

# **DYNAMICALLY RECONFIGURABLE DATA PIPELINES IN THE EDGE NETWORK**

Isuru Nuwanthilaka

(219376N)

Thesis submitted in partial fulfillment of the requirements for the degree  
of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa  
Sri Lanka

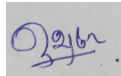
January 2022

## DECLARATION

“I declare that this is my own work, and this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).”

Signature:



Date: 2022/01/07

The supervisor/s should certify the thesis with the following declaration.

The above candidate has carried out research for the Masters thesis under my supervision.

Name of the supervisor: Dr. Gayashan Amarasinghe

Signature of the supervisor:

Date :

## **ACKNOWLEDGMENTS**

First and foremost, I am extremely grateful to my supervisor Dr. Gayashan Amarasinghe for his invaluable advice, continuous support, and patience during my MSc in CS research. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would like to thank all the members in the Department of Computer Science, University of Moratuwa. It is their kind help and support that have made my study and life a wonderful time. Also, I should extend my gratitude for the company I am working for, for providing me the support and being flexible with the work allowing me to finish this research on time.

Finally, I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement in the past year, it would be impossible for me to complete my study.

## ABSTRACT

Pipelines are a highly discussed topic in today's technological world. There are different variations of pipelines; data science pipelines, DevOps pipelines and DevSecOps pipelines etc. A data science pipeline usually comes with a fixed architecture, which can be problematic as it introduces limitations to the users as well as developers. For an example they must clean, pre-process data beforehand and sometimes they have to chunk data before sending to the pipeline as pipelines are not capable of adjusting to the requirements. As the state of the art is to process these machine learning pipelines in the cloud, the users need to migrate the workloads to the cloud as well. However, due to the advancements in processor, networking, battery, and manufacturing technologies, edge computing is becoming a viable option in many aspects such as utilizing local resources, improved privacy, and lower latency. So, these edge resources also need to be considered when executing the pipelines.

So, in this research we first discuss the existing literature in the domain and then we propose a methodology for dynamically reconfigurable data pipeline architecture in the edge network. With the proposed approach we expect to achieve higher efficiency, controllability and scalability of the data across networks. Further we propose a prototype with Raspberry Pi and Android based programs to discuss the effectiveness of our proposed method.

Key words: pipelines, architecture, reconfiguration, edge

# TABLE OF CONTENTS

Declaration	i
Acknowledgements	ii
Abstract	iii
Table of content	iv
List of Figures	v
List of abbreviations	vi
1. Introduction	1
1.1 Problem	1
1.2 Research Objectives	2
2. Literature Review	3
2.1 Data Pipeline Challenges	3
2.1.1 Manufacturing process data analysis pipelines: a requirements analysis and survey	3
2.1.2 Examining the Challenges in Development Data Pipeline	4
2.1.3 Data Pipeline Selection and Optimization	6
2.1.4 Data Life cycle Challenges in Production Machine Learning: A Survey	6
2.2 Data Pipeline Architectures	7
2.2.1 Putting Data Science Pipelines on the Edge	7
2.2.2 Modelling Data Pipelines	8
2.2.3 Scalable data pipeline architecture to support the industrial internet of things	9
2.2.4 Feedback Driven Improvement of Data Preparation Pipelines	9
2.2.5 An Edge-Based Framework for Enabling Data Driven Pipelines for IoT Systems	10
2.2.6 On the Design and Architecture of Deployment Pipelines in Cloud- and Service-Based Computing	12
2.2.7 Edge Based Data-Driven Pipelines -Technical Report	12
2.2.8 Review of social media analytical process and Big Data Pipeline	12
2.2.9 Data Pipeline Architecture for Serverless Platform	13
2.2.10 Pipeline architecture for mobile data analysis	14
2.3 Data Pipeline Security	15
2.3.1 Integration of Security Standards in DevOps Pipelines	15
2.3.2 Security Supporting Continuous Deployment Pipeline	16
3. Methodology	17
4. Conclusion	20
5. References	21

## **LIST OF FIGURES**

- Figure 3.1      Proposed Reconfigurable Architecture
- Figure 3.2      Simplified Switching illustration of the architecture

## LIST OF ABBREVIATIONS

Abbreviation	Description
CD	Continuous Delivery
ETL	Extract Transform Load
IOT	Internet of Things
IIOT	Industrial Internet of Things
JITA-4DS	Just in Time architecture for data science
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
SDK	Software Development Kit
SMBO	Sequential Model Based Optimization

# 1. INTRODUCTION

Data pipelines are playing a significant role in the emerging data science landscape as modern organizations ingest streams and batches of data with high variety, velocity and volume, from different sources. These pipelines mainly focus on extraction, data preparation, cleaning, transforming, training models and visualization. There are many data pipeline requirements and challenges in the field such as data cleansing on large scale data, security implementation to pipelines, visualization on large data sets etc. Apart from that, data pipelines need to comply with the security standards. As a result, the architectural support to implement these requirements is a major requirement.

In the literature review section, we mainly focus on three topics; first we discuss about data pipeline requirements and challenges as it relates to how we design pipelines and what should be in the architecture, secondly, we review on architectures which describe different existing approaches in the domain, and some are academic proposals and thirdly security is highlighted as it cannot be avoided regardless of the context.

In the next section we describe the methodology we are going to use to overcome these issues carefully considering the details extracted from literature review.

## 1.1 Problem

Nowadays, edge devices are resource rich compared to a few years back due to advancements in processor, networking, battery and manufacturing technologies. Also, there are tools/libs popping up which enable effective use of machine learning (ML) (ex: TensorFlow mobile SDK) in the edge devices. Now the developers and the community tend to execute ML tasks on the edge devices which opens us to another research area of ML pipelines in edge networks. So, this research is carried out to explore the possibilities of devising reconfigurable architectures for ML pipelines at edge.



Most of the data pipelines are one-for-all models. So, its efficiency might be reduced for some pipeline tasks and less flexibility. We discuss how to dynamically reconfigure the architecture of data pipelines using the resources available in the edge devices.

## **1.2 Research Objectives**

The following three main objectives are served.

1. Dynamic reconfiguration of the pipeline architecture.

In this objective we will come up with the reconfigurable architecture which will have capabilities to perform a machine learning pipeline. This will be a conceptual design and we will implement that for different edge device – Raspberry Pi and Android Mobile device.

2. Optimal/near-optimal utilisation of edge resources to support proposed architecture.

To support the functionality of proposed dynamically configurable architecture we will come up with few strategies to analyse the resources available in the edge device and propose a method to select functionalities which can be operated based on the available resources. Outcome will be few sets of logical selection criteria of functionalities which fit to that respective edge device.

3. Evaluation of algorithms that have benefited from this approach.

In this section we will do a performance review to identify which category of algorithms will be benefitted from our reconfigurable architecture. All the algorithms will not be run on edge devices so this study will be helpful select appropriate algorithm for the pipeline. Outcome will be a performance evaluation.

## 4. LITERATURE REVIEW

The literature is presented under requirements, architecture and security based on the context of each research.

### 2.1. Data Pipeline Challenges

#### 2.1.1 Manufacturing process data analysis pipelines: a requirements analysis and survey [1]:

In this paper researchers have surveyed about the smart manufacturing pipelines. Smart manufacturing is how the digitization of all the manufacturing activities in the physical process occurs. There are several key activities like data ingestion, communication and visualization but researchers have focused more on storage and analysis phases. Mainly, two questions are being surveyed in this paper.

1. Key requirements for a big data analysis pipeline.
2. Available tools and solutions for pipelines in academic literature. To answer the first question, they have identified key phases and their requirements.

a) Ingestion: This is the main component for entering data to the big data analysis platforms. This opens several other operations like identification, validation, filtering etc tasks. Integrations of too few data sources play a significant role in this domain. It needs to implement several connectors to various platforms/systems and needs to support data formats. As data generated at different locations it may be in different quality, formats and rates. Some systems may generate real-time data and big data platforms need those data in real time or near real time. Some applications may need those data batch-wise. So, all these issues arise in the data ingestion phase.

b) Communication: Communication channels are a much-needed component in the data pipelines as it is the medium where data is routed among the different tools in the big data platforms. In some cases, ingestion and communication bind as one tool in the system. The communication should be reliable and secure. In various environments, the communication architectures may differ. For example, some IoT related systems need publisher-subscriber patterns, some may need pull or push

patterns. Also, these communication channels should be able to handle the utilization of bandwidth efficiently.

c) Storage: In the 1990s, the product processes stored only a few megabytes, but now the situation is much more changed and 10s of gigabytes are being generated daily in the manufacturing pipelines. So, handling this amount of data should be a well thought design. Also, some data should be stored for years as they may be required for legal or privacy policies. So, the real problem is the security and cost of these long-term data storage in the modern-day cloud platforms. The compression algorithms and different databases to operate on these data types and their relations. Apart from that, these databases or storage facilities need to provide high speed read and writes as most of the applications directly depend on data.

d) Analysis: The paper describes the analysis requirements mentioned in Qin [2]. Support for heterogeneous data types, scaling to 10000s of variables, analysis of imperfect data generated by various heterogeneous sources, support for time-series analysis, data mining and machine learning are mentioned as key requirements.

e) Visualization: Big data visualization differs from traditional data representations due to volume, variety and velocity. Even though the data sets are large, to get accepted by the engineers those representations should be interactive, dynamic and well performing. So, handling concurrency, real time visualizations are becoming new challenges. Based on the requirements mentioned above and some other requirements detailed in the paper they have come up with a literature review for the latest tools and solutions in the industry. First, they have defined a criterion of what they are going to look at. In the paper they have mentioned a set of tools according to the topics. In addition, 6 pipelines have no tool setup for ingestion, 19 have no tools for communication and 13 for visualization.

### **2.1.2 Examining the Challenges in Development Data Pipeline [3]:**

Today's developing world is relying more on data driven decisions. Therefore, there are many significant and emerging requirements in the data analysis process. In

this research, authors have explored the various challenges in several phases in data pipeline development. They have conducted several interviews about the development and from the feedback they got, they have identified the common challenges like correcting text fields, extracting textual data from different documents like PDF, merging hierarchical data etc. Authors have identified three main phases ; data collection, data cleansing and data analytics.

a) Data collection: In the data collection phase they have segmented it into two subtopics - data entry and tools. There can be many challenges associated with the data entry. Mainway of data entry is human data entry. So human errors are unavoidable. Due to crowd sourcing, there are errors in the datasets - labelling issues, textual errors etc. In some cases, data must be extracted from PDF files, so there can be extraction errors. Those extractions can be erroneous due to bad writings, tool imperfections etc. Sometimes, those data entry people are less skilled and lacking the training, so these human errors are expected. Another challenge is tool support. When it comes to big data tools should be efficient due to volume, variety and velocity. Most of the time these tools must transform paper form into digitization. Also, in tools, they have described latencies of dashboards and their inability to cater large datasets in visualization.

b) Data cleansing: This is another very important phase. This section includes two subtopics: corrections and structuring. Here authors have mentioned the challenges fixing the issue introduced especially in the data entry and preprocessing for analytics. One challenge is replacing values or masking, so the privacy of the data is preserved. But this should be done consciously as it may make the data useless. Apart from that, removing duplicates, unit conversions, fixing spelling mistakes, and cleaning up open texts are some key challenges. In the structuring authors have mentioned merging data from different sources, restructuring data formats, code conversion/terminology mapping, splitting the aggregations as challenges.

c) Data analytics: In this section, two important subtopics are mentioned; calculations and visualizations. In the calculations the main challenges are to find the missing data, identify the data outliers, calculate accuracy for test data and fine tune values to remove the biases in data sets and training. Authors have mentions visualizations challenges as eyeball data for data accuracy, calculate daily/weekly/real-

time reports generation. As for future work, this paper suggests interviewing more diversified resource persons and tech-savvy organizations.

### **2.1.3 Data Pipeline Selection and Optimization [4]:**

In this paper, the author discusses the data pipeline hyper-parameter optimization problem as a standard optimization problem. Data pre-processing acts as a significant component in the data pipeline. They have focused on selecting data pipelines and optimizing it.

They have discussed about the impact of the data pipeline configurations on classification problems. To identify the impact, they need to define data pipeline selection and optimization problems. So, they have come up with methods to define these topics. Then they analyse the impact of Sequential Model Based Optimization (SMBO) for the above-mentioned problem. They have redefined SMBO and discuss about optimal configurations. In the future work they have identified that this could be evolved on an online version using streaming concepts.

### **2.1.4 Data Life cycle Challenges in Production Machine Learning: A Survey [5]:**

The authors have presented research challenges in the domain of production machine learning taken from their experience working at Google. The focus is on three main areas - data understanding, data validation and cleaning, and data preparation. They have talked about different dimensions of production ML.

User interacting with ML platforms highlight different roles of people involved in data ML pipelines. In the production there are different roles involved in the software tools and they need different set of access

Data lifecycle in an ML pipeline describes different phases of the pipeline like prepare, train and evaluate, validating, clean, serve. They have different objectives in each phase.

Data understanding is also very important to cope with the data correctly. Just by looking at and reading data cannot help to understand the data. User must work on the data set. There are few techniques we can follow.

- a. Sanity check - data in expected shape
- b. Analyses for launch and iterate - feature based, data lifecycle

Next important step is data validation and cleaning. First, we can alert trade-offs in the dataset and then we can identify the categories that results fall into.

Following above steps, these are some other very important steps; data preparation, feature engineering and selection effective most significant data fields which has a significant impact of the results, data enrichment for generating wholesome idea.

In the last section they have identified what are the lessons learned from their study. Firstly, they identified that there are many data management challenges beyond dataflow optimizing. Then discuss about making realistic assumptions when developing machine learning products. This could not be done without recognizing the true user requirements. So, they have mentioned identification of users' diversified needs is a key segment in the process. Finally, they discuss about how to keep smooth integration to the development workflow.

## **2.2 Data Pipeline Architectures**

### **2.2.1 Putting Data Science Pipelines on the Edge [6]:**

In this research paper, authors have come up with a new architecture named as “Just in Time architecture for data science (JITA-4DS)”. Their main concern is that one-for-all architectures are not efficient enough to cater the requirements of different data science pipelines. In addition to that existing architectures are not well designed to provide on-demand infrastructures. Those one-fits-all architectures assume,

1. High reliability and the high availability of the network

## 2. High energy and economically consuming resources

So, considering the agile nature of the pipelines they are trying to provide an algorithm and resource management techniques for reconfiguring disaggregated data centres. Their concern is about the configurations on the cloud infrastructure. This research has a significant impact on the proposed research as it discusses mainly about dynamic reconfiguration in cloud leaving opportunities on the edge network. The work related to this paper is an extended version of two types of approaches.

1. Disaggregated data centres as an alternative for one-fits-all architectures

2. Data science pipelines' execution platforms in the cloud.

Currently they are addressing the challenges in the resource management of simple environments and simulation of different levels of JITA-4DS.

### **2.2.2 Modelling Data Pipelines [7]:**

In this paper, authors mainly propose a new conceptual model of data pipeline. In The first part of the paper, they are discussing the importance of data products- APIs, dashboards etc. Addition to that, data collection in distributed set up, semi structured data sources and ETL/ELT pipelines are identified as important. As a result of the proposed model, they try to improve the efficiency of the pipeline to reduce the latency in the development of data products. This paper has three main parts. In the first part they conduct interviews with professionals and organizations to identify challenges in data collection pipeline for data analytics, building data pipelines for data governance and ML pipeline areas. Then, in the second part they analyse feedback and propose a new conceptual model. That model consists mainly of data sources, data collection, data lakes, data ingestion, data processing, data staging, data warehousing, data labelling, data pre-processing and ML/DL modelling phases. In the last part, authors have discussed the validity of this model taking several cases and have given suggestions to improve.

### **2.2.3 Scalable data pipeline architecture to support the industrial internet of things [8]:**

The Industrial internet of things (IIOT) is a growing field. With the expansion of data sources, data formats, scaling up of data processing and distribution of scalable data pipelines have become challenging. In This research paper, they are proposing an architecture for scalable data pipeline to data processing and distribution data from various sources. When designing the architecture, they have identified a few important facts. One concern is formatting data into standard data types. With this approach they can scale data sources using the same interface to the pipelines. Next, they focus on synchronizing timestamps across multiple sources, so it orders the event accordingly. Also, they try to enable deployment on embedded systems and legacy systems. Apart from these, managing continuity of the data streams, enabling data persistency, route data flows into multiple endpoints are some of other concerns in designing the architecture. In the following section authors have proposed an architecture considering the above-mentioned concerns. This includes middleware which is an architectural pattern that integrate hardware and software to enable two functionalities; moving data from different data sources and distributed in scalable manner. Front part of the architecture consists of different connectors to MTC components and ad-hoc sources, then they talk through MQTT clients to the middleware. The Middleware part consists of MQTT broker and Apache Kafka. Those connect to data applications and different flavours of databases. To evaluate the architecture, they have used Hurco VMX24 device and MTConnect devices. They have not specifically mentioned the result data in the paper.

### **2.2.4 Feedback Driven Improvement of Data Preparation Pipelines [9]:**

Data preparation is laborious work which requires huge effort. This adds upfront cost to the data analysis. Some data preparation tasks are automated, and it is important to have the ability to refine the decision of preparation on the feedback from the users of data products. In this research author explore the approaches to revise the results of diverse data preparation components based on the feedback about the



correctness of the value in the data products. Data preparation takes 80% of data scientist's usual work where they like to spend more time on data analysis part. In preparation phase, according to researchers, it has several steps: discovery, profiling, matching, mapping format transformation and entity resolution. In the next section they have mentioned VADA [10] architectural aspects which is relevant to their proposing feedback driven architecture. Authors have mentioned important aspects like input data sources, target schema, user context and data context. Proposed approach consists of mainly two steps.

1. Given the feedback, identify a collection of hypotheses which explain the feedback.
2. Aspects of a hypothesis in the above collection should be reviewed matching the current preparation pipeline and fine tune.

In the next section, they have come up with mathematical models and pseudo codes of the algorithms to solidify their proposed approach on selecting hypotheses.

#### **2.2.5 An Edge-Based Framework for Enabling Data-Driven Pipelines for IoT Systems [11]:**

The Internet of Things (IoT) generates large amounts of data nowadays as the number of devices gets multiplied day by day. In this research, authors propose a new architecture named R-Pulsar which tries to enable the cloud capabilities in the edge devices. They propose it as an edge-based framework for data pipelines. They have focused on IoT devices because by 2030 there will be 500 Bn Cisco Devices only and nearly half of the data come from the sensor's edge network [18]. They have mentioned that precision medical devices/applications for continuous monitoring of medical instruments, healthcare sensors related to lifestyle and urban mobility applications will be transformative application this era. As the programming abstraction, they have discussed what, when and where data needs to be processed. They have taken smart city and disaster response application as cases. There are two existing models in this field: edge data analytic and cloud-edge programming models. In the edge data analytics, they have mentioned some tools like GeoLytics which is designed for dynamic stream processing, AWS Greengrass, IBM Watson IoT and Azure IoT. For cloud and edge programming models they have taken MapReduce

technique, Apache Hadoop, Apache Spark as tools. Most of the tools are designed to be in the cloud as it is resource rich and scalable. The R-Pulsar is a framework designed to be a software stack that extends cloud capabilities to the edge. This uses a distributed architecture which is implemented in several layers.

a) Location aware overlay network layer: IoT data comes from different locations. So, this temporal and spatial information related to data should be processed properly according to the locations. Nodes in the architecture are considered as Rendezvous Point (RP). Those RPs are grouped according to their geological placements. This layer acts as a lookup to route the messages to relevant RP group.

b) Content-based Routing Layer: This layer builds upon the previous layer. This is responsible for delivery of the messages based on the content. They have used two separate methods, 1. Routing using simple keyword tuples - use IDs 2. Routing using complex keyword tuples - use wildcards, ranges

c) Serverless Messaging Layer: Serverless computing is a cloud computing model which abstracts the management of infrastructure and makes the developer's life easy. In the context, this layer is responsible for posting the AR message with profile routed from the previous layer. This layer has two main components: matching engine and profile manager.

d) Memory-mapped Streaming Analytics Pipeline: This layer is implemented such that it collects data from multiple data sources, processes data and makes them available to use. This consists of few other layers. Data collection layer gathers data from different sources and makes available to the pipeline. Service like Apache Kafka, Google Pub/Sub, Amazon Firehose and Mosquito are mentioned as possible tools for this layer. Next the data processing layer is used to process and perform computation on the collected data. This R-Pulsar is validated using Apache Edgent. Another layer is data storage and a query layer which enables SQL-like querying. R-Pulsar relies on Rock DB which is a key-value database.

e) Rule-based Programming Abstraction: These rules define when data must be sent to cloud. Rule engine consists of IF-THEN which gives developer the ability to conditionally call the processing tasks. The evaluation is done using three environments: RaspberryPi system, android system and cloud system. The system specification is mentioned in the paper. In the result section, authors have mentioned

that they have seen a 20% increase in performance and faster completion of the tasks with bandwidth reduction of 82%. In future work they suggest supporting advanced storage strategies, reduce the cost associated with data movements and improve energy efficiency.

#### **2.2.6 On the Design and Architecture of Deployment Pipelines in Cloud- and Service-Based Computing – A Model-Based Qualitative Study [12]:**

In this research, authors have proposed a formal architecture to DevOps and continuous delivery (CD) pipelines. Even though there is a huge rise regarding this, there is no formal discussion about the design and architecture of DevOps and CD pipelines. So, authors have derived a formal architecture and verified that it increases the precision and the efficiency compared to the informally modelled pipelines. They have collected in-depth study of 25 informal deployment practices from the industrial practitioners. To create the groundwork, they have collected details on three main questions; What are recurring practices for designing deployment pipeline structures? What are the relevant environments deployment pipelines? and What are the architectural elements relevant for building a deployment pipeline infrastructure? In the following sections of the paper, they have modelled mathematically the architecture of the DevOps pipeline.

#### **2.2.7 Edge Based Data-Driven Pipelines (Technical Report) [13]:**

This research report investigates about the edge on-device stream processing platform which enables the serverless computational capabilities across the cloud and edge uniformly. This work has evaluated the previously discussed OR-Pulsar framework [11].

#### **2.2.8 Review of social media analytical process and Big Data Pipeline [14]:**

Social media data analytics is another rising research area. With the current expansion of technologies, previous mere consumers are turned into social data

producers leading to big data use cases. Now it is very important to extract social insights to different fields like business, marketing, advertising, tourism and many more. As the basis for this research, authors have formed two main questions.

1. What types of challenges researchers have in analysing social media data?
2. How could big data technologies be integrated to solve those challenges?

Next, they talked about Big Data and Big Data Vs, volume, velocity, variety. The survey has done database-oriented manner and the inclusion rule was to paper to be in between 2008 and 2018. Next they have summarised their findings in a field-wise manner. In the following sections, authors have described five distinct steps for social media analytics: acquisition and recording, information extraction and cleaning, data integration aggregation and representation, query processing data modelling and analysis, big Data interpretation. Also, authors have summarized the challenges they observed related to the big data 4 V's. Considering those, they have come up with social media analytic steps. Overall, they have studied the joint interaction between social media analytic pipeline and big data.

#### **2.2.9 Data Pipeline Architecture for Serverless Platform [4]:**

In this paper, authors propose a novel data pipeline architecture to decompose monolithic data pipeline into independently deployable pipeline components so that it can be scaled up/down, monitored and managed properly to fulfil the incoming large data sets. This architecture is proposed for the cloud environment. They have come up with a model using TOSCA language which is recently developed standard for cloud-based applications. To implement the model, they have used Apache NiFi and suggested Amazon data pipeline as an alternative tool. Their abstracted Pipeline Block model consists of three main groups: SourcePB, Midway PB and Destination PB. For every pipeline, PB is created, and it acts as a black box. SourcePB used to get data from local and remote data streams. Midway Pipeline Block is for intermediate data processing and analytics. DestinationPB is used to publish data like SourcePB but in a different direction. TOSCA model has relevant node types to match these components. Authors have identified Viarota as a potential use case for this model

which is a mobile and web-based cloud application. In future work they have mentioned about expanding the TOSCA node types to support data movements.

#### **2.2.10 Pipeline architecture for mobile data analysis [15]:**

This paper presents a cloud architecture which provides support to collect and analyse mobile data. With their pipeline architecture users can extract data like multimedia data, location information and barcodes. Also, this system provides easy UI components so anyone coming to this system needs not to have programming knowledge. Authors have identified that even though the advancements in hardware and software, still people struggle to create apps with less lead time or less resources. They have come up with this new project Maritacas, a possible solution for the above issue. It provides the flexibility to create apps easily. Initially researchers have explored the existing tools like App Inventor, Nokia Data Gathering, OpenData Kit (ODK), DoForms, Mafuta Go, Fulcrum etc but as per the authors none of them provide data analysis support. The Maritaca project consists of two main components: mobile component and server component. Mobile component is an Android application, and it is an engine of interpreter design pattern - it interprets the xml description. Next one is the server component. It has several items. Formeditoris drag and drop supported web UI written in HTML and AJAX. Analytics editor is also a web UI which allows to query the collected data. Cassandra database is used as a scalable database. Hadoop file system is used to unstructured data and multimedia file across distributed file system. Solar Engine enables searching in apps. These components provide several features. In the future work, authors have mentioned that they are going to expand its infrastructure with cloud resources without compromising its performance. Addition to that they hope to implement it on iOS as an extension to current Android Development.

## 2.3 Data Pipeline Security

### 2.3.1 Integration of Security Standards in DevOps Pipelines [16]:

In the past few years, DevOps pipelines have been adopted to the industrial control systems (ICS) in order to automate the product swim-lanes and reduce the lead time to the market. So, it can improve the customer expectations and increase the development time rather than focusing on operations. But in the modern technology world there are lots of security gaps in the pipelines so in this paper, authors introduce methods to implement security standards to the DevOps pipelines without affecting lead time much. In the paper two main contributions added to the field,

1. Explore the systematic approach to DevOps pipelines to comply with IEC 62443-4-1 standard
2. Describe the automation capabilities for 4-1 standards with the available tools in the industry. In the ICS domain it is important to adopt IEC 62443-4-1 standard. With security concerns, a new area called ‘DevSecOps’ emerged. In the IEC 62443-4-1 standard, it has proposed eight practices: security management (SM), specification for security requirement (SR), secure design (SD), secure implementation (SI), security validation and verification testing (SVV), management of security related issues (DM), security update management (SUM), security guidelines (SG). This work presents how each practice applied to DevOps phases. To apply above security practices, authors suggest three main activities,

- a) Describe standard requirements as activities: This mainly focuses on detailed analysis of 4-1 standard and its requirements. After identifying the standard activities, they define the 4-1 standard process models.<sup>5</sup>

- b) Determine 4-1 standard automation capabilities: Even though they are trying to adopt standards they might not fit properly in the automation environments thus needed to evaluate the automation capabilities without affecting the lead time.

- c) Map activities into pipeline stages: In this phase they are going to map the 4-1 standard pipeline specifications to Security Standard Complaint (S2C) DevOps pipeline specifications. Finally, they have evaluated their qualitative study at Siemens AG which is main contributor to 4-1 standard in ICS. They have interviewed

persons responsible for security of pipelines and have tried to answer the questions related to precision and usefulness. In future work, they are supposed to build tools to increase the percentage of 4-1 standard requirements that can be fully automated in the ICS domain.

### **2.3.2 Security Supporting Continuous Deployment Pipeline [17]:**

Continuous delivery pipelines (CDP) lack security tactics. In this paper authors have researched on the security tactics that can be used in the CDP without affecting the existing behaviour. They have done penetration test to the CDP components and have observed that it increases the security of the components like repository, continuous integration server and main server. They have identified several security risks in the repository, main server and CI server. For examples, uncontrolled access categorized as a repository risk, poor authentication mechanisms and uncontrolled access mentioned as main server (AWS) risk and starting with previously infected version or replay identified as CI server (Jenkins) risks.

In the next section they have proposed possible security tactics which come after the thorough analysis of previous security risks.

Limiting the access of the repository branches so only selected groups can commit/merge code to the default build branch. Normally we use this general method for most of the projects for example GitHub projects. Next, we can secure main server access with private key along with SSH. AWS/GitHub supports this. Use role-based access control (RBAC) for the main server is also a vastly used method in software systems. AWS has an IAM ecosystem. Other thing is setting up clean VM environments for build phase in Jenkins CI server. So, this can be separated and destroyed after build phase safely. Use Jenkins role plugin to control access so that the users can have limited access to delete, create, modify pipelines. In the final section they have summarized the quantitative and qualitative results of this proposed system. Using tools like OWASP Zed Attack Proxy (ZAP) they have scanned security vulnerabilities in the system and have done some penetration tests.

## 5. METHODOLOGY

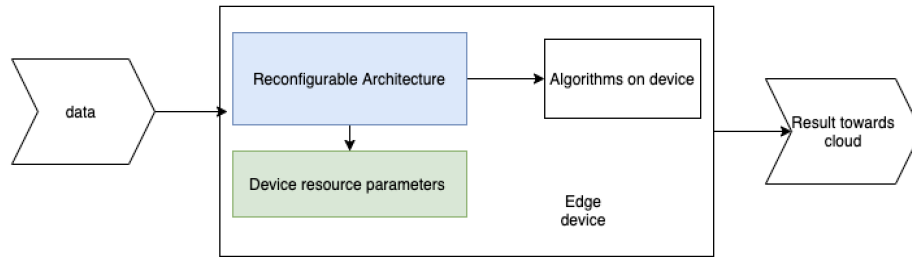
When talking about reconfiguration we must identify the parameters we are going to reconfigure. Therefore, first we carry out the analysis phase on the devices we select. Here we are going to focus on common devices which can be found easily. Edge devices usually have low computation power compared to resource rich devices like high end computers/Laptops or cloud machines. For our research we are going to use Raspberry Pi 3 model A/B and Android 10 Samsung A20s devices.

The Raspberry Pi 3 Model A+ comes with following features extending the Raspberry Pi 3 range into the A+ board format. Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz processor, 512MB LPDDR2 SDRAM memory , 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2/BLE, Extended 40-pin GPIO header, Full-size HDMI, Single USB 2.0 ports, CSI camera port for connecting a Raspberry Pi Camera Module, DSI display port for connecting a Raspberry Pi Touch Display, 4-pole stereo output and composite video port, Micro SD port for loading your operating system and storing data, 5V/2.5A DC power input.

Samsung A20s comes with GSM/LTE, Qualcomm SDM450 Snapdragon 450 (14 nm), Octa-core 1.8 GHz Cortex-A53, Adreno 506, 32GB 2GB RAM Memory and OS Android 9.0 (Pie), upgradable to Android 11, One UI 3.1. These are suitable specifications for commodity edge nodes. Both devices have sufficient memory size and processing power compared to older edge devices.

After analysis of the resources, we will develop an architecture to use those resource parameters and change according to the changes in the environment like data loading, size and formats. The reconfigurable architecture will be optimized to utilize the resources maximally.





*Figure 3.1 Proposed Reconfigurable Architecture*

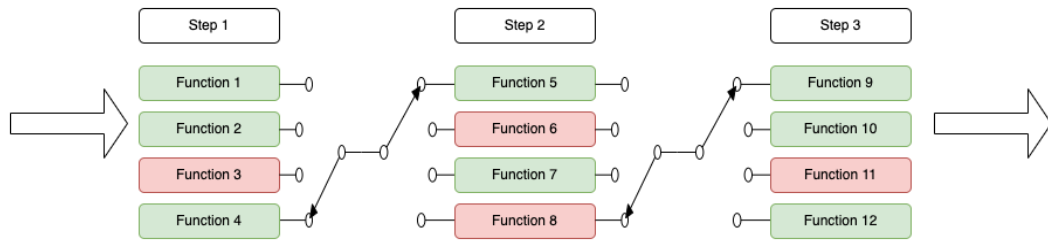
This architecture will be utilising AWS cloud services for the cloud execution components and to manage the platform. (<https://pages.awscloud.com/aws-cloud-credit-for-research.html>).

If needed, we execute small tasks or less-resource needed algorithms on edge devices. So, we can off load some of the cloud utilization to the edge network this may reduce the cost associated with the Cloud platform providers. Architecture components will be subdivided to smaller components like data ingestion, data cleaning, data pre-processing, labelling etc based on the availability of the underlying resources.

To integrate with the different types of data source we will standardize the data ingestions interface with few possible file formats. Initially we will come up with an interface to support CSV data format and eventually we will add support to excel and json formats. Here we will stay more with structured data. After data ingestion data columns will be visible to the user and user will be able to select what they want to proceed with. To clean the data and remove the NaN values, users will be given the opportunity to fine tune the data by adding a few hooks. So, this will act like custom functions which can be added to the execution stack so users can do some manual tasks based on their requirements.

Reconfigurable architecture will look like a mesh of functionalities which can be connected to each other based on the available resource, users will be given the opportunity to build the final system choosing one out of the few different

configurations. Architectural capabilities will be ETL functionalities and some small-scale algorithms which can be easily run with less resources.



*Figure 3.2 Simplified Switching illustration of the architecture*

In the diagram red functionalities are not available due to resource unavailability and green boxes can be selected as the user's preference and can create a sequence of functions.

In the final phase, different algorithms will be analysed against the proposed architecture and how they perform at different resource availability. Finally, we will perform an acceptance test on the security on each subcomponent to check whether they perform as expected or not. Test cases will be built on the various scenarios. For an example, function 1 will be chained with function 5 and function 12 and those will be tested collecting running time/speed. That information will be analysed compared with the traditional method results to evaluate the performance of the proposed methodology.

## 6. CONCLUSION

In this report, we have explored the existing work related to data science pipeline in different perspectives. First thing is the data pipeline challenges and requirements which are presented in existing technologies. So, this is a good starting point for the next tech frameworks. Future technologies should investigate the loopholes in the existing technologies and build expert systems based on them to answer those requirements and challenges. Then we have explored the pipeline design and architectures related literature. This gives a sound understanding about how pipelines work in the current world. Some of those architectures are conceptual designs arise from the academia and some of them are developed by the industry experts. We have looked in both academia and industry related developments in this domain. Next, we have discussed data and delivery pipeline security related work as it is also key concern in the development of pipelines. Pipelines would consume various kind of data – social network data, health data, eCommerce etc therefore security cannot be neglected.

By looking at the literature, few key knowledge gaps are identified. One important area is data pipeline architecture which seems to be having less research done as per our knowledge. We are inspired by the research work of 'Putting Data Science Pipelines on the Edge [6]' and 'An Edge-Based Framework for Enabling Data-Driven Pipeline for IoT Systems [11]'. Following them, we propose research regarding the possibilities of reconfigurable architectures on the edge for data pipelines.

## 7. REFERENCE

1. A. Ismail, H. L. Truong, and W. Kastner, “Manufacturing Process data analysis pipelines: a requirements analysis and survey”, *Journal of Big Data*, vol. 6, no. 1, pp. 1–26, 2019.
2. S. J. Qin, “Process data analytics in the era of big data,” *AIChE Journal*, vol. 60, no. 9, pp. 3092–3100, 2014.
3. F. Pervaiz, A. Vashistha, and R. Anderson, “Examining the Challenges in Development Data Pipeline,” *COMPASS2019 - Proceedings of the 2019 Conference on Computing and Sustainable Societies*, pp. 13–21, 2019.
4. C. Dehury, P. Jakovits, S. N. Srirama, V. Tountopoulos, and G. Giotis, *Data pipeline architecture for serverless platform*, vol. 1269 CCIS. Springer International Publishing, 2020.
5. N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, “Data lifecycle challenges in production machine learning: A survey” *SIGMOD Record*, vol. 47, no. 2, pp. 17–28, 2018.
6. A. Akoglu and G. Vargas-Solar, “Putting Data Science Pipelines on the Edge,” pp. 1–13, 2021.
7. A. Raj, J. Bosch, H. H. Olsson, and T. J. Wang, “Modelling Data Pipelines,” *Proceedings - 46th Euro micro-Conference on Software Engineering and Advanced Applications*, SEAA 2020, pp. 13–20, 2020.
8. M. Helu, T. Sprock, D. Hartenstein, R. Venketesh, and W. Sobel, “Scalable data pipeline architecture to support the industrial internet of things”, *CIRP Annals*, vol. 69, no. 1, pp. 385–388, 2020.
9. N. Konstantinou and N. W. Paton, “Feedback driven improvement of data preparation pipelines”, *Information Systems*, vol. 92, 2020.
10. N. Konstantinou, M. Koehler, E. Abel, C. Civili, B. Neumayr, E. Sallinger, A. A. Fernandes, G. Gottlob, J. A. Keane, L. Libkin, and N. W. Paton, “The vada architecture for cost-effective data wrangling,” *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD ’17, (New York, NY, USA), p. 1599–1602, Association for Computing Machinery, 2017.

11. E. G. Renart, D. Balouek-Thomert, and M. Parashar, “An edge-based framework for enabling data-driven pipeline for IoT systems”, Proceedings - 2019 IEEE 33rd International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2019, pp. 885–894, 2019.6
12. U. Zdun, E. Ntontos, K. Plakidas, A. El Malki, D. Schall, and F. Li, “On the design and architecture of deployment pipelines in cloud-and service-based computing-a model-based qualitative study”, Proceedings - 2019 IEEE International Conference on Services Computing, SCC 2019- Part of the 2019 IEEE World Congress on Services, pp. 141–145, 2019.
13. E. G. Renart, D. Balouek-Thomert, and M. Parashar, “Edge Based Data-Driven Pipelines (Technical Report),”2018.
14. H. Sebei, M. A. Hadj Taieb, and M. Ben Aouicha, “Review of social media analytics process and Big Data Pipeline”, Social Network Analysis and Mining, vol. 8, no. 1, 2018.
15. A. F. Conceição, J. V. S´anchez, B. G. Dos Santos. Vieira, and V. Rocha, “Pipeline architecture for mobile data analysis”, International Conference on Information Networking, no. February 2015, pp. 492–496, 2014.
16. F. Moyón, R. Soares, M. Pinto-Albuquerque, D. Mendez, and K. Beckers, “Integration of Security Standards in DevOps Pipelines: An Industry Case Study”, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12562 LNCS, pp. 434–452, 2020.
17. F. Ullah, A. J. Raft, M. Shahin, M. Zahedi, and M. A. Babar, “Security support in continuous deployment pipeline” ENASE 2017 - Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 57–68, 2017
18. <https://www.cisco.com/c/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf>
19. Mahapatra, T. (2020). Composing high-level stream processing pipelines. Journal of Big Data, 7(1). <https://doi.org/10.1186/s40537-020-00353-2>