properties. We then found out that using only title properties in the search would create the most accurate result. To obtain the first list of papers, we performed a search for papers with title that contain these 4 combination of keywords resulting and put them all to our review list.

*Clean* — Papers from the following categories were omitted from the review list: (i) papers that are duplicates, (ii) papers in other than English authored in source language, (iii) papers with incomplete information of title, year published, and abstract, and (iv) papers that doesn't have full-text availability.

*Filter* — The last step is filtering the unrelated articles from the list resulted from the cleaning. We omitted papers with topic that is not about data streams or stream processing as defined in Section 2.1

### 2.2.3 Findings

Table 2.1 shows the numbers of papers resulted from each steps and for each keyword combinations. The original search results gave us a list of 324 papers with almost half of them reduced in the cleaning steps to 157 papers. After curating the relevance of the paper with our topic scope, we got 120 papers in our final list.

Figure 2.3 shows the trend of number of survey articles on stream topics from 1997 until 2021 (June). The chart illustrates a growing interest throughout the years in performing research surveys on stream related topics. From 2012 to 2013, there was a substantial increase in particular. While we saw less articles in 2016, 2017 and some 2021, it still shows an overall increased trend on stream survey researches in the last decade.

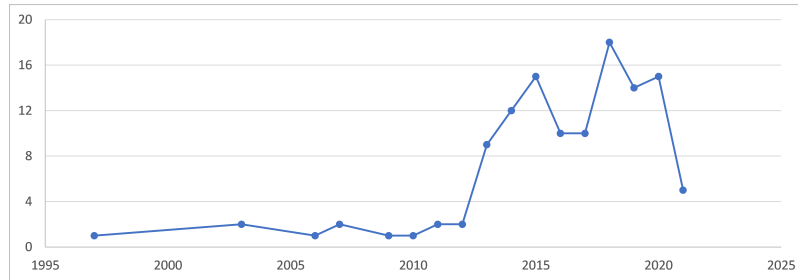


FIGURE 2.3: Growing numbers of survey research papers on stream from 1997 - June 2021

There are substantial variety in the themes of the surveys that have been conducted. The themes of the survey works can be categorized into Stream Processing Techniques, Stream Processing Tools, Stream Analytics, Use Case Studies, and Challenges in Data Streams. One survey work can hold one or more themes. For each of these themes, Figure 2.4 present the number of survey paper that focus on a particular theme.

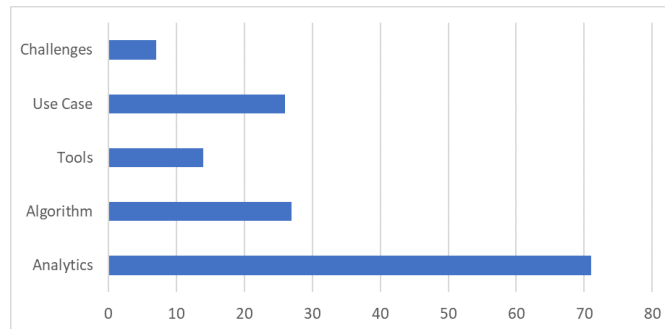


FIGURE 2.4: Number of survey research works done for each survey themes

**Surveys on Stream Processing Techniques** — Dias de Assunção, da Silva Veith, and Buyya, 2018 conducted a survey on state of the art stream processing mechanisms and described how existing solutions exploit resource elasticity features of cloud computing in stream processing. Röger and Mayer, 2019 established a categorization of existing methods for both parallelisation and elasticity in stream processing systems while taking into account various aspects such as system type, programming model, and memory architecture. Liu and Buyya, 2020 produced a comprehensive taxonomy in the context of resource management and scheduling, covering critical research topics such as resource provisioning, operator parallelisation, and task scheduling. These studies focused on techniques for processing data streams. However, none of these survey papers elaborated the use case of processing the streams and how was it related to the techniques they used.

**Surveys on Stream Processing Tools** — A systematic literature review was performed by Kolajo, Daramola, and Adebisi, 2019 where they presented stream processing engines and compared them based on 10 characteristics such as database support, implementation language, application domain, and execution model. Hesse and Lorenz, 2015 analyzed and compared certain data stream processing systems in relation to their architecture and features such as latency, throughput, and message processing. Isah et al., 2019 did a comparative study of distributed data stream processing engines and presented a taxonomy and a critical review of representative

open source and commercial distributed data stream processing frameworks. These works compared stream processing tools based on their features but they did not consider how users utilize the tools and their experience using them.

**Surveys on Stream Analytics** — Surveys on stream analytics were the most popular amongst other themes. These surveys focused on approaches for analysing data contained in the stream, covering topics such as data stream mining (specifically clustering and classifications), outlier detection and concept drift. In their work Ikononovska, Loskovska, and Gjorgjevikj, 2007 examined the theoretical basis for mining a data stream, critically review techniques of data stream mining, and highlight some general problems in this topic. Nguyen, Woon, and Ng, 2015 produced a review paper that offers an extensive overview of the state-of-the-art data stream mining techniques focusing on classification and clustering that highlights mining constraints, proposes a general data stream mining model, and assess benefits and limitations of the algorithms. A survey done by Silva et al., 2013 provided a study of data stream clustering algorithms that reviews the principal design elements of the state-of-the-art algorithms and provides an overview of the commonly used experimental techniques. However, we had not yet found any surveys that examined how these analytic approach being utilized in practice.

**Stream Use Case Studies** — Several surveys focused on some specific applications of streaming technologies. The use cases vary from smart city, mobile networking, educational technologies, privacy preserving, and the most popular one is social media analysis. A study performed by Hasan, Orgun, and Schwitter, 2018 offers a review of a large number of techniques of event detection for Twitter data streaming by categorizing the methods according to common characteristics and then analyzes various elements of the subtasks and challenges existed in event detection. Baccarelli et al., 2016 did a survey on processing of big data streams from resource-limited mobile/wireless devices about its potential applications and key problems of the resource management. The work of Nasiri, Nasehi, and Goudarzi, 2018 presented a review of the stream processing frameworks applicability in data processing layer of Smart City. We had not yet found a survey on data streaming applications related to the financial field.

**Surveys on Data Streaming Challenges** — A study performed by Golab and Özsu, 2003 examined recent work on data stream management systems with a focus on several issues including data models, continuous query languages, query evaluation and optimization techniques. Tidke and Mehta, 2018 stated two open issues in real time processing of big data which are the need of real-time analytics framework that can transform data into decisions and distributed data mining algorithm for analyzing accurate information. A systematic literature review of challenges and its solutions for processing real-time big data stream was done by Mehmood and Anees, 2020 where they identified some key challenges such as in-memory computing, support to semi-structured data streams, machine learning algorithms on un-structured big data, effective resource allocation, and other. These studies presented open challenges in various aspect of data streaming but none focused on the challenges of implementing data streaming concepts in practice.

In summary, the existing survey research on data stream processing focused on its theoretical aspects and none of them explored the practical aspects of it. Examining how a technology is being practiced in the industry is important for closing the gap between academic and industry and it is something that is missing from the stream processing literature. Thus, our survey research would provide an interesting insight to the research community as it analyzes how data stream processing is being implemented by the practitioners from the industry area.



## Chapter 3

# Survey Framework

In what follows, we will discuss extensively the methodology that we used to perform our survey. Taking into consideration that our survey differs greatly from existing surveys on data streams, we opted to follow approaches from other research topics. Therefore, we chose to design our survey methodology based on the survey framework presented by Sahu et al., 2017. Although Sahu et al., 2017 focus on graph processing, the authors have goals similar to ours, i.e, they aim to investigate how graph technology is actually used in practice and which are the challenges and the use cases of the engaged practitioners. Although surveys that focus on the practitioners' point of view are quite common for other communities, they are a rarity for the Data Management community. The work from Sahu et al., 2017 is a pioneer towards such research studies and the proposed methodology is easily adaptable for many data management technologies.

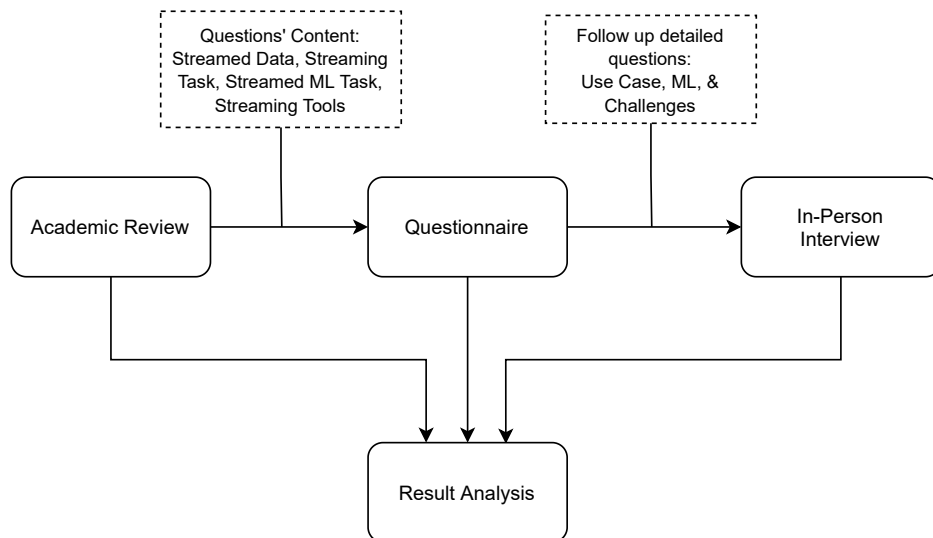


FIGURE 3.1: Diagram of our survey process main steps where the arrow shows what kind of information are being passed between the steps.

As shown in Figure 3.1, there are 3 main steps involved in our process of survey. These steps are Academic Review, Questionnaire and Interview. This process has an iterative nature where it is possible to go back to a previous step depending on the result of the current step. Results from academic review became the basis of designing the questionnaire and results from questionnaire was being followed up in depth in the interviews. All the result from each steps were then analyzed.



### 3.1 Academic Publications Review

The first step is to try to answer the posed research questions through the existing research in the topic of streaming use case, streamed data, streaming task, streaming machine learning computation, and streaming tools. Thus, we needed to perform a review of data stream processing academic publications. Since we needed to map the studies into our topics, we followed the guideline of systematic mapping studies in software engineering from Petersen, Vakkalanka, and Kuzniarz, 2015 and Petersen et al., 2008. Several adaptations were made to the mapping methodology to support the iterative nature of our survey process.

In Chapter 4 we describe the detailed process for the academic publication review process. However, the overview of our systematic review process steps are the followings:

1. *Scope Defining,*
2. *Identification of Studies,*
3. *Keywording,*
4. *Data Extraction,* and
5. *Mapping.*

In the review, we collected papers from 4 top academic conferences in the field of database management for the year of 2018 to 2020. We first defined the scope of literature review to guide us in identifying studies that were relevant to our review goal. For each papers retrieved, we selected the ones that were developing a stream processing framework or directly study a stream computation. We then omitted papers that were using stream technology or computation as a part of solving a non-stream related problem.

From the retrieved 210 papers, we identified: (i) the kind of data streamed in experiments; (ii) the streaming computations performed; (iii) any machine learning computation used upon the streams that appeared in the papers; and (iv) the streaming software and tools used in the papers. We then used keywording technique to find keywords from the papers' abstract that were relevant to our review scope and continued to do the data extraction as we did the complete reading process of the papers. Finally, we map and analyze the data we obtained from the review process to fit the information needed for the questionnaire. Analysis results from the academic publication review were then used to construct the questionnaire questions and answer options.

The steps and the results of the reviewing and the mapping of the academic publications are presented in details in Chapter 4.

### 3.2 Online Questionnaire

The goal in conducting the questionnaire is to gain a quantitative insight into the different aspects of using streaming technologies in practice through the practitioners' perspective. The process of conducting the questionnaire is divide into 3 steps: *Design, Distribution,* and *Result Analysis.*

*Design* — Initially, we constructed a draft version of the questionnaire with questions based on the mapping result of the academic review. We also included some questions regarding information that we would like to know but was not answered in the mapping result. We needed to make sure that the questions make sense to



practitioners especially in the financial technology industry. Thus, before we distributed the questionnaire to ING employees, we ran a test run of our draft questionnaire with 3 practitioners from a financial technology company outside ING to get feedback.

*Distribution* — Our target participants for the questionnaire are ING employees, especially engineers, who was and had been working on data stream processing. We started the questionnaire distribution in June 2021 and used 3 methods to recruit participants from ING employees: Mailing List, Individual Corporate Email, and Slack Channels. The questionnaire was distributed in the format of an official platform of online form in ING.

*Result Analysis* — In the end, there were 45 participants in total where 7 of them answered that they were performing stream-related task in their work. We analyzed the results of the questionnaire to gain insights based on our research questions. Answers from multiple choice and yes-or-no questions were analyzed quantitatively and visualized using data visualization tools. While answers from open-ended questions were analyzed manually where we did a simple categorization process to produce a descriptive summary of the answers. Insights about the streaming use cases and practical challenges were later followed up in the in-depth interview.

Chapter 5 presents in details the complete process and Chapter 6 explains the analysis of the results of the questionnaire.

### 3.3 Individual Interviews

To get a better and deeper understanding of the streaming use cases in the financial sector alongside with their challenges in implementing and maintaining the end product, we invited selected ING employees for an individual online interview. Some of the participants are people who previously have filled in our questionnaire. To include a broader set of interviewees from different and diverse teams across the organization, we reached out to several ING employees who we knew that they were working with streams. The interviews were focused more on questions about the streaming use case within ING, the implementation and the workflow of their streaming applications, and challenges and feature requests regarding the streaming framework that they used.

We managed to organize 5 interviews with engineers from ING. Insights gained from the interviews where then analyzed collectively with questionnaire result. We transcribed the interview and the transcription were then cleaned and coded to extract the information relevant to our interview goal. The extracted information were grouped to generate the desired result of the interview. We revisited our academic review step to investigate if and how these insight are reflected in the existing literature.

The detailed process of designing and performing the interviews are elaborated in Chapter 5. The analysis of the interview result are discussed in Chapter 6.



## Chapter 4

# Academic Publication Review

The goal of the literature review phase is to gain insights into the current state of stream processing research and use it as content for our survey questionnaire. In order to do that, we have to do a systematic review of academic publications in stream processing. In this section, we describe the design and execution of our systematic academic publication review. To give structure to our review process, we mainly followed the design proposed by Petersen, Vakkalanka, and Kuzniarz, 2015 in their guidelines of systematic literature review, and we adapted some of its described steps. Therefore, we used the following steps for our review. First, we define the scope of the review by putting together a detailed description of our literature review questions in Section 4.1. Then we describe our search strategy and process for identifying relevant studies in Section 4.2. Next in Section 4.3 we explain the keywording technique for developing the classification scheme. Afterwards we provide a brief introduction to our data extraction process and the extracted data properties in Section 4.4. Finally in Section 4.5 we present the mapping result of our review based on our classification scheme. Figure 4.1 is adopted from the work of Petersen et al., 2008 that shows each review steps and its outcome where the final outcome of the process is the systematic review.

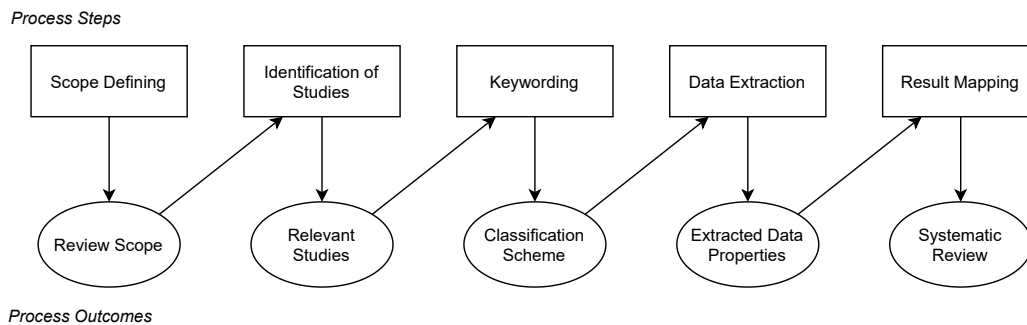


FIGURE 4.1: Systematic literature review process of academic publications in data stream processing.

### 4.1 Scope Definition

The main goal of our systematic mapping study is to build a solid background knowledge of stream processing practices amongst research work in order to appropriately construct our questionnaire and populate the suggested answers. Our secondary goal is to get an overview of the state-of-the-art research on stream processing, and the current research directions that researchers follow. All the collected insights are used for the construction of our questionnaire.

TABLE 4.1: Research questions for the systematic literature review of academic publication.

ID	Literature Review Question	Aim
LRQ1	Which use cases of stream processing are discussed in research?	To gain an overview of the case of using streams in these stream related researches
LRQ2	What kind of streamed data are commonly used in research?	To identify and classify the properties and characteristics of dataset that the researchers process in their streams
LRQ3	Which stream processing tasks and techniques are implemented in research?	To identify a set of categories of stream processing tasks and to obtain an overview of which techniques are usually used for these tasks
LRQ4	Which machine learning computations are performed by researcher in their streaming pipeline?	To identify a set of machine learning computations that are usually used in the researchers' streaming pipeline
LRQ5	Which software and tools are commonly used by researcher to process their streams?	To identify mainly the stream processing engines and other tools of stream processing that are used by researchers and how are these tools being utilized

Thus to reflect these goals, we broke down our research questions from Section 1.1 into 5 literature review questions depicted in Table 4.1 as our scope of the literature review. With *LRQ1*, we intend to obtain an understanding of the use cases of data stream processing discussed in the literature and identify the goals to be achieved or the problem to be addressed. *LRQ2* investigate types of data that are commonly used in stream processing applications such as the properties and the characteristics of these data that are considered important for data stream processing. *LRQ3* explores the kind of tasks usually carried out in stream processing systems. We are also interested to see the techniques that can be used to deliver each tasks. With *LRQ4*, we aim to gain insight on how machine learning computations are applied within a streaming environment and which models are often used in research. Finally, *LRQ5* investigate stream processing engines and other additional tools used to support the stream processing system. We are also intending to identify the usage of these additional tools.

## 4.2 Identification of Studies

In identifying relevant studies for our academic publication review, we started with 4 conferences in the latest years. We carried out the search on SIGMOD, VLDB, ICDE and DEBS for the year of 2018, 2019 and 2020. The objective of choosing these conferences and the year was to narrow down our work in obtaining initial insight on topics by choosing some of the top conferences in the field of database management and took the latest years to understand the latest development in the research area.

TABLE 4.2: Retrieved papers from the study search process in 4 top conferences in database management field.

Conference	Year	Retrieved Paper	Total
SIGMOD	2020	11	31
	2019	10	
	2018	10	
VLDB	2020	11	47
	2019	24	
	2018	12	
ICDE	2020	19	77
	2019	26	
	2018	32	
DEBS	2020	21	55
	2019	22	
	2018	12	
<b>Total</b>			<b>210</b>

This would give enough insight about state-of-the-art development on streaming technologies and how streaming concept is being practiced amongst researcher, for us to start. For the whole process of conducting the academic papers identification, we generally follow the recommended guideline by Kitchenham and Charters, 2007.

### Search Strategy

We used this search strategy to first retrieve papers related to streaming topics:

1. Based on our literature review questions, the main search terms need to be general to simply retrieve any stream related papers from the conferences. Thus, we used "*stream*" as our main search term. We also needed to consider the possible related terms and alternative spellings for the identified main search term. Therefore, we added "*data flow*", "*dataflow*", and "*data-flow*" to our search terms.
2. We looked at papers from the each conferences' papers and proceedings in our years scope. We searched for these search terms within papers' title and abstract to retrieve papers that are related to stream topics.
3. We conducted several pilot searches by checking a subset of our search results against a test set to validate the completeness of the search. The test set are a set of papers belongs to stream related sessions of the conferences.

Search completeness was believed to be at a high degree using this search strategy. The search was conducted in a time range of February until April 2021 and we retrieved a total of 210 papers with streams related topic. Table 4.2 presents the details of papers retrieved from each conferences and in each years.

### Papers Selection

The next step would be the selection process of the retrieved papers. The aim of this paper selection process was to identify relevant papers from the retrieved set, that can provide evidence to answer our literature review questions. The selection

process is a multi-stage approach (see Figure X) where we clean the retrieved papers from duplicate papers and papers that are less than 5 pages, then we incrementally read different parts of the cleaned primary studies and removed irrelevant papers from the primary studies based on the corresponding inclusion and exclusion criteria. The purpose of this multi-stage approach is to select papers in effective way by separating the process into several selection process so that the most unrelated papers are filtered from the beginning before we have to read it in full-text. Since the guideline from Kitchenham and Charters, 2007 does not provide a detailed steps for this process, we follow the selection method in the work of Qin, Eichelberger, and Schmid, 2019. At the end of the process, we obtained the final set of relevant papers.

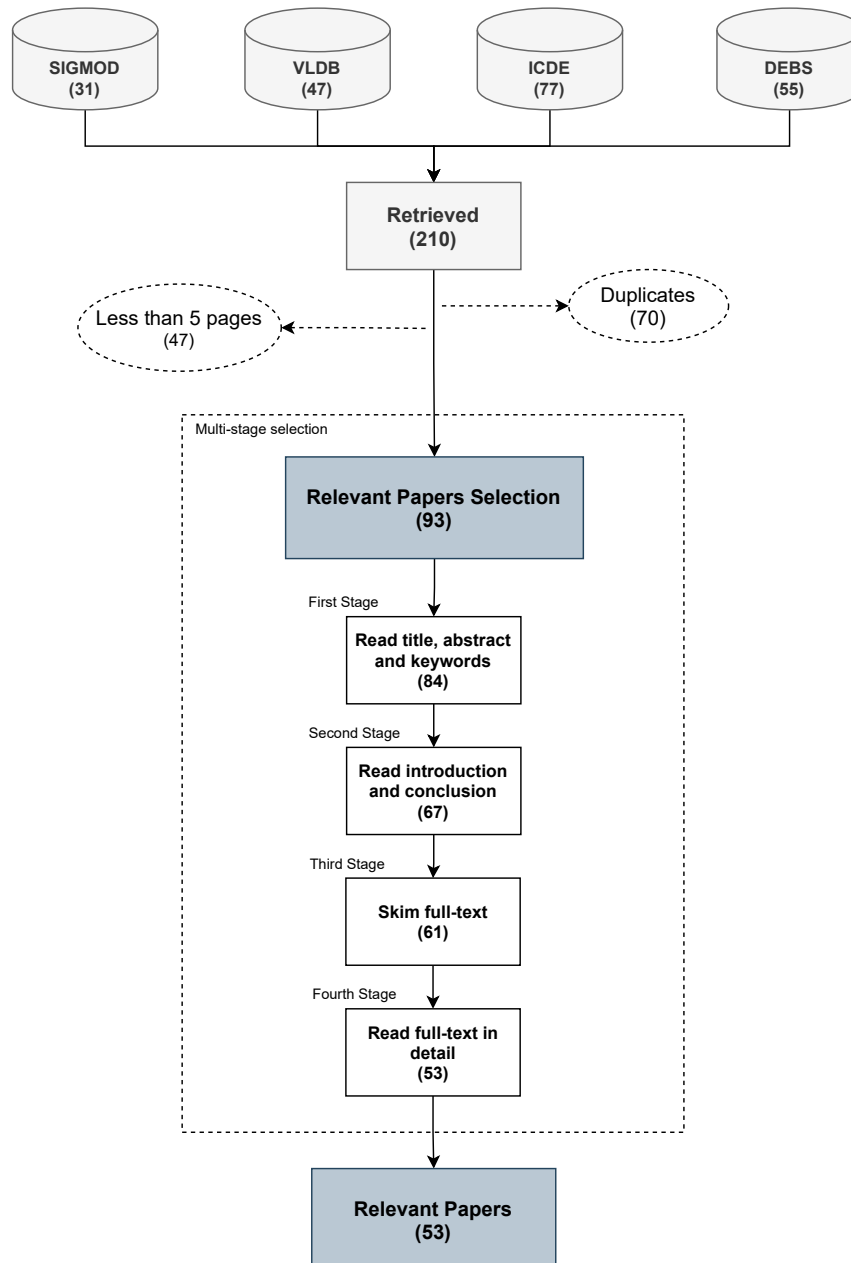


FIGURE 4.2: Results of each steps in the papers selection process.

Figure 4.2 represents the results of the papers selection in each of its steps. From the total retrieved 210 papers, we removed 70 duplicate papers and 47 papers that are less than 5 pages that left us with 93 papers for the selection process. We followed

the multi-stage approach that consisted of four selection stages that will be explained in the following part of this sub-section. Non-relevant papers are deducted on each stages and we got 53 relevant papers at the end.

### *Selection Procedure*

The multi-stage process is divided into four stages. Through each stages, we incrementally read different parts of the paper where we started with title and abstract only then gradually adding up to full text reading in detail. Moreover, we incrementally applied the inclusion and exclusion criteria (see next part of this subsection for its details) to each selection stage. The selection stages are detailed as follows:

- First stage was to read the paper's title and abstract. To avoid excluding papers too early, we applied only IC1 and IC2 at this stage.
- Second stage was reading the introduction and conclusion section of the paper. We added IC3-IC5 and EC1 in this stage to start being selective towards our specific streaming topics.
- Third stage was skimming the whole part of the paper. Here, we added IC6-IC8, relaxed IC9-IC10 and EC2 to start identifying papers that can answer our literature review questions.
- Fourth stage was to read the full text of the paper in detail. We added not-relaxed IC9-IC10 and EC3-EC4 to this final stage.

### *Inclusion and Exclusion Criteria*

Inclusion and exclusion criteria were initially determined based on the literature review questions. It evolved and refined during the process of detailed reading through the stages of selection process. The final inclusion and exclusion criteria are listed below:

- Inclusion Criteria (IC): IC1 and IC2 must be met by all papers while IC3-IC5 and IC6-IC10 are optional within each group in the sense that it is sufficient that a paper holds one of them.
  - IC1: Paper is written in English.
  - IC2: Paper is in the field of data stream processing.
  - IC3: Paper discuss or develop a stream processing algorithm.
  - IC4: Paper discuss an implementation of stream processing concepts to solve a problem.
  - IC5: Paper is about developing a stream processing engine or discussion about an existing stream processing engine.
  - IC6: Paper mention the real world use case or problem addressed by the research.
  - IC7: Paper indicates the dataset being used for their stream research.
  - IC8: Paper indicates the tools and software used in its stream processing work.
  - IC9: Paper presents the techniques used to complete a streaming task they did.

- IC10: Paper explains the machine learning computation implemented within their stream processing system.
- Exclusion Criteria (EC): All EC are mandatory to all papers.
  - EC1: Paper is using stream processing concept but it is not the main focus of the study.
  - EC2: Paper only mention stream processing usage but details are not sufficiently described.
  - EC3: Paper focuses on hardware-related approaches and no software solution is provided.
  - EC4: Paper is about theoretical discussion of streaming concept.

### 4.3 Keywording

When designing a classification scheme, we used the keywording technique as a time-saving approach and to guarantee that the scheme took relevant papers into consideration. Keywording technique is a method to create classification scheme by first identifying relevant keywords from the papers' abstracts which then refined as the reading process continues. In the original guideline from Petersen, Vakkalanka, and Kuzniarz, 2015, classification scheme is used to categorize papers into the scheme for the purpose of quantitative analysis. Being that our main goal of this literature review is to obtain a framework of stream processing practices for our questionnaire, our classification scheme was used to qualitatively collect relevant information that will later function as a structure for our questionnaire.

Our keywording method followed these steps:

1. Already from the paper selection process, we started to search for keywords and concepts in the papers' abstracts that are representative of our literature review questions.
2. We refined these initial keywords and concepts while we were skimming and detail-reading the papers to collect more related keywords and concepts.
3. Final set of keywords and concepts were then analyzed. Similar keywords and concepts were clustered to form a series of categories for our review result. This series of categories were then acted as the classification scheme in the data extraction process.

We compiled 12 keywords of properties of study that we would like to collect for our questionnaire. These properties were derived from the literature review questions. Table 4.3 presents the properties, its description and the literature review questions it relates to.

### 4.4 Data Extraction

We then use Table 4.3 as the classification scheme in our data extraction process For each 53 relevant papers, we collected data that explain any property and classify this data to its related property e.g. if a paper explained that they process 10 record of data per second and use Apache Spark as their stream processing engine in their experiment, then we classify these extracted data to Data Speed (P6) and Stream



TABLE 4.3: Properties of the study as keyword references

ID	Properties	Description	LRQ
P1	Application Category	Applications that are powered by data stream processing technologies	LRQ1
P2	Field of Industry	Real world field of industry that the stream applications is being applied	LRQ1
P3	Dataset Type	Type of streamed dataset based on its data characteristic	LRQ2
P4	Data Format	Format used to structure the data in the stream	LRQ2
P5	Data Attributes Type	Type of data of the attributes contained in the dataset	LRQ2
P6	Data Speed	Amount of data record per second processed within the stream	LRQ2
P7	Data Representation	Real world representation of the data in the stream	LRQ2
P8	Streaming Task	Common tasks done in streaming practices	LRQ3
P9	Streaming Task Techniques	Techniques used to do the streaming tasks	LRQ3
P10	Machine Learning Task	Machine learning tasks implemented in streaming environment	LRQ4
P11	Stream Processing Engine	System or framework that support developers in writing code to process streaming data	LRQ5
P12	Stream Supporting Tools	Tools or software that commonly integrated with stream processing engines to support the complete processing of streaming data	LRQ5

Processing Engine (P11) property. The data extraction focused on identifying the taxonomy or content of each properties from the obtained academic studies so that we gain insight of how the questionnaire content could be and how it should be structured.

## 4.5 Result Mapping & Discussion

For each property from the data extraction step, we mapped the collected information to generate its taxonomy or content. In this section, we present and discuss findings from academic publication review for each literature review questions.

### LRQ1: What is often the use case of stream implementation in research?

Respectively from data extracted for Application Category (P1) and Field of Industry (P2) properties, we found a total of 8 applications over 13 fields of industry. Table 4.4 shows the data streaming applications, its example of use case and the fields of industry in which the application covered.

We identified three most popular applications as follows:

- *Anomaly Detection*: This was the most popular applications of data streaming as it was discussed in 7 stream papers. These papers wrote about the implementation of streaming analytic to detect anomalies within the streamed data. Anomaly detection in streaming environment was used within various field such as social media, transportation & logistic, and electricity. Tam et al., 2019 introduce an incremental method to do rumour detection in streaming data of social media posts where it identifies anomalies in both social entities and relations. Kontopoulos et al., 2020a present an event-based classification approach of vessel activity from real-time data streams to identify suspicious vessel activities.
- *Event Detection*: The second most popular stream application from our findings was event detection. Event detection is the process of examining event streams in order to identify collections of events that fit patterns of events in an event context. Zhao et al., 2020 discuss about detection of local popular topics in a stream of geo-textual social network data using a subscription matching technique. Chen et al., 2019 explains about detecting special event related to abrupt changes of electrical consumption by implementing sequential incremental event detection algorithm on a stream of electrical smart meter measurements.
- *Continuous Recommendation System*: Continuous recommender system placed third on the list of most mentioned applications in stream papers. Similar to classic recommender system applications, continuous recommender system aims to generate suggestions relevant items to the user of a system but in a continuous manner rather than static. Karimov et al., 2018 used an application of online video game item advertisement as their research workload use case, where the streaming system aims to give personalized gem packs suggestion as the game progress.

TABLE 4.4: Data streaming applications and example use of stream in various fields identified from academic papers

Application	Example	Fields
Anomaly Detection	Detecting unusually crowded areas in a city using streamed mobile phone connection data collected in the city	Transportation & Logistics, Social Media, Electricity, Urbanism, Stock Trading, Life Science
Event Detection	Detecting misleading online retail product reviews in real-time manner to provide a better shopping experience for users	Social Media, Electricity, Health, Transportation & Logistics, Oil & Gas, Retail
Continuous Recommender System	Giving real-time recommendation to users of online video sharing social media platform about they favourite idols' performance once it's available	News & Entertainment, Social Media, Online Game, Advertisement
Finding Significant Items	Calculating the $k$ most popular taxi routes in a rolling window from a streamed data of taxi trips in a city	Social Media, IT & Telecommunication, Retail
Monitoring System	Processing streams of data of electric smart meter in high-throughput and low-latency for the purpose of building real-time energy consumption monitoring application	Electricity, IT & Telecommunication, Health
Graph Processing	Performing queries in a network traffic graph streams to locate certain topology structures in the telecommunication network	IT & Telecommunication, Social Media
Pattern Recognition	Future movement predictions of humans, vehicles, and animals from the streamed location data of GPS-equipped devices	Urbanism, IT & Telecommunication
Search in Streams	Real-time data retrieval of both fresh and historical data from streamed sensor devices data of a smart city system	Urbanism

## LRQ2: What kind of streamed data commonly used in research?

We analyzed data extracted for Dataset Type (P3), Data Format (P4), Data Attributes Type (P5), Data Speed (P6) and Data Representation (P7) properties and we obtained some observations that will be discussed in the following part.

### Dataset Type

We identified that there are several dataset type commonly used in data streaming experiments as follows:

- *Temporal Dataset* — Temporal dataset stores data relating to time and the state according to the time, such as temperature measurement in every minute from the last 3 months. By definition, data used in streaming environment must have temporal characteristic because the timestamp defines the incoming order of the data.
- *Spatio-Temporal Dataset* — Spatio-temporal dataset is a dataset that contain data collected across both time and space dimension describing an event in a particular location and a period of time. This type of dataset is used in stream use case where time and spatial aspect is crucial content for problem solving. An example of this dataset is a trajectory of taxi trip data containing taxi ID, latitude and longitude of its position, and the timestamp of this location snapshot.
- *Spatio-Textual Dataset* — Spatio-textual or geo-textual dataset store data describing an entity with its geographical aspect representing the location and textual aspect that represent some kind of context of the entity. Streamed spatio-textual data arrives in high rate so it is commonly used in applications that aims to detect changes in events at a specific range of location. An example of streamed spatio-textual data is geo-tagged tweets.
- *Relational Dataset* — Relational dataset structured in a way that is able to store data objects and also the relation within these objects. In streaming environment, relational data is often stored as streamed graph where the data stored in its vertices and nodes keeps changing over time. An example of relational dataset is social graph where the nodes represent a person and the vertices represent relationship status between two person.
- *Image Dataset* — We identified some stream studies that focus on image processing in streaming fashion. This type of dataset contain images stored as its pixels data and other metadata.

### Data Format

We could not extract enough information about all types of data format used within streamed dataset as most papers often not describing the format of data in the dataset used in the research. We identified some papers that are using *JSON* and *graph* as the format of their streamed data. As a schema-free data format, JSON can be really practical in stream implementation as it gives flexibility to it's data structure. This is useful in stream environment because it is highly possible that data is retrieved from varying sources. Several paper that focuses on graph processing in streaming environment use streamed graph dataset where it receive real-time changes to the data in the nodes and vertices.

### Data Attributes Type

For data types contained in the attributes of streamed data, we recognized five common data types as follows:

- *Timestamp*: e.g., transaction time, taxi pick-up time, user login time
- *String*: e.g., social media username, messages, product review
- *Integer*: e.g., web clicks count, social network user's follower amount, mobile phone call frequency
- *Double*: e.g., voltage measurement, taxi trip distance, location's latitude & longitude
- *Boolean*: e.g., connectivity flag, spammer email flag, suspicious account flag

### Data Speed

Most of the study we investigated does not always state the throughput of their data streams being used in their experiment. There were 4 stream papers that mentioned the speed rate of their streams. Based on this, we gained insight on the amount of record per second that the research experiment can handle which ranges from 1,000 to 33 million data record per second.

### Entities Representation

We gathered the content of datasets used in stream papers and associated our findings with a real world entities that it represents. For example, a social graph dataset represent human entity specifically human interaction and product review dataset represent business entity. The followings are entities we identified as a common representations of streamed data:

- *Humans*: e.g., customers, patients health, social interactions, social networks
- *Business & Finance* : e.g., products, advertisements, stock tradings
- *Knowledge*
  - Scientific Knowledge: e.g., physical experiments, life sciences, environment
  - Linguistic Knowledge: e.g., words, definitions
- *Infrastructure*
  - Physical Infrastructure: e.g., household gas, oil wells, wireless sensors
  - Telecommunication Infrastructure: e.g., call records, wi-fi router network, IP packets
  - Electrical Infrastructure: e.g., electrical power consumption, smart meter sensors
  - Transportation Infrastructure: e.g., public transports, road network
- *Digital Information*
  - Digital Object: e.g., videos, files, emails
  - Digital Activities: e.g., log files, website links, website clicks

### LRQ3: Which stream processing tasks and techniques are implemented in research?

The purpose of this literature research question is to identify tasks done in data stream processing work and to derive a taxonomy of techniques used to carry out these identified tasks. We categorized the streaming task into two high-level categories which are data operator tasks and general tasks. Figure 4.3 shows the taxonomy of data operator streaming tasks and Figure 4.4 shows the taxonomy of general streaming tasks.

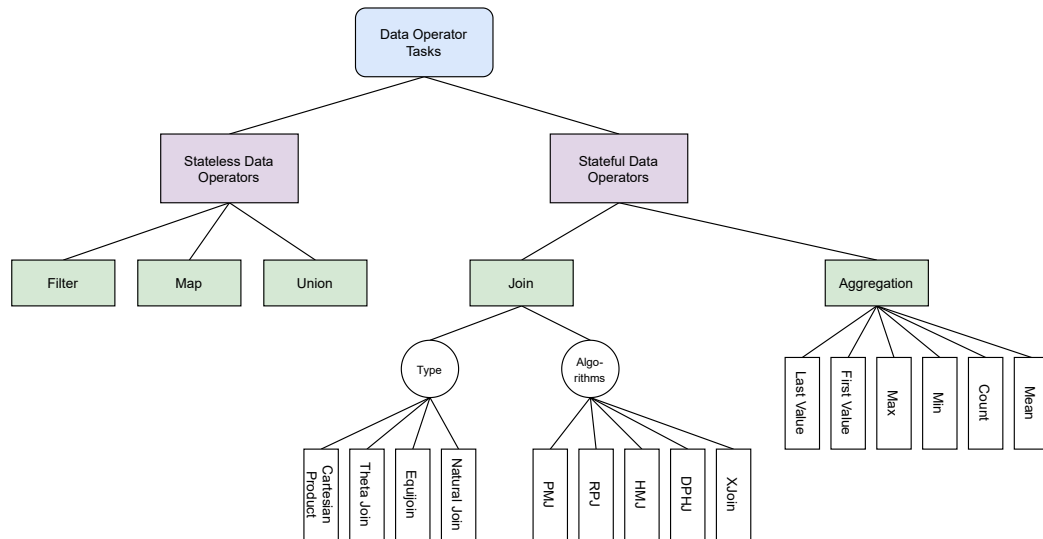


FIGURE 4.3: Taxonomy of identified data operator streaming tasks and its technique

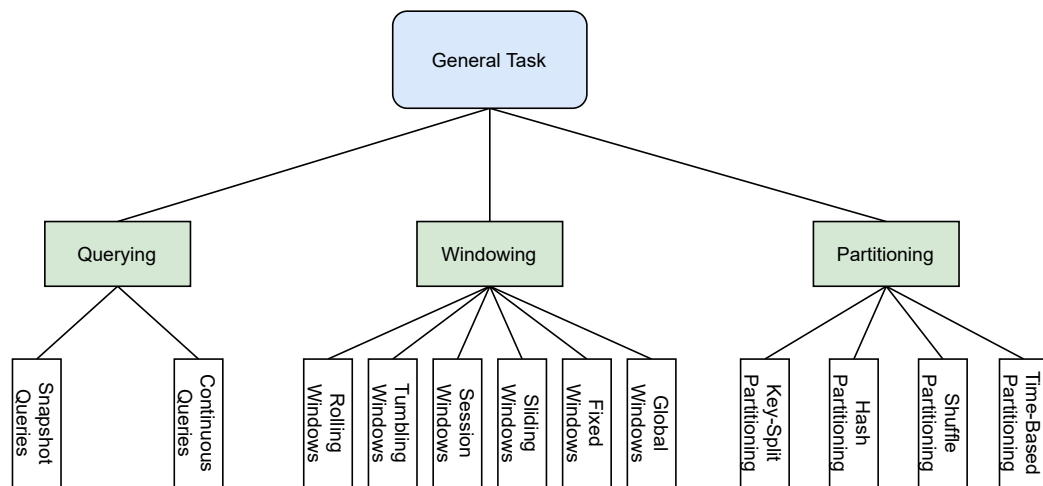


FIGURE 4.4: Taxonomy of identified general streaming tasks and its technique

Explanations about these identified streaming tasks are as follows:

- **Data Operator Tasks:** These are the tasks in streaming process that deals with transformation of data in the streams. Data operators can be divided into two types which are stateful data operators and stateless data operators.

- Stateful Data Operators: In order to produce an output, stateful data operators maintain the states as input streamed data is being processed. There are two data operators tasks that operates in stateful way which are Join and Aggregation.
  - \* *Join*: Join task is used to match tuples from two distinct input streams by defining terms referring to an equality between two fields of the different input streams. Several types of Join are Natural Join, Equi-join, Theta Join, and Cartesian Product. There are several Join algorithm such as XJoin, DPHJ, HMJ, RPJ, and PMJ.
  - \* *Aggregation*: Aggregation task aims to execute aggregate function such as computing value of mean, count, minimum, maximum, and first and last value.
- Stateless Data Operators: Stateless operators perform a one-by-one processing of input tuples where each tuple is processed individually and the output is produced without maintaining any state.
  - \* *Filter*: Filter task is a generalized selection operator used either to discard or to route tuples from one input stream to multiple output streams. Filter can be seen as the data streaming equivalent of the Select function in relational database.
  - \* *Map*: Map task is a projection operator used to transform the schema of the input tuples. Map is the data streaming counterpart of the Projection function in relational database.
  - \* *Union*: Union task merges tuples from multiple input streams into a single output stream where all the input streams and the output stream tuples share the same schema.
- General Tasks: These are the tasks in data streaming process that interact with data but does not do any form of transformation to the data in the stream.
  - *Querying*: Querying is a task to request specific data to be retrieved from the streams. There are two types of querying in stream which are Snapshots Query that return data from the cache as it exists at a moment in time and Continuous Query that continues to gather and return data when changes are made until you stop the query.
  - *Windowing*: Windowing is an approach to break the data in streams into mini-batches or finite streams to process the data in it. There are several types of windowing which are Global Windows, Fixed Windows, Sliding Windows, Session Windows, Tumbling Windows, and Rolling Windows.
  - *Partitioning*: Partitioning task aims to break up a large data set into smaller subsets within a single instance typically for the purpose of dividing load and scaling. Several types of Partitioning are Time-Based Partitioning, Shuffle Partitioning, Hash Partitioning, and Key-Split Partitioning.

#### **LRQ4: Which machine learning computations are performed by researcher in their streaming pipeline?**

The insight we got from the academic publication review on machine learning implementation within streams are:

- There are two ways of doing learning in streaming environment which are online learning and offline learning.

- *Online learning* is a method of machine learning where the data to be trained becomes available in a sequential order and is used to update the best predictor for future data at each step of data ingestion.
- *Offline learning* is a machine learning approach that ingests all the data at one time to build a model.
- Machine learning models used for data training and learning within streaming environment are the common learning model used in static data environment such as artificial neural network, decision trees, support-vector machines, etc.
- Clustering and classification were the common machine learning tasks performed in the researches that we have reviewed
  - *Clustering* is a type of unsupervised learning method of machine learning in which the goal is dividing the data sets into a certain number of clusters in such a manner that the data points belonging to a cluster have similar characteristics. A work done by Gangineni et al., 2019 implemented object recognition system from high-speed light detection data stream where the streaming system includes data learning process of filtering, object segmentation, noise reduction, and multi-class object classification using Convolutional Neural Network. Gong, Zhang, and Yu, 2017 dealt with stream clustering challenges where they proposed a solution on how to incrementally update their clustering results efficiently and capture the cluster evolution activities. Their work provided efficient data structures and filtering schemes to ensure that the data abstraction is in real-time thus making online clustering possible.
  - *Classification* is a supervised learning approach that learn how to assign predefined labels or classes to the data given. The work of Kontopoulos et al., 2020b presented a novel approach is for the behaviour classification of vessel activity from real-time data streams of maritime events where they implement a real-time stream classification system using Akka Streams as the engine and XGBoost as the learning model. Wang et al., 2019 provided an incremental learning strategy in data streaming environment by proposing a Convolutional Neural Network based effective learning framework for novel class detection and correction.

#### **LRQ5: Which software and tools commonly are used by researchers to process their streams?**

From our academic publication review, we identified several stream processing frameworks that were used by the researchers to handle their streams. The common frameworks are the followings:

- Apache Storm — Storm is an open source, low latency, data stream processing system. It is the oldest open source streaming framework and one of the most mature and reliable one. It is true streaming and is good for simple event based use cases. It has the ability to integrate with other queuing and bandwidth systems. Storm implements the data flow model in which data flows continuously through a network of transformation entities. The abstraction of data flow is called streams and the transformation entities are called bolts. Bolts in Storm can implement operations such as filtering, aggregation, mapping, etc.



TABLE 4.5: Comparison of stream processing engines

Criteria	Storm	Kafka	Flink	Spark Streaming
Streaming Model	Native-Streaming	Native-Streaming	Native-Streaming	Micro-Batching
Message Delivery	At-Least-Once	Exactly-Once	Exactly-Once	Exactly- and At-Least-Once
Language	Any Language	Java, Scala	Java, Scala, Python	Java, Scala, Python
Fault Tolerance	Checkpointing & Stream re-playing	Stream Re-playing	Checkpointing & Stream Re-playing	Checkpointing
Deployment Model	Clustered	Not Clustered	Clustered	Clustered
Documentation	Good	Extensive with Stack Overflow coverage	Good with Stack Overflow coverage	Extensive with Stack Overflow coverage
Community	Oldest	Newest but fast growing	Small but fast growing	Small

- Apache Kafka — Kafka is a framework implementation of a publish-subscribe using stream processing concept originally developed by LinkedIn. Kafka maintains the feeds of messages in topic categories where each category has several partitions. Every message is assigned a unique sequential id for identifying the message in a partition. Kafka retains the published messages for a configurable period of time. When the time is due, the messages are discarded no matter they have been consumed or not.
- Apache Flink — Flink is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Flink has been designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. It is a framework for stateful computations over unbounded and bounded data streams. Flink provides multiple APIs at different levels of abstraction and offers dedicated libraries for common use cases. It has low latency with high throughput which are both configurable according to requirements. Similar to Storm, Flink operators are able to do function like mapping, filtering, reduce, etc.
- Spark Streaming — Spark Streaming is extended from Apache Spark by adding the ability to perform online processing through a similar functional interface to Spark, such as map, filter, reduce, etc []. fully support the Lambda architecture where both batch and streaming are implemented. Spark Streaming runs streaming computations as a series of short batch jobs on RDDs, and it can automatically parallel the jobs across the nodes in a cluster. Thus, Spark Streaming supports fault recovery for a wide array of operators.

Table 4.5 presents the comparison of Apache Storm, Apache Kafka, Apache Flink, and Spark Streaming as a combined result of comparison made by Isah et al., 2019, Gorasiya, 2019 and Cloudera, 2020.



## Chapter 5

# Questionnaire & Interview Approach

In this chapter, we elaborate the approach we did in carrying out the questionnaire and interview for our research. Performing both questionnaire and interview were meant to gather inside of the state of data streaming practices in practitioners in ING. Insight on data streaming practices that we obtained from academic publication review were used to structure and construct our questionnaire in a way that it would make sense to our potential participants. The design process of the questionnaire, distribution and execution of the questionnaire, and how we analyzed the questionnaire's results are explained in Section 5.1. Based on the results of the questionnaire, we then decided on what our interview should focus on. In Section 5.2, we explain the questions guideline of the interview, how we select our interview participants and the processing of our interview results.

### 5.1 Questionnaire Methodology

The goal of performing questionnaire was to collect data from wide range of practitioners in ING on their data streaming practices. We aimed to obtain insight on themes and pattern existed in topics relating to our research questions.

#### Designing The Questionnaire

The questionnaire was organized into seven sections with one section aimed to gather the respondents' demographic information (i.e. the department they worked in, team size, their job role) and six other sections with questions related for our research. Table 5.1 shows the mapping between the sections in our questionnaire with its correlating research questions and properties obtained from the academic publication review. We used these properties to build our content for the questions in the respective sections.

Demographic section was meant to collection information about our participants department, job role and the size of their team. Stream Usage section meant to gain insight on how many of our participants used streams and how many of them did not. For participants who practiced data streaming in their work, we asked them their experience with streams. For participants who did not practice data streaming, we asked them their reason of not using streams. To address *RQ1: What use cases do users implement their streaming pipeline for?*, in Use Case section we asked our participants their data streaming use case in multiple choice question and we also asked them to give a simple description of their streaming workflow in an open ended question. To address *RQ2: What types of streamed data do users have?*, we asked

TABLE 5.1: Mapping our the questionnaire's sections to research questions and properties resulted from academic publication review

Questionnaire Section	Research Question	LR Properties
Demographics	-	-
Stream Usage	-	-
Use Case	RQ1	P1, P2
Streamed Data	RQ2	P3, P4, P5, P6, P7
Streaming Task & ML	RQ3, RQ4	P8, P9, P10
Software & Tools	RQ5	P11, P12
Challenges	-	-

the participants in multiple choice questions to describe the data contained in the streaming environment in Streamed Data section. We used properties (P3-P7) of dataset type, data format, data attributes type, data speed and entities representation we found from academic publication result to guide us in providing the answer options. For RQ3: *What kind of streaming task & computations do users run on their streams?*, we provided questions about streaming task performed by our participants and which technique they used for those task in Streaming Task & ML section and we used properties of Streaming Task (P8) and Streaming Task Techniques (P9) to structure our questions. For RQ4: *Which machine learning task users perform in their streaming pipeline?*, we also asked whether or not they implement machine learning computation within their streams in Streaming Task & ML section where we use insight from properties Machine Learning Task (P10) to construct our questions. In section Software & Tools, we aim to address RQ5: *What software & tools do users use to perform their streaming processes?*. We asked about what stream processing engines and other additional tools that was used by our participants in multiple choice questions and properties Stream Processing Engine (P11) and Stream Supporting Tools (P12) from academic review to structure the answer options. In the last section Challenges, we asked in open ended questions on challenges face by our participants while they were practicing data streaming. We also asked for the ideal solution for these challenges and other expectation they have for data streaming technologies.

Based on the mapping, we developed 38 questions grouped into these six sections which are: (i) demographic questions; (ii) streamed data; (iii) streaming tasks & machine learning; (iv) streaming software & tools; and (v) streaming use case. All of our questions were either *mandatory* or *optional* for our participants to answer and there were two types of questions:

- *Multiple Choice Question* — For multiple choice questions, there were 2 types of question based on the way to answer it which are: (a) questions that allowed only a single answer as a response; (b) questions that allowed multiple answers as a response; and (c) yes or no answer. We provided an "Other" option in most of our multiple choice questions for when our participants answer did not match any of the provided answer choices.
- *Open-Ended Question* — For these type of questions, participants had to enter their response in a text box in a form of several words or sentences.

Before finalizing the composition of the questions, we did a feedback session on our questionnaire draft with three practitioners outside ING who had experience with data streaming and worked on financial technology company. In the feedback session, the participating practitioners gave their perspective on whether or not the

questions and answer options made sense to them and what kind of improvement we can do to the questionnaire draft. We then came up with our final version of questionnaire in which its overview is shown in Table 5.2 and the full version is presented in Appendix A. In Table 5.2, "Type" column explains the type of the questions and the values are "O" for open-ended questions, "S" for single answered multiple choice questions, "M" for multiple answered multiple choice questions and "Y" for yes or no questions. "Necessity" column explains whether the question is mandatory or optional to be answered.

### Questionnaire Implementation

The questionnaire was uploaded onto Microsoft Form which was the official internal questionnaire management platform at ING. Based on our discussion with several people in ING, we decided to distribute our questionnaire within departments of engineering and analytic as these were the part of the company that most likely to practice data streaming. We conducted the survey from June 1 to July 5, 2021 where respondents had four weeks to participate in the survey. We sent the original invitation emails in the first week and a reminder emails in the third week. Candidate participants were invited using an invitation email where we also explained the purpose of the questionnaire and how the results can help us to gain more insight about current data streaming practices and how it can be improved. We sent invitation to a total of 597 active emails out of 826 emails in several mailing list across the targeted departments and received 45 responses (7.5% response rate). Compared to the on-line surveys conducted by Punter et al., 2003 for their software engineering online surveys guidelines, our questionnaire's response rate is below theirs which are within 14—20% range. In Section 6.9, we explain the reasoning of our survey response rate.

### Results Analysis

Based on the type of questions, we analyzed our questionnaire results in two ways as follows:

- **Quantitative Analysis** — For type of questions such as multiple choice and yes or no questions, we used tables or statistical visualizations to present the results. There were 22 questions that we analyzed quantitatively and we used Microsoft Power BI desktop software to visualize some of the question's answers. When necessary, we grouped or filtered the answers based on whether or not our respondents did data streaming practices in their work.
- **Manual Analysis** — For open-ended questions, we needed to do manual analysis since the answers in sentences and had not been categorized by nature. We manually analyzed 11 questions in total. For short answers, we either present the result individually or put them in a range or group. We then used simple manual categorization process to present the summary of descriptive answers. We identified the main theme for each questions, examined which description can be labeled as a category and created a summary for each category based on the descriptions from our participants' answers.

TABLE 5.2: Overview of our questionnaire's questions

Section	#	Question	Type	Necessity
Demographics	1	In which category of department at ING do you work?	M	Mandatory
	2	What is your role/job position?	O	Mandatory
	3	How big is your team?	S	Mandatory
Stream Usage	4	Do you (or your team) implement any kind of data streaming practices in your work?	Y	Mandatory
	5	(If not using streams) What is the reason that data streaming technologies is not used within your team?	M	Mandatory
	6	(If not using streams) What kind of data processing technologies that you and your team currently using?	O	Mandatory
	7	What is the estimated number of the users of your stream(s)?	S	Mandatory
	8	How long (in years) have you been working with streaming data? Including also your experience outside ING.	O	Mandatory
	9	How would you consider your knowledge level about streaming data?	S	Mandatory
Streamed Data	10	How many stream(s) do you handle?	O	Mandatory
	11	Which entities are handled in your stream?	M	Mandatory
	12	How many data points does your stream(s) processes per second, on estimated average?	S	Mandatory
	13	How many data points does your stream(s) processes per second, at an estimated maximum?	S	Mandatory
	14	What is the format of the data processed on your stream(s)?	M	Mandatory
	15	Which category best described the data in your stream(s)?	M	Mandatory
	16	Which data type(s) are contained in your streams' data point?	M	Mandatory
Streaming Task & ML	17	What kind of streaming task(s) do you perform?	M	Mandatory
	18	What kind of join operation do you perform?	M	Optional
	19	What kind of aggregate queries do you perform?	M	Optional
	20	If you have time-bounded computation, which type of window you use?	M	Optional
	21	Following up the question above, what is the window interval?	O	Optional
	22	What are the sources of your stream(s)?	M	Mandatory
	23	Do you perform any machine learning computation in the stream or at the end of the stream?	Y	Mandatory
	24	Please explain the machine learning computation that you do and its involvement with the stream.	O	Optional
Software & Tools	25	What stream processing engine do you use?	M	Mandatory
	26	What other type of tools/services do you use alongside the stream processing engine?	M	Optional
Use Case	27	For what use case(s) are you using your stream?	M	Mandatory
	28	Describe the workflow of your use case of stream	O	Mandatory
	29	Why are you using data streaming technologies for your use case?	M	Mandatory
Challenges	30	What are the challenges, problems, or constraints that you faced while you're working with data streams?	O	Mandatory
	31	How do you try to solve these challenges currently?	O	Mandatory
	32	How would you picture the ideal solution for these challenges?	O	Mandatory
	33	What feature that you think is missing from the data streaming software or tools that you're using?	O	Mandatory

## 5.2 Interview Methodology

Based on the results of the questionnaire, we then can observe which topics we need more detailed information of. Thus, through the interview we aimed to get a detailed answers from the interview by obtaining deeper details of foreseen information from the questionnaire results and bringing out unexpected information.

### Interview Questions

Based on the explanation by Seaman, 1999 on software engineering qualitative study, the interview should be designed to be semi-structured because of its exploratory nature. The interview were divided into three main part which are the introduction, main questions, and closing part.

From the analysis result of the questionnaire's answers, we observed that we would be able to dig deeper several topics such as use case, machine learning implementation, and challenges. Thus, in the main question part of our interview we put focus on questions for:

- streaming use case and its workflow,
- machine learning implementation in the streams,
- and challenges and expectation in data streaming practices.

Questions around other topics such as streamed data, streaming tasks, and streaming tools were still posed during the interview although it will not be the main discussion. Table 5.3 presents the overview of questions guideline for the interview with estimated time for each section and related research questions.

### Participants Selection

Our target interview participants were people in ING who practices data streaming in their work. Our questionnaire were done by our participants anonymously so it was not possible to follow up the questionnaire participants for interview invitation unless they reached out to us. Thus, we had to reach out directly to people at the company who implemented or had experience working with data streaming. We discussed with the our lab partners from ING to gather information on the potential interviewees.

Based on the gathered information, we sent 10 interview invitations and one questionnaire participant reached out to us for the interview. At the end, we interviewed 5 people in total. Table 5.4 shows the work area or job role of our interview participants.

### Interview Execution

Our interviews were held online using the official meeting platform used within ING to ensure that the process compiled with ING non-disclosure agreement. We recorded both video and audio of our interviews where the record is stored in the internal storage system of ING that can only be accessed through ING network. Each interviews took around more or less 60 minutes. Table 5.5 shows the duration of each interview records resulting in around 310 minutes of recorded video.

TABLE 5.3: Interview questions guideline with estimated time per section and related research questions

Interview Questions	Estimated Time	RQ
Introduction of our research - Explanation of confidentiality and how information would be processed	3 min	-
What is your role/job position? - How is your experience working with streaming data?	2 min	-
For what use case are you using your stream? - Can you describe how do you handle the use case by using stream? - How many streams do you handle?	15 min	RQ1
Can you describe the data contained in your stream? - How are the data organized within your stream system? - How many data points does your stream(s) processes per second?	5 min	RQ2
What kind of streaming task(s) do you perform? - Do you perform any machine learning computation within your streaming environment?	10 min	RQ3, RQ4
Which stream processing engine do you use? - How was your experience using this engine? - Do you use other tools to support your stream processing?	5 min	RQ5
What are the challenges that you faced while working with data streams? - How would you picture the ideal solution for these challenges? - What feature that you think is missing from the data streaming software and tools?	15 min	-
Closing	5 min	-

TABLE 5.4: Overview of the work area of the interviewees

Work Area	Count
Information Architect	1
Software Engineer	1
Product Owner	1
Data Engineer	1
Data Scientist	1



TABLE 5.5: Overview of the duration of recorded interviews written in hh:mm:ss format

Interview #	Duration
1	01:02:14
2	01:13:38
3	00:55:52
4	01:02:13
5	00:55:57
Total	05:09:54

### Audio Transcribing

The recorded interview videos were then transcribed for the content to be processed later on. The transcribing process took several stages which are the followings:

1. Initially, the transcription was resulted by an automated tool provided by ING video storage platform. This step was resulting a complete baseline text of the conversations that captured most of the conversation correctly but some parts were still incorrect such as technical terms, abbreviations, etc.
2. Corrections were made to the mistakes that were made by the automated transcription tools. We did this by reading the automated generated transcription while listening to the audio of the recordings. The transcription now contained the actual conversations between us and the interviewees.
3. We then need to create structure to the transcriptions where we separate questions from the interviewers and answers from the interviewees. We also separate interviewees answers were too long into smaller paragraph based on the theme of the answer content. that By doing this, we were able to see the overview of the answers content.
4. Lastly, we did a cleaning process to the transcription to remove expression words such as "Oh", "Uhm", "Yeah", etc and restructure the sentence to be more formal.

### Code Extraction & Grouping

During this stage, the final version of transcription were coded using an analytic tools for qualitative research data. The purpose of the coding phase was to extract desired information in a structured way. Coding was performed a the level of sentences or paragraphs, instead of words, because the content or message contained in the interview conversations were not always delivered in the form of exact words but need to be inferred from the whole sentence or paragraph.

This resulted in quotations from the transcriptions with codes assigned to it. We could also write comments on each quotation to write the summary of message we understand from the quoted sentence or paragraph. Table 5.6 shows a dummy example of a quotation from a sentence, its assigned codes and its comment about the inferred message.

Table 5.7 depicts the number of codes extracted from each of the performed interviews. After all transcriptions were coded, the number of appearance of the code within all the interview were then aggregated for each codes. Next, the individual codes were grouped in a round of thematic grouping. The criteria for grouping

TABLE 5.6: A dummy example of quotation with the assigned codes and commented content inferred from the sentence

Dummy Quotation	Codes	Comments
After we train our system log data in the machine learning model, we were mainly ingesting the categorized log data resulted in JSON to our stream topics.	JSON System Log Offline ML	Machine learning model was done outside the stream environment and act as a source to the streams

TABLE 5.7: Overview of number of codes processed from each interviews

Interview #	Amount of codes
1	36
2	40
3	24
4	28
5	38
Total	166

the codes were loosely following the questions theme on the guideline in Table 5.3 where we allowed space for new theme to came up. At the end, we obtained 160 quotations, 166 codes, and 18 groups. Table 5.8 provides a numerical overview of the grouping process.

TABLE 5.8: Overview of the process of the code grouping for each themes

Theme	Group	Amount of codes
Stream Use Case	Use Case	23
	Frequency	2
	Why Use Stream	5
Streamed Data	Entities	20
	Format	11
	Speed	9
	Type	2
	Stream Count	4
Task & ML	Stream Task	10
	Stream Output	1
	ML Task	5
	ML Workflow	15
Streaming Tools	ML Model	1
	Stream Engine	8
	Other Tools	7
Challenges	Reason	6
	Challenges	20
Total	Expectations	17
		166



## Chapter 6

# Results

This chapter presents the results and the insights derived from the questionnaire responses and interview results on data stream processing from a practitioner's point of view. Firstly on Section 6.1, we discuss the percentage of respondents who used streams in their data processing workflows and those who did not. Then in Section 6.2, we give an overview of the demographic details of our survey participants. From Section 6.3 to Section 6.6, we try to address our research questions, as stated in Section 1.1, based on the gathered results. Next in Section 6.7 we explain the challenges and expectation of respondents towards streaming practices in ING. Lastly in Section 6.9 we reflect on our survey methodology, we discuss lessons learned and we provide suggestions for studying further the data streaming practices.

We distributed our questionnaire to 597 active emails from the technical infrastructure, analytics and wholesale banking department at ING and we got a 7.5% response rate with 45 respondents (in Section 6.9 we explain why this could be the case). On top of that, five employees of ING participated in our series of interviews where they were able to provide detailed information on their use cases and the employed data stream processing practices. Our survey produce insightful results on how streams are being used within ING. However, a higher response rate and more interviews would be needed to get a complete overview of data streaming practices at ING.

### 6.1 Stream versus Non-Stream Practices

In our questionnaire, we asked the participants whether or not they implement any data stream processing technique in the scope of their use cases with *Question 4: Do you (or your team) implement any kind of data streaming practices in your work?*. The results, as depicted in Figure 6.1, show that out of the 45 respondents, 7 respondents (15,56%) replied that they use data stream practices in their work. During the interviewing phase, one interviewee [15], who was part of the team leaders in ING's in-house stream processing engine, mentioned that it can be estimated that there are around 100 people within ING who used data streams in their work. Assuming that these people are from the engineering and analytics departments, we can consider that 100 out of total 597 targeted participants (16.7%) of our questionnaire would have answered that they used data streams in their work. The number of positive answers in our questionnaire matches the expected number based on the experience of the interviewee. This result shows that data streaming practices are not prevalent within ING employees as there were only a handful of people who worked with data streams.

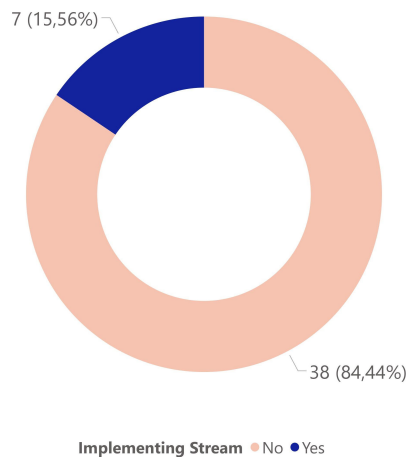


FIGURE 6.1: Distribution of the number of respondent who practiced data stream processing and who did not

### Reason of Using Stream

We wanted to see what motivates employees in ING to use stream so we asked our participants *Question 29: Why are you using data streaming technologies for your use case?* where they could pick several reasons. Figure 6.2 shows the reasons and its percentage of total number of all reasons being chosen by the participants. There are five reason why data stream was used within ING which are:

1. Need of real-time data processing (41,67%)
2. Faster decision making (16,67%)
3. Improvement in processing speed (16,67%)
4. Optimization of processes and resources (16,67%)
5. Impossibility to solve the problem without using data streaming technologies (8,33%)

From this result, it can be seen that the main reason of why our participants use data stream is because of their need to process their data in a real-time manner. Faster decision making, processing speed improvement and processes and resources optimization could also be a supporting reason of why data stream was used. There's also 1 case where it's impossible to solve the problem without streaming indicating that data streaming is very much needed because it was the only solution. In our questionnaire, none of the participants chose improvement in accuracy of data processing as the reason why they used data stream. This indicates that people who use data stream are more concern with the time aspect rather than accuracy in their data processing.

### Reason of Not Using Stream

For our respondents who answered that they did not practice data streaming, we asked them *Question 5: What is the reason that data streaming technologies is not used*

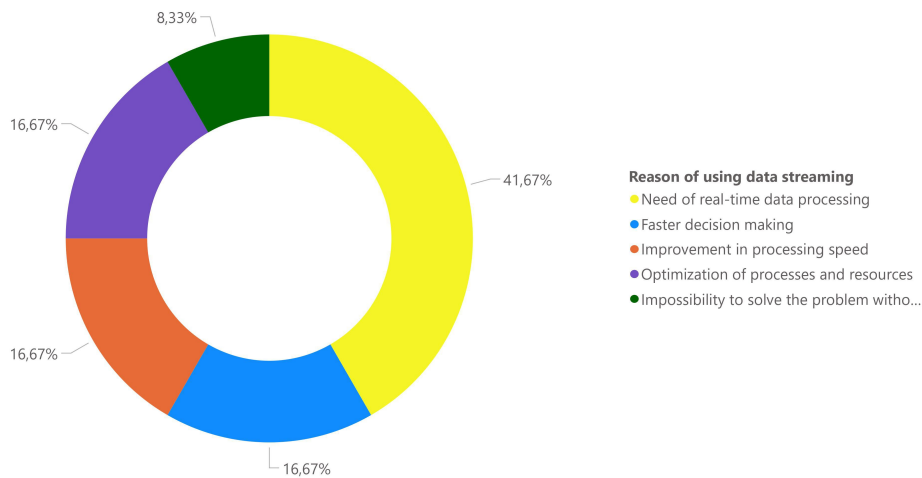


FIGURE 6.2: Distribution of the reason stated by the respondent on why they used data stream in their work

*within your team?*. Respondents could pick several reasons as their answer to the question. Figure 6.3 shows the reasons and its percentage of total number of all reasons chosen.

There are top three reason of why the respondents did not implement data streaming practices in their work which are:

1. Not applicable (25%), means that data streaming would not be applicable for the respondent's use case of their work because there's not need for real-time data processing
2. Data streaming technologies is not the right solution (20,83%), means that data streaming technologies was considered not the right solution for their data processing needs
3. The current solution works just fine (20,83%), means that they used a non-stream solution for their data processing needs and they were not thinking of changing it because it works just fine

Other reason of not using data streams are that there's other solution alternative that is more doable at the time (12,5%), unfamiliarity with data streaming technologies (10,42%), not enough resource to implement data streaming technologies (8,33%), and no availability of all the data we need as streaming (2,08%).

We then asked our respondents *Question 6: What kind of data processing technologies that you and your team currently using?* to see what non-stream data processing technologies that they were using. 16 respondents did not give answer to this thus we took this as an indication that data processing is not relevant to their scope of work. Seven respondents answered that they use DBMS for data processing. There's 1 respondent answered that they used batch processing and 4 respondents mentioned Spark in their answer, thus we can infer that they processed data in batch fashion. We also saw some other answers such as Oracle DB, IBM MQ, Tibco EMS, Power Query and Power BI. However, we found an interesting comment made by

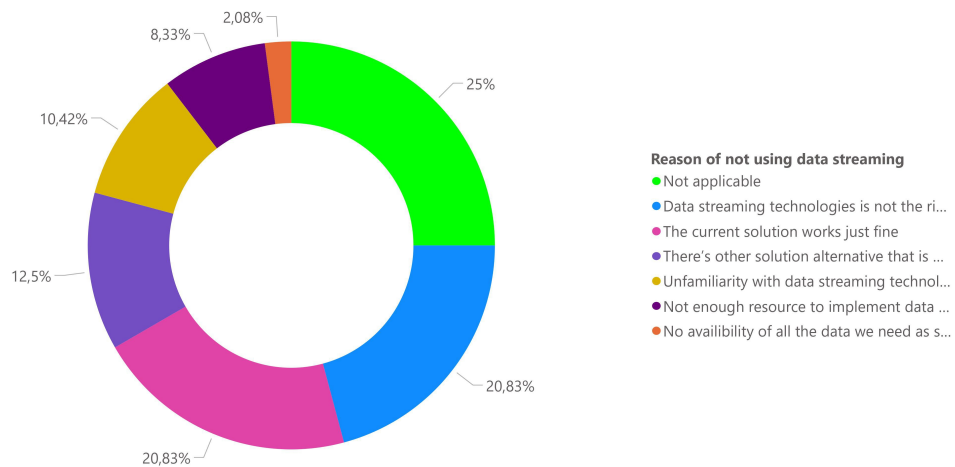


FIGURE 6.3: Distribution of the reason stated by the respondent on why they did not use data stream in their work

one respondent *"For automating evidencing/compliance and continual security/ compliance monitoring, streaming could be an option. It would also lower costs for the bank and risk work updates within teams."* [r47] which indicates that there are some interest on using streaming solution for use cases dealing with compliance.

## 6.2 Demographics of Survey Participants

We would like to see the representative of the sample for both participants who used and did not use data streams. Thus we collected several demographical information about the participants, namely the department at ING that they were working in, their team size, and their job role at ING. For those who said that they worked with data streaming technologies, we also asked their experience with data streams.

### Department at ING

We asked our participants *Question 1: In which category of department at ING do you work?*. Figure 6.4 shows the distribution of the answers group by whether or not they used stream. It can be seen that most of our participants worked at the Information Technology department at ING. Most participants who used stream also work in Information Technology department although only 4 out of 31 participants from Information Technology that used data streaming in their work. Departments that uses stream are Information Technology, Retail Banking Experties, Sales/Relationship Management, and Administration/Operations. Our participants from Wholesale Banking, Analytics, Human Resource, Risk Management, and Facilities/Procurement department claimed that they did not implement data streaming in their work.



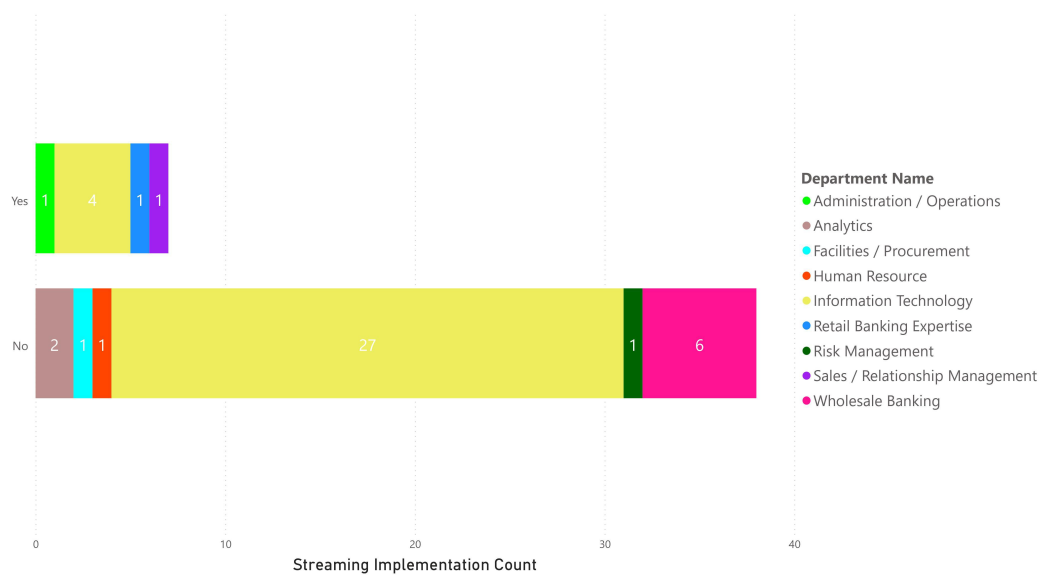


FIGURE 6.4: Distribution of the department at ING where the participants worked at

TABLE 6.1: Distribution of the job role of our participants

Job Role	Stream	Non-Stream
DevOps Engineer	2	7
Data Scientist	2	5
Product Owner	-	5
Software Engineer	-	4
Chapter Lead	-	3
Proficient Engineer	-	2
Other	3	13

### Job Role

Table 6.1 shows the answer distribution to *Question 2: What is your role/job position?*. From our questionnaire result, the most popular job role from our participants was DevOps Engineer and it is also the role that used data streaming the most. We gained insight that job role that practiced data streaming in their work are DevOps Engineer, Data Scientist and other role such as Data Engineer, Information Architect and Tech Lead. Based on the result, we also understood that roles such as Product Owner, Software Engineer, Chapter Lead, and Proficient Engineer did not use data streaming framework in their work. We also have Data Analyst, Feature Engineer, Network & Security Engineer, Product Manager, and Advisory Architect stated that they did not use stream.

### Team Size

We then asked our respondents *Question 3: How big is your team?* and Figure 6.5 shows the distribution of the answers. Our respondent mostly comes from medium size team (5-10 people) with total 31 answer count. The next most common team size is 10-20 people with the total of 13 answers and 10 respondents came from a

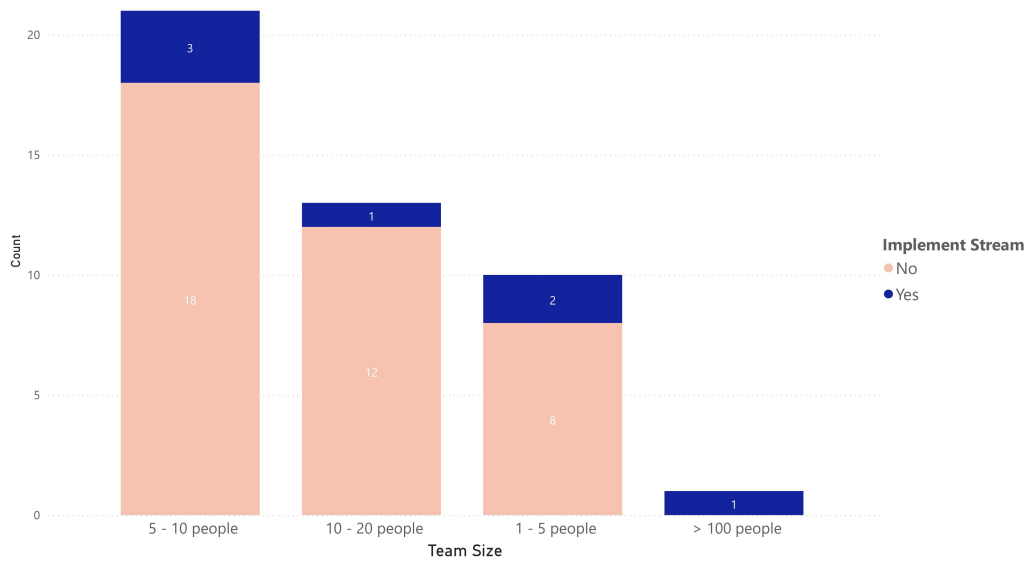


FIGURE 6.5: Distribution of the size of our participants' working team

team of 1-5 people. One respondent answered with 130 team member because the respondent was the chief of a big IT support department at ING. It can be seen from Figure 6.5 that in general the percentage of stream usage in all team size are relatively low. However, stream usage was more common to exist in small team of size 1-10 people compare to team with more than 10 people.

### Experience with Data Streaming

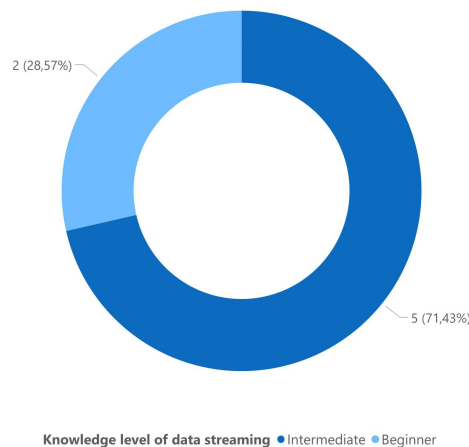


FIGURE 6.6: Distribution of the data streaming knowledge level of the participants who used streams

Figure 6.6 shows the distribution of answers to *Question 9: How would you consider your knowledge level about streaming data?*. Most of our respondents (5 out of 7) who used streams claimed that their knowledge level of data streaming is on the Intermediate level meaning that they had experiences in setting up a streaming pipeline

and making major changes to the streaming architecture. Two of the respondents answered with Beginner level which means they have experiences in working with an existing streaming set up and making minor changes to the streaming architecture. None of our respondents said that they had Expert knowledge level of data streaming where they had experience in customizing or creating streaming algorithms.

We then asked our stream using respondents *Question 8: How long (in years) have you been working with streaming data?* where we instructed them to also include experience with streams outside ING. Most respondents had experience with data streaming for 2-4 years. One respondents answered with 10 years experience working with data streaming and one respondents said that he/she had been working with monitoring, which can be a lot correlated with data streaming, for around 20 years.

### 6.3 RQ1: Streaming Use Cases

For Research Question 1 about streaming use cases, we looked into the use cases of stream practices at ING, their workflow and the frequencies of events related to the use case to happen. Additionally, we also looked into the number of users of their streams.

#### Streaming Use Cases

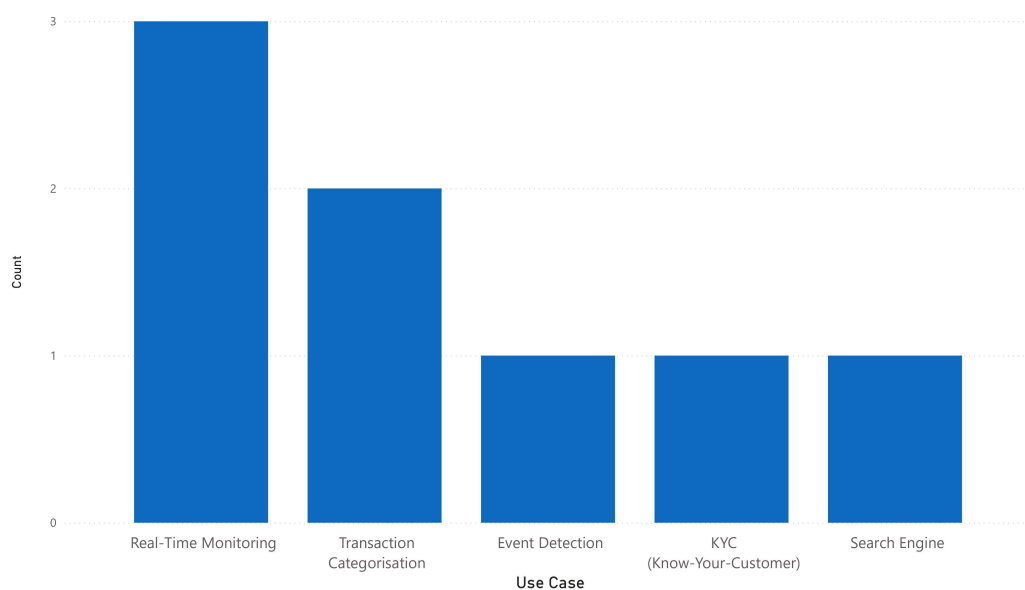


FIGURE 6.7: Distribution of the streaming use cases at ING that our questionnaire participants worked on

We posed our questionnaire participants with *Question 27: For what use case(s) are you using your stream?* to gain information on what purpose was data streaming used for within ING and *Question 28: Describe the workflow of your use case of stream* to understand how streams are being used for this use case. Figure 6.7 shows the distribution of our questionnaire participants answer about their streaming use case. We got more insights from the interviews on the use case of Transaction Categorization, Search Engine, and KYC and some additional use cases which is Trading Prediction.

Combined insight from questionnaire and interview results are explained in the following:

- **Real-time Monitoring** is the most common use case of data streaming practices in ING with 3 questionnaire participants chose this answer. Events were collected from various system through system logs, application alerts, etc. These collected events were then ingested to the streaming environment to be filtered and enriched before being forwarded to as notification to corresponding stakeholders. Events in the streams were filtered based on specific monitoring purposes. Filtered original events were then being enriched through the streaming process by combining them with related information from other sources such as database, queue system, and logs. Finally, enriched data were forwarded to the monitoring tools where data is visible and can be used to analyzed target incidents.
- **Transaction Categorization** aims to provide real-time money management service to the customers. Transaction and product interaction data were ingested to streams as soon as it happened. Ingested transactions data in streams were first transformed to have uniform structure and applied customer rules such as consent were checked before processing. Uniformed data were then categorized in real-time manner where known category were stored in the state of the streams. If the categorization already exist within the state then the transaction can be categorized right away. Otherwise, the transaction data will be forwarded to a separate machine learning model where categorization were made based on mapping rule of transactions data and product categories. Categorization result where then ingested back to streams that forwarded it to relevant stakeholders.
- **Know-Your-Customer** is identification and verification process of customers to make sure that the customer relationship with the bank is in compliance with the laws and regulations. Real-time processing was needed for KYC use cases to reduce the time required to verify customers. All kinds of data needed for the KYC process were collected from different sources and ingested to streams. The data can be customer information, regulation data, transactions, customer activities, etc. Any updates of these data will be processed as soon as possible within the streams. The results will be reported to relating teams where they could see the processed data in a form of real-time review.
- **Search Engine** aims to built an internal centralized search library of data sources, such as documents, where users can find internal information needed for their work. Any creation, updates, or deletion of documents were detected from various sources. These events were then ingested to the streaming environment. Documents were processed to retrieved its metadata such as title, owner, date created, security information, etc. Document components such as text and tables were also extracted from the original documents. Streams were used to coordinate the ingestion and curation of documents and events related to the documents.

### Users of Streams

We asked *Question 7: What is the estimated number of the users of your stream(s)?*. The definition of users may vary thus we provided a brief guideline of what we considered as users in our question. Users are people who uses (consume data from) your

TABLE 6.2: Distribution of the answer of number of streams users

Streams Users Range	Count
1 - 10	2
10 - 50	0
50 - 100	1
100+	2
Invalid	2

TABLE 6.3: Overview of the answer for entities represented in data within streams from both questionnaire and interview

Entities Represented	
Transactions	Business Data
Security Events	Business Process
Software/Application Activity	Business Product
Customer Information	Business Companies
Customer Activity	Country Domain
Credit and Loans	Customer Security Rule
Product Interaction	Document Metadata
System Machine State	Technical Master Data
System Failure Log	Technical Reference Data
System Network Traffic	

stream(s) for their own purposes i.e. data engineers from a business intelligence team use your stream(s) to their visualization platform. Table 6.2 shows that two respondents answered that they had 1-10 users for their streams, 1 respondent with 50-100 users, and 2 respondents with around 100 users. Two of the respondents gave invalid answers.

## 6.4 RQ2: Streamed Data Characteristics

To answer our Research Question 2 about characteristics of data contained in the streams, we looked into the numbers of streams that our participants handled, entities represented in their streams data, average throughput of their streams, maximum throughput of their streams, format of data in the streams, data category, and data types contained in their streams data points.

### Number of Streams Handled

For *Question 10: How many stream(s) do you handle?* in our questionnaire, the answers we got from the participants ranges from 1-10 streams except for one participant that answered with 29 streams. These insight are aligned with the information we got from our interviews, two of our interviewees mentioned that they handled around 10 streams and two other interviewees answered with around 30 streams.

### Entities Represented

Table 6.3 shows the answers for our questionnaire's *Question 11: Which entities are handled in your stream?* and entities mentioned by our interviewees. Based on the

TABLE 6.4: Distribution of the answer of average streams throughput

Average Streams Throughput	Count
<10	1
10 - 100	2
100 - 1,000	3
Fluctuates	1

participants responses, Transactions, Security Events, Software/Application Activity, Customer Information, and Customer Activity were most represented in their streams.

Combining answers from questionnaire and interview, the summary of entities represented in our participants streamed data are the following:

- **Business:** The streams contained business related data such as Business Data of external business companies profile across the Netherlands, Business Product of external companies, Business Process within the ING, Product Interaction that represent the interaction within products and customers, and Country Domain of the business companies.
- **Customer:** Their data streams also contain data that represent people as a customer such as Customer Information which hold profile data of a customer within ING, Customer Rule that illustrate different rules applied for each customer, and Customer Activities within ING.
- **Finance:** Some financial data were being utilized in the stream processing practices. The data represent entities such as Transactions data within ING, Credits and Loan data, and Financial Markets data.
- **IT Infrastructure:** Data that represent the states and activities of the IT infrastructure were also being processed within the streams such as System Machine State data, System Failure Log, System Network Traffic data, Security Events data, Tech Master Data that represent important technical data and Tech Reference Data that represent relation between the technical data.

### Average & Maximum Streams Throughput

We asked our participants *Question 12: How many data points does your stream(s) processes per second, on estimated average?* and Table 6.4 shows the answers. Most of our participants (three) were dealing with streams with throughput of 100 to 1,000 data points per second. Two participants stated that they were dealing with 10 - 100 data points per second and only one participants answered with less than 10 data points per second. From our interviews, the insight on data point processed per second are aligned with the result of our questionnaire. Our interviewees handled streams with speed as low as 0.002 and 0.02 data points per second. Within the range 10—100, there were streams with around 50 data points per second on average. For the range 100—1,000, our interviewees mentioned that they handled streams with 115 and 350 data points per second. There were also streams with speed above 1,000 mentioned by our interviewees which are streams with 3,400 and 10,000 data points per second.

We also asked our participants *Question 13: How many data points does your stream(s) processes per second, at an estimated maximum?* and Table 6.5 shows the distribution

TABLE 6.5: Distribution of the answer of maximum streams throughput

Maximum Stream Throughput	Count
10 - 100	2
1,000 - 10,000	2
Unknown	3

of their answers. Three participants stated that the maximum throughput of the streams that they were dealing with was not known to them. Two participants stated that their streams maximum velocity was in the range of 10 to 100 data points per second and two other participants answered with the range of 1,000 to 10,000 data points per second. Maximum streams throughput were not discussed in our interviews.

### Streamed Data Format

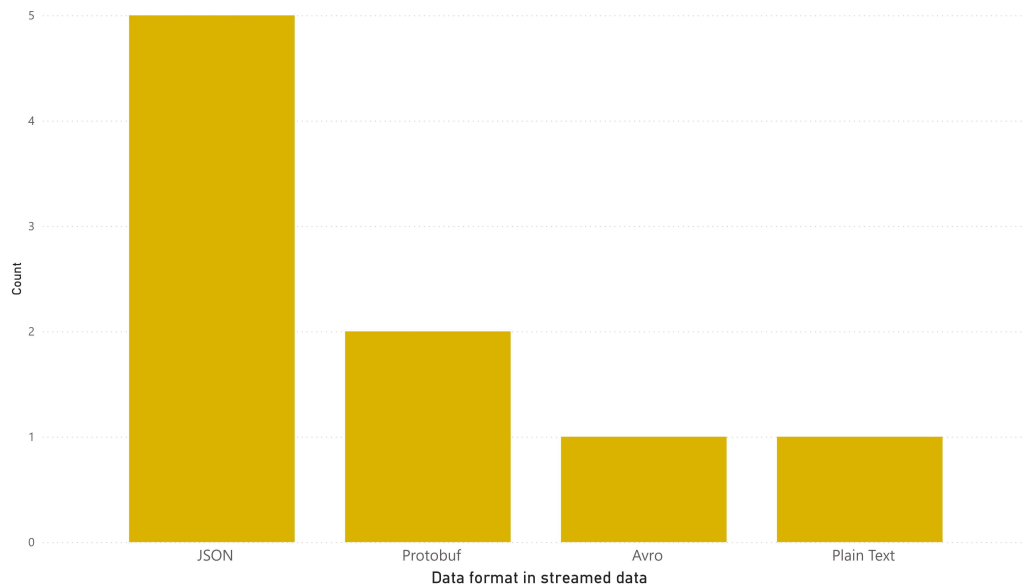


FIGURE 6.8: Distribution of the questionnaire response for data format in streams

In the questionnaire, our participants was posed with *Question 14: What is the format of the data processed on your stream(s)?*. Figure 6.8 shows the distributions of our participants answers. JSON format was used most within our questionnaire participants where 5 of our respondent said that their streams contained data in JSON format. Two participants responded with Protobuf, one participant with Avro and another one responded with Plain Text.

Insights we got on data format from the interviews were aligned with our questionnaire result. In one of our interview, it was explained that there were not necessary any standardized data format for stream processing within ING since they could not really enforce various system of record on how to extract its data because of its different capabilities and limitations. However in all of our interview, JSON was mentioned as the format of data they used to ingest to their stream. Another

TABLE 6.6: Distribution of the answers of streamed data category

Data Category	Count
Free Text	3
Time Series	3
Relational Data	2

data format used by our interviewees are Avro and Protobuf. Avro is a language neutral data serialization protocol that uses JSON format to store row-oriented data. Similar to Avro, Protobuf is also a protocol to serialize structured data but it has its own JSON like schema that can be easily compiled for open-source language such as Java or Python. One of our interviewees mentioned that they processed a lot of XML formatted data from the HTTP and FTP connection that they did to get data. These XML formatted data were then transformed into JSON as well before being ingested to their streams.

### Streamed Data Category

We asked our participants *Question 15: Which category best described the data in your stream(s)?* in our questionnaire. Free Text, Time Series and Relational Data category was chosen to describe their streams data by respectively three, three and two respondents. Spatial Data, Multimedia Data and Graph Data was not chosen by any of our respondents.

### Data Points Type

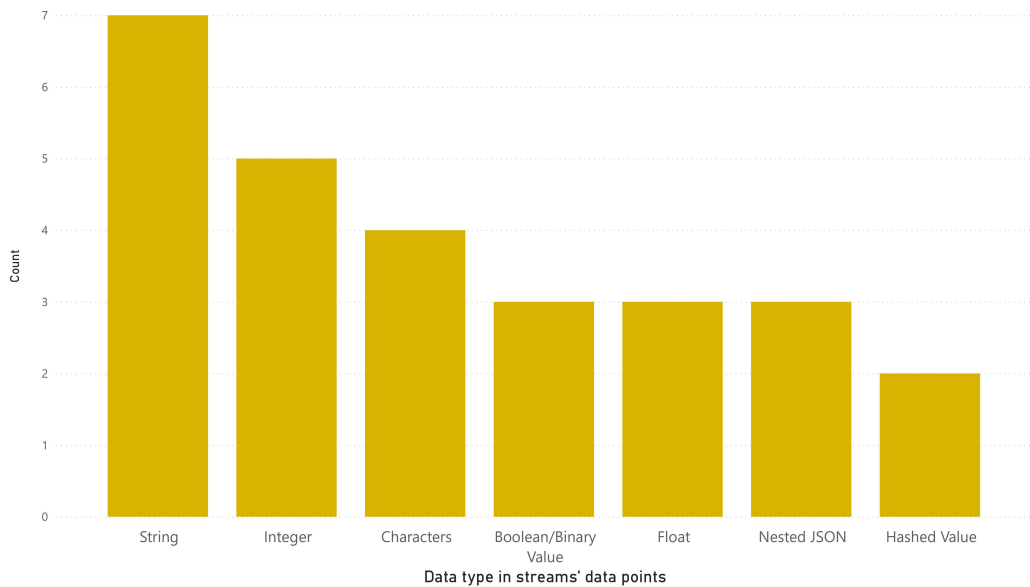


FIGURE 6.9: Distribution of the questionnaire response for type of data in streams data points

We then also posed the participants with *Question 16: Which data type(s) are contained in your streams' data point?* and Figure 6.9 shows the distribution of our participants answers. String data type was most contained within our participants streams.



All seven respondents who used streams in their work answered that their streams contained string data points. The second most contained data type within our respondents streams was integer with five respondents answers and the third was characters with four respondents. Boolean/binary Values, float and hashed value data types were also contained in our participants data streams although not as much as string, integer and characters. We got three respondents said that their data point could also be another JSON data structure. This indicates that there could be nested JSON structure in our participants streams.

### Data Source

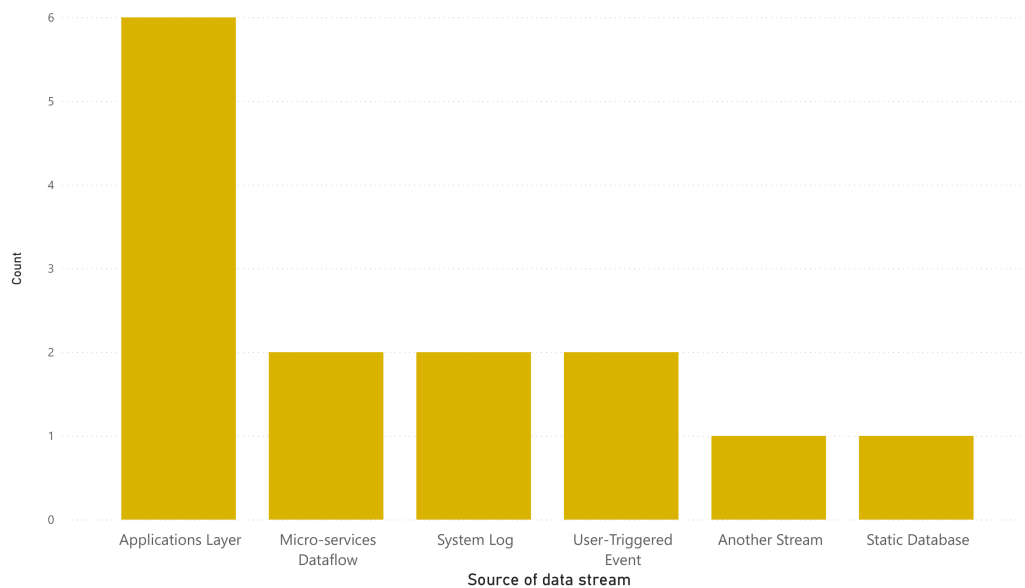


FIGURE 6.10: Distribution of the questionnaire response for their streams data source

We wanted to see from where does data of our participants' streams came from so we asked our participants *Question 22: What are the sources of your stream(s)?*. Six out of seven of our participants who used stream in their work answered with application layer as their source of streams. Micro-services dataflow, system log, and user-triggered event answer options were chosen by two participants each. One person said that their source of streams is another streams and one other person answered with static database.

## 6.5 RQ3 & RQ4: Streaming Task & Machine Learning Computation

In this section, we will discuss about the results of our questionnaire that answer our Research Question 3 & 4. We looked into the stream processing tasks that our participants did and what they did with the streams output. For each tasks, we also investigate what the type of tasks commonly carried out by our participants. We then explored if our participants implement any kind of machine learning computations with their streams.

## Streaming Tasks

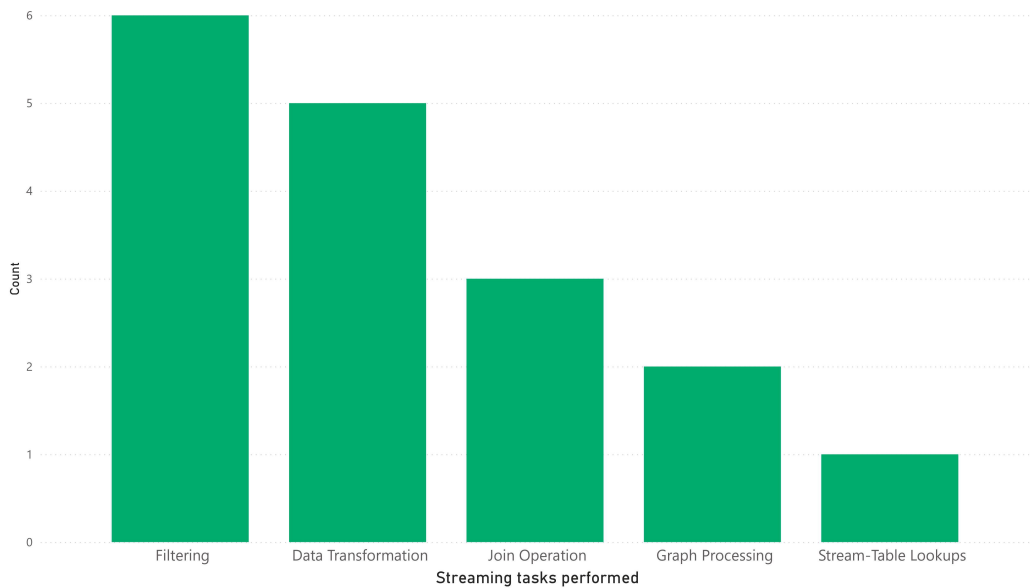


FIGURE 6.11: Distribution of the respondents' answer for the streaming tasks they performed

Based on our academic review result on streaming tasks from Section 4.5, we wanted to see which of these tasks were commonly performed by our participants thus we asked them *Question 17: What kind of streaming task(s) do you perform?* in our questionnaire. Figure 6.11 shows the answers distribution of our participants. Three streaming tasks that were most performed by our participants are:

1. Filtering (with 6 respondents),
2. Data Transformation (with 5 respondents), and
3. Join Operation (with 3 respondents).

We got answers with two new streaming tasks that was not included in our academic review results which are graph processing and stream-table lookups. Two respondents said that they did graph processing tasks in their stream and one respondents answered with stream-table lookups task.

We asked our participants on what kind of join operation, aggregate queries, and windowing technique that they used. These are some insights that we got from the answers:

- Similarity and equality join were being used by our participants. Two of our participants said that they used similarity join and one of our participants used equality join for their join operation.
- Aggregation queries that were performed within our participants was sum aggregation. One of our participants said that they were doing sum aggregation within their monitoring window interval.
- Sliding window was used by one of our participants. Two other participants said that they applied windowing to their streams but the technique was not known. Our participants window interval ranges from 1 to 5 minutes.

TABLE 6.7: Distribution of the participants response on implementing machine learning computation within their streams

Implement Machine Learning	Count (Percentage)
Yes	5 (72%)
No	2 (28%)

### Machine Learning Computations

We also asked our participants *Question 23: Do you perform any machine learning computation in the stream or at the end of the stream?* to investigate how common that machine learning computations are implemented in our participants streaming environment. Table 6.7 shows the response distribution to Question 23. It can be seen that machine learning implementations in streams were frequent as 72% of our participants answered with "Yes".

To follow up the response from participants who implemented machine learning computation in their streams, we asked them to explain the machine learning computation that you do and its involvement with the stream. Four out of five respondents used machine learning computation in their stream for classification purpose such as transaction categorization, data tagging based on certain criteria. From these respondents, two of them used Naive Bayes model for the classification. One other respondent said that machine learning computation was not done within the streams but the streams acted as an input for the machine learning model. From the interview, one of our interviewees used machine learning model that is integrated to the streaming environment. The machine learning model were implemented separated from the streams flow. The model needs to be uploaded as a specific file format to an ING in-house stream analytic platforms where it can be updated in batch. The stream operators were then able to forward data to this model and receive learning result from the model in low latency.

## 6.6 RQ5: Streaming Software & Tools

Research Question 5 is about streaming software and tools used by practitioners thus we looked into which stream processing engines that our participants used and other kinds of tools or services being used alongside the stream processing engines.

### Stream Processing Engines

We posed our questionnaire participants with *Question 25: What stream processing engine do you use?* and Figure 6.12 shows the distribution of our participants' answers. It can be seen that within our participants, Apache Kafka is the most used stream processing engines with 6 response from our participants. Apache Flink was used by 3 of our participants, one participant used Akka Streams and one participant used an in-house stream processing engine. Insights we got from interviews are also aligned with the questionnaire results, stream processing engines that were used by our interviewees are Apache Kafka, ING in-house stream processor, Apache Flink, and Akka Streams.

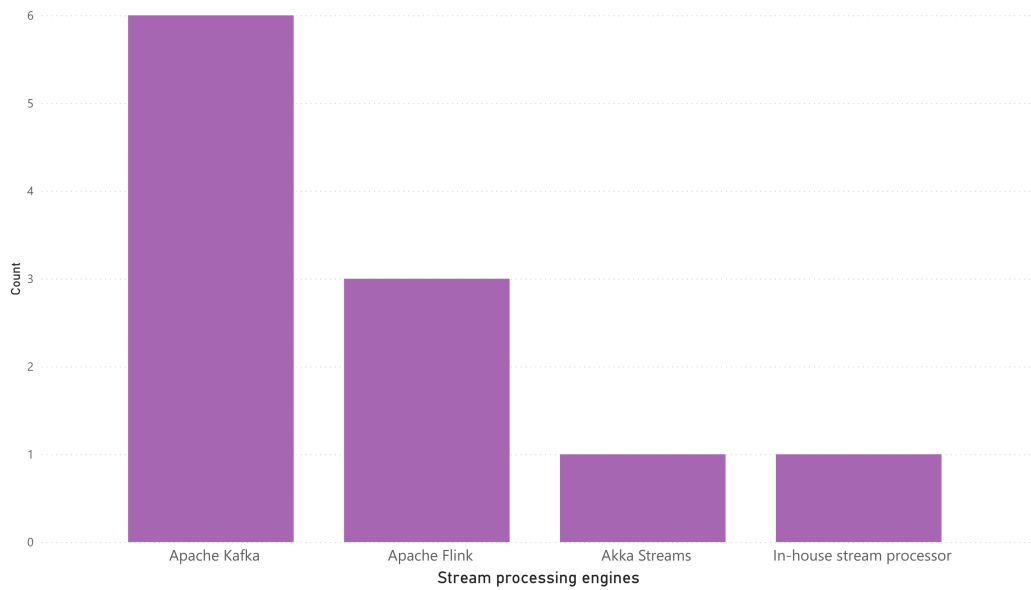


FIGURE 6.12: Distribution of the respondents' answer for stream processing engines that they used

### Other Supporting Tools

We then asked *Question 26: What other type of tools/services do you use alongside the stream processing engine?* to our participants. Our participants used workflow management platform such as Airflow, data visualization tools, static database management systems such as Postgres, message broker services, monitoring platforms such as IBM Tivoli Monitoring and search engine services such as Elasticsearch.

## 6.7 Challenges & Expectation in Streaming Practices

We wanted to investigate the challenges and expectation that practitioners faced while working with data streams to gain insight on what can be improved in the approaches of data stream processing. Thus, we looked into any challenges, problems or constraints encountered, how our participants dealt with these challenges and what they perceived as the ideal solution for these challenges. We also asked our participants on features they would like to see in stream processing engines. Here are some remarks we gained from our participants answers:

### Simplicity of Configuration

The most mentioned challenges by our participants were the complexity in setting up the streams. Setting up process of stream processing engines could involve a lot of technical work that was beyond our participants skill range such as configuring password vaults, certificates, etc. Two of our participants [I2 & I4] stated that it was a challenge to learn the technical knowledge in addition to setting up the platform itself. Thus, our participants wished for a simpler way of setting up stream processing engines. Ideally, the set up process should require little to no technical work.

### User-Friendly Implementation

One of our questionnaire participants [R42] mentioned that using a fairly low level stream processing engine was a challenge for the team. It was not perceived as user-friendly engine as it require a deep technical understanding and knowledge to smoothly utilize the engine. We got an example of this case from one interviewee [I4] who said that because of the low level nature of the streams implementation, join operation was a difficult thing to do. They had to process the same data that through different streams at the same time. It was not promised that the data order in different streams will be kept the same. When they had to combine processing results from these streams, it was challenging to join the exact same data as there were no idem-potency identifier between the data.

Most of the open-source stream processing engines were using Java or Scala programming language for its implementation. Another insight from the questionnaire result is that the in-house processing stream processor platform used by the participants required them to use a custom programming language. From Section 6.2, we can see that practitioners of data streaming were not always engineers. Compared to DevOps Engineers and Data Engineers, users such as Data Scientist might not have as much technical experience. One interviewee [I2] said that it was a bit of a learning curve for users with less technical role because they were not really familiar with the low-level technical knowledge.

The expected solution that we obtained from our questionnaire participant and interviewees is that as a users they would like to be able to do stream processing with a less low level programming language such as Python. Additionally, they also wished for clear documentation and greater availability of support on both programming language and features of the stream processing engines.

### Centralized Streaming-As-A-Service

One of our questionnaire participants [R48] stated that it was a challenge for them that there was no central streaming engine at the company. They stated that they had to use a rather low level stream processing engine. This statement was supported by the insights we got from our interviews. One interviewee [I3] said that for their use case, they only needed a simple usage of stream processing engine. They were having a dilemma in choosing stream processing engines because they needed the capabilities of a more advanced stream processing engines but their team did not have the capacity to do the maintenance required for these engines. At the end they chose the less advanced stream processing engines that require little to no maintenance from their team but with less feature and capabilities. Another interviewee [I5] said that there were a lot of needs around the company to use stream technologies for their work so the demand is quite high.

So having a centralized streaming-as-a-service platform was one of the thing in the wish-list of our questionnaire participant and interviewees. They would like to be able to use it as a service where you can use as a tools out of the box and is easily integrated with the architecture and other services used within the company. An interviewee [I4] explained an example of the ideal central streaming service feature that allows them to simply choose the input for their streams, easily schedule and publish jobs to clusters, and given their own private network for clusters that they do not have to manage themselves.

### Embedded Machine Learning Model in Stream Processing Engines

Another challenge expressed by our interviewees were that the procedure on implementing machine learning model in the streaming environment comes with several limitations. One interviewee [I2] said that the file size of the machine learning model that they could upload to the stream processing engine was limited. It made them unable to do any kind of deep learning in their streaming environment. Another interviewee [I1] also stated that the machine learning model available to be used with the stream processing engines were still limited. Another challenges mentioned during our interviews was the customized language used in ING in-house stream processor to create the training model. Since it's a customized language, it was harder for the user to look for support when faced with programming challenges for instance they would have to ask to their colleague instead of being able to look for the solution in the internet.

In that case, our participants expected stream processing engines to be able to provide more machine learning model and increase the size limitation of model file that can be uploaded to the streaming environment. They also wished for the ability to create the model in popular programming language such as Python and to upload the model to streaming environment in pickle files.

## 6.8 Observations

During the process of analyzing our survey results, we came upon several interesting observations:

- *High Desire, Challenging in Practice:* We saw that high desire and interest on using data streaming technologies were expressed multiple times during our questionnaire and interview. Most of our participants interest towards data streaming was driven by their enthusiasm to improve the data processing speed and produce data processing results in real-time manner. However popular data streaming was within our participants, the implementation itself was challenging to the practitioners. Transforming their current system legacy to provide streaming infrastructure and maintaining the streams itself required a lot of work both from organizational and technical perspective. This factor affected the decision whether or not streams should be used and also how the implementation would be.
- *Data Enrichment Application:* Among our participants use cases, data streaming technologies were mostly used for enriching the data as it being ingested to the streams. The data enrichment process was typically done by collecting reference data from different source and using it to add value to the main data in the streams by utilizing standard streaming tasks such as join, filter, aggregation, and map.
- *Uniformity in Streamed Data:* We observed that there were a degree of uniformity of data in our participants' streams in the aspect of streams amount, velocity, format, type and entities represented based on several findings below:
  - The common number of streams handled by our survey participants are within the range of 10-30 streams.
  - Within our survey participants, their average streams throughput can be as low as <1 data point per second and can be as high as 10,000 data

points per second while the maximum stream velocity vary from 10 to 10,000 data point per second. Most participants did not really know the exact range of the throughput. Streams handled by our participants has lower range in throughput compared to streams performed in academic work which range from 1,000 to 33 million data points per second.

- Most common format of data contained in streams is JSON. This insight on data format was aligned within our findings on academic publication review, questionnaire answers and interviews results.
  - Temporal or Time Series data were prevalence on being processed using data streaming in both academic and practitioner works. In academic research works there were more usage on Spatio-Temporal, Spatio-Textual, Graph and Image dataset, while these categoris of dataset were not processed by practitioner in our survey. Our participants did process Textual data in their streaming environment, however no graph, image, or spatial data were processed.
  - Entities represented in our participants streamed data were focused on business, finance, IT infra and digital information which was much aligned with the insight we got from academic publication review.
- *Prevalence of Machine Learning in Stream Processing:* Usage of machine learning computation in data stream processing was popular within practitioners participated in our survey and researchers from the academic publications we reviewed. Most of their stream use cases hold analytic purposes where learning is needed such as categorization, event detection, data extraction, etc. While several papers mentioned that they did online learning, our survey participants implement their machine learning model offline from the streams. Practitioners indicated their expectations on stream processing engines to improve their machine learning feature so that it has more learning model and become easier to utilize.
  - *Easy-To-Use Wish:* Most challenges of data stream processing that were expressed by our participants are around the themes on easiness of implementing data streaming. From setting it up to the maintenance of the streams, practitioners pointed out that these processes still required a lot of technical knowledge and skill. Not having the correct resource to implement data streaming in their team has also been one of the reason of why some practitioners choose to not use streams. Thus, our participants wished that stream processing engines could be easier to use so that non-technical people can utilize it as well.

## 6.9 Reflection on Survey

### Response Rate

We thought of some possibilities of why the response rate of our questionnaire was not as high as we expected it to be. First, it could be because of the unfamiliarity with the concept of data streaming itself within the invited potential participants. They invitees were either never heard of the term data streaming or that they familiar with the term but did not know what it is exactly. We got one reply for our invitation stating that they would like to participate in the questionnaire but did not know what data streaming is. A similar case to this was that some potential participants have various understanding of data streaming making them hesitated to fill in the

questionnaire. One of the reply to our invitation was saying that the definition of data streaming was not clear to them although they would be happy to participate in the questionnaire. We then described our meaning of data stream processing to them and explained that they were still able to participate in the survey since we also have section for people who did not practice data streaming.

Secondly, there were no exact way to confirm that we have sent the invitation exhaustively to every practitioner at ING who could potentially use data streaming in their work. As stated by Sahu et al., 2017, it is a known challenge for researcher in academia to acquire direct access to practitioner from industry. To the best of our knowledge, specific mailing lists or forums for data streaming practitioners were non existence within the company. Thus, we could only depend to the networking extension of our colleagues from ING.

Another factor that could contribute to our response rate was the potential participants inability to answer the questionnaire completely. We got two replies for our invitation explaining that they could not fill our questionnaire until the end because they could not answer some of the mandatory questions. They could not answer these questions because it was too technical for them or they simply did not know the answer.

## Biases

A survey research provides challenges to the methodology of recruiting participants, selecting a list of answer options for the questions and executing the academic publication reviews. We wanted to be as comprehensive as possible when dealing with these aspect and to prevent impromptu choices.

We recognize these biases when we report our findings in previous sections. The data we present, in particular, should not be statistically evaluated. Our objective was not to study the statistical properties of the data streaming practices in industry or the practitioners themselves. Despite these biases, we have discovered insightful observation from several of our findings that provide us an understanding into how data streaming are utilized in practice.

We also acknowledge the bias that is produced from the different understanding of what data streaming is. As explained in Section 2.1, there are several other concepts that commonly misunderstood as data streaming practices. It indicates that there could exist a degree of misalignment between data streaming definition within practitioners in the industry and researchers in academia.

While reading the result of this survey, it should also be taken into account the scope of the survey and its participants. The range of this survey participants are limited to the industry of financial service and to the environment of one company. Thus, these results should not be perceived as a full representation of data streaming practice in the industry but merely as an insight that initiate a bridge of knowledge in data streaming between the industry and academia.

## Lessons from the Survey Methodology

We highlight several lessons from our experience of implementing our methodology for both questionnaire and interview. The lessons we got are the followings:

- Lesson 1: It would probably a good idea to use a more trivial statement for the survey title to attract more participants for our questionnaire i.e. *To stream or*



*not to stream?*. Increasing the clarity of data streaming definition in the questionnaire would help reducing hesitation of potential participants to fill in our questionnaire.

- Lesson 2: We could get richer insight on why data stream processing was not practiced by some practitioners by asking the use case of the work of those who did not use stream. This way we could have more observation on the reason of why streams was not used by linking the reason and the use case.
- Lesson 3: Performing a discussion on the initial questionnaire design with some practitioners really helped improving our questionnaire to be perceived better from the practitioner's perspective. The discussion provided useful feedback on which questions and answer options that do and do not make sense for practitioners.
- Lesson 4: Adding guidance or increasing the specificity to some questions would help our participants answer our questions correctly. A good example of this are open ended questions. Open ended questions can either provide rich answers or leads to various answers that's hard to understand. By providing enough guidance to the question, it would increase the chance of getting the answer we wanted.
- Lesson 5: Questionnaire participants and interviewees could have different understanding on technical terms. This resulted in more time required to understand the meaning of their answers. We learned to thoroughly study our participants use case and area of work to put context to terms they were using.
- Lesson 6: Interview questions should have only including questions related to the topics we wanted to focus on the interview. In our case, questions about streaming data and streaming task are not our focus for the interview. These questions turned out to be not as effective as we thought it would be and could even mislead the focus of the interview.



## Chapter 7

# Conclusion

This thesis aims to provide insight and observation on data streaming practices in the industry. In this section, we present the summary of our survey research by providing concluded answer for each research question in Section 7.1. In Section 7.2, we also explained the possibility of future work that can be continued from our research.

### 7.1 Summary

Our survey research were divided into 3 main phase which were academic publication review, survey questionnaire and individual interview of the practitioners. In the academic publication review phase, we obtained 210 papers to be reviewed from 4 top data management conference of year 2018 to 2020. The review process resulted in 12 literature properties extracted for our research questions. These literature properties were then used to design the questionnaire for the practitioners. Next we sent our questionnaire to 597 active emails ING potential stream practitioners and received a 7.5 percent response rate with 45 respondents. Based on the questionnaire result, we identified area of questions that we still need to dig deeper. These area were then become the focus of our interview. We interviewed 5 ING's employee where posed questions about streaming use cases, machine learning implementation, and challenges and expectations in data streaming practices. Based on the obtained results we summarized the answers for each our research questions as follows:

#### **Research Question 1: What use cases do users implement their streaming pipeline for?**

As financial service industry practitioners, data streaming was practiced within our survey participant for several use cases such as real time monitoring system, transaction categorization, know-your-customer (KYC), search engine and event detection. The motivations to use data streaming technologies in their work were to fulfill their need of real-time data processing, faster decision making, improvement in processing speed and optimization of processes and resources. For some use cases, it was impossible to solve the problem without using data streaming technologies. Furthermore, from the academic publication review we obtained insight that researchers did data streaming practices for use cases of anomaly detection, monitoring system, event detection, search in streams, continuous recommender system, finding significant items, graph processing, and pattern recognition to solve problems in various fields of industry such as transportation & logistics, IT & telecommunication, health, social media, urbanism, retail, etc.

**Research Question 2: What types of streamed data do users have?**

We observed the types of data contained in our users streams based on its velocity, data format, data category, data points type, and entities represented by the streamed data. Streams handled by practitioners of our survey participants had the average of velocity from as low as <1 data point per second to 10,000 data points per second. It is comparably low than the average velocity of streams handled by researchers from our academic publication review which was ranging from 1,000 to 33 million data points per second. For data format in streams, the most common format used by both practitioners and researchers in our survey scope is JSON. Temporal and Textual data were categories of data that was commonly used within practitioners streaming practices while researchers in academic had broader categories of data in their streams such as Temporal, Spatio-Temporal, Spatio-Textual, Graph Dataset and Image Dataset. Types of data in the streams data points were commonly string, integer, and float. Finally, data contained in our practitioners streams represent real world entities such as business, finance, IT infrastructure and digital information that focused on financial service industry. Meanwhile for researchers within our academic publication review that was not limited to financial service industry, their streamed data represent more entities such as human, knowledge, and other kind of infrastructure for instance physical and electrical infrastructure.

**Research Question 3: What kind of streaming task & computations do users run on their streams?**

From our academic publication review, we conclude that streaming tasks can be divided into two groups of Data Operator Task and General Task. Data operator task group includes Join and Aggregation tasks that are stateful and tasks such as Filter, Map, and Union that are stateless. General operator task group include tasks such as Querying, Windowing, and Partitioning. We obtained an insight from our survey that Filter, Map, Union, and Join are the most common streaming tasks done by practitioners. Practitioners used Similarity Join and Equality Join techniques for their join tasks and Sliding Window technique in implementing windowing for their streams.

**Research Question 4: Which machine learning task users perform in their streaming pipeline?**

Machine learning implementation in streaming environment were prevalent in both researchers and practitioners within our scope survey. We understood based on our review of academic publications that clustering and classification were the most common machine learning computation performed in academic work. 72% of practitioners from our questionnaire participants implemented machine learning computation within their streams. From our questionnaire and interview, we recognized that practitioners implement their machine learning computation offline from the streaming environment. They expected for stream processing engines to have more advance machine learning feature that allows them to use more machine learning model and utilize machine learning computation easily within streams.

**Research Question 5: What software & tools do users use to perform their streaming processes?**

Apache Kafka, Apache Flink, Apache Storm and Spark Streaming were the popular stream processing engines within researcher from our academic publication review and practitioners from our survey. Apache Flink was most used amongst researchers while practitioners mostly used Apache Kafka. Additionally, practitioners at ING also used their own in-house stream processor. Aside from stream processing engines, other tools were also used to support the processing of data streams such as workflow management platform, data visualization tools, static database management systems, message broker services, monitoring platforms, search engine services and visualization tools. Although practitioners were generally satisfied using these stream processing engines, they hoped for easier set up and implementation of the engines as it still required a lot of technical knowledge to do.

## 7.2 Future Work

### Deeper Survey Research

In this survey, we opt to focus on a practitioners aspect on data stream processing. As this is the first research about data streaming practices, the purpose of the survey is to get general observation about data streaming practices in ING. The result will serve as an initial insight to be continued in future research. Based on the results we get for each research questions, we figured that each research questions can be made as a new independent survey research such as survey about data processed in streams, streaming task & machine learning computations within streams, and stream processing engines experience within practitioners. It would also be interesting survey research topic to focus on why people use and don't implement streaming practices in their work to understand what motivates people.

### Extend Survey to a Broader Audience

Our survey research can also act as a pilot study of a more generalized survey on stream processing techniques and technologies from the practitioner's point of view, as its scope was still limited to stream practitioners from ING. Thus, the insight we got from our survey was bounded to the financial service industry. Although our survey managed to accomplish the desired result, executing another survey with participants from various industrial areas will provide a richer insight on how data stream processing is being used in practice. Extending our survey to a broader audience is easy and straight forward, since our survey methodology is already proven to be applicable in the small scope of participants.

### Prototype for The Wishlist

We obtained various insights on the challenges practitioners face when implementing data streaming in their work such as the complexity in setting up and configuring the streaming systems, the big overhead of maintenance work, and the difficulties in embedding machine learning models in their streaming pipelines. Based on our survey, participants could come up with specific ideas of how these challenges could be solved from their point of view. This creates an opportunity to do a research that explore the challenges and examine the requirements from both academic and industrial practitioners which can in the end lead into producing a prototype.



## Appendix A

# Data Stream Processing Questionnaire

### DATA STREAM PROCESSING SURVEY

- Multiple choice
- Single choice
- \* Required to be answered

#### Section 1

<p>In which category of department at ING do you work? <i>(This list is retrieved from ING HR Site, please state in the "Other" option if the department you're working on is not on the list)</i></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Information Technology</li> <li><input type="checkbox"/> Sales / Relationship Management</li> <li><input type="checkbox"/> Risk Management</li> <li><input type="checkbox"/> Wholesale Banking</li> <li><input type="checkbox"/> Administration / Operations</li> <li><input type="checkbox"/> Legal / Tax / Compliance</li> <li><input type="checkbox"/> Retail Banking Expertise</li> <li><input type="checkbox"/> Accounting / Finance</li> <li><input type="checkbox"/> Human Resource</li> <li><input type="checkbox"/> Contact Centre</li> <li><input type="checkbox"/> Audit</li> <li><input type="checkbox"/> Economic Research / Strategy</li> <li><input type="checkbox"/> Marketing / Communication</li> <li><input type="checkbox"/> Project / Programme Management</li> <li><input type="checkbox"/> Asset / Portfolio Management</li> <li><input type="checkbox"/> Facilities / Procurement</li> <li><input type="checkbox"/> Sales Support &amp; Operations</li> <li><input type="checkbox"/> Lending</li> <li><input type="checkbox"/> Websites Management, Internet &amp; Mobile</li> <li><input type="checkbox"/> Project / Program Management</li> <li><input type="checkbox"/> Sales Support / Internal Account</li> <li><input type="checkbox"/> Other: ....</li> </ul>
<p>What is your role/job position? ...</p>
<p>If you're willing to share, what is the name of your team? ...</p>
<p>How big is your team?</p> <ul style="list-style-type: none"> <li><input type="radio"/> 1 – 5 people</li> <li><input type="radio"/> 5 – 10 people</li> <li><input type="radio"/> 10 – 20 people</li> <li><input type="radio"/> Other: ...</li> </ul>
<p>Do you (or your team) implement any kind of data streaming practices in your work?</p> <ul style="list-style-type: none"> <li><input type="radio"/> Yes</li> <li><input type="radio"/> No</li> </ul>

If you answer no, please continue filling in this section only. If you answer yes, please continue from the next section.

<p>What is the reason that data streaming technologies is not used within your team?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> The current solution works just fine</li> <li><input type="checkbox"/> Data streaming technologies is not the right solution</li> </ul>
--

<input type="checkbox"/> There's other solution alternative that is more doable at the time <input type="checkbox"/> Unfamiliarity with data streaming technologies <input type="checkbox"/> Not enough resource to implement data streaming technologies <input type="checkbox"/> Other: ...
What kind of data processing technologies that you and your team currently using? <i>(i.e., Message broker, DBMS, etc)</i> ...

This section onward is about data streaming technologies & practices.

What is the estimated number of the users of your stream(s)? <i>Users are people who uses (consume data from) your stream(s) for their own purposes. For example, data engineers from a business intelligence team use your stream(s) to their visualization platform.</i> <input type="radio"/> 1 – 10 <input type="radio"/> 10 – 50 <input type="radio"/> 50 – 100 <input type="radio"/> Other: ...
How long (in years) have you been working with streaming data? Including also your experience outside ING. ...
How would you consider your knowledge level about streaming data? <input type="radio"/> Beginner (i.e., have been working with an existing streaming set up, have made minor changes to the streaming architecture) <input type="radio"/> Intermediate (i.e., experienced in setting up a streaming pipeline, made major changes to the streaming architecture) <input type="radio"/> Expert (i.e., experienced in customizing or creating a streaming algorithm)

## Section 2: Streamed Data

How many stream(s) do you handle? ...
Which entities are handled in your stream? <input type="checkbox"/> Transaction <input type="checkbox"/> Credit and Loans <input type="checkbox"/> Tax <input type="checkbox"/> Assets <input type="checkbox"/> Liabilities <input type="checkbox"/> Equity <input type="checkbox"/> Customer Information <input type="checkbox"/> Customer Activity <input type="checkbox"/> Product Interaction <input type="checkbox"/> Trade Flow <input type="checkbox"/> Economic <input type="checkbox"/> News <input type="checkbox"/> Social Media <input type="checkbox"/> System Network Traffic <input type="checkbox"/> Machine State <input type="checkbox"/> System Failure Log <input type="checkbox"/> Software/Application Activity <input type="checkbox"/> Other: ....
How many data points does your stream(s) processes per second, on estimated <u>average</u> ? <input type="radio"/> 0 – 10 <input type="radio"/> 10 – 100



<ul style="list-style-type: none"><li><input type="radio"/> 100 – 1K</li><li><input type="radio"/> 1k – 10K</li><li><input type="radio"/> 10K – 100K</li><li><input type="radio"/> 100K – 1M</li><li><input type="radio"/> 10M – 100M</li><li><input type="radio"/> Other: ...</li></ul>
<p>How many data points does your stream(s) processes per second, at an estimated <u>maximum</u>?</p> <ul style="list-style-type: none"><li><input type="radio"/> 0 – 10</li><li><input type="radio"/> 10 – 100</li><li><input type="radio"/> 100 – 1K</li><li><input type="radio"/> 1k – 10K</li><li><input type="radio"/> 10K – 100K</li><li><input type="radio"/> 100K – 1M</li><li><input type="radio"/> 10M – 100M</li><li><input type="radio"/> Other: ...</li></ul>
<p>What is the format of the data processed on your stream(s)?</p> <ul style="list-style-type: none"><li><input type="checkbox"/> Plain Text</li><li><input type="checkbox"/> Array</li><li><input type="checkbox"/> Linked List</li><li><input type="checkbox"/> Java Object</li><li><input type="checkbox"/> JSON</li><li><input type="checkbox"/> XML</li><li><input type="checkbox"/> RDF</li><li><input type="checkbox"/> CSV</li><li><input type="checkbox"/> GraphViz DOT</li><li><input type="checkbox"/> GDF</li><li><input type="checkbox"/> GML</li><li><input type="checkbox"/> GraphML</li><li><input type="checkbox"/> ISO BMFF</li><li><input type="checkbox"/> RIFF</li><li><input type="checkbox"/> Other: ...</li></ul>
<p>Which category best described the data in your stream(s)?</p> <ul style="list-style-type: none"><li><input type="checkbox"/> Time-series</li><li><input type="checkbox"/> Spatial Data</li><li><input type="checkbox"/> Relational Data</li><li><input type="checkbox"/> Free Text</li><li><input type="checkbox"/> Multimedia Data</li><li><input type="checkbox"/> Graph Data</li><li><input type="checkbox"/> Other: ...</li></ul>
<p>Which data type(s) are contained in your streams' data point?</p> <ul style="list-style-type: none"><li><input type="checkbox"/> String</li><li><input type="checkbox"/> Integer</li><li><input type="checkbox"/> Float</li><li><input type="checkbox"/> Characters</li><li><input type="checkbox"/> Boolean/Binary Value</li><li><input type="checkbox"/> Hashed Value</li><li><input type="checkbox"/> Nested JSON</li><li><input type="checkbox"/> Other: ...</li></ul>

## Section 3: Streaming Task &amp; Algorithm

<p>What kind of streaming task(s) do you perform?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Join Operation</li> <li><input type="checkbox"/> Similarity Search</li> <li><input type="checkbox"/> Aggregate Queries</li> <li><input type="checkbox"/> Filtering</li> <li><input type="checkbox"/> Data Transformation</li> <li><input type="checkbox"/> Graph Processing</li> <li><input type="checkbox"/> Other: ...</li> </ul>
<p>If you answer "Other" on the above question, please explain the task (i.e., <i>Randomized streams merging where one merges stream in random ways using self-built algorithm</i>)</p> <p>...</p>
<p>Based on your answer for question 16, what kind of join operation do you perform?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Similarity Join</li> <li><input type="checkbox"/> Equality Join</li> <li><input type="checkbox"/> Theta Join</li> <li><input type="checkbox"/> Other: ...</li> <li><input type="checkbox"/> I don't know</li> <li><input type="checkbox"/> I don't do join operation</li> </ul>
<p>Based on your answer for question 16, what kind of aggregate queries do you perform?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Count</li> <li><input type="checkbox"/> Sum</li> <li><input type="checkbox"/> Average</li> <li><input type="checkbox"/> Min-Max</li> <li><input type="checkbox"/> Percentile Calculation</li> <li><input type="checkbox"/> Other: ...</li> <li><input type="checkbox"/> I don't know</li> <li><input type="checkbox"/> I don't do aggregate queries</li> </ul>
<p>If you have time-bounded computation, which type of window you use?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Tumbling Window</li> <li><input type="checkbox"/> Sliding Window</li> <li><input type="checkbox"/> Hopping Window</li> <li><input type="checkbox"/> Session Window</li> <li><input type="checkbox"/> Other: ...</li> <li><input type="checkbox"/> I don't know</li> </ul>
<p>Following up the question above, what is the window interval? <i>If there's more than one, write it in this specific format of 1 second, 2 minutes, 5 hours. If the number of streams is too much, please put the estimated average of window interval of all streams</i></p> <p>...</p>
<p>What are the sources of your stream(s)?</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Database</li> <li><input type="checkbox"/> File System</li> <li><input type="checkbox"/> System Log</li> <li><input type="checkbox"/> User-Triggered Event</li> <li><input type="checkbox"/> Applications</li> <li><input type="checkbox"/> Micro-services</li> <li><input type="checkbox"/> Sensors</li> <li><input type="checkbox"/> Other: ...</li> </ul>
<p>What do you do with your stream output?</p>

...
Do you perform any machine learning computation in the stream or at the end of the stream?
<input type="radio"/> Yes <input type="radio"/> No
If your answer yes for the question above, please explain the machine learning computation that you do and its involvement with the stream.
...

#### Section 4: Streaming Software & Tools

What stream processing engine do you use?
<input type="checkbox"/> Apache Kafka <input type="checkbox"/> Apache Flink <input type="checkbox"/> Apache Storm <input type="checkbox"/> Spark Streaming <input type="checkbox"/> Faust <input type="checkbox"/> Amazon Kinesis <input type="checkbox"/> IBM Streaming <input type="checkbox"/> In-house stream processor <input type="checkbox"/> Other: ...
What other type of tools/services do you use alongside the stream processing engine?
<input type="checkbox"/> Job Manager <input type="checkbox"/> Cloud Service <input type="checkbox"/> Visualization <input type="checkbox"/> Graph Processor <input type="checkbox"/> Message Broker <input type="checkbox"/> Other: ...
For your answer above (if you use other tools), please specify the name of the tools that you use.
...

#### Section 5: Use Case

For what use case(s) are you using your stream?
<input type="checkbox"/> Fraud Detection <input type="checkbox"/> Visualization <input type="checkbox"/> Real-Time Monitoring <input type="checkbox"/> Event Prediction <input type="checkbox"/> Forecasting <input type="checkbox"/> Credit Scoring <input type="checkbox"/> KYC (Know-Your-Customer) <input type="checkbox"/> Macro-economic Insights <input type="checkbox"/> Financial Reporting <input type="checkbox"/> Other: ...
Describe the workflow of your use case of stream
<i>(i.e., Our team main work is API monitoring dashboard. We track our API hit in Redis and store the API parameter input in MongoDB database. We use Kafka as our stream processing engine and set up Kafka configuration to detect any changes in the database. Redis data is streamed real-time to Kafka data pipeline using Kafka Connect.</i>

<i>In the pipeline, we set up a consumer that stored the data in PostgreSQL that is being used for the monitoring dashboard.)</i>
...
If you're willing to share, how frequent is the event occurrence of your use case? <i>(i.e., Fraudulence problem happens 3 times a day, our monitoring dashboard is up 20 hours per day, we generate audit report once a week)</i>
...
Why are you using data streaming technologies for your use case?
<input type="checkbox"/> Impossibility to solve the problem without using data streaming technologies <input type="checkbox"/> Improvement in processing speed <input type="checkbox"/> Improvement in accuracy of data processing <input type="checkbox"/> Faster decision making <input type="checkbox"/> Optimization of processes and resources <input type="checkbox"/> Need of real-time data processing <input type="checkbox"/> Other: ...

### Section 6: Challenges & Expected Features

What are the challenges, problems, or constraints that you faced while you're working with data streams?
...
How do you try to solve these challenges currently?
...
How would you picture the ideal solution (wish list) for these challenges?
...
What feature that you think is missing from the data streaming software or tools that you're using (related or non-related to the challenges above)? Please specify the software and tools for each expected feature mentioned.
...

# Bibliography

- Baccarelli, Enzo et al. (2016). “Energy-efficient dynamic traffic offloading and re-configuration of networked data centers for big data stream mobile computing: review, challenges, and a case study”. In: IEEE Network 30.2, pp. 54–61. DOI: [10.1109/MNET.2016.7437025](https://doi.org/10.1109/MNET.2016.7437025).
- Botan, Irina et al. (Sept. 2010). “SECRET: A Model for Analysis of the Execution Semantics of Stream Processing Systems”. In: Proc. VLDB Endow. 3.1–2, 232–243. ISSN: 2150-8097. DOI: [10.14778/1920841.1920874](https://doi.org/10.14778/1920841.1920874). URL: <https://doi-org.tudelft.idm.oclc.org/10.14778/1920841.1920874>.
- Chang, P. Y., P. Damodaran, and S. Melouk (2004). “Minimizing makespan on parallel batch processing machines”. In: International Journal of Production Research 42.19, pp. 4211–4220. DOI: [10.1080/00207540410001711863](https://doi.org/10.1080/00207540410001711863). eprint: <https://doi.org/10.1080/00207540410001711863>. URL: <https://doi.org/10.1080/00207540410001711863>.
- Chen, Lisi et al. (2019). “Cluster-Based Subscription Matching for Geo-Textual Data Streams”. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 890–901. DOI: [10.1109/ICDE.2019.00084](https://doi.org/10.1109/ICDE.2019.00084).
- Cloudera (2020). White paper: Choose The Right Stream Processing Engine For Your Data Needs. Tech. rep. 395 Page Mill Road Palo Alto CA 94306 USA: Cloudera Inc., p. 13. URL: <https://www.cloudera.com/campaign/choose-the-right-stream-processing-engine-for-your-data-needs.html>.
- Cugola, Gianpaolo and Alessandro Margara (June 2012). “Processing Flows of Information: From Data Stream to Complex Event Processing”. In: ACM Comput. Surv. 44.3. ISSN: 0360-0300. DOI: [10.1145/2187671.2187677](https://doi.org/10.1145/2187671.2187677). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/2187671.2187677>.
- Demers, Alan et al. (Jan. 2007). “Cayuga: A General Purpose Event Monitoring System.” In: pp. 412–422.
- Dias de Assunção, Marcos, Alexandre da Silva Veith, and Rajkumar Buyya (2018). “Distributed data stream processing and edge computing: A survey on resource elasticity and future directions”. In: Journal of Network and Computer Applications 103, pp. 1–17. ISSN: 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2017.12.001>. URL: <https://www.sciencedirect.com/science/article/pii/S1084804517303971>.
- Fragkoulis, Marios et al. (2020). “A Survey on the Evolution of Stream Processing Systems”. In: CoRR abs/2008.00842. arXiv: 2008.00842. URL: <https://arxiv.org/abs/2008.00842>.
- Gangineni, Sambasiva Rao et al. (2019). “Real-Time Object Recognition from Streaming LiDAR Point Cloud Data”. In: Proceedings of the 13th ACM International Conference on Distributed DEBS '19. Darmstadt, Germany: Association for Computing Machinery, 214–219. ISBN: 9781450367943. DOI: [10.1145/3328905.3330297](https://doi.org/10.1145/3328905.3330297). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3328905.3330297>.
- Golab, L. and M. Tamer Özsu (Jan. 2003). “Data stream management issues D a survey”. In: Sigmod Record.

- Golab, Lukasz and M. Tamer Özsu (June 2003). "Issues in Data Stream Management". In: *SIGMOD Rec.* 32.2, 5–14. ISSN: 0163-5808. DOI: [10.1145/776985.776986](https://doi-org.tudelft.idm.oclc.org/10.1145/776985.776986). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/776985.776986>.
- Gomes, Eliza HA et al. (2019). "A survey on data stream, big data and real-time". In: *International Journal of Networking and Virtual Organisations* 20.2, pp. 143–167.
- Gong, Shufeng, Yanfeng Zhang, and Ge Yu (2017). "Clustering Stream Data by Exploring the Evolution of Density Mountain". In: *Proc. VLDB Endow.* 11.4, 393–405. ISSN: 2150-8097. DOI: [10.1145/3186728.3164136](https://doi-org.tudelft.idm.oclc.org/10.1145/3186728.3164136). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3186728.3164136>.
- Gorasiya, Darshankumar (Sept. 2019). *Comparison of Open-Source Data Stream Processing Engines: S*. DOI: [10.13140/RG.2.2.16747.49440](https://doi-org.tudelft.idm.oclc.org/10.13140/RG.2.2.16747.49440).
- Hasan, Mahmud, Mehmet A Orgun, and Rolf Schwitter (Aug. 2018). "A Survey on Real-Time Event Detection from the Twitter Data Stream". In: *J. Inf. Sci.* 44.4, 443–463. ISSN: 0165-5515. DOI: [10.1177/0165551517698564](https://doi-org.tudelft.idm.oclc.org/10.1177/0165551517698564). URL: <https://doi-org.tudelft.idm.oclc.org/10.1177/0165551517698564>.
- Hesse, Guenter and Martin Lorenz (2015). "Conceptual Survey on Data Stream Processing Systems". In: *2015 IEEE 21st International Conference on Parallel and Distributed Systems* pp. 797–802. DOI: [10.1109/ICPADS.2015.106](https://doi-org.tudelft.idm.oclc.org/10.1109/ICPADS.2015.106).
- Ikonomovska, Elena, Suzana Loskovska, and Dejan Gjorgjevikj (Sept. 2007). "A survey of stream data mining". In:
- Isah, Haruna et al. (2019). "A Survey of Distributed Data Stream Processing Frameworks". In: *IEEE Access* 7, pp. 154300–154316. DOI: [10.1109/ACCESS.2019.2946884](https://doi-org.tudelft.idm.oclc.org/10.1109/ACCESS.2019.2946884).
- Karimov, Jeyhun et al. (2018). "Benchmarking Distributed Stream Data Processing Systems". In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1507–1518. DOI: [10.1109/ICDE.2018.00169](https://doi-org.tudelft.idm.oclc.org/10.1109/ICDE.2018.00169).
- Kitchenham, B. and S Charters (Jan. 2007). *Guidelines for performing Systematic Literature Reviews in*
- Klar, Rainer (1992). "Event-Driven Monitoring of Parallel Systems". In: *in Workshop on Performance M* Elsevier.
- Kolajo, Taiwo, Olawande Daramola, and Ayodele Adebisi (2019). "Big data stream analysis: a systematic literature review". In: *Journal of Big Data* 6.1, p. 47. ISSN: 2196-1115. DOI: [10.1186/s40537-019-0210-7](https://doi-org.tudelft.idm.oclc.org/10.1186/s40537-019-0210-7). URL: <https://doi-org.tudelft.idm.oclc.org/10.1186/s40537-019-0210-7>.
- Kontopoulos, Ioannis et al. (2020a). "Classification of Vessel Activity in Streaming Data". In: *Proceedings of the 14th ACM International Conference on Distributed and Event-Based DEBS '20*. Montreal, Quebec, Canada: Association for Computing Machinery, 153–164. ISBN: 9781450380287. DOI: [10.1145/3401025.3401763](https://doi-org.tudelft.idm.oclc.org/10.1145/3401025.3401763). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3401025.3401763>.
- (2020b). "Classification of Vessel Activity in Streaming Data". In: *Proceedings of the 14th ACM International Conference on Distributed and Event-Based DEBS '20*. Montreal, Quebec, Canada: Association for Computing Machinery, 153–164. ISBN: 9781450380287. DOI: [10.1145/3401025.3401763](https://doi-org.tudelft.idm.oclc.org/10.1145/3401025.3401763). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3401025.3401763>.
- Liu, Xunyun and Rajkumar Buyya (May 2020). "Resource Management and Scheduling in Distributed Stream Processing Systems: A Taxonomy, Review, and Future Directions". In: *ACM Comput. Surv.* 53.3. ISSN: 0360-0300. DOI: [10.1145/3355399](https://doi-org.tudelft.idm.oclc.org/10.1145/3355399). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3355399>.
- Luckham, David (2008). "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems". In: *Rule Representation, Interchange and Reasoning*

- Ed. by Nick Bassiliades, Guido Governatori, and Adrian Paschke. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–3. ISBN: 978-3-540-88808-6.
- Mansouri-Samani, Masoud and Morris Sloman (June 1997). “GEM: A generalized event monitoring language for distributed systems”. In: *Distributed Systems Engineering* 4, pp. 96–108. DOI: [10.1088/0967-1846/4/2/004](https://doi.org/10.1088/0967-1846/4/2/004).
- Martin, Niels et al. (Dec. 2015). “Batch processing: definition and event log identification”. In:
- Mehmood, Erum and Tayyaba Anees (2020). “Challenges and Solutions for Processing Real-Time Big Data Stream: A Systematic Literature Review”. In: *IEEE Access* 8, pp. 119123–119143. DOI: [10.1109/ACCESS.2020.3005268](https://doi.org/10.1109/ACCESS.2020.3005268).
- Moser, Oliver, Florian Rosenberg, and Schahram Dustdar (2010). “Event Driven Monitoring for Service Composition Infrastructures”. In: *Web Information Systems Engineering – WISE 2010*. Ed. by Lei Chen, Peter Triantafillou, and Torsten Suel. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 38–51. ISBN: 978-3-642-17616-6.
- Nasiri, Hamid, Saeed Nasehi, and Maziar Goudarzi (2018). “A Survey of Distributed Stream Processing Systems for Smart City Data Analytics”. In: *Proceedings of the International Conference on SCIOT '18*. Mashhad, Iran: Association for Computing Machinery. ISBN: 9781450365321. DOI: [10.1145/3269961.3282845](https://doi.org/10.1145/3269961.3282845). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3269961.3282845>.
- Nguyen, Hai-Long, Yew-Kwong Woon, and Wee-Keong Ng (2015). “A survey on data stream clustering and classification”. In: *Knowledge and Information Systems* 45.3, pp. 535–569. ISSN: 0219-3116. DOI: [10.1007/s10115-014-0808-1](https://doi.org/10.1007/s10115-014-0808-1). URL: <https://doi.org/10.1007/s10115-014-0808-1>.
- Petersen, Kai, Sairam Vakkalanka, and Ludwik Kuzniarz (2015). “Guidelines for conducting systematic mapping studies in software engineering: An update”. In: *Information and Software Technology* 64, pp. 1–18. ISSN: 0950-5849. DOI: <https://doi.org/10.1016/j.infsof.2015.03.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0950584915000646>.
- Petersen, Kai et al. (2008). “Systematic Mapping Studies in Software Engineering”. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering EASE'08*. Italy: BCS Learning & Development Ltd., 68–77.
- Punter, T. et al. (2003). “Conducting on-line surveys in software engineering”. In: *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings*. Pp. 80–88. DOI: [10.1109/ISESE.2003.1237967](https://doi.org/10.1109/ISESE.2003.1237967).
- Qin, Cui, Holger Eichelberger, and Klaus Schmid (2019). “Enactment of adaptation in data stream processing with latency implications—A systematic literature review”. In: *Information and Software Technology* 111, pp. 1–21. ISSN: 0950-5849. DOI: <https://doi.org/10.1016/j.infsof.2019.03.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0950584919300539>.
- Querzoni, Leonardo and Nicolo Rivetti (2017). “Data Streaming and Its Application to Stream Processing: Tutorial”. In: *Proceedings of the 11th ACM International Conference on Distributed Computing DEBS '17*. Barcelona, Spain: Association for Computing Machinery, 15–18. ISBN: 9781450350655. DOI: [10.1145/3093742.3095108](https://doi.org/10.1145/3093742.3095108). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3093742.3095108>.
- Röger, Henriette and Ruben Mayer (Apr. 2019). “A Comprehensive Survey on Parallelization and Elasticity in Stream Processing”. In: *ACM Comput. Surv.* 52.2. ISSN: 0360-0300. DOI: [10.1145/3303849](https://doi.org/10.1145/3303849). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3303849>.
- Sahu, Siddhartha et al. (Dec. 2017). “The Ubiquity of Large Graphs and Surprising Challenges of Graph Processing”. In: *Proc. VLDB Endow.* 11.4, 420–431. ISSN:

- 2150-8097. DOI: [10.1145/3186728.3164139](https://doi-org.tudelft.idm.oclc.org/10.1145/3186728.3164139). URL: <https://doi-org.tudelft.idm.oclc.org/10.1145/3186728.3164139>.
- Seaman, C.B. (1999). "Qualitative methods in empirical studies of software engineering". In: *IEEE Transactions on Software Engineering* 25.4, pp. 557–572. DOI: [10.1109/32.799955](https://doi.org/10.1109/32.799955).
- Shahrivari, Saeed (2014). "Beyond Batch Processing: Towards Real-Time and Streaming Big Data". In: *Computers* 3.4, pp. 117–129. ISSN: 2073-431X. DOI: [10.3390/computers3040117](https://doi.org/10.3390/computers3040117). URL: <https://www.mdpi.com/2073-431X/3/4/117>.
- Silva, Jonathan A. et al. (July 2013). "Data Stream Clustering: A Survey". In: *ACM Comput. Surv.* 46.1. ISSN: 0360-0300. DOI: [10.1145/2522968.2522981](https://doi.org/10.1145/2522968.2522981). URL: <https://doi.org/10.1145/2522968.2522981>.
- Stankovic, John A. and R. Rajkumar (2004). "Real-Time Operating Systems". In: *Real-Time Systems* 28.2, pp. 237–253. ISSN: 1573-1383. DOI: [10.1023/B:TIME.0000045319.20260.73](https://doi.org/10.1023/B:TIME.0000045319.20260.73). URL: <https://doi.org/10.1023/B:TIME.0000045319.20260.73>.
- Stankovic, John A. and Krithi Ramamritham (1995). "A Reflective Architecture for Real-Time Operating Systems". In: *Advances in Real-Time Systems*. USA: Prentice-Hall, Inc., 23–38. ISBN: 0130833487.
- Tam, Nguyen Thanh et al. (May 2019). "From Anomaly Detection to Rumour Detection Using Data Streams of Social Platforms". In: *Proc. VLDB Endow.* 12.9, 1016–1029. ISSN: 2150-8097. DOI: [10.14778/3329772.3329778](https://doi.org/10.14778/3329772.3329778). URL: <https://doi.org/10.14778/3329772.3329778>.
- Tidke, Bharat and Rupa Mehta (2018). "A Comprehensive Review and Open Challenges of Stream Big Data". In: *Soft Computing: Theories and Applications*. Ed. by Millie Pant et al. Singapore: Springer Singapore, pp. 89–99. ISBN: 978-981-10-5699-4.
- Turaga, Deepak et al. (2010). "Design principles for developing stream processing applications". In: *Software: Practice and Experience* 40.12, pp. 1073–1104. DOI: <https://doi.org/10.1002/spe.993>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.993>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.993>.
- Wang, Zhuoyi et al. (2019). "Robust High Dimensional Stream Classification with Novel Class Detection". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1418–1429. DOI: [10.1109/ICDE.2019.00128](https://doi.org/10.1109/ICDE.2019.00128).
- Zhao, Junzhou et al. (2020). "Continuously Tracking Core Items in Data Streams with Probabilistic Decays". In: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 769–780. DOI: [10.1109/ICDE48307.2020.00072](https://doi.org/10.1109/ICDE48307.2020.00072).
- Zhao, Xinwei et al. (2017). "Chapter 11 - A Taxonomy and Survey of Stream Processing Systems". In: *Software Architecture for Big Data and the Cloud*. Ed. by Ivan Mistrik et al. Boston: Morgan Kaufmann, pp. 183–206. ISBN: 978-0-12-805467-3. DOI: <https://doi.org/10.1016/B978-0-12-805467-3.00011-9>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128054673000119>.