

Improvement of protein model scoring using grouping and interpreter for machine learning

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and
Engineering

By

NETTEM SAI SUMANTH (Reg. No. 37110507)

N.SHIVA(Reg. No. 37110502)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NETTEM SAI**

SUMANTH (Reg. No. 37110507) , N.SHIVA(Reg. No. 37110502)

who carried out the project entitled **“Improvement of protein model scoring using grouping and interpreter for machine learning”** under our supervision from Dec 2020 to Mar 2021.

Internal Guide

Mr.R.Vignesh.M.E.,(Ph.D
).,

Head of the Department

Dr.S. Vigneshwari M.E., Ph.D.

Dr.L. Lakshmanan M.E.,

Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We, NETTEM SAI SUMANTH (Reg. No. 37110507) , N.SHIVA(Reg. No. 37110502) hereby declare that the Professional Training Report on **“Improvement of protein model scoring using grouping and interpreter for machine learning”** done under the guidance of **Mr. R.Vignesh M.E., (Ph.D.)**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.Vignesh.R M.E.,(Ph.D)**., for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Protein prediction is the deduction of the three-dimensional structures of proteins from their amino acid sequences, which involves the prediction of its folding and the secondary structure from its primary sequence. New proteins on discovery are subject to structure prediction. A linear protein structure is derived from the protein in the form of chains which is the primary structure. The chain provides information about arrangement of the amino acids that is helix, strands, loops and turns. Although plenty of protein sequence data is available online, it was not easy to acquire data that was pretty classified with labels. We scraped information from various websites allowing us to get class labels for our unsupervised data set. Our attempt uses Feature Vectors methodology to generate a transformed view of the protein sequences, which are then used as input to classifiers like Random Forest and Naive Bayesian. With this we are able to compare and contrast the precision and accuracy of the predicted sequences. Results of the comparison will provide deep insights in methodologies used for structure prediction in the future. Our research yields the best result using the Random Forest Classifier with an astounding accuracy of 9fi to.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	V
	LIST OF FIGURES	IX
	LIST OF ABBREVIATION	X
1	INTRODUCTION	1
	1.1 OVERVIEW OF BIG DATA	1
2	LITERATURE SURVEY	4
	2.1 ON THE AVERAGE LOCALITY OF LOCALLY REPAIRABLE CODES	4
	2.2 MANUFACTURING ANALYTICS AND INDUSTRIAL INTERNET OF THINGS	5
	2.3 AN EFFICIENT INDUSTRIAL BIG-DATA ENGINE	5
	2.4 CROSS PLATFORM RESOURCE SCHEDULING FOR SPARK AND MAP-REDUCE ON YARN	6
	2.5 ENUMERATING MAXIMAL BICLIQUES FROM A LARGE GRAPH USING MAP-REDUCE	6
3	METHODOLOGY	8
	3.1 DESIGN METHODOLOGY	8
	3.1.1 EXISTING SYSTEM METHODOLOGY	8
	3.1.2 PROPOSED SYSTEM	9
	3.2 ARCHITECTURE	11
	3.3 EXISTING ALGORITHM	11
	3.3.1 BATCH PROCESSING ALGORITHM	11
	3.4 PROPOSED ALGORITHM	12
	3.5 DATA ACQUISITION	13
	3.6 DATA STORAGE	14
	3.7 TECHNOLOGIES USED	15
	3.8 RELATIONAL DATABASE MANAGEMENT	22
	3.9 PANDAS	27
	3.10 NUMPY	30
	3.11 FRONTEND TECHNOLOGIES	32
	3.12 ADVANTAGES OF PROPOSED ALGORITHM	32

4.	RESULTS AND DISCUSSIONS	31
	4.1 DATA SET	31
	4.2 MODIFIED TABLE	33
	4.3 GRAPH PRESENTATION	33
	4.4 BAR GRAPH	33
	4.5 ACCIDENT IN INDUSTRIES	34
5.	CONCLUSION AND FUTURE WORK	38
	REFERENCES	40
	APPENDIX	40
	A. SOURCE CODE	40
	B. SCREEN SHOTS	40
	C. PUBLICATION	46

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.2	SYSTEM ARCHITECTURE	11
3.4	INDUSTRIAL DATA ANALYSIS	13
3.6	DATA STORED FOR PROCESSING DATA	14
3.7.1	PYTHON	15
3.7.2	OPERATION IN PYTHON	16
4.1	DATA SET	34
4.2	MODIFIED TABLE USING DATA	35
4.3	GRAPH PRESENTATION	35
4.4	BAR PRESENTATION	36

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF BIG DATA

In today's e-Business environment, ERP, CRM, collaboration tools, and networked sensors may be characterized as data generators resources. Business Intelligence (BI) is a term that incorporates a range of analytical and decision support applications in business including data mining, decision support systems, knowledge management systems, and online analytical processing; processing data within these systems produce new data that are characterized to grow rapidly.

Collectively these structured and unstructured data are referred to as Big Data. Successful and efficient handling of Big Data requires deployment of specific IT infrastructure components as well as adopting an emerging service model. In this research we introduce a conceptual model that abstracts the processing scheme of big data processing lifecycle. The model addresses the main phases of the lifecycle: data acquisition, data serialization, data aggregation, data analysis, data mining, knowledge representation, and information dissemination.

Big data, in general, are grouped into three types of data: archived data, repository data, and flowing data. Flowing data (aka in motion) are dependent five factors: velocity (rate of data flow), variability (change of rate of data flow, structure, and refresh rate), accessibility, transport format, and transport protocols.

Big data based Acquisition and Storage System (ASS) plays an important role in the design of industrial data platform. Many big data frameworks have been integrated compression and serialization method. These methods cannot meet the needs of industrial production information management for requiring the time-consuming and mass storage. Based on the existing big data frameworks, we proposes an enhanced industrial big data platform in order to reduce the data processing time while require less data storage space. Specifically, this study focuses on evaluating the impact of multiple compression and serialization methods on big data platform performance and tries to choose optimal compression and serialization methods for the industrial data platform. Compared to the methods integrated in Hadoop and Spark, the experimental

the data compression time of the platform has been reduced by 73.9% with a less than 96% the size of data compressed, furthermore, the data serialization time has been reduced by 80.8%. With the increasing amount of data, it takes less time to compare with benchmark method.

Big data is changing how companies function across all industries. Both large and small organizations can benefit from the insights gained from the large volumes of data that is out there. If data management is not handled correctly it can cost a business up to 35 percent of their operating revenue. So, in a snapshot, it seems like recognizing the role big data can play in your company is the smart thing to do.

But how do we understand big data? What makes data 'big' and does it relate to how project managers get their jobs done? Our blog today has the answers you need.

First, we must understand what makes data 'big'. The measurements for big data are often set out as 'the five Vs'.

Volume

There is a vast amount of data being created and collected every day. The data produced in 2020 will be [44 times](#) greater than it was in 2009. That shows you the increase we are looking at and it isn't slowing down. Traditional data tools are unable to handle the enormity of the volume and so big data tools have emerged as their replacements – these tools are helping businesses to make more incisive decisions about their processes.

Velocity

Data is being generated at incredibly high speeds. The way we use information today is faster than at any other time. Think of applying for a loan or another banking transaction. Data analysis and processing now take seconds rather than hours or days.

Variety

There are many different types of data out there. Previously, data was usually something that fits easily into tables and charts. Today, data is much more unstructured. Data could mean the photos we take and share, the social media.

Veracity

With so much data and information floating around it can be difficult to ensure what is quality, true, false, accurate, etc. Data contains typos, slang, and unreliable sources. Again, big data tools can take veracity into account when data is being analyzed.

How big data is transforming project management

So, what about big data in project management? Big data can have a positive effect on how the project management office (PMO) goes about its job. From reducing project costs and increasing project efficiencies to improving resource allocation, big data is having a major impact on the improvement of project and resource management.

Let's look at some of the ways big data can help an organization accelerate its project processes to meet its goals

Boost to project intelligence

Business intelligence and the tools that enable effective analysis were rather restricted before the emergence of big data. Big data is the rocket fuel that allows business intelligence tools to work to their best. For project managers, it means having access to much better knowledge of each project they are working on, from timelines to resource allocation. This leads to better decision-making.

Reduces project costs

By collecting more and more data, project managers can more easily predict trends and future events within their industries. Resource forecasts and planning can be made more efficient. This leads to more efficient output; workers are given the jobs they are best at and better information to complete their tasks.

Improves project efficiency

Big data gives organizations important information about their processes, their products, or their services. Project bottlenecks and other issues can be discovered quickly and resolved, helping to keep each project and the business working optimally.

CHAPTER 2

LITERATURE SURVEY

This chapter gives an insight to different works done by different researchers about “Improved data Acquisition and Capacity for Designing Modern Information Stage”

2.1 : - TITLE: On the Average Locality of Locally Repairable

Codes **AUTHORS:** Mostafa Shahabinejad and Masoud Ardakani

PUBLISHED YEAR: 1 NOV 2017

EFFICIENCY: A novel approach to find a general lower bound on of erasure codes with arbitrary parameters.

DRAWBACKS: The non-zero elements can be randomly selected from a sufficiently large finite field. Constructing explicit r -optimal LRCs over small finite fields as well as generalizing our proposed lower bound on r for vector codes and erasure codes with multiple groups of locality remains as open problems.

DESCRIPTION: A linear block code with dimension k , length n , and minimum distance d is called a locally repairable code (LRC) with locality r if it can retrieve any coded symbol by at most r other coded symbols. LRCs have been recently proposed and used in practice in distributed storage systems (DSSs) such as Windows-Azure storage and Face book HDFS-RAID. Theoretical bounds on the maximum locality of LRCs (r) have been established. The *average* locality of an LRC (r) directly affects the costly repair bandwidth, disk I/O, and number of nodes involved in the repair process of a missing data block. There is a gap in the literature studying r . In this paper, we establish a lower bound on r of arbitrary $(n; k; d)$ LRCs. Furthermore, we obtain a tight lower bound on r for a practical case where the code rate ($R = kn$) is greater than $(1 - \frac{1}{pn})^2$. Finally, we design three classes of LRCs that achieve the obtained bounds on r . Comparing with the existing LRCs, our proposed codes improve the average locality without sacrificing such crucial parameters as the code rate or minimum distance.

2.2: -PUBLICATION: Manufacturing Analytics and Industrial Internet of Things

AUTHORS: Prasanth Lade and Rumi Ghosh, Robert Bosch LLC Soundar Srinivasan

PUBLISHED YEAR: 8 NOV 2017

EFFICIENCY: This paper provides flexibility and tailored functionalities that make the implementation more efficient than a standard commercial solution.

DRAWBACKS: In future work, we plan to use machine learning as a collaborative tool for process engineers. This will also highlight the need for engineers to pursue lifelong education and expansion of skills

DESCRIPTION: This paper describes the opportunities for the use of analytics in manufacturing and presents a case study of a successful application at one of Bosch's large-scale manufacturing facilities. Building on our experience in more than two scores of such applications, we present our recommendations for replications in other operations and share our preferred technology stack. Finally, we provide a view of future developments in this area, including the co-evolution of technologies and solutions in the broader Internet of Things (IOT) space.

2.4. PUBLICATION: Cross-platform Resource Scheduling for Spark and Map-Reduce on YARN

AUTHORS: Dazhao Cheng, Xiaobo Zhou, Pal den Lama, Jun Wu and Changjun Jiang

PUBLISHED YEAR: 4 NOV 16

EFFICIENCY: Emerging Apache Spark fills in this gap, which can run on an established Hadoop cluster and take advantages of existing HDFS.

DRAWBACKS: In future work, we will explore more cross-platform resource scheduling approaches while deploying other processing paradigms (e.g., Shark, Pig, Storm and Hive) on Hadoop YARN

DESCRIPTION: Map Reduce is inherently designed for batch and high throughput processing workloads, there is an increasing demand for non-batch processes on big data, e.g., interactive jobs, real-time queries, and stream computations. Emerging Apache Spark fills in this gap, which can run on an established Hadoop cluster and take advantages of existing HDFS. As a result, the deployment model of Spark-on-YARN is widely applied by many industry leaders. However, we identify three key challenges to deploy Spark on YARN, inflexible reservation-based resource management, inter-task dependency blind scheduling, and the locality interference between Spark and Map Reduce applications. The three challenges cause inefficient resource utilization and significant performance deterioration. We propose and develop a cross-platform resource scheduling middleware, iKayak, which aims to improve the resource utilization and application performance in multi-tenant Spark-on-YARN clusters. iKayak relies on three key mechanisms: reservation-aware executor placement to avoid long waiting for resource reservation, dependency-aware resource adjustment to exploit under-utilized resource occupied by reduce tasks, and cross-platform locality-aware task assignment to coordinate locality competition between Spark and Map Reduce applications. We implement iKayak in YARN. Experimental results on a test bed show that iKayak can achieve 50% performance improvement for Spark applications and 19% performance improvement for Map Reduce applications, compared to two popular Spark-on-YARN deployment models, i.e., YARN-client model and YARN-cluster model.

2.5 PUBLICATION: Enumerating Maximal Bicliques from a Large Graph using Map Reduce

AUTHORS: Arko Provo Mukherjee, Arko Provo Mukherjee

PUBLISHED YEAR: 2 NOV 16

EFFICIENCY: 1) The redundancy in work between different sub-graph explorations is minimized through a careful pruning of the search space, and

2) The load on different reducers is balanced through a task assignment that is based on an appropriate total order among the vertices.

DRAWBACKS: The first problem is the overlap in work within different tasks. For biclique enumeration, it is usually not possible to assign disjoint sub graphs to different tasks, and sub graphs assigned to different tasks will overlap, sometimes significantly.

DESCRIPTION: The enumeration of maximal bipartite cliques (bicliques) from a large graph, a task central to many data mining problems arising in social network analysis and bioinformatics. We present novel parallel algorithms for the Map Reduce framework, and an experimental evaluation using Hadoop Map Reduce. Our algorithm is based on clustering the input graph into smaller sub graphs, followed by processing different sub graphs in parallel. Our algorithm uses two ideas that enable it to scale to large graphs: (1) the redundancy in work between different sub graph explorations is minimized through a careful pruning of the search space, and (2) the load on different reducers is balanced through a task assignment that is based on an appropriate total order among the vertices. We show theoretically that our algorithm is work optimal i.e. it performs the same total work as its sequential counterpart. We present a detailed evaluation which shows that the algorithm scales to large graphs with millions of edges and tens of millions of maximal bicliques. To our knowledge, this is the first work on maximal biclique enumeration for graphs of this scale.

3.1 DESIGN METHODOLOGY

C
HA
PT
ER
3
ME
TH
OD
OL
OG
Y

Big Data Analytics is a multi-disciplinary open-access, peer reviewed journal, which welcomes cutting-edge articles describing original basic and applied work involving computational accounts of all aspects of big data science analytics. Spanning the life sciences, social sciences, engineering, physical and mathematical sciences, *Big Data Analytics* aims to provide a platform for the dissemination of research, current practices, and future trends in the emerging discipline of big data analytics.

Big Data Analytics invites high-quality original research articles and timely reviews on current developments in the field, covering all aspects of big data analytics, including, but not limited to the following topics:

Algorithmic, theoretical and computational approaches (such as deep learning networks, nature-inspired and brain-inspired cognitive computation, statistical and mathematical analytics, visualization and informatics).

Implementations and platforms (such as neuromorphic, GPUs, clusters and clouds, and open-source software).

Applications in domains as diverse as genomics, medicine, healthcare, clinical, biological and neuro-informatics, natural robotics, language processing, meteorology, geosciences, multimedia and business intelligence, social media and network analytics, trend discovery, opinion mining, smart cities, surveillance, transportation, power, energy and economic management, internet search, biological, chemical, physical, environmental, oceanic and planetary sciences, and e-Science in general.

3.1.1 EXISTING SYSTEM METHODOLOGY

There are massive multi-source heterogeneous data inside and outside the business. The data platform needs to connect multiple types of data in equipments, production lines, products and industrial software. At present, industrial data lack effective interoperability standards or low-cost acquisition schemes. This leads to high costs of data collection and integration of equipments

and systems. Thus, it is difficult to support functions of industrial data analysis and application development.

There are a large number of data analysis requirements in the plant. The data platform needs to support multiple types of data analysis and data visualization services, such as Finite Element Analysis, Optimization, Deep Learning, Knowledge Mapping, Digital Twins. Generally, big data platform includes three modules, which are data acquisition module, data storage module and data computation module.

The data acquisition module provides a data source for data analysis of the big data platform, and the data storage module provides data source and storage

space of the data computation module. It provides data calculation, machine learning, graph analysis, data query, which is the core component of data analysis. The computation module computes massive amounts of data, mines useful data value information, and provides decision-making grounds for industrial decision makers. Both Sqoop and Flume are the main framework for data acquisition modules.

DRAWBACKS OF EXISTING SYSTEM

- Cost of data representation is small as the data is split in Spark.
- Overlap in work within different tasks.
- Costly repair bandwidth.
- Disk I/O is costly
- Number of nodes involved in the repair process of a missing data block is Expensive

3.1.2 PROPOSED SYSTEM

At Big Data scale you constantly need to serialize and de-serialize objects.

This occurs when

- Mappers write output to local disk.
- Mappers read output from local disk for Map side sorting
- When Reducers pull sorted/shuffled output from Mapper node

In each of the above instances, you are working with considerably more data than represented by the raw byte representation. Storage overhead of Big Data objects impacts how efficiently the memory hierarchy is utilized. The proposed system performs split ability for easier to achieve splits on byte streams rather than JSON or XML files. It also performs portability for schema can be consumed by different languages and versioning for flexibility to define fields with default values or continue to use old schema version data integrity for serialization schemas enforces data correctness. Thanks to them, errors can be detected earlier, when data is written.

The proposed system features three attributes: visibility, interaction, and effect. Visibility refers to the provider's and requester's ability to interact and accomplish

the requested service capability. This attribute introduces opportunities for

matching the requester's needs to the provider's capabilities and vice versa. We can decompose visibility according to related attributes or functional requirements, such as awareness, willingness, reach-ability, and self-serviceability. The second attribute is interaction when the requester uses one of the provider's capabilities. Interaction is mediated through message exchange and produces an effect (or set of effects). An effect can take the form of information resulting from using a capability, a state change in entities (defined or undefined), or a combination of both.

ADVANTAGES OF PROPOSED SYSTEM

- It split work into smaller pieces that may be more efficiently processed in parallel.
- Can run on an established Hadoop cluster and take advantages of existing HDFS.
- Redundancy in work is minimized.
- Provides flexibility and tailored functionalities that make the implementation more efficient.
- Improved performance.

3.2 ARCHITECTURE



Fig:3.2 System Architecture

3.3 EXISTING ALGORITHM

3.3.1 : Batch processing algorithm

Scheduling problems of a batch processing machine are solved by efficient algorithms. On a batch processing machine, multiple jobs can be processed simultaneously in a batch form. We call the number of jobs in the batch the batch size, which can be any integer between 1 and k , a predetermined integer of the maximum batch size. The process time of a batch is constant and independent of the batch size. Preemption is not allowed. Given n jobs with release time's r_i and due dates d_i , $i = 1, \dots, n$, we give efficient algorithms to find a feasible schedule, if any, which minimizes the final completion time under the assumption such that for $r_i > r_j$, $d_i \geq d_j$. Some industrial applications are discussed.

3.4 PROPOSED ALGORITHM

3.4.1 Streaming processing algorithm

Stream processing is the processing of data in motion, or in other words, computing on data directly as it is produced or received. The majority of data are born as continuous streams: sensor events, user activity on a website, financial trades, and so on – all these data are created as a series of events over time. In computer science, streaming algorithms are algorithms for processing data streams in which the input is presented as a sequence of items and can be examined in only a few passes (typically just one). ... They may also have limited processing time per item. Stream processing is the processing of data in motion, or in other words, computing on data directly as it is produced or received. The majority of data are born as continuous streams: sensor events, user activity on a website, financial trades, and so on – all these data are created as a series of events over time.

ADVANTAGES OF PROPOSED ALGORITHM

- Gain more value from streaming data ingest with Kafka.
- Support DB streaming by turning databases into live feeds.
- Accelerating delivery of data to enable real-time analytics.

- Reduce the skill and training requirements for managing data stream processing.

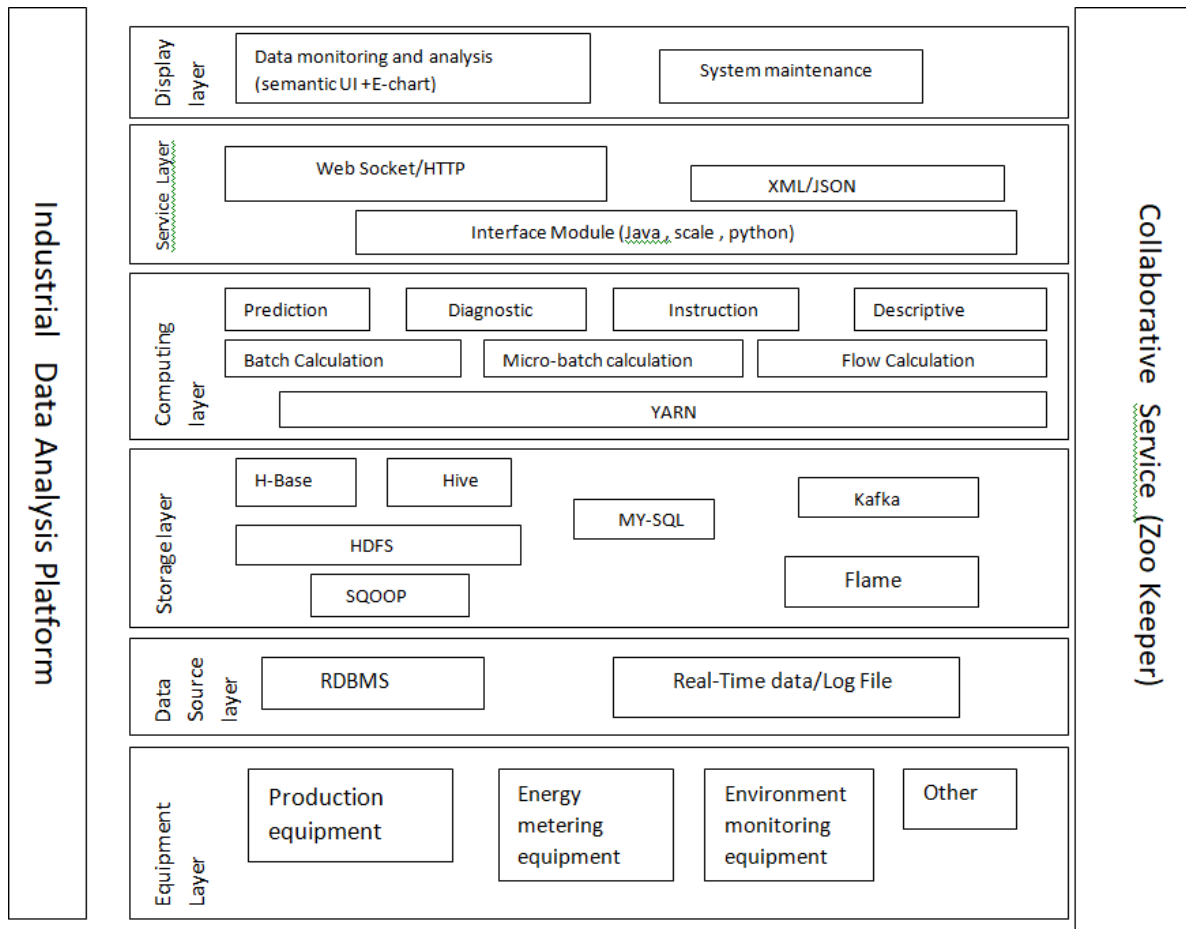


Fig: 3.4 Overall architecture of industrial data analysis platform.

3.5 DATA ACQUISITION

The system transmits data by three channels. Firstly, data flow is a timeless computing module consisting of Flume, Kafka, Structure Streaming and Flink. Secondly, data flow is a data query module consisting of Flume, HBase, Spark SQL and Table. Thirdly, dataflow is a visual analysis module consisting of Flume, HBase, Spark Streaming and Flink. The data serialized method of the acquisition framework is upgraded. The specific acquisition scheme is as follows. The module for Flume data acquisition is shown in Figure 2. The information configured of

Flume requires three parts: Source, Channel, and Sink. The Source receives the event from the server or log file and puts it into the Channel in bulk. Data transmission of Flume is divided into two data transmission channels, so you need to set KafkaChannel, HBase- Channel, KafkaSink and HBaseSink. The information configured of Sources requires to configure type, Channel, Internet Protocol Address (IP), Port and selector type. The specific configuration information is shown in Table 1. The data acquisition method of Sqoop relational database is shown in Figure 3. After receiving the shell command or Java API command from the client, the Sqoop converts the command to the corresponding MapReduce task by task translator in Sqoop, and then transfers the data from relational database to Hadoop to complete the copy of the data. The Sqoop operates on relational databases by four ways. First of all, the Sqoop gets data from all MySQL databases. Then Sqoop sends these data to HDFS. Next, Sqoop imports data from MySQL to Hive. Last, Sqoop exports data from Hive to MySQL.

3.6 DATA STORAGE

Hadoop is adopted by the data storage framework. Both file storage and inter-cluster communication on HDFS require data compression. The former can increase the amount of industrial data stored and reduce the cost. And the latter can reduce the bandwidth consumption of data transmission between networks and reduce data transmission time. As we know, industrial big data platform needs high stability.

Dataset.csv - Microsoft Excel

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Data	Country	Local	Industry	Accident	Potential	Genre	Employee	Critical	Description												
2	0	Country_0	Local_01	Mining	I	IV	Male	Third Party	Pressurized	While removing the drill rod of the Jumbo 08 for maintenance, the supervisor proceeds to loosen the support of the intermediate centralizer to fac												
3	1	Country_0	Local_02	Mining	I	IV	Male	Employee	Pressurized	During the activation of a sodium sulphide pump, the piping was uncoupled and the sulfide solution was designed in the area to reach the maid. Im												
4	2	Country_0	Local_03	Mining	I	III	Male	Third Party	Manual To	In the sub-station MILPO located at level +170 when the collaborator was doing the excavation work with a pick (hand tool), hitting a rock with the												
5	3	Country_0	Local_04	Mining	I	I	Male	Third Party	Others	Being 9:45 am, approximately in the Nv. 1880 CX-695 OB7, the personnel begins the task of unlocking the Saquet bolts of the BHB machine, when t												
6	4	Country_0	Local_04	Mining	IV	IV	Male	Third Party	Others	Approximately at 11:45 a.m., in circumstances that the mechanics Anthony (group leader), Eduardo and Eric Fernández-injured-the three of the Co												
7	5	Country_0	Local_05	Metals	I	III	Male	Third Party	Pressurized	During the unloading operation of the ustulado Bag there was a need to unclog the discharge mouth of the silo truck. In performing this procedure,												
8	6	Country_0	Local_05	Metals	I	III	Male	Employee	Fall prevent	The collaborator reports that he was on street 09 holding in his left hand the volumetric balloon, when he slipped and when placing his hand on the												
9	7	Country_0	Local_04	Mining	I	III	Male	Third Party	Pressurized	At approximately 04:50 p.m., when the mechanic technician JosãO of the Tecnomin verified the transmission belts of the HM-100 pump at the Ac												
10	8	Country_0	Local_02	Mining	I	IV	Male	Third Party	Others	Employee was sitting in the resting area at level 326 (raise bore), when he suffered sudden illness, falling and suffering excoriation on the face.												
11	9	Country_0	Local_06	Metals	I	II	Male	Third Party	Chemical s	At the moment the forklift operator went to manipulate big bag of bixolite in section 70 and just in front of the ladder that leads to the area of &E												
12	10	Country_0	Local_03	Mining	I	III	Male	Employee	Others	While installing a segment of the polyurethane pulley protective lyner - 60x45cm weighing 1.2 kg - on the head pulley of the ore winch, when the												
13	11	Country_0	Local_03	Mining	I	IV	Male	Third Party	Others	While preparing the rice for the lunch of the day, when moving the pot # 60 - 35 Kg of weight including the contents - to evacuate the residual wat												
14	12	Country_0	Local_05	Metals	I	I	Male	Employee	Liquid Met	The collaborator reports that he was working in the Ustulaciã'n and realized that the cyclone duct was obstructed and opened the door to try to u												
15	13	Country_0	Local_01	Mining	IV	V	Male	Third Party	Electrical i	In moments that the operator of the Jumbo 2, tried energize your equipment to proceed to the installation of 4 split set at intersection 544 of Nv 3												
16	14	Country_0	Local_05	Metals	I	III	Male	Employee	Confined s	Due to the accumulation of Waelz on the conveyor and trailer of the Filter 08FI0502, the employee performed the cleaning of the shutter using the												
17	15	Country_0	Local_05	Metals	I	IV	Male	Employee	Liquid Met	The employee was working in the When a thermal shock caused a splash of zinc in his direction, the employee, despite using all the indicated PPE, v												
18	16	Country_0	Local_04	Mining	III	IV	Male	Third Party	Others	At Rp 050 of level 1620, in circumstances where the workers of the company were performing the task of diamond drilling, the assistants Jhonatan												
19	17	Country_0	Local_06	Metals	I	II	Female	Third Party	Others	Due to the overheating of 2 bars in row 5 of cell 7 a spark is produced, which is projected and manages to reach the Chief of guard who was in the												
20	18	Country_0	Local_06	Metals	I	II	Male	Employee	Others	An auxiliary wheel of the cathode crane G2133 was changed in area 75, when when a bearing was heated and hit with a hammer and chisel at one												
21	19	Country_0	Local_06	Metals	I	III	Male	Employee	Others	The worker Manuel was making the disconnection of the power cables of the gate that is at the intersection of Manco streets with Cajamarquilla i												
22	20	Country_0	Local_06	Metals	I	II	Male	Third Party	Others	At 14:16, the operator Samuel was about to open the container for its subsequent loading with silver concentrate, and opened the first gate. After												
23	21	Country_0	Local_03	Mining	I	IV	Male	Third Party	Others	In the Raise chamber, when the operator and assistant removed the drilling bar from the 71R drilling machine - weight approx. 500 Kg - holding it a												
24	22	Country_0	Local_03	Mining	I	IV	Male	Third Party	Others	While aligning the right bracket of tower N 4 32, when releasing the tension applied by the tirdorf of 1.5 Tn, when pushing the lever towards the t												
25	23	Country_0	Local_02	Mining	II	II	Male	Third Party	Others	By loosening the A30-20 truck's steering cylinder bolt using power cable and socket, with the force exerted in favor of the equipment, the bolt sudc												
26	24	Country_0	Local_06	Metals	I	II	Male	Third Party	Others	An operator of the G-2142 deslaminadora in section 75 was unlocking a sheet of zinc stuck so that it can enter the slide of the stacking table, for th												
27	25	Country_0	Local_06	Metals	I	II	Male	Employee	Manual To	The operator was willing to manually displace the zinc sheet adhered to the aluminum cathode. At the moment when the blade was detached, the												
28	26	Country_0	Local_01	Mining	I	I	Male	Employee	Others	At the end of lunch in an enabled place on the side of the winche control room. He gets up and after a short walk he slips and sits with the floor an												
29	27	Country_0	Local_03	Mining	I	III	Male	Third Party	Others	At the end of the rock break at the intersection of the ramp -1055 and the opening of the access to the ventilation chimney, the master loader ider												
30	28	Country_0	Local_01	Mining	I	IV	Male	Employee	Others	The worker carried out his work at level 2940, stop 895, and placed metal mesh on the basket (SOIL HALF) of the Scissor the mesh falls and hits the												
31	29	Country_0	Local_07	Mining	II	III	Male	Employee	Others	When using the griff wrench to unscrew the rod from the probe the key came to move by pressing the employee's finger against the probe.												
32	30	Country_0	Local_03	Mining	I	II	Male	Employee	Others	When the plant operator was removing the lid of the distributor box of the second mill No. 4 and No. 5, his right												

Ready Type here to search

1937 26-04-2020

Fig: 3.6 Data stored for processing the data

3.7 TECHNOLOGIES

USED BACKEND

TECHNOLOGIES

- Python
- Numpy
- Sci learn
- Eclipse IDE

ABOUT PYTHON: Python is a free, open-source programming language.

Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these

operating systems.

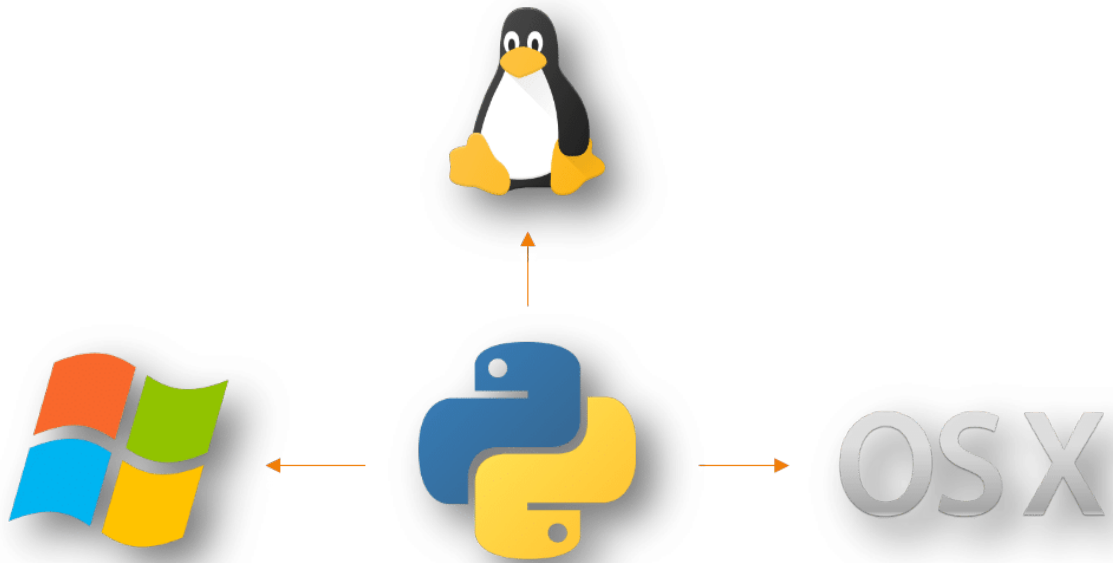


Fig: 3.7.1 Python

Python is also a great visualization tool. It provides libraries such as Matplotlib, sea born and boke to create stunning visualizations.

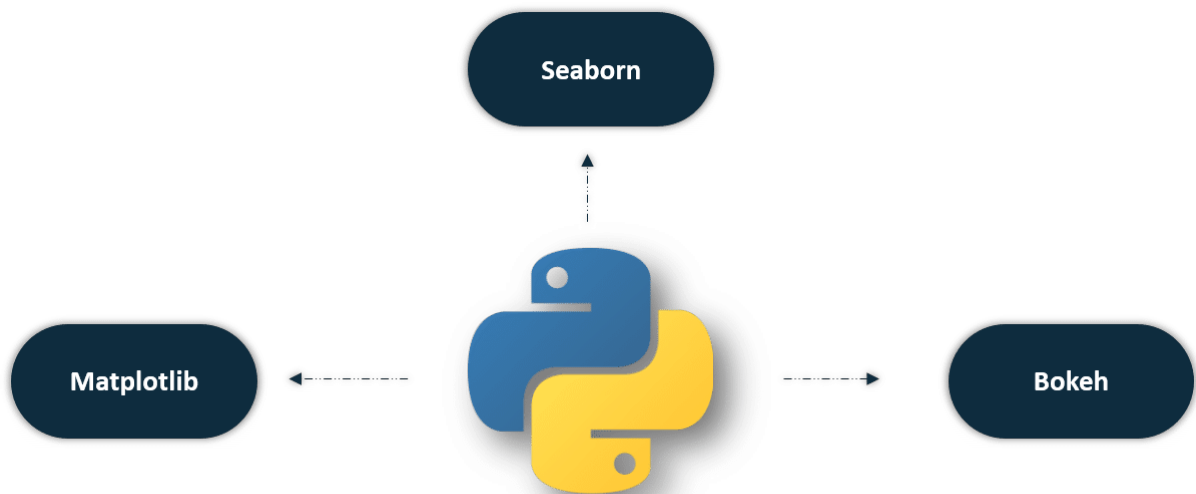


Fig: 3.7.2 Operations in python

In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

Python Line Structure

Python coding style comprises physical lines as well as logical lines or statements.

A physical line in a Python program is a sequence of characters, and the end of

the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semi-colon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although it's not mandatory. The NEWLINE token denotes the end of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines.

Explicit Line Joining

In explicit line joining, we use a backward slash to split a statement into a multiline statement.

Implicit Line Joining

Statements that reside inside [], {}, or () parentheses can be broken down into two or more physical lines without using a back slash.

Multiple Statements on a Single Line

In Python, it is possible to club multiple statements in the same line using a semi-colon; however, most programmers do not consider this to be a good practice as it reduces the readability of the code.

Whitespaces and Indentation

Unlike most of the programming languages, Python uses indentation to mark a block of code. According to Python coding style guideline or PEP8, we should keep an indent size of four.

Most of the programming languages provide indentation for better code formatting and don't enforce to have it. But in Python it is mandatory. This is why indentation is so crucial in Python.

Comments in any programming language are used to increase the readability of

the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It

is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project.

The code can only explain how it does something and not why it does that, but Python comments can do that. With Python comments, we can make documentations for various explanations in our code itself. Comments are nothing but tagged lines of codes which increase the readability of a code and make it self-explanatory. There are different ways of creating comments depending on the type of comment we want to include in our code. Following are different kinds of comments that can be included in our Python program:

1. Single Line Comments
2. Doc string Comments
3. Multiline Comments

Single line Python comments are marked with `#` character. These comments end at the end of the physical line, which means that all characters starting after the `#` character (and lasts till the end of the line) are part of the comment.

Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules. Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator.

It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Doc-strings can be one-liners as well as multi-liners.

Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code.

Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that

follows the Block comment. Python Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Python's PEP8 style guide since Block comments are ignored by Python interpreter or parser. However, nothing is stopping programmers from using the second 'non-real' way of writing multiline comments in Python which is explained below.

Using Docstrings: Docstrings are largely used as multiline comments in Python by many programmers since it is the closest thing to having a multiline comment feature in Python. While it is not wrong to use doc-strings when we need to make multiline comments, it is important to keep in mind that there is a significant difference between docstrings and comments. Comments in Python are totally ignored by the Python Interpreter, while docstrings, when used inside the Python function, can be accessed at the run time.

Python Data Types

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing.

In Python, unlike statically typed languages like C or Java, there is no need to specifically declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself predicts the data type of the Python Variable based on the type of value assigned to that variable.

Advantages of Python

- Universal Language Construct
- Support both High Level and Low Level Programming
- Language Interoperability
- Fastest Development life cycle therefore more productive coding environment
- Less memory used because a single container hold multiple data

types and each type doesn't require its own function

- Learning Ease and open source development
- Speed and user-friendly data structure
- Extensive and extensible libraries.
- Simple & support IoT
- And many more

Based on the 3 languages that are described above, we decided to use Python as programming language for developing this e-Voting web based application. The important motives are

- Easy to learn, even non experienced programmers can use it.
Ex: spacing and tabbing instead of extra syntax
- Interactive mode
- Large and comprehensive standard libraries
- Python programs resemble to that of pseudo-code. This makes it a basis and a must have for beginner programmers due to its extreme ease a difficulty when compared to C++ Java, Perl, and so forth

Selection of Integrated Development Environment

IDE – Integrated Development Environment. It is software suite it contains different instruments for web application software engineers to compose and test applications.

Programming Web application software engineers utilize wide scope of devices habitually all through application source code creation, building and application testing. Programming Development apparatuses frequently incorporate word processors for source code creation, code libraries for reuse, compilers and test stages.

An Integrated Development Environment (IDE) is a product application that gives a programming domain to streamline creating and troubleshooting programming. Instead of playing out every one of the means required to make an executable program as disconnected individual assignments, it brings every one of the devices required into one application and workspace. Every one of the devices has a familiarity with the earth, and they cooperate to display a consistent improvement set for the designer.

Without an IDE, a product web application software engineer must choose, send, incorporate and deal with these instruments independently. An IDE brings huge numbers of those improvement related instruments all together system, application or administration. The incorporated toolset is intended to streamline programming advancement and can distinguish and limit coding missteps and grammatical errors.

- Komodo
- Net Beans
- PyCharm
- Python for Visual Studio.NET
- PyDev
- PyStudio
- LiClipse (Eclipse)

Liclipse

LiClipse is at the core for all Python based application development activities. It's been available for many years. LiClipse is a set of plugins to enhance Eclipse. It improves the overall Eclipse experience. It is best IDE tool to develop Python Applications.

Advantages of LiClipse

- Perfect Debugger
- Error checking features
- An enhanced ability to analyze the code for quality and security concerns
- Auto complete feature

Based on the various IDE that are described above, we decided to use LiClipse as IDE (Integrated Development Environment) for developing this e-

Voting web based application.

Important Features That Make LiClipse our IDE of Choice

- Rich Reusable Libraries are available
- Model Driven Development
- Enterprise Python Tooling
- Powerful Debugging options

3.8 Selection of Relational Database Management System (RDBMS)

There are various open source and commercial database management systems are available in the market. Normally, databases are divided into three categories

- Hierarchical databases
- Network databases
- Relational databases
- Object-oriented databases
- Graph databases
- ER model databases
- Document databases

Out of which Relational database management system is widely used. RDBMS represents social database the board framework. A social model can be spoken to as a table of lines and segments.

RDBMS is a data the board framework which is arranged on an information model. Here all the data is appropriately put away as tables. It offers data to be spared in the tables.

A relational database contains tables which store information that is connected here and there. SQL is the language that permits recovery and control of table information in a social database.

SQL can be used for storing and retrieving information from one spot considered database and utilize that information at whatever point it required in other programming applications.

There are approximately 121 relational database management systems are available in the market.

We have decided to use SQL Server for our e-Voting application.

SQL Server is a relational database technology developed by Microsoft. SQL Server can be used to develop from small scale to large enterprises with complex data requirements, data warehousing and Web-enabled databases.

SQL Server uses Transact-SQL (T-SQL) which is an extension of SQL.

Important Features That Make SQL Server our Database of Choice

- Better Performance Features
- Better Security Features
- Lower Cost Of Ownership
- Excellent Data Restoration and Recovery Mechanism

Selection of Operating System

There are various open source and commercial Operating systems are available in the market. Computer Operating systems can be broadly classified into two types:

- Normal Operating System
- Real Time Operating System

We have decided to use Windows 10 for our e-Voting application.

Windows 10 is a is the most powerful Microsoft operating system it can be used for desktop/laptop computers as well as IoT devices.

Important Features That Make Windows 10 our Operating System of Choice

- The loading time is very fast compare to other OS
- Supports built-in mobile device executive facilities
- Stability and performance
- Have hologram technology

1.2 Selection of Web Application Framework

Framework provides a structure for application development. This makes web application programmers' lives easier when developing consistent, accessible, and workable enterprise wide web applications. They automate the implementation of redundant tasks or extensions for common operations, reduces the development and testing time and allowing programmers to concentration more on application

logic instead of routine works.

Python framework can be classified as two types:

Full-Stack Framework

The full-stack systems give total help to engineers, including vital components, for example, structure approval, structure generators, and format designs.

- Web2py
- Django
- Flask
- Turbo Gears
- Zope2
- Cubic Web
- Grok
- Pylon

Non-Full Stack

The non-full-stack systems don't give extra functionalities and highlights to the clients. Designers need to include a great deal of code and different things physically.

Micro framework

Miniaturized scale structures are little, straightforward, and simple to utilize. They are laconic and have straightforward documentation. URL directing is frequently RESTful web services. Miniaturized scale systems use WSGI and work through HTTP request/Response. They are a decent decision for little applications, or as a feature of a bigger undertaking.

- CherryPy
- Flask
- Bottle
- Pyramid

Asynchronous Framework

In situations where solicitation handling pace assumes a significant job or an undertaking should manage long reaction times, non concurrent structure ought to work.

- Quart
- Tornado
- Fast-API
- Sanic
- Vibora

We have decided to use Flask Micro framework for our e-Voting application.

Flask is a lightweight popular Python WSGI web application framework. It is intended to make beginning fast and simple, with the capacity to scale up to complex applications. It started as a basic wrapper around Werkzeug and Jinja and has turned out to be one of the most well known Python web application systems.

It is viewed as more Python than the Django web system on the grounds that in like manner circumstances the identical Flask web application is increasingly express.

Flask offers proposals, yet doesn't uphold any conditions or venture format. It is dependent upon the engineer to pick the apparatuses and libraries they need to utilize. There are numerous expansions given by the network that make including new usefulness simple.

Applications that use the Flask framework include interest LinkedIn and the community web page for Flask itself.

Important Features That Make Flask Micro framework our micro framework of Choice

- Easy to get started
- Little Boilerplate code needed
- built-in development server and fast debugger

- Provides simplicity, flexibility and fine-grained control.

- Jinja2 templating

Selection Browser

There are various open source and commercial Operating systems are available in the market. There are five different Types of Web Browsers:

1. Web-Kit Browser Types
2. Blink Browser Types
3. Gecko Browsers
4. Goanna Browsers
5. Edge HTML Browsers

We have decided to use Google Chrome for our e-Voting application. Google Chrome is a free and one of the popular web browser created by Google that uses the WebKit layout engine.

Important Features That Make Google Chrome our Browser of Choice

- Extremely fast web browser; it loads and displays pages very quickly.
- Tab independence
- Support for latest EcmaScript
- Modern Layout
- Cross Platform Browser

3.9 Pandas

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The Data-Frame is one of these structures.

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data-Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed

statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

Jupyter Notebooks offer a good environment for using pandas to do data exploration and modeling, but pandas can also be used in text editors just as easily.

Jupyter Notebooks give us the ability to execute code in a particular cell as opposed to running the entire file. This saves a lot of time when working with large datasets and complex transformations. Notebooks also provide an easy way to visualize pandas' Data-Frames and plots. As a matter of fact, this article was created entirely in a Jupyter Notebook.

There are two types of data structures in pandas: Series and Data-Frames.

1. Series: a pandas Series is a one dimensional data structure ("a one dimensional ndarray") that can store values — and for every value it holds a unique index, too.
2. Data-Frame: a pandas Data-Frame is a two (or more) dimensional data structure – basically a table with rows and columns. The columns have names and the rows have indexes.

Those who are familiar with R know the data frame as a way to store data in rectangular grids that can easily be overviewed. Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

Now, Data Frames in Python are very similar: they come with the Pandas library, and they are defined as two-dimensional labeled data structures with columns of potentially different types.

In general, you could say that the Pandas Data Frame consists of three main components: the data, the index, and the columns.

Firstly, the Data Frame can contain data that is:

- A Pandas Data Frame
- A Pandas Series: a one-dimensional labeled array capable of holding any

data type with axis labels or index. An example of a Series object is one

column from a Data Frame.

- A NumPy ndarray, which can be a record or structured
 - A two-dimensional ndarray
 - Dictionaries of one-dimensional ndarray's, lists, dictionaries or Series.
- Some of the key features of Python Pandas are as follows:
It provides Data Frame objects with default and customized indexing which is very fast and efficient.

- There are tools available for loading data of different file formats into in-memory data objects.
- It is easy to perform data alignment and integrated handling of missing data in Python Pandas.
- It is very simple to perform pivoting and reshaping of data sets in Pandas.
- It also provides indexing, label-based slicing, and sub-setting of large data sets.
- We can easily insert and delete columns from a data structure.
- Data aggregation and transformations can be done using group by.
- High-performance merging and joining of data can be done using Pandas.
- It also provides time series functionality.
- Inserting and deleting columns in data structures.
- Merging and joining data sets.
- Reshaping and pivoting data sets.
- Aligning data and dealing with missing data.
- Manipulating data using integrated indexing for Data-Frame objects.
- Performing split-apply-combine on data sets using the group by engine.
- Manipulating high-dimensional data in a data structure with a lower dimension using hierarchical axis indexing.
- Sub-setting, fancy indexing, and label-based slicing data sets that are large

in size.

- Generating data range, converting frequency, date shifting, lagging, and other time-series functionality.
- Reading from files with CSV, XLSX, TXT, among other formats.
- Arranging data in an order ascending or descending.
- Filtering data around a condition.
- Analyzing time series.
- Iterating over a data set.

3.10 Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB, you might find this tutorial useful to get started with Numpy.

A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

NumPy is, just like SciPy, Scikit-Learn, Pandas, etc. one of the packages that you just can't miss when you're learning data science, mainly because this library provides you with an array data structure that holds some benefits over Python lists, such as: being more compact, faster access in reading and writing items, being more convenient and more efficient.

NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices. To work with these arrays, there's a vast amount of high-level mathematical functions operate on these matrices and arrays.

An array is basically nothing but pointers. It's a combination of a memory address, a data type, a shape, and strides:

- The data pointer indicates the memory address of the first byte in the array,
- The data type or dtype pointer describes the kind of elements that are contained within the array,
- The shape indicates the shape of the array, and
- The strides are the number of bytes that should be skipped in memory to go to the next element. If your strides are (10,1), you need to proceed one byte to get to the next column and 10 bytes to locate the next row.

Or, in other words, an array contains information about the raw data, how to locate an element and how to interpret an element.

With NumPy, we work with multidimensional arrays. We'll dive into all of the possible types of multidimensional arrays later on, but for now, we'll focus on 2-dimensional arrays. A 2-dimensional array is also known as a matrix, and is something you should be familiar with. In fact, it's just a different way of thinking about a list of lists. A matrix has rows and columns. By specifying a row number and a column number, we're able to extract an element from a matrix.

We can create a NumPy array using the `numpy.array` function. If we pass in a list of lists, it will automatically create a NumPy array with the same number of rows and columns. Because we want all of the elements in the array to be float elements for easy computation, we'll leave off the header row, which contains strings. One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

NumPy has several advantages over using core Python mathematical functions, a few of which are outlined here:

1. NumPy is extremely fast when compared to core Python thanks to its heavy use of C extensions.
2. Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library. Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.

- NumPy comes with a variety of built-in functionalities, which in core Python

would take a fair bit of custom code.

3.11 FRONTEND TECHNOLOGIES

- Web Technologies
- Bootstrap

3.12 ADVANTAGES OF PROPOSED ALGORITHM

- Gain more value from streaming data ingest with Kafka.
- Support DB streaming by turning databases into live feeds.
- Accelerating delivery of data to enable real-time analytics.

CHAPTER 4

RESULT AND DISCUSSIONS

Big data analysis of industry is considered as a necessary aspect for further improvement in order to improve the profit margin of industrial production and operation, and represents the next frontier of innovation, competition and productivity. Nowadays, industrial data platform is the core component of industrial data storage, computation and analysis for the management of intelligent plant. With the increasing number of intelligent equipments used in intelligent plant, however, intelligent plant can acquire a large quantity of data of Radio Frequency Identification (RFID) and intelligent equipments, thus providing rich data sets for manufacturing industry. The current trend in industrial systems is to use different big data engines as a means of processing a large quantity of data that cannot be processed by ordinary infrastructures. Industrial infrastructure faces a large number of problems, including challenges such as defining different efficient architecture settings for different applications and defining specific models for industrial analysis.

In recent years, Spark and Flink have been well studied and applied on the Internet. The MapReduce is a computing framework used by most industrial enterprises at present. Iterative computing is involved in many aspects of industrial data analysis, which is used to seek optimal control and management solutions. However, the Map Reduce framework is ineffective in iterative computing and the performance of the Spark framework has some obvious advantages over the MapReduce. There have been many hot issues and difficult problems in the research of industrial big data platforms: reducing the data processing time and the size of space for big data acquisition and storage, optimizing the performance of computing framework, providing a wealth of function, improving system stability. Our motivation for this study is to look for an efficient plan for acquiring and processing industrial data.

Moreover, we need to find compression and serialization method which have good perform and the space of data storing. Furthermore, we aim at designing an industrial data platform with higher real-time performance and higher compression ratio. Based on the above considerations, we design and implement an industrial data platform, which integrated both LZ4 and Protobuf method for data processing. The LZ4 is used for data compression, and the Protobuf is used for data serialization. Based on Flume and Sqoop, we build up a data acquisition module in the industrial data platform. Meanwhile, the Hadoop is adopted to store data acquired by intelligent equipments in intelligent plant. In order to enhance the data analysis capabilities of the industrial data platform, both the Spark and the Flink are integrated into the computing system. In addition, the industrial data platform adopts Django as front-end framework for the sake of system stability and system maintenance management. The structure of this paper is as follows. In Section I, we introduced the motivation of this research. In Section II, we introduce the relevant background. We build the data platform for industrial data analysis based on big data in Section III. Then, we show some results of the acquisition and processing system in Section IV. In Section V, we discuss the conclusions of this research.

4.1 DATA SET

3]:

Unnamed: 0		Data	Countries	Local	Industry Sector	Accident Level	Potential Accident Level	Genre	Employee or Third Party	Critical Risk	Description
0	0	2016-01-01 00:00:00	Country_01	Local_01	Mining	I	IV	Male	Third Party	Pressed	While removing the drill rod of the Jumbo 08 f...
1	1	2016-01-02 00:00:00	Country_02	Local_02	Mining	I	IV	Male	Employee	Pressurized Systems	During the activation of a sodium sulphide pum...
2	2	2016-01-06 00:00:00	Country_01	Local_03	Mining	I	III	Male	Third Party (Remote)	Manual Tools	In the sub-station MILPO located at level +170...
3	3	2016-01-08 00:00:00	Country_01	Local_04	Mining	I	I	Male	Third Party	Others	Being 9:45 am. approximately in the Nv. 1880 C...
4	4	2016-01-10 00:00:00	Country_01	Local_04	Mining	IV	IV	Male	Third Party	Others	Approximately at 11:45 a.m. in circumstances t...

Fig: 4.1 Dataset collected for presenting the data

Above dataset is a collection of information related to different industries which is taken to analyze different type of accident and try to minimize the accident rate by taking the reading into consideration. Information of different industries also have different sections by which they can be separated.

4.2 MODIFIED TABLE

	Data	Countries	Local	Industry Sector	Accident Level	Potential Accident Level	Genre	Employee or Third Party	Critical Risk	Description
0	2016-01-01 00:00:00	Country_01	Local_01	Mining	I	IV	Male	Third Party	Pressed	While removing the drill rod of the Jumbo 08 f...
1	2016-01-02 00:00:00	Country_02	Local_02	Mining	I	IV	Male	Employee	Pressurized Systems	During the activation of a sodium sulphide pum...
2	2016-01-06 00:00:00	Country_01	Local_03	Mining	I	III	Male	Third Party (Remote)	Manual Tools	In the sub-station MILPO located at level +170...
3	2016-01-08 00:00:00	Country_01	Local_04	Mining	I	I	Male	Third Party	Others	Being 9:45 am. approximately in the Nv. 1880 C...
4	2016-01-10 00:00:00	Country_01	Local_04	Mining	IV	IV	Male	Third Party	Others	Approximately at 11:45 a.m. in circumstances t...

Fig: 4.2 Modified table using data in above table

The dataset in the above table is the modified one from the previous table used for analyzing the data perfectly by removing the unwanted data in the table. The dataset is analyzed by taking different type of graphs.

4.3 GRAPH PRESENTATION

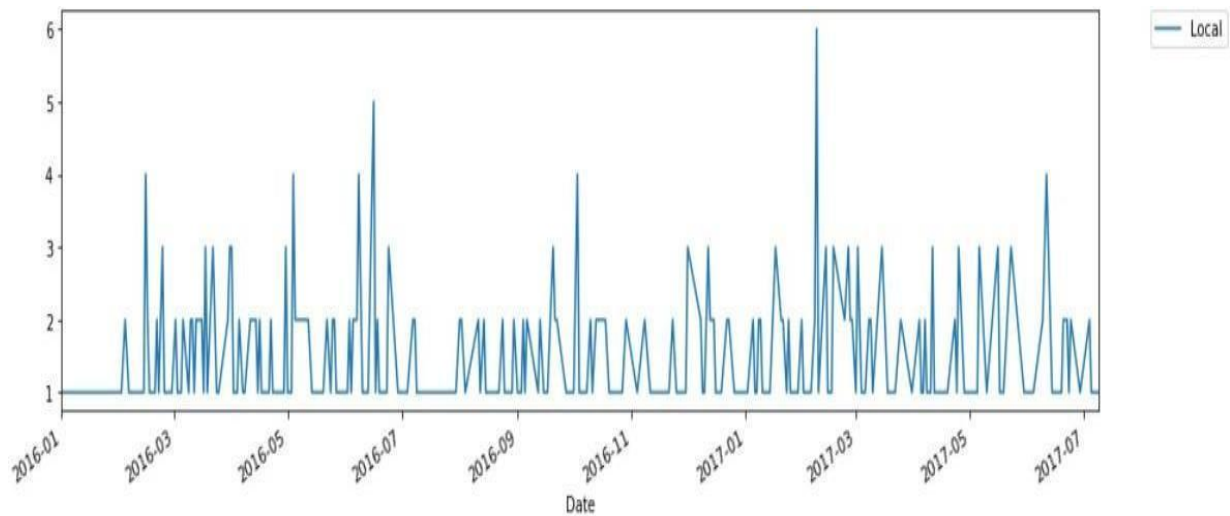


Fig: 4.3 Graph presentation for above dataset

Above graph is drawn by using the table data for different years. The data consists of different industry are plotted and used for analyzing the accident rate that take place in different years.

4.4 BAR GRAPH

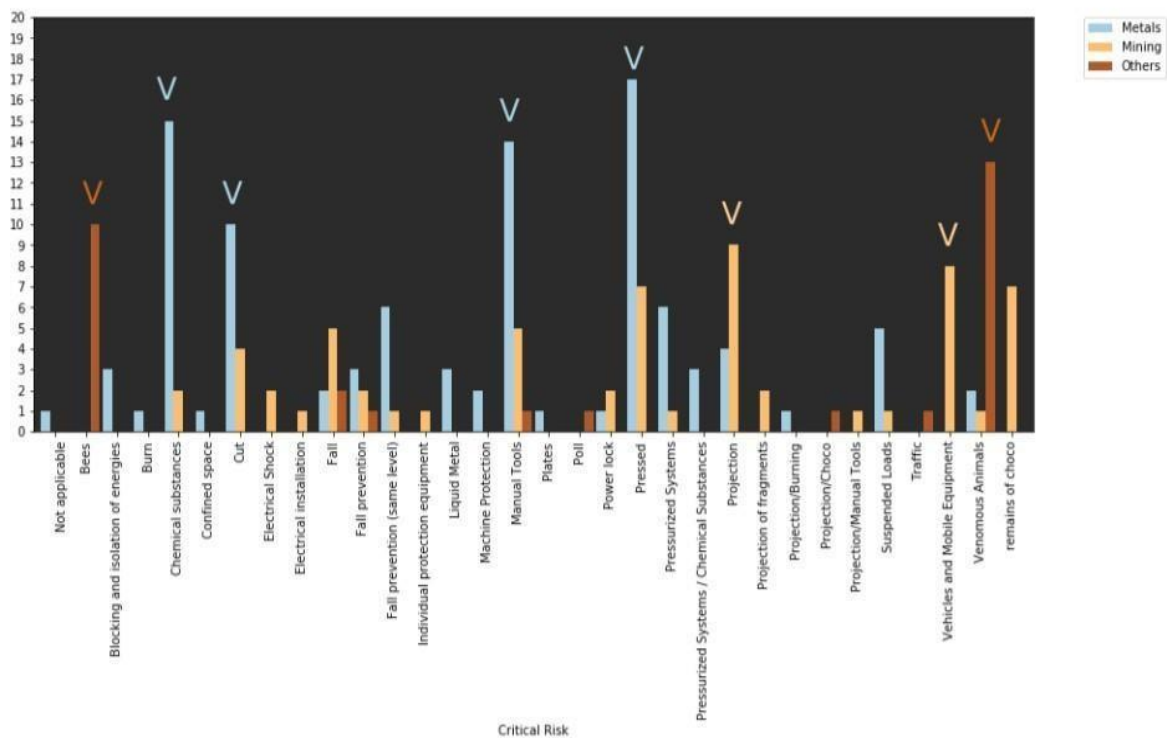


Fig: 4.4 Bar graph for different type of critical risk

The graph shows different type of critical risks faced in the industries using the bar graph. Risk in the industries are faced due to some of this factors in the industries. By analyzing the problems various precautions are taking to prevent this not happen in regular interval bases.

4.5 ACCIDENT IN INDUSTRIES

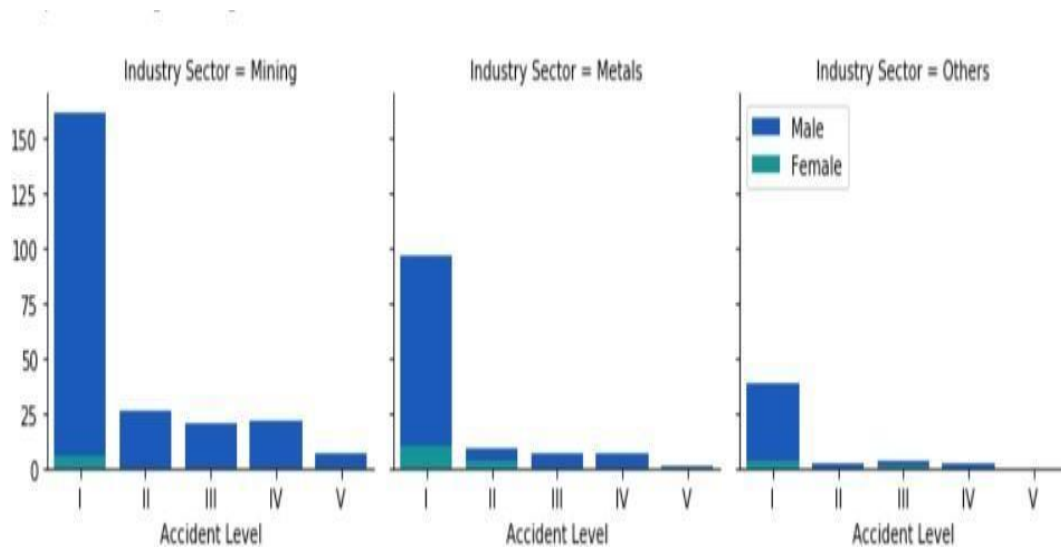


Fig: 4.5 Accident in industries

The graph consists of accident in industries in different levels. This accident values are differentiated using male/female by taking different value into consideration. By using this information various measurement can be taken to prevent this accidents.

CHAPTER CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In conclusion, there are mainly two stages in big data storage process. First to acquire, preprocess and analyses the big data. This model is abstract in the sense that it has not been designed assuming any particular scenario nor application, but can be easily adapted to address all requirements and constraints of any specific scenario in the fields of big and complex data management. We have also illustrated the concept of big data value chain along with its four main stages like data generation, data acquisition, data storage and data analysis. During storing and generating the data some variations may be possible, i.e., whether data is in video, audio, images, etc. For easy understanding of big data analytics, researchers have divided data into various big data applications like web analytics, text analytics, etc. Finally, to improve efficiency of government sectors, industries, there is an urgent need for advanced data management, analysis and acquisition mechanisms on typical big data applications to generate profit.

Right now, will in general start with the technique for business preparing and along these lines the interest of business data investigation. High comfort lupus keep framework is made to dodge single reason disappointment. Inside the technique for building lupus keep framework, the presentation of LZ0 gave with the structure is inferior to LZ4 inside the time and consequently the size, and along these lines the time taken by LZ4 is scarcely a year of that taken by LZ0. The size of information compacted by LZ4 is ninety six of which is packed by LZ0. We look at the best distributing ways furnished by enormous data system with various superior approaches to enhance the overall structure's method. Contrasted and the contrary distributing ways coordinated by Hadoop and Spark, Protobuf performs ever more elevated in preparing in light of the fact that the assortment of serialized items will increment. Contrasted and Java defaults, Protobuf takes exclusively seven membered of the opportunity to strategy data. At same time, our information stage has a few information investigation works just as AI, Finite part Analysis, advancement and information Mapping.

5.2 FUTURE WORK

Recently, there are a number of studies on data acquisition and optimization of computing framework methods, and most of these methods provided good performance in the field of Internet. We consider the characteristic of real-time and stability in industrial production in our research. Therefore, Data platform optimization includes data acquisition, data compression, data serialization, and system stability. With the development of intelligent plant, the data acquisition volume of plants increases rapidly. Flume can meet the needs of data acquisition in plants. In the research of the method of data stored, we compress data by LZ4 and serialize data by Protobuf. Compared with the original method of Hadoop integration, the time of processing data is greatly reduced and the processing effect is better. Both Spark and Flink are used in computing framework. According to the size of sliding window, we adopt appropriate window mechanism to compute the data for certain time periods. Furthermore, the selection method is given.

REFERENCES

- [1] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing," *Knowl.-Based Syst.*, vol. 79, pp. 3 17, May 2015.
- [2] R. Y. Zhong, C.Xu, C. Chen, and G. Q. Huang, "Big data analytics for physical Internet-based intelligent manufacturing shop doors," *Int. J. Prod. Res.*, vol. 55, no. 9, pp. 2610 2621, 2017.
- [3] P. Lade, R. Ghosh, and S. Srinivasan, "Manufacturing analytics and industrial Internet of Things," *IEEE Intel. Syst.*, vol. 32, no. 3, pp. 74 79, May/Jun. 2017.
- [4] X. Yi, F. Liu, J. Liu, and H. Jin, "Building a network highway for big data: Architecture and challenges," *IEEE Network.*, vol. 28, no. 4, pp. 5 13, Jul. 2014
- [5] B. Meredig, "Industrial materials informatics: Analyzing large-scale data to solve applied problems in R&D, manufacturing, and supply chain," *Current Opinion Solid State Mater. Sci.*, vol. 21, no. 3, pp. 159 166, 2017.
- [6] P.Basanta-Val, "An efficient industrial big-data engine," *IEEE Trans. Ind. Information.*, vol. 14, no. 4, pp. 1361 1369, Apr. 2018.
- [7] X. He, K. Wang, H. Huang, and B. Liu, "QoE-driven big data architecture for smart city," *IEEE Communication. Mag.*, vol. 56, no. 2, pp. 88 93, Feb. 2018.
- [8] D. Cheng, X. Zhou, P. Lama, J. Wu, and C. Jiang, "Cross-platform resource scheduling for spark and map reduce on YARN," *IEEE Trans. Computer.*, vol. 66, no. 8, pp. 1341 1353, Aug. 2017.
- [9] X. Yan, J. Zhang, Y. Xun, and Q. Xiao, "A parallel algorithm for mining constrained frequent patterns using Map Reduce," *Soft Computer.*, vol. 21, no. 9, pp.2237 2249, 2017.
- [10]. H. Huang, S. Guo, and K. Wang, "Envisioned wireless big data storage for low-earth-orbit satellite-based cloud," *IEEE Wireless Communication.*, vol. 25, no. 1, pp. 26 31, Feb. 2018.
- [11] M.Shahabinejad, M.Khabbazzian, and M.Ardakani, "On the average locality of

locally repairable codes,"*IEEE Trans. Commun.* vol.66,no.7, pp. 2773–2783, Jul. 2018.

- [12] A.P.Mukherjee and S.Tirthapura, "Enumerating maximal bicliques from a large graph using Map Reduce," *IEEE Trans. Services Computer.*, vol. 10, no. 5, pp. 771–784, Sep./Oct. 2017.
- [13] N. Kari, K. Latif, Z. Anwar, S. Khan, and A. Hayat, "Storage schema and ontology-independent SPARQL to Hive-QL translation," *J.Super Computer.*, vol. 71, no. 7, pp. 2694–2719, 2015.
- [14] M. Maas, K. Asanovic, T. Harris, and J. Kubiawicz, "Taurus: A Holistic language runtime system for coordinating distributed managed-language applications," *Acme SIGPLAN Notices*, vol. 51, no. 4, pp. 457–471, 2016.
- [15] D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "A comparison on scalability for batch big data processing on Apache Spark and Apache Flink," *Big Data Anal.*, vol. 2, no. 1, pp. 1–11, 2017.
- [16] E. A. Lee, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC)*, 2008 11th IEEE International Symposium on, 2008(363-369).
- [17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communication. Surv. Tutor.*, vol. 17, no. 4, (2015)2347-2376.

APPENDIX

A] PROJECT SCREENSHOT

```
Anaconda Prompt (data) - jupyter notebook
(base) C:\Users\Admin>cd D:\myProject\Big Data Acquisition
(base) C:\Users\Admin>:

(base) D:\myProject\Big Data Acquisition>jupyter notebook
[I 11:48:01.248 NotebookApp] JupyterLab extension loaded from C:\Anaconda\Python\data\lib\site-packages\jupyterlab
[I 11:48:01.248 NotebookApp] JupyterLab application directory is C:\Anaconda\Python\data\share\jupyterlab
[I 11:48:01.256 NotebookApp] Serving notebooks from local directory: D:\myProject\Big Data Acquisition
[I 11:48:01.258 NotebookApp] The Jupyter Notebook is running at:
[I 11:48:01.258 NotebookApp] http://localhost:8888/?token=ca380f34c8e1a7f011a2a6df4e7d44a21dec7574eaa5606
[I 11:48:01.258 NotebookApp] or http://127.0.0.1:8888/?token=ca380f34c8e1a7f011a2a6df4e7d44a21dec7574eaa5606
[I 11:48:01.259 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:48:01.349 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Admin/AppData/Roaming/jupyter/runtime/nbserver-7656-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=ca380f34c8e1a7f011a2a6df4e7d44a21dec7574eaa5606
or http://127.0.0.1:8888/?token=ca380f34c8e1a7f011a2a6df4e7d44a21dec7574eaa5606
[I 11:48:17.114 NotebookApp] Kernel started: 17b2c622-9a6f-4118-b2ad-63c760651bd6
[I 11:48:21.635 NotebookApp] Starting buffering for 17b2c622-9a6f-4118-b2ad-63c760651bd6:b8c991d098ca40b492b9f01181a26638
[I 11:48:22.050 NotebookApp] Kernel restarted: 17b2c622-9a6f-4118-b2ad-63c760651bd6
[I 11:48:23.120 NotebookApp] Restoring connection for 17b2c622-9a6f-4118-b2ad-63c760651bd6:b8c991d098ca40b492b9f01181a26638
[I 11:48:23.120 NotebookApp] Replying 6 buffered messages
[I 11:49:05.265 NotebookApp] Kernel started: 4fc8cb94-259c-43e5-8a62-6881142907d6
[I 11:49:18.358 NotebookApp] Starting buffering for 4fc8cb94-259c-43e5-8a62-6881142907d6:e23ac8409fab46a3a304dd60dfa919af
[I 11:49:18.780 NotebookApp] Kernel restarted: 4fc8cb94-259c-43e5-8a62-6881142907d6
[I 11:49:19.868 NotebookApp] Restoring connection for 4fc8cb94-259c-43e5-8a62-6881142907d6:e23ac8409fab46a3a304dd60dfa919af
[I 11:49:19.861 NotebookApp] Replying 6 buffered messages
[I 11:50:17.336 NotebookApp] Saving file at /BigDataAcquisitionForIndustrialData.ipynb
[I 11:51:05.361 NotebookApp] Saving file at /BigDataAcquisitionForIndustrialData - Part2.ipynb
```

B] SOURCE CODE

```
umentation.pc | Home Page - Select or creat | BigDataAcquisitionForIndustrialData | x + v
localhost:8888/notebooks/BigDataAcquisitionForIndustrialData.ipynb
jupyter BigDataAcquisitionForIndustrialData Last Checkpoint: 01/22/2020 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from pandas import DataFrame, Series
%matplotlib inline
import calendar

In [ ]: M dataset=pd.read_csv('Dataset.csv')

In [ ]: M dataset.head()

In [ ]: M
dataset.size

In [ ]: M dataset.isnull().sum()

There are no null values but we need to do some clean up:

In [ ]: M del dataset['Unnamed: 0']

In [ ]: M dataset.head()

In [ ]: M dataset.rename(columns={'Data': 'Date', 'Genre': 'Gender', 'Employee or Third Party': 'Employee type'}, inplace=True)

In [ ]: M dataset.Date.max()

In [ ]: M dataset.Date.min()
```

```
jupyter BigDataAcquisitionForIndustrialData Last Checkpoint: 01/22/2020 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: ▶ dset.rename(columns={'Data': 'Date', 'Genre': 'Gender', 'Employee or Third Party': 'Employee type'}, inplace=True)
In [ ]: ▶ dset.Date.max()
In [ ]: ▶ dset.Date.min()
In [ ]: ▶ dset['Date'] = pd.to_datetime(dset['Date'])
In [ ]: ▶ month_order={
    'January':1,
    'February':2,
    'March':3,
    'April':4,
    'May':5,
    'June':6,
    'July':7,
    'August':8,
    'September':9,
    'October':10,
    'November':11,
    'December':12
}
In [ ]: ▶ dset.groupby('Date').count()['Local'].plot(figsize=(15,4))
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
In [ ]: ▶ def month_n(a):
    b=calendar.month_name[a]
    return b
In [ ]: ▶ x=[]
    for i in range(0, len(dset.Date)):
```

```
In [ ]: ▶ dset_trend[2016].loc[month_order].plot(kind='bar', figsize=(15,4), width=0.9, cmap='cool', title='2016')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
dset_trend[2017].loc[month_order].plot(kind='bar', figsize=(15,4), width=0.9, cmap='hot', title='2017')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

In [ ]: ▶ order={'I':1, 'II':2, 'III':3, 'IV':4, 'V':5}

In [ ]: ▶ fig=sns.FacetGrid(dset,aspect=1.2,palette="winter", hue='Gender',col='Industry Sector', legend_out=True)
fig.map(sns.countplot, 'Accident Level', order=order)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

In [ ]: ▶ sns.countplot('Employee type',data=dset,palette='cool' )
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

In [ ]: ▶ dset['Potential Accident Level'].unique()

In [ ]: ▶ order2={'I': 1, 'II': 2, 'III': 3, 'IV': 4, 'V': 5, 'VI': 6}

In [ ]: ▶ q=Series()
    for i in range(0, len(dset)):
        q.q.append(Series(order2[dset.loc[i]['Accident Level']]*order2[dset.loc[i]['Potential Accident Level']], ignore_index=True))

In [ ]: ▶ dset['Accident Impact']=q

In [ ]: ▶ dset.head(5)
```

```
jupyter BigDataAcquisitionForIndustrialData Last Checkpoint: 01/22/2020 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: wd=[]
        for i in range(len(dset)):
            wd.append(dset['Date'].loc[i].weekday())

In [ ]: weekday={'0':'Monday',
                '1':'Tuesday',
                '2':'Wednesday',
                '3':'Thursday',
                '4':'Friday',
                '5':'Saturday',
                '6':'Sunday'}

In [ ]: wd_order={'Monday':1,
                 'Tuesday':2,
                 'Wednesday':3,
                 'Thursday':4,
                 'Friday':5,
                 'Saturday':6,
                 'Sunday':7}

In [ ]: wwd=[]
        for i in wd:
            wwd.append(weekday[str(i)])

In [ ]: dset['weekday']=wwd

In [ ]: week_d=dset.pivot_table(index='weekday', columns='Industry Sector', aggfunc='count')['Accident Level']

In [ ]: week_d.loc[wd_order].plot(figsize=(10,4), xticks=range(7), cmap='Dark2', kind='line')
        plt.ylabel('number of Accident')

In [ ]: dset.groupby('weekday').count()['month'].loc[wd_order].plot(kind='line', figsize=(9,4), xticks=range(0,8), color='#FF6A6A')
```

```
jupyter BigDataAcquisitionForIndustrialData Last Checkpoint: 01/22/2020 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: dset_cr.nlargest(6, 'total').style.background_gradient(cmap='winter')

In [ ]: dset_ind_risk=dset.pivot_table(index='Critical Risk', columns='Industry Sector', aggfunc='count')['Accident Level']
        dset_ind_risk.drop('Others', axis=0, inplace=True)
        dset_ind_risk.replace(n, 0, inplace=True)
        dset_ind_risk['total']=dset_ind_risk.sum(axis=1)
        dset_ind_risk

In [ ]: dset_ind_risk.nlargest(6, 'total').plot(kind='bar', xticks=range(30), figsize=(15,5), cmap='summer')
        plt.xticks(rotation=90)

In [ ]: dset_ind_risk_nt=dset_ind_risk.drop('total', axis=1)

In [ ]: fig=dset_ind_risk_nt.plot(kind='bar', xticks=range(30), yticks=range(0,21), figsize=(15,6), cmap='Paired', width=0.9)
        fig.set_facecolor('#2B2B2B')
        plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
        plt.text(x=0.9, y=11, s='V', color='#CD661D', fontsize=25)
        plt.text(x=3.3, y=16, s='V', color='#ADD8E6', fontsize=25)
        plt.text(x=5.4, y=11, s='V', color='#ADD8E6', fontsize=25)
        plt.text(x=14.4, y=15, s='V', color='#ADD8E6', fontsize=25)
        plt.text(x=18.4, y=17.5, s='V', color='#ADD8E6', fontsize=25)
        plt.text(x=21.6, y=10, s='V', color='#FFD39B', fontsize=25)
        plt.text(x=28.6, y=9, s='V', color='#FFD39B', fontsize=25)
        plt.text(x=30, y=14, s='V', color='#CD661D', fontsize=25)

In [ ]: dset_risk_other=dset.loc[dset['Critical Risk']=='Others']

In [ ]: sns.countplot('Employee type', data=dset_risk_other)

In [ ]: sns.distplot(dset_risk_other['Accident Impact'])

In [ ]:
```

jupyterBigDataAcquisitionForIndustrialData - Part2Last Checkpoint: 03/04/2020 (unsaved changes)

FileEditViewInsertCellKernelWidgetsHelp

Python 3

RunCode

```
In [ ]: !pip install neologdn

In [ ]: import numpy as np

In [ ]: import pandas as pd

In [ ]: import seaborn as sns

In [ ]: import matplotlib.pyplot as plt

In [ ]: from sklearn.ensemble import RandomForestClassifier

In [ ]: from sklearn.model_selection import train_test_split

In [ ]: from sklearn.preprocessing import LabelEncoder

In [ ]: sklearn.metrics import accuracy_score, f1_score, roc_auc_score

In [ ]: import os

In [ ]: from sklearn.feature_extraction.text import TfidfVectorizer

In [ ]: from nltk.corpus import stopwords

In [ ]: from nltk.tokenize import word_tokenize
```

jupyterBigDataAcquisitionForIndustrialData - Part2Last Checkpoint: 03/04/2020 (unsaved changes)

FileEditViewInsertCellKernelWidgetsHelp

Python 3

RunCode

```
In [ ]: import neologdn

In [ ]: data = pd.read_csv("Dataset.csv")

In [ ]: data.drop("Unnamed: 0", axis=1, inplace=True)

In [ ]: data.rename(columns={'Data': 'Date', 'Genre': 'Gender', 'Employee or Third Party': 'Employee type'}, inplace=True)

In [ ]: print(data.columns)

In [ ]: data.head()

In [ ]: data.Date = pd.to_datetime(data.Date)

In [ ]: year_feature = []

In [ ]: month_feature = []

In [ ]: day_feature = []

In [ ]: weekday_feature = []

In [ ]: for i in range(0, len(data.Date)):
    _year, _month, _day, _weekday = data.Date.iloc[i].year, data.Date.iloc[i].month, data.Date.iloc[i].day, data.Date.iloc[i].weekday
    year_feature.append(_year)
    month_feature.append(_month)
    day_feature.append(_day)
    weekday_feature.append(_weekday)
```

Jupyter BigDataAcquisitionForIndustrialData - Part2 Last Checkpoint: 03/04/2020 (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [ ]: day_feature = []

In [ ]: weekday_feature = []

In [ ]: for i in range(0, len(data.Date)):
    _year, _month, _day, _weekday = data.Date.iloc[i].year, data.Date.iloc[i].month, data.Date.iloc[i].day, data.Date.iloc[i].weekday
    year_feature.append(_year)
    month_feature.append(_month)
    day_feature.append(_day)
    weekday_feature.append(_weekday)

In [ ]: data['Year'] = year_feature

In [ ]: data['Month'] = month_feature

In [ ]: data['Day'] = day_feature

In [ ]: data['Weekday'] = weekday_feature

In [ ]: data.head()

In [ ]: stop_words = stopwords.words('english')

In [ ]: stop_words.extend(["cm", "kg", "mr", "wa", "nv", "ore", "da", "pm", "am", "cx"])

In [ ]: print(stop_words)

In [ ]: def text_preprocessor(document):
    #convert all characters to lowercase
    lowered = document.lower()
```

Jupyter BigDataAcquisitionForIndustrialData - Part2 Last Checkpoint: 03/04/2020 (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [ ]: data['Weekday'] = weekday_feature

In [ ]: data.head()

In [ ]: stop_words = stopwords.words('english')

In [ ]: stop_words.extend(["cm", "kg", "mr", "wa", "nv", "ore", "da", "pm", "am", "cx"])

In [ ]: print(stop_words)

In [ ]: def text_preprocessor(document):
    #convert all characters to lowercase
    lowered = document.lower()
    normalized = neologdn.normalize(lowered)
    tokenized = word_tokenize(normalized)
    wnl = WordNetLemmatizer()
    lemmas = list(map(wnl.lemmatize, tokenized))

    alphas = [w for w in lemmas if w.isalpha()]
    # remove stopwords
    removed_stops = [r for r in alphas if r not in stop_words]
    return removed_stops

In [ ]: docs = np.array(data["Description"])
docs_without_stopwords = []
for doc in docs:
    doc_without_stopwords = " ".join(text_preprocessor(doc))
    docs_without_stopwords.append(doc_without_stopwords)
vec_tfidf = TfidfVectorizer(max_df=1.0, min_df=1, max_features=100, norm='l2', stop_words='english', lowercase=True, use_idf=True)
X = vec_tfidf.fit_transform(docs_without_stopwords)

print('Vocabulary size: {}'.format(len(vec_tfidf.vocabulary_)))
```

```
documentation.pc | Home Page - Select or crea | BigDataAcquisitionForIndu | BigDataAcquisitionForI x + v
localhost:8888/notebooks/BigDataAcquisitionForIndustrialData%20-%20Part2.ipynb
jupyter BigDataAcquisitionForIndustrialData - Part2 Last Checkpoint: 03/04/2020 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: tfs = pd.DataFrame(x.toarray(), columns=["TFIDF_" + n for n in vec_tfidf.get_feature_names()])
feature_df = pd.concat([data, tfs], axis=1)
feature_df.head()

In [ ]: target_columns = ["Countries", "Local", "Industry Sector", "Accident Level", "Potential Accident Level", "Gender", "Employee
for i in range(0, len(feature_df.columns)):
    if feature_df.columns[i] in target_columns:
        print("##### " + data.columns[i] + " #####")
        print(data.iloc[:,i].value_counts())

In [ ]: data.pivot_table(index='Month', columns=['Year', 'Accident Level'], aggfunc='count')['Countries'].head()

In [ ]: data_trend = data.pivot_table(index='Month', columns=['Year', 'Accident Level'], aggfunc='count').replace(np.nan,0)
data_trend["Countries"].head()

In [ ]: data_trend["Countries"][2016].plot(kind='bar', figsize=(15,4), cmap='cool', title='Counts by Accident Level in 2016')
plt.legend()

In [ ]: data_trend["Countries"][2017].plot(kind='bar', figsize=(15,4), cmap='cool', title='Counts by Accident Level in 2017')
plt.legend()

In [ ]: fig=sns.FacetGrid(data, aspect=1.2, palette="winter", hue='Gender', col='Industry Sector', legend_out=True)
fig.map(sns.countplot, 'Accident Level', order={'I':1, 'II':2, 'III':3, 'IV':4, 'V':5})
plt.legend()

In [ ]: model_features = ["Local", "Accident Level", "Gender", "Employee type", "Critical Risk", "Month", "Weekday", "TFIDF_assistant", "TFID
"TFIDF_face", "TFIDF_fall", "TFIDF_finger", "TFIDF_floor", "TFIDF_foot", "TFIDF_glove", "TFIDF_hand", "TFIDF_heigh
"TFIDF_weight", "TFIDF_drill", "TFIDF_cleaning", "TFIDF_arm"]
extracted_feature_df = feature_df[model_features]
extracted_feature_df.head()
```

```
jupyter BigDataAcquisitionForIndustrialData - Part2 Last Checkpoint: 03/04/2020 (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [ ]: data_trend["Countries"][2017].plot(kind='bar', figsize=(15,4), cmap='cool', title='Counts by Accident Level in 2017')
plt.legend()

In [ ]: fig=sns.FacetGrid(data, aspect=1.2, palette="winter", hue='Gender', col='Industry Sector', legend_out=True)
fig.map(sns.countplot, 'Accident Level', order={'I':1, 'II':2, 'III':3, 'IV':4, 'V':5})
plt.legend()

In [ ]: model_features = ["Local", "Accident Level", "Gender", "Employee type", "Critical Risk", "Month", "Weekday", "TFIDF_assistant", "TFID
"TFIDF_face", "TFIDF_fall", "TFIDF_finger", "TFIDF_floor", "TFIDF_foot", "TFIDF_glove", "TFIDF_hand", "TFIDF_heigh
"TFIDF_weight", "TFIDF_drill", "TFIDF_cleaning", "TFIDF_arm"]
extracted_feature_df = feature_df[model_features]
extracted_feature_df.head()

In [ ]: def lb_enc(df, label):
    le = LabelEncoder()
    le.fit(df[label])
    df[label] = le.transform(df[label])
    return df

In [ ]: enc_target = ["Local", "Accident Level", "Gender", "Employee type", "Critical Risk", "Weekday"]
for t in enc_target:
    extracted_feature_df = lb_enc(extracted_feature_df, t)
    extracted_feature_df.head()

In [ ]: def make_model_and_predict(df):
    y = np.array(df['Accident Level'])
    x = df.drop('Accident Level', axis=1).values
    x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.9, random_state=123)

    model = RandomForestClassifier(n_estimators=300, random_state=0)
    model.fit(x_train, y_train)
    y_score = model.predict(x_test)
```

Improved Information Acquisition and Capacity for Designing Modern Information Stage

By- Vineeth Kumar and Pavan Kumar Guided By- Deepa D, M.E(Ph.D)

Abstract— Huge information based obtaining and capacity framework (ASS) assumes a significant job in the plan of mechanical information stage. Numerous huge information structures have been coordinated pressure and serialization strategy. These techniques can't address the issues of modern creation data the executives for requiring tedious and mass stockpiling. In light of the current large information structures, we propose an upgraded modern huge information stage so as to lessen the information preparing time while requiring less information extra room. Specifically, this paper centers around assessing the effect of numerous pressure and sequential methods on bigdata stage execution and attempts to pick ideal pressure and sequential strategies for the modern information stage. Contrasted with the techniques incorporated in Hadoop and Spark, the trial results time of the stage has been diminished by 73.9% with an under 96% the size of information stuffed, also, the information serialization time has been diminished by 80.8%. With the extending proportion of data, it puts aside reduced exertion to complexity and standard strategies.

Keywords:

Huge information, obtaining and capacity framework (ASS), Industrial information stage, Information procurement, Information stockpiling, Information pressure, calibrated.

I. INTRODUCTION

Enormous information assessment of large business is considered as a key issue for besides improvement with a reason to upgrade the edge of benefit of business creation and activity, and speaks to progressive outskirts of development, rivalry and productivity. Nowadays, mechanical information stage is that the center piece of business information stockpiling, calculation and investigation for the administration of canny plant. With the developing type of astute supplies used in savvy plant, be that as it may, knowledge plant will procure huge amount of records of oftenness Identification(RFID) and shrewd types of gear, therefore introducing well off. The main edge pattern in mechanical frameworks is to utilize totally distinctive gigantic information motors as a method for process a tremendous measure of realities that can't be prepared. Modern foundation faces a larger than usual measure of difficulties, just as difficulties along the edge of procedure unmistakable practical structure settings for one in everythingabout applications and forming explicit styles for mechanical examination. In current years, Spark and Flink are all around contemplated and regulated on the web. The Map-Reduce could be a figuring system utilized by most mechanical firms as of now. Redundant processing considers in a few components of financial information examination, that is acclimated are endeavoring to look out ideal oversee and the executives arrangements. In any case, the guide cut back structure is pointless in

redundant figuring and in this way the presentation of the Spark system has some conspicuous favors over the Map-Reduce. There had been a few hot issues and extreme difficulties inside the examination of business tremendous data stages diminishing the measurements time interim and along these lines the elements of a gigantic amount of realities that can't be handled by standard infrastructures. Industrial foundation faces an immense measure of issues, complete of difficulties related to molding outstanding efficient structure settings for thought of one among a structure projects and shaping. In in vogue days/years, flash and Flink are pleasantly considered and regulated at the web. The Map- Reduce could be a registering system utilized by most modern realities investigation, this is frequently acclimated zone unit looking for ideal control and the board arrangements. Be that as it may, the Map cut back system is pointless in dull processing and thusly the general execution of the sparkle structure envelops a couple of evident gifts over the Map cut back. There are a few hot issues and intense issues inside the investigations of business huge data stages diminishing the information time interim and in this way the measurements.

II LITERATURE REVIEW

An Efficient Industrial Big-information Engine

Current patterns in mechanical frameworks choose the work of different huge information motors as an intend to technique huge measures of data that can't be prepared with a typical infrastructure. The amount of issues A modern foundation must face is monster and incorporates difficulties like the meaning of different efficient plan arrangements for different applications, and furthermore during this unique situation, the article investigates the occasion of a medium size huge information motor (i.e.implementation) ready to improve execution in map-lessen registering by tearing the logical into entirely unexpected portions that will be handled.

by the motor in equal utilizing a hierarchic model. This sort of office decreases start to finish calculation time for all fragments with their outcomes at that point bound together with various data from various sections when their procedure in parallel. This sort of arrangement will expand execution of current groups up I/O activities amazingly as exact outcomes found.

Cross-stage Resource Scheduling for Spark and Map- Reduce on YARN

Cross-stage Resource programming for Spark and Map- Reduce on YARN Map-Reduce is innately intended for group and high outturn process remaining tasks at hand, there's Associate in Nursing expanding interest for non-clump forms on gigantic information, e.g., intelligent employments, timeframe inquiries, rising Apache Spark fills during this hole, which may run on a long-lasting Hadoop bunch and take profits of current HDFS. Accordingly, the preparing model of Spark on YARN is wide applied by a few business leaders. However, we will in general build up 3 key difficulties to convey Spark on YARN, firm reservation-based asset the executives, between task reliance dazzle programing, and thusly the area obstruction among Spark and the 3 difficulties cause wasteful asset use and indispensable execution disintegration. we will in general propose and build up a cross-stage asset programming middleware, iKaya, that plans to upgrade the asset usage and application execution in multi-occupant Spark on YARN groups. iKaya relies upon 3 key components: reservation- mindful guardian situation to keep away from long anticipating asset reservation, reliance mindful asset change in accordance with use under-used asset involved by cut back errands, and cross-platform section- mindful undertaking dole out and Map-Reduce applications. Test results on a testbed show that iKaya can do five hundredth execution advancement for Spark applications and nineteen execution advancement for Map-Reduce applications, contrasted with 2 in style Sparkon YARN preparing models, that is, YARN-customized model and YARN bunch model.

Listing Maximal Biclques from a Large Graph utilizing MapReduce

The identification of outside bipartite clubs (biclques) from a larger than usual chart, an assignment integral to a few information handling issues emerging in interpersonal organization investigation and bioinformatics. we will in general blessing novel equal calculations for the Map cut back structure, and a trial investigation abuse Hadoop Map cut back. Our algorithmic program is predicated on group the info diagram into littler sub charts, trailed by process totally extraordinary sub charts in equal. Our algorithmic program utilizes 2 ideas that adjust it to scale to enormous charts: (1) the repetition in work between totally extraordinary sub diagram investigations is decreased through a cautious pruning of the inquiry house, and (2) the heap on totally various reducers is adjusted through an errand task that is upheld a relevant all out request among the vertices. We appear on paper that our algorithmic program is figure ideal for example it performs steady absolute work as its sequential partner. we will in general blessing a top to bottom investigation that shows that the algorithmic program scales to gigantic diagrams with multitudinous edges and many countless outside biclques. To our data, this can be the essential work on outside

biclique en ment to facilitate territory rivalry between Spark umeration for

diagrams of this scale.

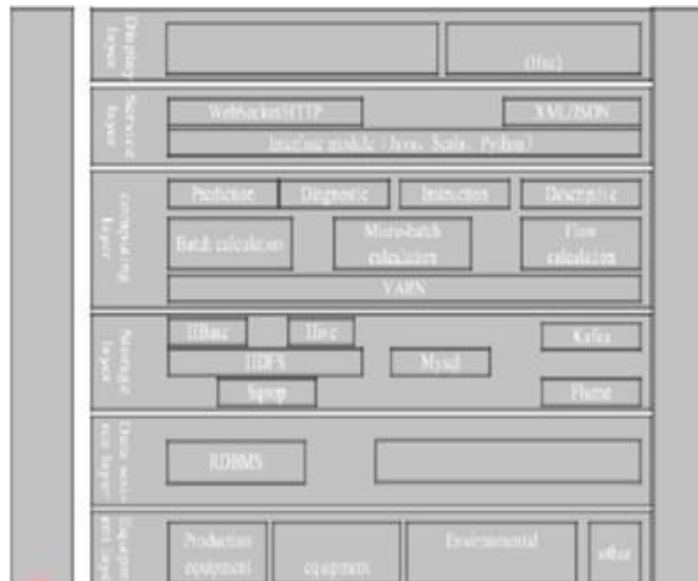
Manufacturing Analytics and Industrial Internet of Things

This paper describes the opportunities for the utilization of analytics in producing and gift a case study of a thriving application at one among Bosch's large-scale producing facilities. Building on our expertise in additional than 2 voluminous such applications, we tend to gift our recommendations for replications in alternative operations and share our most well-liked technology stack. Finally, we offer a read of future developments during this space, as well as the coevolution of technologies and solutions within the broader net of Things (IoT) house.

On the mean territory of Locally repairable Codes

A LRCs (r) are built up. the regular neighborhood of Associate in Nursing LRC (r) legitimately influences the costly fix data measure, plate I/O, and scope of hubs worried inside the fix strategy for a missing data square. there's a specialty inside the writing learning r . during this paper A direct square code with measurement k , length n , and least separation d is named a locally fixable code (LRC) with neighborhood r on the off chance that it will recover any coded picture by at the most r elective coded images. LRCs are as of late anticipated and utilized in follow in disseminated capacity frameworks (DSSs) like Windows Azure stockpiling and Facebook HDFS-RAID. Hypothetical limits on the most extreme neighborhood of, we will in general build up a limit on r of discretionary ($n; k; d$) LRCs. furthermore, we will in general get an OK limit on r for a reasonable case any place the code rate ($R = kn$) is greater than $(1 - \frac{1}{p})^2$. At long last, webdesign 3 classes of LRCs that get through the got limits on r . assessment with the present LRCs, our anticipated codes improve the basic neighborhood while not giving up such vital parameters on the grounds that the code rate or least separation. As per mechanical data procurement and procedure needs, this paper styles A modern immense data plat- structure. the general structure of the monetary data stage is appeared in Figure one. the information stage remembers 6 layers for terms of information ow. These 6 layers are gadget layer, recovery floor, stockpiling floor, figuring floor, administration floor and show layer, that relate progressively to data procurement, data stockpiling, data examination, administration bundle and

side. There are a few styles of modern data appropriation, hence the data securing elements of this stage embraces Flume and sqoop. The Sqoop obtains a lot of information hang on in relative databases inside the business, and along these lines the timeframe request of those data isn't high. furthermore, consequently the Flume secures timeframe data nonheritable by the clever types of gear. inside the business, dynamic data, similar to that of wellbeing recognition which of prepare ment activity standing, is nonheritable by Flume. at that point these information is transmitted to the message line of Kafka. The Kafka turns into the supplier of timeframe utilization of dynamic information for Structured Streaming and Spark



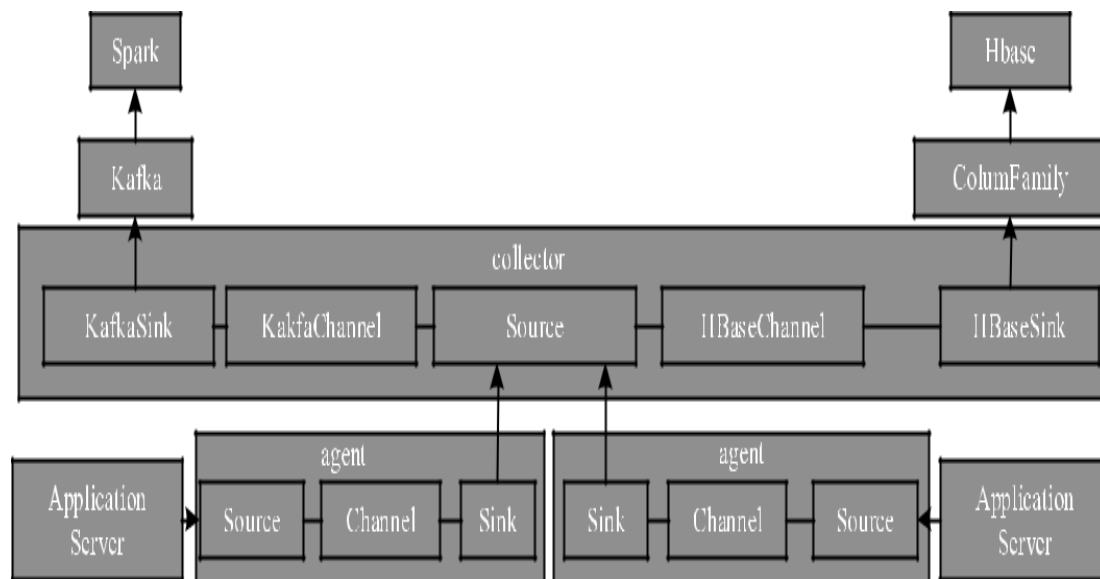
Streaming. A few supervisors of the plant must be constrained to address authentic information, along these lines the information nonheritable from Flume must be hang on in HDFS or HBase. however one more Resource treater (YARN) is embraced by the asset programming system, it bolsters all processing structures and encourages reconciliation of Flink and spark. The registering system utilizes Spark SQL for bunch processing, Spark-Streaming for miniaturized scale group process, Flink and Structured Streaming for stream process. The outcomes ar typified by net administrations inside the administration layer, at that point the exemplified data is transfered to the show floor that coordinates Echart, Hue and H5 by Django. At that point the client will get the information investigation outcome by the show alliance. Moreover, Zoo-Keeper finishes the helpful oversee ment and high handiness examples of some disseminated tremendous information systems of the full mechanical information stage.

INFORMATION ACQUISITION

The framework transmits information by 3 channels. Initially, information flow could be an unaltered figuring module comprising of Structure Streaming, Flink, Flume and Kafka. Furthermore, information flow could be an information question module comprising of HBase, Flume, SparkSQL and Table. Thirdly, information flow could be a visual investigation module comprising of Spark Streaming, Flink, Flume and HBase. The information serialized system of the securing structure is updated. The configuration obtaining subject is as per the following. The component for Flume information procurement is appeared. The data consists of Flume needs 3 sections Sink, Channel, and Source. The inventory gets the occasion from the log or server immune system malady and places it into the tunnel in mass. Information transference of Flume is part into 2 information transference tunnels, consequently you wish to line HBase Channel, Kafka Channel, HBase source and Kafka sink. the information configured of Sources needs to arrange kind, Channel, web Protocol Address (IP), Port and selector kind. The special configuration data is appeared up in Table one. The data getting system of Sqoop PC database is showed up in Fig 3 . While tolerating the shell request or Java API request from the client, the Sqoop changes over the request to the relating Map-Reduce task by task translator in Sqoop, therefore moves the information from PC database to Hadoop to finish the copy of the data. The Sqoop takes a shot at relative databases by four habits by which. early on of all, The Sqoop gets information from My-SQL database. By then Sqoop moves these data to HDFS. Next, Sqoop takes information from MySQL to Hive. Sqoop exchanges information from Hive to MySQL.

B. INFORMATION STORAGE

Hadoop is received by the information stockpiling system. every lupus stockpiling and between group correspondence on HDFS need data pressure. The previous will increase the number of commercial information keep and scale back the price. and therefore the latter will scale back the information measure consumption {of information|of knowledge|of information} transmission between networks and scale back data UTC. As we know, industrial huge information platform desires high stability.



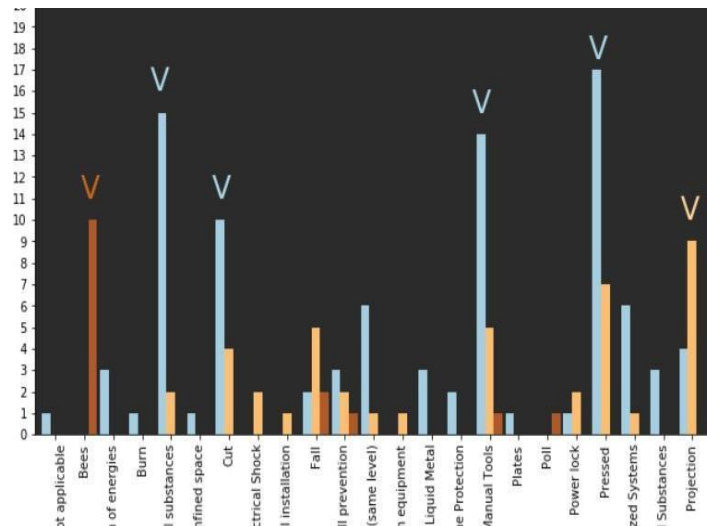
INFORMATION ACQUISITION SCHEME OF THE DATA PLATFORM BY SGOOP.

HIGH AVAILABILITY DATA STORAGE SYSTEM.

We are able to watch two Name-Node (NN) and 3 DataNode (DN) inside the information stage because of all variants of Hadoop2 exclusively bolster 2 high-accessibility modes for NN. The customer gets meta details through the NN once the informationn is kept in touch with the DN special brokenness by the data. we tend to pick LZ4 on the grounds that the information pressure approach by assessing differed ways. As we probably am aware, the information compacted time and size of the house packed should be considered all through the decision of pressure ways. mammoth plate capacity implies that high estimation of circle, contrasted and period examination of business information, notwithstanding, practicality is a ton of fundamental than the cost of circle. Besides, the practicality is one among the chief essential markers in mechanical creation, as canny plant wellbeing perception and disappointment assignment. There-fore, we tend to give that is require to the data packed time. There are more than 20 types of varies for the evaluation of distribution. Among them, numerous ways, as Proto-buf, Java default technique, Proto-stuff, jackson, Fasterxml and kryo, ar hand- picked to pass judgment on the distribution. during this paper, we tend to mostly pass judgment on the practicality of

the data serialized procedure. beside the Java default system, various ways don't have any significant differentiation inside the size of the house just as the memory house and along these lines the cabinet space. Because of the terrible showing of JDK, it's even as a simple reference. registering framework PC framework programmed information preparing framework ADP framework ADPS framework needs to help numerous sorts of information investigation and information perception benefits, that the information figuring module of the financial monstrous information stage needs to help a spread of data process ways. Thus, we tend to pass judgment on four unrivaled preparing outline works. As we probably am aware, the Spark is that the third-age computing system, whose work is more and dependable. The Flink is named the fourth-age registering system and has elite. Thinking about the common sense and in this manner the preparing execution of different registering systems, we tend to finally chosen Flink and spark. In Table 2, we will see that the Spark and in this way the Flink bolster a great deal of functions of data investigation than the contrary 2 frameworks. In this framework, if your application handles individual or delicate client data, if it's not too much trouble moreover observe the additional necessities inside the "Individual and Sensitive Information" area beneath. These Google Play necessities are moreover to any necessities as endorsed by any material security or data assurance laws making certain the insurance of the client's application. we tend to anticipated, A client UN office needs to place in Associate in Nursing the utilize any outsider application doesn't see the significance and that methods for the authorizations mentioned by an application, and along these lines only awards all the consents to all or any the applications that are out there inside the stage in this way as a consequences of that hurtful applications furthermore get put in and play out their noxious movement behind the scene. Future works include making of this methodology out there at a greater scale.

RESULTS AND SCREENSHOTS



V. PROPOSED SYSTEM

The framework transmits information by 3 channels. Right off the bat, information ow could be an unaltered processing module comprising of Flume, Kafka, Structure Streaming and Flink. Furthermore, information ow could be an information question module comprising of Spark-SQL, H-Base, Flume and Table. Thirdly, information ow could be a visual examination module comprising of Flume, H-Base, Spark Streaming and Flank..The information serialized technique of the we can see 2 Name Node (NN) and 3 Data Node (DN) inside the information stage because of all forms of Hadoop2 exclusively bolster 2 high-accessibility modes for NN. The customer gets meta information through the NN once the data is kept in touch with the DN speci brokenness by the data. we tend to pick LZ4 on the grounds that the information pressure approach by assessing fluctuated ways. As we probably am aware, the data packed time and size of the house compacted should be contemplated all through the decision of pressure ways. monster circle capacity implies that high estimation of plate, contrasted and period investigation of business information, in any case, practicality is a great deal of indispensable than the cost of circle. Moreover, the practicality is one among the principal crucial markers in modern creation, as astute plant wellbeing perception and disappointment assignment. There-fore, we will in general offer need to the information pressed time. There ar in excess of twenty kinds of ways for the evalu- ation of appropriation. Among them, various ways, as Protobuf, Kryo, Protostuff, Fasterxml, Jackson and Java default philosophy, ar hand- picked to pass judgment on the production. during this paper, we tend to mainly pass judgment on the practicality of the information serialized philosophy. beside the Java default system, various ways don't have any significant qualification inside the size of the house just as the memory house and along these lines the pantry space. because of the horrible showing of

JDK, it's even as a simple reference. registering framework PC framework programmed information handling framework ADP framework needs to help numerous sorts of information investigation and information representation benefits, that the information com-puting module of the monetary monstrous information stage needs to help a spread of data process ways. In this way, we tend to pass judgment on four predominant preparing outline works. As we probably am aware, the Spark is that the 3rd age computing structure, whose presentation is large and dependable. The Flink is named the 4th-age registering system and has elite. Thinking about the common sense and along these lines the handling execution of different processing systems, we tend to nally chosen Flink and spark. In Table 2, we will see where the Spark and along these lines the Flink bolster a ton of elements of data examination than the contrary 2 systems.

VI CONCLUSION

Right now, will in general start with the technique for business preparing and along these lines the interest of business data investigation. High comfort lupus keep framework is made to dodge single reason disappointment. inside the technique for building lupus keep framework, the presentation of LZO gave with the structure is inferior to LZ4 inside the time and consequently the size, and along these lines the time taken by LZ4 is scarcely a year of that taken by LZO. the size of information compacted by LZ4 is ninety six of which is packed by LZO. We look at the best distributing ways furnished by enormous data system with various superior approaches to enhance the overall structure's method. Contrasted and the contrary distributing ways coordinated by Hadoop and Spark, Protobuf performs ever more elevated in preparing in light of the fact that the assortment of serialized items will increment. Contrasted and

Java defaults, Protobuf takes exclusively seven membered of the opportunity to strategy data. At same time, our information stage has a few information investigation works just as AI, Finite part Analysis, advancement and information Mapping.

REFERENCES

- C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing," *Knowl.-Based Syst.*, vol. 79, pp. 3–17, May 2015.
- R. Y. Zhong, C. Xu, C. Chen, and G. Q. Huang, "Big data analytics for physical Internet-based intelligent manufacturing shop floors," *Int. J. Prod. Res.*, vol. 55, no. 9, pp. 2610–2621, 2017.
- P. Lade, R. Ghosh, and S. Srinivasan, "Manufacturing analytics and industrial Internet of Things," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 74–79, May/Jun. 2017.
- X. Yi, F. Liu, J. Liu, and H. Jin, "Building a network highway for big data: Architecture and challenges," *IEEE Netw.*, vol. 28, no. 4, pp. 5–13, Jul. 2014.
- B. Meredig, "Industrial materials informatics: Analyzing large-scale data to solve applied problems in R&D, manufacturing, and supply chain," *Current Opinion Solid State Mater. Sci.*, vol. 21, no. 3, pp. 159–166, 2017.
- P. Basanta-Val, "An efficient industrial big-data engine," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1361–1369, Apr. 2018.
- X. He, K. Wang, H. Huang, and B. Liu, "QoE-driven big data architecture for smart city," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 88–93, Feb. 2018.
- D. Cheng, X. Zhou, P. Lama, J. Wu, and C. Jiang, "Cross-platform resource scheduling for spark and mapreduce on YARN," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1341–1353, Aug. 2017.
- X. Yan, J. Zhang, Y. Xun, and Q. Xiao, "A parallel algorithm for mining constrained frequent patterns using MapReduce," *Soft Comput.*, vol. 21, no. 9, pp. 2237–2249, 2017.
- [10] H. Huang, S. Guo, and K. Wang, "Envisioned wireless big data storage for low-earth-orbit satellite-based cloud," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 26–31, Feb. 2018.
- [11] M. Shahabinejad, M. Khabbazi, and M. Ardakani, "On the average locality of locally repairable codes," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 2773–2783, Jul. 2018.
- [12] A. P. Mukherjee and S. Tirthapura, "Enumerating maximal bi-cliques from a large graph using MapReduce," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 771–784, Sep./Oct. 2017.

