



Computer Engineering and Informatics (CEI)

CST4599- Postgraduate Project Dissertation Title: Performance evaluation of Apache Hadoop and Apache Spark using Electronic Health Records Dataset and X-Ray Images dataset to detect Pneumonia.

Supervisor: Jaspreet Singh Sethi

Module Leader: Dr. Fehmida Hussain

Author: Shaik Mohammed Suhail

Date: 30/9/2022

Student ID Number: M00815369

A thesis submitted in partial fulfilment of the requirements for the degree
of Master of Science in Network Management and Cloud Computing

Performance evaluation of Apache Hadoop and Apache Spark using Electronic Health Records Dataset and X-Ray Images dataset to detect Pneumonia

MISIS

MSc Network Management and Cloud Computing

Middlesex University

Statement of originality of submitted work

I hereby confirm that the work presented here in this report and all other associated material;
is wholly our own work. And I agree to assessment for plagiarism

ABSTRACT

This thesis evaluates the performances of different Apache Spark frameworks in a healthcare environment. EHR Vendors have integrated systems that fulfill hospitals' requirements to comply with HIPAA regulations. A major hospital chain management may not afford all the applications related to the EHR vendor applications, making integration more difficult. A platform where data can be pulled from multiple storage locations must be used to make generating data reports and providing different forms of media seamless. This thesis shows that minimum configurations in cloud services such as AWS virtual machines can be used to provide the extra boost required for processing big data. This thesis contributes to the field of selecting the accurate data framework or processing engine required for the data analysis of EHR data and detecting Pneumonia using X-Rays through Transfer Learning.

TABLE OF CONTENTS

Abstract.....	2
1 Introduction	4
2 Aim	6
3 Objectives.....	6
4 Background	7
4.1.1 Different Big Data SQL Engines	8
5 Literature Review	11
5.1 Structure of EHR.....	11
5.1.1 Current Research in EHR field	11
5.2 Detection of Pneumonia by transfer learning of X-ray images.....	13
6 Related Work	16
7 Approach.....	17
7.1 Electronic Health Records Dataset.....	17
7.2 Chest X-Ray Dataset.....	18
8 Implementation	19
8.1 Transfer Learning in Apache Spark	19
8.2 Observations	19
8.2.1 Using SQL engines	21
8.2.2 COUNT.....	22
8.2.3 JOIN	23
8.2.4 ORDER	24
9 DISCUSSION.....	25
9.1.1 Data Analysis in EHR.....	25
9.1.2 Transfer Learning	25
9.2 Limitations.....	25
9.3 Future Recommendations	26
9.4 Conclusion.....	27
10 References	28
11 Appendices.....	31
12 ETHICS FORM	46

1 INTRODUCTION

The rate of patient health data is increasing exponentially. A proper data storage solution provides better data analytics with vast data available. By 2025 it is estimated that over 463 exabytes of data will be created, out of which 28 Petabytes are just from health-related wearable devices. Although other forms of data are not necessary to be stored, healthcare records cannot be disregarded in such a category. The data of each patient is necessary to analyze his/her case appropriately by a doctor or as an institution, for example, immunization or previous health disorders. Current estimates indicate that a patient generates 80 MB of health-related data annually. Moreover, it is to be noted that this data is from legacy health systems data, and with every year moving forward, there will always be new parameters in the healthcare domain to be considered among multiple patients. Healthcare data has the highest data growth business sector, with a compound annual growth rate of 36% by 2025. With this much more data, it has become much more critical to have redundancy in data apart from the databases provided by the EHR vendors. It has become more critical than ever to reduce maintenance costs on-premises. With the above increasing data, NoSQL databases are preferred over RDBMS since they are easily accessible and scalable. Since they are easier to program and manage due to their flexibility, however, they are limited features regarding security and providing technical support. File management systems present in operating systems which are file servers are not capable of expanding horizontally. At the core design, the OS file system is supposed to respond to the application-level calls and is unsuitable for data storage at the design level.(Ergüzen and ünver, 2018). Suppose data anonymity can be easily achieved and not a significant issue for performing data analytics. In that case, it can be stored in the cloud as burst storage or just for more computing power due to one of the attributes being easy and decommissioning of service within a short amount of time.(Harmony, 2020) . The aim is to use a combination of Apache Spark, Hadoop Yet Another Resource Negotiator, and Transfer Learning to enable effective recognition of X-Rays and therefore encourage the ability to train the model for detection of Pneumonia and provide an effective way for data analysis. The objectives are to evaluate if the existing EHR vendors, such as Cerner EHR data format, can be used to analyze data on the cloud.

Since Hadoop was intended for bare metal systems, there may be doubt arising if there may be any specific components that would not work as intended. However, VMware has mentioned that Hadoop distributions such as Hortonworks, IBM, MapR, and Pivotal work very well on Virtual Machines on VMs present in the vSphere environment.('Virtualizing Hadoop[®] on VMware vSphere[®] Virtualizing Hadoop on VMware vSphere', no date) Advancements in Technological devices require high-quality datasets and produce high-quality graphical output, which in turn means more storage must be provisioned. For example, Computed Radiography, Digital mammography, and Digital Radiography which a few patients go through require storage of approximately 20-30MB. The main reason why hospitals require Hadoop are the following:

- Newer technologies provide higher amounts of resolution and quality such as Digital Radiography and this takes up more space and demands for more storage.
- Utilizing more storage and, more importantly, the computing power present where machines are mostly idle can be used to the hospital's advantage.

- Almost every data collected about a patient is here to stay, and the key is to access this data as quickly as possible without any harm or damage to the infrastructure; Hadoop has tools such as HDFS and applications such as Spark and Drill that provide these features.
- When approached according to the political aspect of providing data and the current stance on using Machine Learning (ML) algorithms, U.S Government Accountability Office recently conducted a report regarding the use of Machine Learning in hospitals, and here is a summarization of the official findings:
 - There have already been technologies used to detect diseases such as Cancer, and Alzheimer's Disease using MRI scans, heart diseases using ECG, and more.
 - The technology for the detection of diseases is relatively new and has not been widely implemented yet.
 - The report ultimately stressed the importance of using ML algorithms can only help in the early detection of diseases rather than providing an assured decision about if the patient is affected or not.
 - The main aim for most emerging ML technologies was to provide a probability score of if the patient is affected by the disease or not
 - The report suggested ways to improve the Machine Learning Algorithms for diagnosing diseases is by providing more Data Access, Collaboration, and evaluation of the data.('United States Government Accountability Office Report to Congressional Requesters Artificial Intelligence in Health Care Benefits and Challenges of Machine Learning Technologies for Medical Diagnostics With content from the National Academy of Medicine, 2022)
- The current way of detecting Pneumonia is not discussed in the report mentioned above, but this opens the possibility of detecting Pneumonia using images of X-Ray.
- In most cases, the data present is extremely sensitive and vulnerable. Therefore, data redundancy and replication are an utmost priority to which Hadoop has attributes.
- Simple TensorFlow libraries present in Python are not capable of using data from multiple sources and processing it. However, Spark and Drill can pull data from multiple sources at any given time.
- A decentralized network is one of the most important features for Hadoop as one broken virtual machine slave node may not result in permanent data loss, and there can be multiple master nodes using Zookeeper, reducing the possibility.

The paper is organized in – parts which Aims, Objectives, and Background, giving the introduction on the topic so far and the research involved, followed by a discussion on which libraries available for deep learning are suitable for transfer learning. After this, the architecture of the model being used is discussed with next topic being related work in terms of this research and the approach taken. The results are discussed in accordance with graphs measured in terms of time, efficiency, and cost. Discussions, limitations, future recommendations, and a Conclusion is provided to give insight into how the research had some limitations and observations and how this research can be continued in the future.

2 AIM

With reference to the above introduction, here are the five main aims of this research

- With the increase in demand for electronic health Data and the requirement for Artificial Intelligence Assisted to be taken in terms of Pneumonia Detection, a colossal hospital environment can take advantage of the infrastructure present
- To utilize the existing infrastructure present along with integrating cloud infrastructure such as Virtual Machines to quickly provide burst storage or extra processing capabilities for tasks such as transfer learning and data analysis based on Electronic Health Record
- To integrate infrastructure which can be capable of predictive data analysis in the future with the help of cloud computing IaaS components, mainly Virtual Machines
- To integrate the existing database systems with distributed database management such as Hadoop Database File System and provide data redundancy and data replication across a coordinated cluster of virtual machines and physical infrastructure.
- To provide a suitable environment where minimal resources of compute power can be used to achieve the goals as mentioned earlier in cloud computing environments.

3 OBJECTIVES

The objective of this research is to compare the performances of two different Apache Spark environments which may or may not use YARN. YARN is integrated along with the Hadoop package and plays a crucial role in allocating resources to the applications using the cluster for raw compute power which would otherwise make Hadoop just a package for HDFS. Apache Spark has multiple libraries, providing a single platform to work off, such as data analysis and transfer learning.

These are the main objectives of this experiment:

- Creating virtual machines with a single configuration of 8GB RAM and 20GB memory in Amazon Web Services and creating Hadoop Clusters with YARN and a spark layer on top of them
- Evaluate the performance between Apache Spark in terms of transfer learning capabilities and data analysis using tools such as:
 - Spark SQL with/without YARN
 - spark-deep-learning (pip library)
- In terms of SQL performance, compare performance with other SQL engines that use HDFS as the backend for the compute power, such as Apache Drill.
- The integration of Apache Drill with Zookeeper and YARN to keep up with configuration requirements required for operating this Particular Engine.
- Comparing the performance in terms of time and cloud credits being used and concluding with whether the virtual machines' specifications were sufficient enough to provide a satisfactory output with a minimal budget.

4 BACKGROUND

Spark Introduction

Spark is a cluster computing technology used as a standalone or with Hadoop YARN. It uses the MapReduce model to perform more computations, including queries and stream processing. It uses in-memory cluster computing for better performance in an application. Spark Core is the execution engine to which all other functionalities branch. It manages the memory, scheduling, distribution, and monitoring and is responsible for fault recovery. Spark Streaming leverages this core to provide the functionality of streaming analytics. It takes data in mini-batches and does Resilient Distributed Datasets transformations on the data. To use the functionality of transfer learning in the spark deep learning library, Machine Learning of Spark, known as MLlib, is used. MLlib is machine learning framework that is distributed due to Spark's use-case and architecture.

The fundamental data structure is RDD. This is an immutable collection of objects where each dataset is divided into logical partitions that can be parallelized and computed in different cluster nodes simultaneously.

In current frameworks, MapReduce is slow since reading and write operations are performed on this disk. This makes it slow in operations crucial in a Machine Learning process such as replication, iterations, and serialization.

The iterative operations span across multiple computations and nodes in multi-stage applications. This is possible in short time because the intermediate results are stored in distributed RAM instead of the disk. In case the storage is not enough in distributed RAM, it will be stored in the disks of each slave node. (Apache Spark - Introduction, 2020)

YARN Introduction

To achieve proper utilization of resources between the machines and achieve abstraction and uniformity between other applications that may use Hadoop MapReduce, a resource negotiator is required. YARN divides the functions of resource management and job scheduling into two different applications into ResourceManager(RM) and Application Master(AM).

ResourceManager manages the resources among the available applications running in the system. ResourceManager requires two components which are Scheduler and ApplicationsManager

Scheduler allocates resources that various applications require according to constraints such as queues and capacities. Scheduler is not responsible for failing tasks or monitoring any application and is only responsible for scheduling the resources required from each application.

The ApplicationsManager accepts job submissions and negotiates with containers for executing the application. Upon any failure, it is responsible for restarting the ApplicationMaster, providing the tracking status, and monitoring the process.

NodeManager is present in each machine to manage the local containers and the resource being used in these containers.

ApplicationMaster is responsible for gathering and communicating the resources required for the applications running from the ResourceManager. ResourceManager, in turn, negotiates with the NodeManager for the required resources. (Apache Hadoop 3.3.4 – Apache Hadoop YARN, no date; Spark SQL and DataFrames - Spark 3.3.0 Documentation, no date)

4.1.1 Different Big Data SQL Engines

SQL provides the capability of acquiring specific and complex data that may not otherwise be available by an EHR vendor. In combination with health informatics, SQL provides the process of obtaining knowledge from an EHR. When a particular problem is identified, the data from EHR is extracted using SQL for that hospital's use case. (Moerbe and Kelemen, 2014)

Different SQL languages can work parallelly, as HDFS is its base data source. Here, the SQL engines which use the worker nodes and work parallelly are considered. Most of the time, the case is that there is an application on each worker node that takes advantage of the part of the data present within it by HDFS. In this project, there are two different SQL engines with four different ways of allocating resources to the worker nodes and acquiring data.

4.1.1.1 Apache Drill

Apache drill is a distributed query engine for large-scale datasets, which include structured and semi-structured data. It can scale up to thousands of nodes to improve the speeds analytics such as Tableau or Power BI require. It uses the "Drillbit" service responsible for accepting requests from the client queries and providing results. Although Drill is not entirely dependent on the Hadoop cluster environment, it requires a zookeeper. However, it can provide better results when the data is divided into multiple nodes. It queries from HDFS to where the data is so that the local Drillbit for that data can access it, therefore making it takes advantage of HDFS but is not entirely dependent on it for performance improvement.

Drillbit is a process that runs on worker nodes to coordinate and plan for executing the queries while also ensuring to maximize data locality. The below diagram shows the communication between the client, application, and Drillbits.

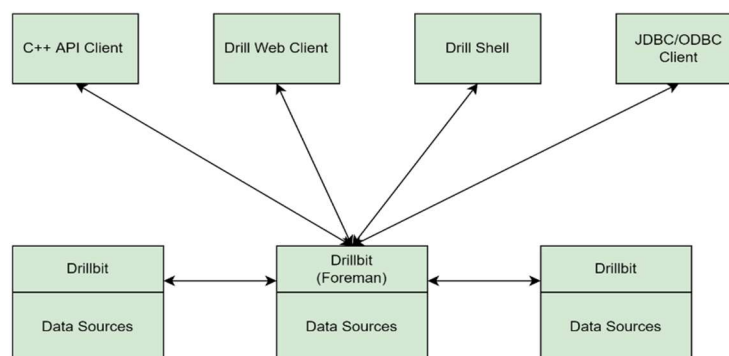


Figure 1: Architecture of Apache Drill

Foreman is the Drillbit that receives the query from the client and parses the SQL for creating custom rules to convert into the syntax for Drill to understand. It forms a collection of logical operators known as a logical plan that describes the work that needs to be done to provide the query results, where the data sources are, and what operation is to be applied. According to the cost optimizer, the rules applied are to rearrange operators to create an optimal plan from which it is converted to a physical plan to execute the query. Foreman generates multiple fragments from this physical plane to create a multi-level execution for rewriting the query and executing in parallel with the configured data sources.(Apache Drill, 2015)

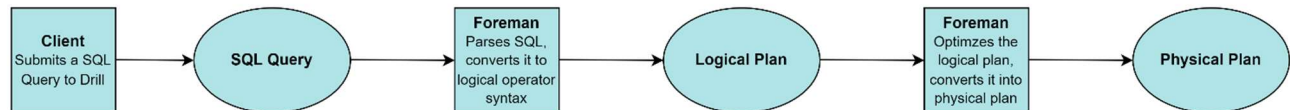


Figure 2: Process of creating a physical plan in Apache Drill

Apache Drill Major Fragment

This representation of a concept in the query execution cannot be modified. These do not perform any tasks but are divided later into multiple fragments to perform the operations. It has one or multiple operations that must be performed in the phase for the Drill to perform the general query and assign a MajorFramingID. To perform a GROUP BY operation of two tables, hash aggregations must be performed. Drill divides the execution into two significant phases where one would be about scanning the files mentioned in the execution, and the second phase would be dedicated to the aggregation.

Apache Minor Fragment

Minor fragments are parallelized from each fragment, where the execution can be run as root, intermediate, or leaf fragments. Every minor fragment is a unit of work that runs inside a thread. Each Minor fragment is also assigned a MinorFragmentID. In reference to the above-mentioned two major fragments, the minor fragments from the major fragments are further divided to perform functions parallel, such as hash aggregation operations.

The root fragment receives the queries, performs metadata reading from tables, routing and rewriting the queries to the up coming level in the serving tree. All this occurs in the Foreman.

Intermediate fragments are necessary to perform the work as soon as the data is made available from other fragments. Their main purpose is to perform operations on data and transfer the data.

The tables are scanned in parallel by the leaf fragments from either the storage layer or the local data and passes this data to the intermediate fragments, where intermediate results are produced.

The key takeaway from the fragments is that Drill always plans to run queries concurrently in fragments, i.e., it assumes that data can be continuously computed parallelly. If there are 20 slices of data in the cluster, it performs 20 minor fragments in a major fragment.

Core Modules

There are 3 main key components of a Drillbit:

RPC endpoint:

In order to communicate with clients, Drill uses a protobuf-based RPC interface. C++ and Java API layers also provide interfaces for the client applications to interact directly with Drill. These APIs can be used to find if a particular Drillbit is available before assigning its functions to perform. Zookeeper is required to perform and is crucial for clients' applications to communicate as Zookeeper keeps the complications of cluster provisioning and management away from the clients.

SQL Parser:

Calcite is an SQL parser that is used by Drill for incoming queries. This parser provides computer friendly, logical plan for the said query.

Storage Plugin Interface:

Drill is capable of querying data from multiple data sources; therefore, storage data plugins are essential for interacting with the data sources by providing abstraction. It provides Drill with information such as Metadata, Interfaces from Drill that have to perform modify (read and write) operations, and the location of data. When Hadoop is integrated with Drill, it will have storage plugins for distributed files and HBase.

Zookeeper is required to maintain highly distributed coordination in the cluster. It is a service that maintains configuration information and synchronization and provides group services across the cluster. Zookeeper is used to communicate between the machines in the cluster and determine which cluster is available for performing the task. It is also responsible for returning results to the client. (Apache ZooKeeper, 2018)

4.1.1.2 Apache Spark SQL

Spark SQL is a module in Spark for data processing. Spark SQL does not use Spark RDD API but instead uses SQL and Dataset API to provide more information about the structure of the data and the computation being performed to return the results in the form of a dataset. The following terms explain the architecture of Spark SQL

The architecture consists of three layers which are

Language API: Since Spark is compatible with languages such as Python, SCALA, and Java, Language API provides the abstraction required for the translation between the language.

Schema RDD: RDD is the unique data structure in Spark Core that provides a temporary table whenever the functions are based on schemas, tables, and records.

Data Sources: Since Spark is compatible with multiple data sources such as Parquet File, JSON format, Cassandra database, etc., it requires a Data Source layer. (*Spark SQL and DataFrames - Spark 3.3.0 Documentation*, no date)

5 LITERATURE REVIEW

5.1 STRUCTURE OF EHR

Electronic Health records store patients' health-sensitive information such as medical visits, medical tests, allergies, medications, immunization status, and age and weight. (Vuppalapati, Ilapakurti and Kedari, 2016)

EMR systems streamline the patient and doctor relationship and communication between interested parties such as insurance providers. They are designed to track all the interested parties in the healthcare system ranging from the actual patients' data to insurance providers. This helps reduce misdiagnosis and convey important information to all interested parties in different data formats.

Some of the software features which EMR provides are:

- **Patient Medical History**
This includes data such as patient visits, allergies, patient referrals, biodata, and lab results
- **Prescription and Medication Data**
This includes data such as medication prescribed to a patient, and inventory management of these medications in a particular hospital.
- **Insurance due and balance.**
Real-time information on patient's insurance fees and how much each insurance provider owes the hospital
- **Data Analytics**
Generating valuable insights on what inventory management of medical supplies and reports on patient data and appointment history
- **Secure remote access**
Usage of SSL technology when accessing the EMR server such as (epic health services, Cerner) . Doing this also helps in complying with local government regulations. (ehrintelligence.com, 2020)

The foundation for information for exchanging healthcare information is done by a standard called FHIR. It uses an HL7-defined set of resources to send and receive information that may document messages, services, etc, through RESTFUL interfaces. These files are stored in formats such as XML, JSON, and RDF syntaxes. (Lloyd, 2015)

5.1.1 Current Research in EHR field

In 2013, a cloud-based approach was proposed for interoperable EHR systems. Public cloud computing can provision resources on pay-per-use pricing models and with minimum start-up investment. Using a SaaS application, patients, doctors and hospitals could access data and manage their inventories. (Bahga and Madiseti, 2013). But this is not the case as most of the patient's data is held in hospitals on private cloud servers. Organizations using the private cloud have more granular control in which machine the data is stored and what action can be taken. This allows for being compliant with local government

health regulations and, most importantly, more fully compliant with HIPPA regulations. (HITInfrastructure, 2017).

Nevertheless, this may not rule out the possibility of hosting some part of the hospital's business in the public cloud achieving the hybrid model in the process, which can be complex. In the case of this particular paper (Alharbi et al., 2016), it was found that hospitals usually already have the underlying IT infrastructure to host a private cloud, although migration may be a challenge. However, this can be achieved with a proper set of guidelines.

Intercept, Cerner and Epic are the Big 3 EHR vendors that many hospitals use extensively. Even if these hospitals do not use the vendors mentioned above, they will migrate to them if they plan to expand because of their good patient safety and clinical process scores.(Beauvais *et al.*, 2021) . With these EHR vendors, in most cases where the access to these services from certain parts of the department in the hospital is not provided to these Epic services or databases, the data entered is mostly in excel format following the HL7 standard of.

EHR vendors have already integrated the required tools for performing data analytics. Cerner is a popular EHR vendor where data analytics tools are not only integrated into the software but can also be integrated with software such as Tableau. They also provide solutions for integrating and pulling data from multiple sources to perform data analytics. APIs can be developed to also integrate with other existing systems for effective transfer of information and communication with other APIs such as FHIR API. (Cerner Oracle, 2022) So this could mean that if there were any particular data analytics report which needs to be generated and it is not provided by the EHR vendor or has a paywall behind it, an API can be developed to communicate with the data analytics framework to perform analysis.

Massive IT infrastructure is often only accessible to a well-established health institution. This also means the number of patients the hospital will be attending is large. A single computer may not have the computational power to access and process the data on the go at that moment if some data analysis is to be done. Here the volume of the data is enormous. Also, considering there is a different type of data stored in the form of images, text format, structured, semi-structured, and unstructured, which contributes to the fact that the data present has variety. The hospital also has limited time because the statistics are required more often within the next day; for example, the number of infectious cases in one day contributes to the velocity factor. This fills in the criteria of naming such cases as Big Data, which fulfills the criteria of velocity, volume, and variety.

The cost of losing the EHR of any patient or any health record is immeasurable. Therefore, it is essential to have redundancy where the same data is present in multiple sources and fail-overs where if one of the systems fails, there is already another system ready to take the job.

Hadoop Distributed File System (HDFS) is useful in this scenario as data is distributed among multiple machines and consists of properties such as fault-tolerant and redundancy. It can store any form of data, but the data present cannot be modified once it is present in the system; therefore, having data integrity once it is present in the system

5.2 DETECTION OF PNEUMONIA BY TRANSFER LEARNING OF X-RAY IMAGES.

Lung X-rays are also stored in the databases of the same hospitals. However, no proprietary or industrial solutions are known yet for detecting Pneumonia. There has been extensive research being done in this field. This is because a model has to be trained using the existing data, which is time-consuming, and a large amount of data must be available in a single file. These models are trained using TensorFlow, and extensive research work has been done using Convolution Neural Networks (CNNs) to train a model using the training data set, validation data set, and testing data set. Each data set serves its purpose in creating a model, then adjusting the model's parameters by validating with the validation data and testing the model's accuracy by the testing dataset. A single machine must produce a model with the necessary parameter with the accurate sample size, i.e., the testing and validation dataset with the required hardware configuration.

Transfer Learning is another method of creating a model, but instead of creating a model from scratch, it is based on an existing model. Transfer learning models are already working model which is trained on numerous amounts of datasets. Here most of the weights in the neural network are frozen, and other parts are trained according to the training and validation data sets provided. This reduces the amount of sample size required for training a model. Transfer learning models are built-in packages in TensorFlow. When the data is present in HDFS, it takes more time for training on TensorflowOnSpark than it takes when the same data is present locally. But this was done using only 3 nodes (IEEE Staff, 2018)

The scope of this research covers the transfer learning and SQL potential in regard to EHR Data. These options can be used for creating a model using Spark or/and TensorFlow as their platform.

Systems based on Spark / YARN Platform	Advantages	Support Or Active Community Support
SparkNet	<ul style="list-style-type: none">Targeted where there are limited network resources.Runs separate optimizers in isolation of each worker, and the average is broadcasted.	✗
DeepSpark	<ul style="list-style-type: none">It is an attempt at the implementation of EASGD on commodity hardware.It bypasses Spark by using a custom communication protocol.	✗
CaffeOnSpark	<ul style="list-style-type: none">It is a data-parallel optimizer on top of Spark.It bypasses the spark execution model by combining MPI-all Reduce communication and RDMA.	✗
BigDL	<ul style="list-style-type: none">Similar to CaffeonSpark but does not use MPIInstead of MPI, parameters are exchanged using Spark Block manager.	✗

	<ul style="list-style-type: none"> This is the only Spark Compliant way of exchanging parameters compared to the above.(Langer <i>et al.</i>, 2018) 	
TensorflowOnSpark	<ul style="list-style-type: none"> Runs on top of Spark. It uses RDDs to provide the workers and supports RDMA over InfiniBand. 	✓
Tensorflow-on-YARN	<ul style="list-style-type: none"> It uses service assembly by managing a C++ TensorFlow The YARN client is provided with ClusterSpec, which will launch the application remotely. 	✗
tf-yarn	<ul style="list-style-type: none"> It is a python library where TensorFlow runs solely on YARN, not Spark. Tf-yarn supports Keras API and Estimator API(Johansson, 2018) 	✓
Sparkdl	<ul style="list-style-type: none"> It uses HorovodRunner to run distributed deep learning training jobs. It is based on DataBricks Runtime 5.0 ML It uses various libraries such as tensorflowonspark, py4j, NumPy, pandas, and tensorframes. 	✓

Among the options mentioned above, spark deep learning uses a combination of different Spark-based platforms and has active community support with the disadvantage of being dependent on different out-of-date python libraries, such as having support for TensorFlow and not TensorFlow 2.0. The latest available version of spark deep learning was in 2020 from Databricks. During the report, the tools used for transfer learning are Spark Deep Learning and the built-in library from Spark, Spark Machine Learning.

Spark Machine Learning provides tools such as ML Algorithms (classification, regression), Featurization (Feature extraction), pipelines, persistence, and utilities (linear algebra, statistics). MLlib supports RDD-based API, which can be lightweight, and each RDD can be distributed among the workers to improve the performance of spark 2.0. Among these ML algorithms, Logistic Regression has the second lowest execution time.(IEEE Staff, 2018)

Spark Deep Learning comes with Keras as a built-in library. There are different pre-trained models in the deep learning libraries like Keras, which have Inception V3, ResNet 50, and VGG16. Among these, Inception V3 has the highest accuracy of 96% in pneumonia detection (Manickam *et al.*, 2021) .

The model's architecture has the following features:

- The inception model is a combination of modules that consists of 48 layers.
- Each basic module is four parallelly aligned layers where 1x1 convolution, 3x3 convolution, 5x5 convolution, and 3x3 max pool layer.

- It uses Factorized convolution layers with the operation of intra-channel spatial convolutions, and linear projection of channels is put into 2D convolutions, and subsequent linear projection is made.
- Smaller convolutions replace the more extensive convolution layers, allowing the computation to occur faster.
- CNN layers have auxiliary layers during the training, adding the loss incurred to the main network loss.

5.2.1.1 SPARK SQL OPERATION

Spark SQL return results in Dataset/DataFrame, a collection of data arranged in a distributed manner. Python does not natively support datasets in the earlier versions of Spark. Java Virtual Machine objects are used to construct datasets for being manipulated with functions such as map, flatMap, and filter. It is similar to a table present in Python or R.

To perform a query, the query must be divided into a set of operations which need to be divided into different operations in an execution plan. Spark Executors use Directed Acyclic Graph, a cyclic graph created from scheduler known as DAGScheduler in Spark. The optimizer in Spark is shown below and is named "Catalyst." At first, there will be a Logical Plan responsible for checking basic syntaxes and performing semantic analysis. The next step is to generate the Analyzed Logical Plan, which would be the actual plan for the RDD. This is followed by the optimized Logical plan based on operations such as filters, aggregation, etc. After the optimized logical plan has been created, the catalyst generates many physical plans that are different in strategies, cost and time taken. Considering all the factors, a single physical plan will be selected and executed (Laurent Leturgez, 2020)

Bunsen allows loading, transforming, and analyzing FHIR data with Apache Spark. It uses Java and Python APIs for converting into Spark Datasets. However, the Bunsen API, which Cerner created, is not maintained since 2020 and lacks community support.

SQL Engine Name	Platform Integration	Analytics Tools	Data Source Integration
Apache Drill	Standalone with Zookeeper, YARN	Tableau	HBase, Hive, MongoDB, AWS S3, Kafka, Azure Blob Storage, Cassandra, Splunk, Druid, Dropbox.
Apache Spark	Standalone, YARN, Mesos	Tableau, Power BI	DB2, MariaDB, MS SQL, Oracle SQL, PostgreSQL, Bunsen (FHIR) API

Cloud Computing virtual machines have many advantages in terms of lower maintenance, storage, networking, and operating systems. Provisioning virtual machines and disabling them when not needed is more manageable. Higher-end instances can be provisioned within minutes, and the model can be trained within a short amount of time. As Spark is a memory-intensive application, it needs ample RAM. AWS uses the Xen Virtualization technique to manage the virtual machines in one physical server. Each Xen virtual machine is known as an AWS EC2 instance. The smallest instance starts from 1.7Gb of RAM to up to 160 GB of RAM. For this experiment, the EC2 instance t2.large will be used, consisting of 8GB RAM and 4 vCPU cores with 20GB as ROM storage.

6 RELATED WORK

An integration with Vancouver jasdask with HBase and HDFS was proposed in (Chrimes and Zamani, 2017) , Five worker nodes and one master node were used, and these were the VMs created from their local OpenStack Cloud. Jupyter and Zeppelin used a GUI interface to enable the analyst to query data instead of using the CLI command from the servers using Apache Drill and Spark. The goal of integration, maintenance, and data analysis was accomplished regardless of how complex the data was within a few seconds. It was concluded that Spark and Drill were significantly faster than HBase in ingesting files. Between Spark and Drill, Drill was more capable of emulating complex data more efficiently.

There has been extensive research in terms of detecting Pneumonia with a single high-end configuration machine to using Apache spark-based Machine Learning using Databricks. In (Chouhan *et al.*, 2020), using a single high-end configuration computer, the average computational time for the model took around half a minute for an average of 100 epochs. Rather than using Virtual Machines in the cloud, Databricks workspace was used in (Awan *et al.*, 2021). Databricks provides apache spark using clusters in the cloud-ready environments with GUI similar to Jupyter and Zeppelin. The detection of Chest X-rays was whether if the patient had covid-19, Pneumonia or a normal health condition making it a three-class classifier. They achieved over 97.10% accuracy by using Inception V3 and logistic regression. Although the approach is same, It is not mentioned what the configuration of the machines used in the databricks platform was. The version of Spark and what tools such as YARN or standalone were integrated being used are not in the scope of the mentioned paper. The main goal of the paper was to achieve maximum accuracy and explain the 3-way classifier. Apart from the accuracy measures, they were no performance measures of the system or how much time was taken to achieve for the model to be trained and tested. This paper will focus on the performance measures by using the time taken to execute and for creating the model. The environment will be Amazon EC2 machines where Spark Environments will be created, which are Spark standalone and Spark with YARN and Apache Drill for SQL functions.

7 APPROACH

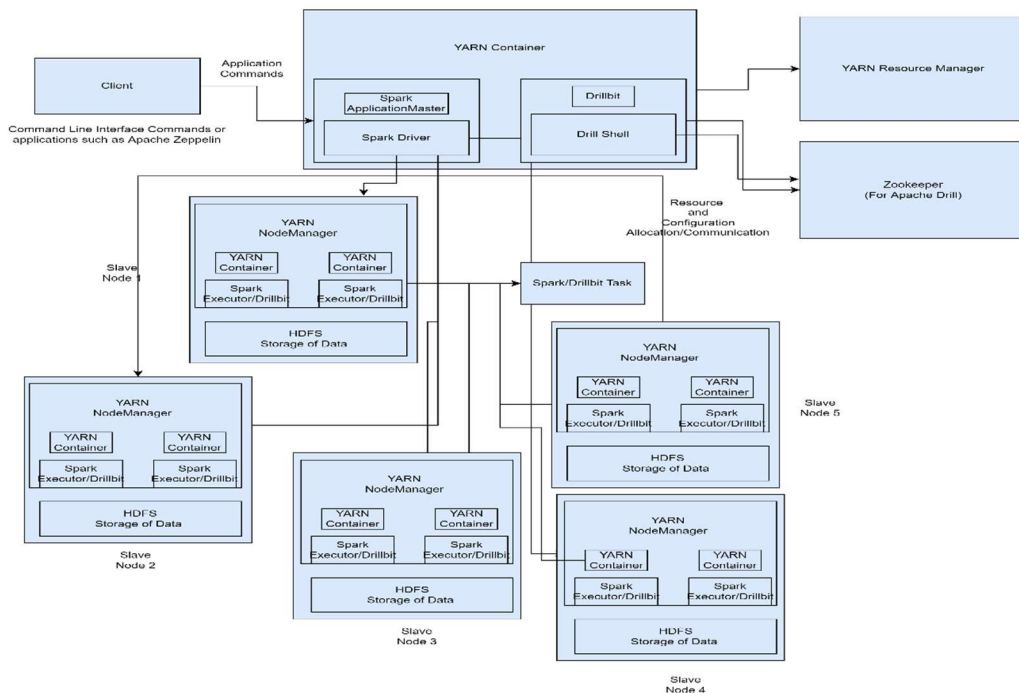


Figure 3: The architectural diagram of running processes using YARN(*Running Spark Jobs on YARN. When running Spark on YARN, each Spark... | by saurabh goyal | Medium*)

In the above diagram, the commands are directly sent to the application master, which communicates with YARN as it is responsible for allocating resources. YARN is already preconfigured with specific amounts of resources to a particular application like Spark or Drill running over it. Hadoop comes with YARN out of the box and is responsible for HDFS, communicating with YARN, and indirectly with other applications to which data block is present in which slave node for localized parallel processing.

7.1 ELECTRONIC HEALTH RECORDS DATASET

High-Quality Electronic Health Record Datasets are required for the developers to create and maintain new software and for testing. Although some healthcare providers may provide the EHR dataset, this may not have the necessary quality or quantity. The EHR datasets are often anonymized but even if the records are anonymized, there are reports of these anonymized datasets traced back to the real patients. Therefore, an open-source system was developed to generate realistic synthetic EHRs.(Walonoski *et al.*, 2018)

The dataset is hosted in SyntheticMass, which Synthea generated. Since the data is synthetic, the data present is free from any expense and security restrictions. The dataset is present in CSV format. It contains one million synthetic patient records with a population of around 200,000 patients. It is divided into different files: allergies, encounters, observations, care plans, immunizations, patients, conditions, medications, and procedures in CSV format. The patient's file contains sensitive information of each

unique patient, such as ID, SSN, Name, Date of Birth, Drivers License Number etc. Observations contain over a million records of these patients with values such as Body weight, Body Height, Cholesterol Levels etc. These two files will be used as these are the routinely collected data as per the clinical requirements. These are contributed to the Informative presence (IP) and information observations. Both IP and IO can be used to develop Clinical Prediction Models (CPMs)(Sisk et al., 2020). Apart from the prediction models, the basic motive is to provide data analytics of a particular patient within a short amount of time.

7.2 CHEST X-RAY DATASET

The images are split into two parts: a training set and a testing set of independent patients. Each set has two sets of conditions, where the X-ray is of either a normal patient or of a patient infected with Pneumonia. The size of the dataset is 2GB. The metadata of the patient's ID number is randomized.

These two cases, i.e., EHR Data Analytics and Detection of Pneumonia using Chest X-ray, will be done on Apache Spark framework with pyspark programming. There will be one master node, and the number of slave nodes will be incremented to 5 machines to recognize the performance impact in different case scenarios. The files will be stored in a distributed file system, i.e., Hadoop Distributed File System.

In the case of EHR Data Analytics, different SQL engines were used to compare the performances while including YARN and excluding the same. Meanwhile, since Inception V3 has a reputable accuracy rate for Pneumonia Detection, it was used for vectorization of the images and Logistic Regression with different amounts of iterations with and without using YARN as the underlying Resource Negotiator.

YARN is a resource negotiator responsible for assigning containers both in Spark and in YARN. The container is the basic unit of processing in YARN which consists of elements of resources such as RAM and CPU in an encapsulated form. It is generally recommended to have two containers per disk and per core for optimum utilization of the resources.

Following the Reserved Memory Recommendations chart from HortonWorks, each 8GB node 2 GB memory was reserved for system memory and HBase memory. The calculation for the number of containers allowed per node was as follows

Number of Containers= minimum of (2*Cores, 1.8 *Disks, (Total available RAM/Minimum Container Size))

Here, the total minimum container size, according to Hortonworks, should be around 1GB

The calculation for RAM for each container is

RAM-per-Container=maximum of (Minimum Container Size, (Total Available RAM/Containers))

(11. Determine YARN and MapReduce Memory Configuration Settings - Hortonworks Data Platform, 2017)

From the below table, the resources available were analyzed and applied in the YARN Configuration file.

yarn.nodemanager.resource.memory-mb	18432
yarn.scheduler.minimum-allocation-mb	2048

yarn.scheduler.maximum-allocation-mb	18432
mapreduce.map.memory.mb	2048
mapreduce.reduce.memory.mb	4096
Number of containers	9
Total RAM	18432
Node 1	6144
Node 2	6144
Node 3	6144
Node 4	6144
Node 5	6144
Disks	5
Cores	10

8 IMPLEMENTATION

8.1 TRANSFER LEARNING IN APACHE SPARK

The transfer learning is used from the spark deep learning library managed by Databricks. Spark Deep Learning uses different packages such as TensorFlow, Keras, tensorflowonspark etc. The data present is divided where 60% is for training the data, and 40% will be used for testing the data. Vectorizer Inception V3 is used to generate the feature map of each image. This follows the Multi-stage processing where all the images with Inception V3 model feature maps are generated, and a new model with a single softmax layer is added to the final layer of the Inception V3 model. The process mentioned above is already present in function i.e, DeepImageFeaturizer from the spark deep learning library.

Pyspark has Machine Learning Classification libraries that can be used for the classification of these feature maps through iterations. With the given Literature Review, Logistic Regression was chosen for performing Machine Learning. Below are the graphs related to the time and cost taken for training a model through transfer learning.

8.2 OBSERVATIONS

It should also be noted that during experimentation, the number of nodes had to be 1 master node and 5 slave nodes due to the limitations faced during the execution with issues such as not enough main memory present in slave machines despite the machines being t2.large with 8GB RAM when lesser than 5 slave nodes were used.

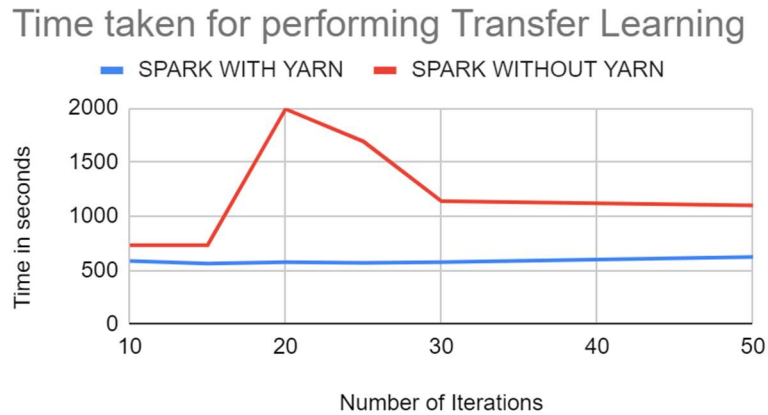


Figure 4: Time taken for creating Transfer Learning Model between Spark with and without YARN

From the above graph, it can be observed that Spark with YARN takes almost equal time for all the iterations up to 50. Nevertheless, Spark without YARN vastly fluctuates the time taken during 20 and 30 iterations and, after that, maintains a steady time but still takes more time taken by Spark without YARN Machines.

The charges are according to the machines' time and utilization rate when utilizing Virtual Machines. Following the prices charged by Amazon Web Services for the Virtual Machines, the below graph demonstrates the charges that are implied for each iteration.

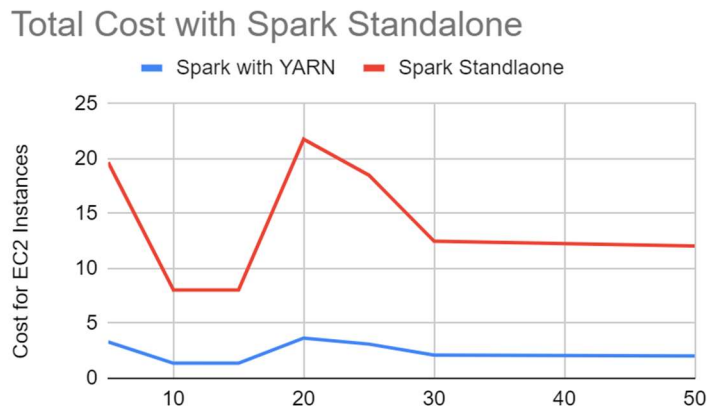


Figure 5: Total Cost in AED for creating the model in Spark Standalone using AWS EC2 Virtual Machines

The cost of 20 iterations is the highest. During the experimentations, the number of iterations heavily determined the time taken for the model to be trained. However, time was not directly proportional to the number of iterations due to how different standalone Pyspark utilizes resources than Pyspark with YARN with multiple machines. Nevertheless, it should also be considered that more iterations could lead to overfitting and, therefore, lesser accuracy.

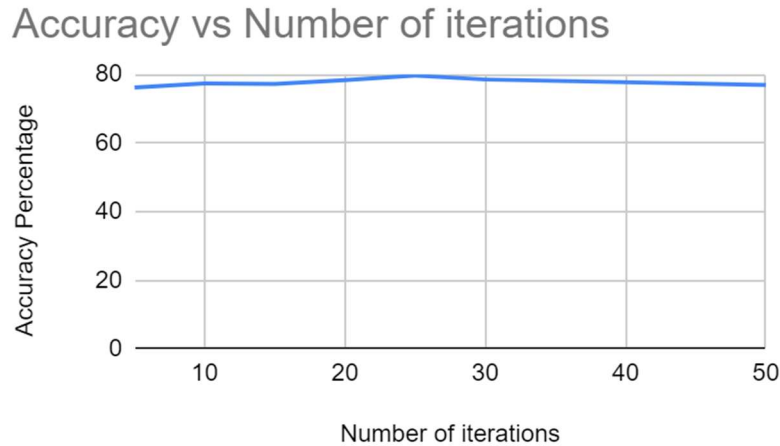


Figure 6: Accuracy of the Pneumonia Detection Model

From the above graph, the curve from 10 iterations to 30 iterations is reaching its peak at 25 iterations with an accuracy of 79.65%. The portion from 10 to 25 shows the effect of underfitting, where the model requires more iterations for it to be accurate and the portion from 25 to 30 shows the effect of overfitting. Considering the above graphs, 25 iterations is perfect in terms of accuracy, cost, and time taken.

8.2.1 Using SQL engines

SQL engines can usually process records up to 100,000 in less than a second. The dataset, in this case, has over a million observations and 150,000 patients. In EHR systems, the FHIR data format must be present in the CSV table to perform data analytics. The names of patients are not present in the observations table but present in the patients' table. This is how relational databases are created; any query is typically required to be completed within seconds. For this reason, the performance between SQL Engines will be compared in terms of seconds.

Types of commands being used:

COUNT

This function demonstrated the number of times patients would have to record their values.

JOIN

This function consisted of joining two tables, the Observations table, and the patients table, of determining which patient's data was in a particular value.

ORDER

This function was used to find the number of patients encountering specific diseases from different months in a particular year using the functions COUNT, GROUP, and ORDER.

Apache Spark SQL with YARN/without

Apache Spark SQL was used with YARN and as a standalone. The performance differences are shown below.

8.2.2 COUNT

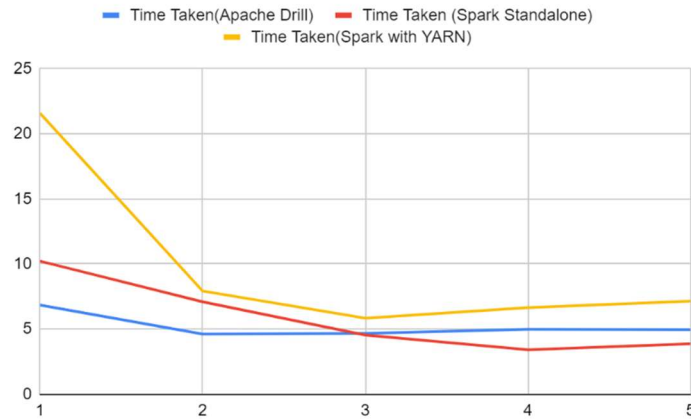


Figure 7: Graph for the time taken (in seconds) in accordance with a number of nodes used.

Since this combination of functions is simple, the performance differences between the three systems are minuscule when more nodes are used. Here, when using a single slave node, Spark with YARN cannot keep up with the performances of Apache Drill and Apache Spark Standalone for this particular function because the simple function took about 20+ seconds.

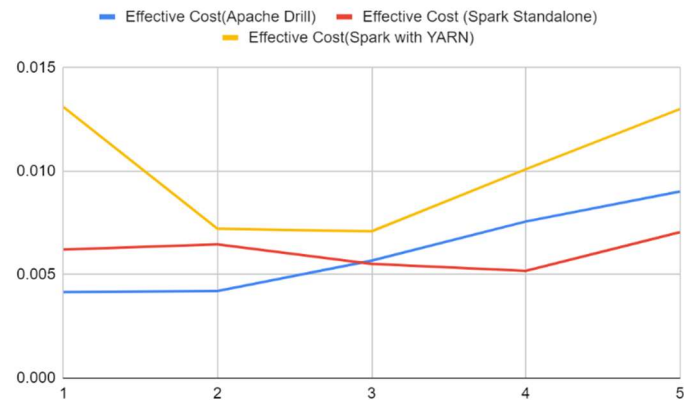


Figure 8: Total Cost in AED for executing the SQL function in accordance with the number of nodes used

The disadvantage of using fewer nodes is exacerbated when the cost factor is added. For this particular function, it was observed that the Spark Standalone was the most economical one.

8.2.3 JOIN

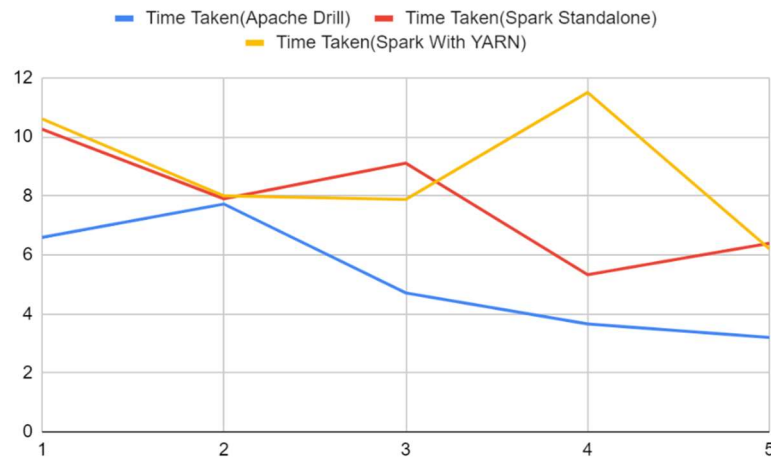


Figure 9: Time taken for executing the particular function per the number of nodes used.

From the above graph, the number of nodes utilized linearly is directly proportional to the time taken in the case of Apache Drill. On the other hand, Apache Spark with and without YARN remains inconsistent, with the four slave nodes taking the highest time in Spark with YARN and consuming the lowest time in Spark without YARN. This graph demonstrates that the Apache Drill is stable enough with its predictable behavior in this particular function.

Considering the cost taken for performing this particular function. Here the cost has two factors: the number of nodes being used and the utilization rate. The utilization rate has been assumed as average here. The value may be minuscule, but in real applications, there may be repeated queries, which could be a factor in deciding.

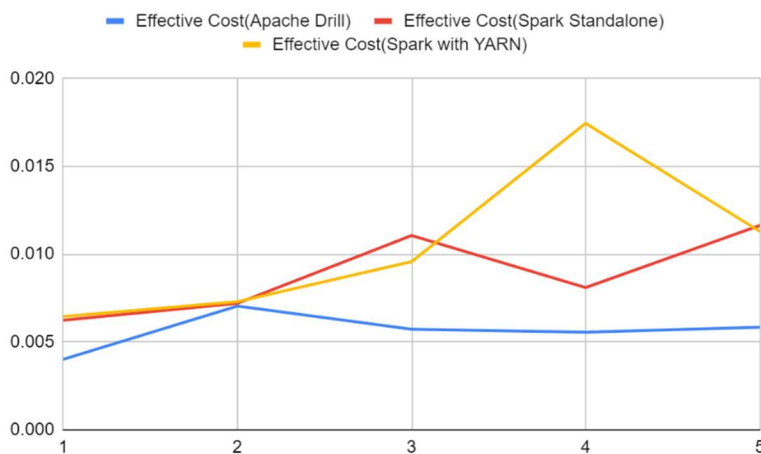


Figure 10: Total Cost in AED for executing SQL functions under the number of nodes used

In the above graph, since the Apache Drill has a linear line along the number of node and the cost, it takes lesser cost when compared to Apache Spark in both cases.

8.2.4 ORDER

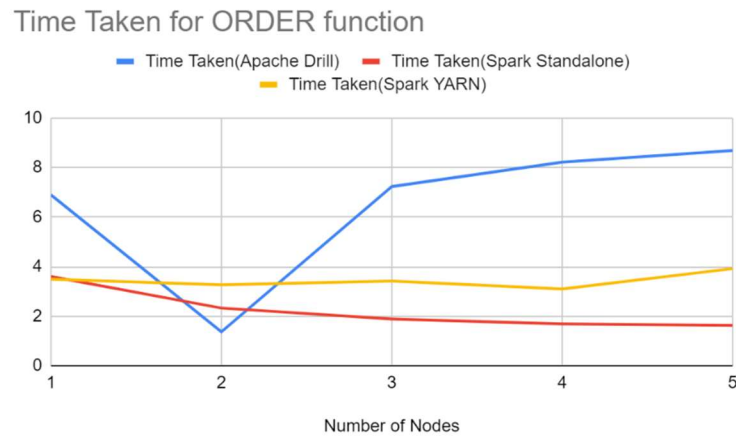


Figure 11: Total time taken for executing the function by the number of nodes used

In the particular set of functions, the increase in the number of nodes used was observed to reduce the execution time of Spark with YARN and Spark Standalone. In this case, Apache Drill has not been consistent with the increasing number of nodes because the execution time increased with more nodes, which is unfavorable.

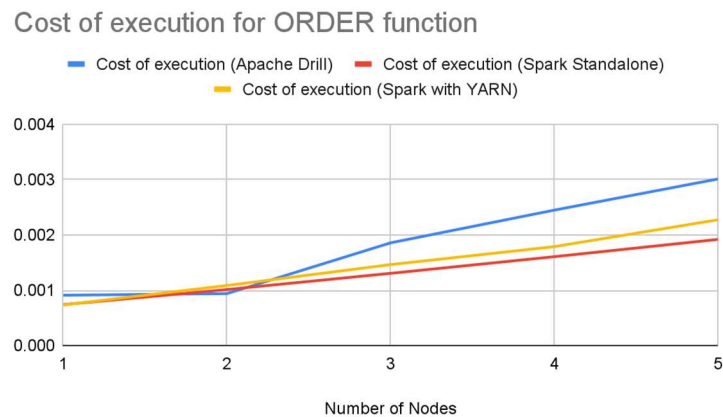


Figure 12: Total Cost taken for executing the function following a number of nodes used.

This graph demonstrates that the differences between the SQL execution times are minuscule at the start but continue as time goes on. It was also observed that, since more nodes are being used, the execution cost was on a steady increase which is expensive and is unlike the other two functions provided in this experiment.

9 DISCUSSION

The virtual machines with configuration could keep up with the processes such as handling the Hadoop distributed file system cluster and providing compute power required for data analysis and transfer learning.

9.1.1 Data Analysis in EHR

The execution time was only a few seconds for all the queries. When using YARN as the basis for Spark, it could not perform as well as its native counterpart Apache Drill, which is solely meant to perform SQL operations, outperforms Spark SQL on both occasions. Although YARN is a vital component and makes it easy to assign resources to Spark containers, the execution time is slower when Spark is used with YARN. Regarding the introduction in the research paper, the main objectives were to provide a minimal cost and optimized environment to perform data analysis and Spark SQL without YARN and Apache Drill having consistent performance in both of these functions.

9.1.2 Transfer Learning

On the other hand, Spark with YARN has proven consistent training models using transfer learning. Transfer learning is a compute-intensive process that usually leads to several hours of training a model to get the required accuracy for the application. Spark with YARN also had fewer failures of executor drivers during the model's training. This proves that YARN provides a stable environment where it allocates a certain number of resources, containerizes the effects of failures, and provides a suitable environment for training the model using transfer learning.

9.2 LIMITATIONS

During the transfer learning process through Spark Deep Learning, the executor drivers present in Spark kept failing and reinstating due to the lower RAM space available. Although it was decided that 8GB would be sufficient for this process, the RAM limitations severely affected the iteration process. Due to the size of the training model, the training process could not parallelize effectively, leading to executors failing around 40% of progress. The training process was smooth for the first 5 minutes, but the executors kept failing after a particular duration. Due to the limitations mentioned above, the training data size had to be reduced to 500 MB. This took a toll on the model's accuracy and could not achieve more than 80%, hitting the peak accuracy rate at 79.65%. A higher specification virtual machine was not possible as the main objective of this experiment was to check if this lower specification of machine could be utilized to create a model.

From the observations taken from the application column in the YARN application during the creation of the model, the process in DAGVisualization it can be observed that process from binary file to tree aggregation the process was quick. The process was time-consuming for tree aggregation.

Tree aggregate is an implementation of aggregate, which functions as a combined function on a subset of partitions. The main aim of tree aggregate is not to provide partial results to the spark driver. The partitions present in Spark have to send a reduced value to the driver machine. To achieve parallelism, the file is divided into partitions. In some cases, due to the file size being too huge or insufficient RAM, the starting partitions are divided evenly, but the last partitions are divided unevenly and are large,

creating a bottleneck. (TreeReduce and TreeAggregate Demystified - Apache Spark - Best Practices and Tuning, 2020)

Executors

► Show Additional Metrics

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(6)	0	262.5 MB / 16.6 GB	0.0 B	5	5	0	0	5	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(6)	0	262.5 MB / 16.6 GB	0.0 B	5	5	0	0	5	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0

Executors

Show 20 entries

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs	Thread Dump
driver	amazon-1:39983	Active	0	87.5 MB / 384.1 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B		Thread Dump
1	amazon-6:45575	Active	0	87.4 MB / 3.2 GB	0.0 B	1	1	0	0	1	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
2	amazon-5:42229	Active	0	51.5 KB / 3.2 GB	0.0 B	1	1	0	0	1	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
3	amazon-3:43245	Active	0	51.5 KB / 3.2 GB	0.0 B	1	1	0	0	1	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
4	amazon-2:38241	Active	0	51.5 KB / 3.2 GB	0.0 B	1	1	0	0	1	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump
5	amazon-4:33345	Active	0	87.4 MB / 3.2 GB	0.0 B	1	1	0	0	1	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr	Thread Dump

Figure 13: Result page of Apache Spark during the Transfer Learning Process about memory usage

The first four executors have evenly divided storage memory. However, the last executor has a large file of almost 100MB, which created a bottleneck during the experiment and reduced the execution speed.

To further provide evidence that there was not enough for Spark, the below image shows Shuffle Spill

Aggregated Metrics by Executor

Executor ID	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Input Size / Records	Shuffle Write Size / Records	Shuffle Spill (Memory)	Shuffle Spill (Disk)	Blacklisted	
0	<div>stdout stderr</div>	172.31.14.189:33625	2.8 min	1	0	0	1	256.1 MB / 1705	6.0 MB / 1	992.0 MB	762.4 MB	false
1	<div>stdout stderr</div>	172.31.9.144:33765	2.6 min	1	0	0	1	391.5 MB / 1052	6.0 MB / 1	0.0 B	0.0 B	false
2	<div>stdout stderr</div>	172.31.4.230:40439	6.2 min	2	0	0	2	256.8 MB / 444	12.0 MB / 2	0.0 B	0.0 B	false
3	<div>stdout stderr</div>	172.31.3.101:35281	2.8 min	2	0	0	2	143.2 MB / 262	11.7 MB / 2	0.0 B	0.0 B	false
4	<div>stdout stderr</div>	172.31.10.146:41877	2.9 min	1	0	0	1	256.2 MB / 1769	6.0 MB / 1	992.0 MB	766.2 MB	false

Figure 14: Result page of Aggregated Metrics of different Executors present.

In Spark, the process of moving data from RAM to disk and vice versa is known as Spill. This occurs when a partition is insufficient for the available RAM.

9.3 FUTURE RECOMMENDATIONS

This thesis provided insight on whether YARN is necessary or not in terms of providing the solution for the best accurate model for detecting Pneumonia. From the above-discussed limitations, it is clear that a thorough calculation of the management of resources must be done beforehand to utilize the spark framework for transfer learning applications. Having more memory than the dataset is crucial, as Spark only runs on RAM. Regarding data analysis, resource allocation does not play a massive role in performance in a way that may make Spark not even suggestible; it depends on only the type of functions being used regardless of the specification of machines at a particular time. YARN is essential when using applications based on Hadoop and parallel processing, and it is highly recommended for

Spark and Drill to be run alongside YARN for better resource utilization. To further explore this field, it is suggested to compare performance between applications that are cloud-based, such as Amazon EMR and Databricks, with solutions provided by Hortonworks.

9.4 CONCLUSION

The experiment provided insights on the performance of Spark with and without YARN and Apache Drill. From the above observations, there are a majority of cases where Spark with YARN outperformed Spark without YARN. Applications which are mainly meant for SQL, such as Apache Drill, are consistent with SQL driver applications, as shown in this experiment. The experiment on transfer learning shows that Spark with YARN is recommended over Spark without YARN compared to the time taken. From the above shortcoming, properly configured clusters and resource provision must be provided when using IaaS elements for creating Hadoop-based applications in the cloud. In terms of SQL engines, Spark SQL with YARN and without YARN proved to be on par with Apache Drill in terms of performance and is recommended to be integrated with data analytics-driven applications such as Tableau or PowerBI.

10 REFERENCES

11. *Determine YARN and MapReduce Memory Configuration Settings - Hortonworks Data Platform* (no date). Available at: https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.0.9.0/bk_installing_manually_book/content/rpm-chap1-11.html (Accessed: 22 September 2022).

Alharbi, F. *et al.* (2016) 'Strategic Value of Cloud Computing in Healthcare Organisations Using the Balanced Scorecard Approach: A Case Study from a Saudi Hospital', in *Procedia Computer Science*. Elsevier B.V., pp. 332–339. Available at: <https://doi.org/10.1016/j.procs.2016.09.050>.

Apache Drill (no date) *Drill Query Execution*. Available at: <https://drill.apache.org/docs/drill-query-execution/> (Accessed: 10 August 2022).

Apache Hadoop 3.3.4 – Apache Hadoop YARN (no date). Available at: <https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html> (Accessed: 29 September 2022).

Apache Spark - Introduction (no date). Available at: https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm (Accessed: 22 September 2022).

Apache ZooKeeper (no date). Available at: <https://zookeeper.apache.org/> (Accessed: 22 September 2022).

Awan, M.J. *et al.* (2021) 'Detection of COVID-19 in Chest X-ray Images: A Big Data Enabled Deep Learning Approach', *International journal of environmental research and public health*, 18(19), p. 10147. Available at: <https://doi.org/10.3390/ijerph181910147>.

Bahga, A. and Madiseti, V.K. (2013) 'A cloud-based approach for interoperable electronic health records (EHRs)', *IEEE Journal of Biomedical and Health Informatics*, 17(5), pp. 894–906. Available at: <https://doi.org/10.1109/JBHI.2013.2257818>.

Beauvais, B. *et al.* (2021) 'Association of Electronic Health Record Vendors With Hospital Financial and Quality Performance: Retrospective Data Analysis.', *Journal of medical Internet research*, 23(4), p. e23961. Available at: <https://doi.org/10.2196/23961>.

Cerner Oracle (no date) *Cerner Data Analytics*. Available at: <https://www.cerner.com/solutions/analytics> (Accessed: 7 August 2022).

Chouhan, V. *et al.* (2020) 'A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images', *Applied sciences*, 10(2), p. 559. Available at: <https://doi.org/10.3390/app10020559>.

Chrimes, D. and Zamani, H. (2017) 'Using Distributed Data over HBase in Big Data Analytics Platform for Clinical Services', *Computational and mathematical methods in medicine*, 2017, pp. 6120820–16. Available at: <https://doi.org/10.1155/2017/6120820>.

Ergüzen, A. and ünver, M. (2018) 'Developing a File System Structure to Solve Healthy Big Data Storage and Archiving Problems Using a Distributed File System', *Applied Sciences* 2018, Vol. 8, Page 913, 8(6), p. 913. Available at: <https://doi.org/10.3390/APP8060913>.

Harmony (2020) *Health Data Volumes Skyrocket, Legacy Data Archives On the Rise*.

HITInfrastructure (2017) *Choosing Between Healthcare Public Cloud, Private Cloud*. Available at: <https://hitinfrastructure.com/news/choosing-between-healthcare-public-cloud-private-cloud> (Accessed: 7 August 2022).

IEEE Staff (2018) *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE.

Johansson, T. (2018) *Managed Distributed TensorFlow with YARN Enabling Large-Scale Machine Learning on Hadoop Clusters, DEGREE PROJECT COMPUTER SCIENCE AND ENGINEERING*.

Langer, M. et al. (2018) 'MPCA SGD - A Method for Distributed Training of Deep Learning Models on Spark', *IEEE Transactions on Parallel and Distributed Systems*, 29(11), pp. 2540–2556. Available at: <https://doi.org/10.1109/TPDS.2018.2833074>.

Lloyd (no date) *FHIR Core, HL7*. Available at: <http://www.hl7.org/Special/committees/fiwig/docs.cfm> (Accessed: 7 August 2022).

Manickam, A. et al. (2021) 'Automated pneumonia detection on chest X-ray images: A deep learning approach with different optimizers and transfer learning architectures', *Measurement: Journal of the International Measurement Confederation*, 184. Available at: <https://doi.org/10.1016/j.measurement.2021.109953>.

Moerbe, M. and Kelemen, A. (2014) 'Turning electronic health record data into meaningful information using SQL and nursing informatics', *Computers, informatics, nursing : CIN*, 32(8), pp. 366–377. Available at: <https://doi.org/10.1097/CIN.0000000000000079>.

Running Spark Jobs on YARN. When running Spark on YARN, each Spark... | by saurabh goyal | Medium (no date). Available at: <https://medium.com/@goyalsaurabh66/running-spark-jobs-on-yarn-809163fc57e2> (Accessed: 29 September 2022).

Sisk, R. et al. (no date) 'Informative presence and observation in routine health data: A review of methodology for clinical risk prediction', *Journal of the American Medical Informatics Association*, 28(1), pp. 155–166. Available at: <https://doi.org/10.1093/jamia/ocaa242>.

Spark SQL and DataFrames - Spark 3.3.0 Documentation (no date). Available at: <https://spark.apache.org/docs/latest/sql-programming-guide.html> (Accessed: 19 July 2022).

Spark's Logical and Physical plans ... When, Why, How and Beyond. | by Laurent Leturgez | datalex | Medium (no date). Available at: <https://medium.com/datalex/sparks-logical-and-physical-plans-when-why-how-and-beyond-8cd1947b605a> (Accessed: 21 September 2022).

TreeReduce and TreeAggregate Demystified - Apache Spark - Best Practices and Tuning (no date). Available at: https://umbertogriffo.gitbook.io/apache-spark-best-practices-and-tuning/rdd/treereduce_and_treeaggregate_demystified (Accessed: 22 September 2022).

'United States Government Accountability Office Report to Congressional Requesters Artificial Intelligence in Health Care Benefits and Challenges of Machine Learning Technologies for Medical Diagnostics With content from the National Academy of Medicine' (2022).

'Virtualizing Hadoop[®] on VMware vSphere[®] Virtualizing Hadoop on VMware vSphere' (no date).

Vuppalapati, C., Ilapakurti, A. and Kedari, S. (2016) 'The Role of Big Data in Creating Sense EHR, an Integrated Approach to Create Next Generation Mobile Sensor and Wearable Data Driven Electronic Health Record (EHR)', in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, pp. 293–296. Available at: <https://doi.org/10.1109/BigDataService.2016.18>.

Walonoski, J. *et al.* (2018) 'Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record', *Journal of the American Medical Informatics Association*, 25(3), pp. 230–238. Available at: <https://doi.org/10.1093/JAMIA/OCX079>.

11 APPENDICES

Initiating PySpark SQL Functions

In [1]:

```
import pyspark
from pyspark.sql.functions import split,array_remove,when,concat_ws
#from pyspark.sql.functions import split_col
import pyspark.sql.column
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
from pyspark.sql import SparkSession
spark=SparkSession.builder.getOrCreate()
sc
```

Out[1]:

SparkContext

[Spark UI](#)

Version

v2.4.8

Master

yarn

AppName

PySparkShell

Reading the file from the HDFS cluster and analyzing the columns present in Patients.csv file

In [2]:

```
rc = spark.read.csv('hdfs://amazon-1:9000/data/csv/patients.csv',header=True)
rc.columns
```

Out[2]:

```
['ID',
 'BIRTHDATE',
 'DEATHDATE',
 'SSN',
 'DRIVERS',
 'PASSPORT',
 'PREFIX',
 'FIRST',
 'LAST',
 'SUFFIX',
 'MAIDEN',
 'MARITAL',
 'RACE',
 'ETHNICITY',
 'GENDER',
 'BIRTHPLACE',
 'ADDRESS']
```

Formatting the available columns present in the patients file

In [3]:


```

from pyspark.sql.functions import to_timestamp,col,lit
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, DateType,
BooleanType, DoubleType, IntegerType,FloatType
fields=[('ID',StringType()),
        ('BIRTHDATE',DateType()),
        ('DEATHDATE',DateType()),
        ('SSN',StringType()),
        ('DRIVERS',StringType()),
        ('PASSPORT',StringType()),
        ('PREFIX',StringType()),
        ('FIRST',StringType()),
        ('LAST',StringType()),
        ('SUFFIX',StringType()),
        ('MAIDEN',StringType()),
        ('MARITAL',StringType()),
        ('RACE',StringType()),
        ('ETHNICITY',StringType()),
        ('GENDER',StringType()),
        ('BIRTHPLACE',StringType()),
        ('ADDRESS',StringType())]
schema = StructType([StructField (x[0],x[1],True) for x in fields])
schema

```

Out[3]:

```

StructType(List(StructField(ID,StringType,true),StructField(BIRTHDATE,DateType,true),StructField(DEATHDATE,DateType,true),StructField(SSN,StringType,true),StructField(DRIVERS,StringType,true),StructField(PASSPORT,StringType,true),StructField(PREFIX,StringType,true),StructField(FIRST,StringType,true),StructField(LAST,StringType,true),StructField(SUFFIX,StringType,true),StructField(MAIDEN,StringType,true),StructField(MARITAL,StringType,true),StructField(RACE,StringType,true),StructField(ETHNICITY,StringType,true),StructField(GENDER,StringType,true),StructField(BIRTHPLACE,StringType,true),StructField(ADDRESS,StringType,true)))

```

Removing excess and anonymous data present

In [4]:

```

rc = spark.read.csv('hdfs://amazon-1:9000/data/csv/patients.csv',schema=schema,header=True)
rc=rc.where("ID!='33f33990-ae8b-4be8-938f-e47ad473abfe'")
rc=rc.where("ID!='64066 Johns Bridge Suite 317 Eastham MA 02642 US'")
rc.show(5)

```

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
|          ID| BIRTHDATE| DEATHDATE|          SSN|  DRIVERS|  PASSPORT|
PREFIX|      FIRST|      LAST|SUFFIX|  MAIDEN|MARITAL|  RACE|      ETHNICITY|
GENDER|      BIRTHPLACE|      ADDRESS|
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

```

```
|660bec03-9e58-47f...|1996-07-26|      null|999-70-3315|S99945940|      false|
  Ms.|Geovany567| Reichert456|  null|      null|      null|white|      irish|
  F|  Fitchburg MA US|20810 Bart Inlet ...|
|5125d2b2-3aef-4ae...|1996-09-24|      null|999-89-6289|S99991246|      false|
  Ms.| Tianna156|  Kuphal267|  null|      null|      null|white|french_canadian|
  F|Westborough MA US|295 Walter Mill D...|
|26626faf-cbd5-48d...|1944-09-01|2015-09-04|999-79-2204|S99913823|X19963891X|
  Mr.|Ryleigh341|      Mraz432|  null|      null|      M|white|      irish|
  M| Fall River MA US|23401 Gerhold For...|
|f509a0f0-77ef-477...|1964-05-14|2010-07-11|999-70-3377|S99930834|      false|
  Mrs.| Amparo640|Bergstrom813|  null|Green619|      M|white|      french|
  F|  Cambridge MA US|55368 Suzanne Via...|
|4c763cac-b1df-4bc...|1946-03-05|1967-01-10|999-52-5432|S99989461|      false|
  Ms.|Demarco886|  Osinski65|  null|      null|      null|white|      irish|
  F| Framingham MA US|63493 Madison Str...|
```

```
+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+
only showing top 5 rows
```

Creating a dictionary of patients with their patient ID to use it in other files

In [5]:

```
rc = spark.read.csv('hdfs://amazon-
1:9000/data/csv/patients.csv', schema= schema, header=True)
rc=rc.where("ID!='64066 Johns Bridge Suite 317 Eastham MA 02642 US'")

split_col = pyspark.sql.functions.split(rc['FIRST'], '[0-9]+')
df = rc.withColumn('FIRST', split_col.getItem(0))
split_col = pyspark.sql.functions.split(df['LAST'], '[0-9]+')
df = df.withColumn('LAST', split_col.getItem(0))
split_col = pyspark.sql.functions.split(df['MAIDEN'], '[0-9]+')
df = df.withColumn('MAIDEN', split_col.getItem(0))
te=df.select(concat_ws(' ',df.FIRST,df.LAST).alias("NAME"),df.ID)

name_dict = {row['ID']: row['NAME']
              for row in te.collect()}
broadcast=spark.sparkContext.broadcast(name_dict)
def nameid_convert(code):
    return broadcaste.value[code]

name_dict
```

Out[5]:

```
{'33f33990-ae8b-4be8-938f-e47ad473abfe': 'S false',
 '660bec03-9e58-47f2-98b9-2f1c564f3838': 'Geovany Reichert',
 '5125d2b2-3aef-4ae2-aa5c-335f7e206b92': 'Tianna Kuphal',
 '26626faf-cbd5-48d5-a3bf-a7b21ae08e4b': 'Ryleigh Mraz',
 'f509a0f0-77ef-477f-977d-e2784a241b52': 'Amparo Bergstrom',
 '4c763cac-b1df-4bcc-b89c-834942c4d3d6': 'Demarco Osinski',
 '8be2ce98-4bdd-4f50-b119-0e83811fc73a': 'Esperanza Koss',
 'eed62b4a-1099-47ec-a2ac-d953830b44d6': 'Dejah Towne',
```

In []:

Reading a new file called observations.csv

In [6]:

```
prod=spark.read.csv('hdfs://amazon-1:9000/data/csv/observations.csv',header=True)
```

Creating a another column for Patient ID

In [27]:

```
col=['DATE',
     'PATIENT',
     'ENCOUNTER',
     'CODE',
     'DESCRIPTION',
     'VALUE',
     'UNITS']
col.append("PATIENT_ID")
prod=spark.read.csv('hdfs://amazon-1:9000/data/csv/observations.csv',header=True).limit(10000)
```

Using the broadcast function. linking the Patient ID and Patient Name into this observations table.

In [28]:

```
prod=prod.where("PATIENT != 'db04ab6a-83d4-4b7a-a492-050133c04662'")
prod1=prod.rdd.map(lambda
x:(x[0],nameid_convert(x[1]),x[2],x[3],x[4],x[5],x[6],x[1]))).toDF(col)
prod1=prod1.select(prod1.DATE,prod1.PATIENT_ID,prod1.PATIENT,prod1.DESCRPTION,prod1.VALUE,prod1.UNITS)
prod1.show(truncate=False)
```

```
+-----+-----+-----+-----+-----+-----+
-----+
|DATE      |PATIENT      |ENCOUNTER      |CODE      |DE
SCRIPTION      |VALUE |UNITS|PATIENT_ID
|
+-----+-----+-----+-----+-----+-----+
-----+
|2014-08-17|Brooke Bartell |a7e46792-5260-4c01-acca-7a33bd2d361a|2085-9 |Hi
gh Density Lipoprotein Cholesterol |80.0 |mg/dL|81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2014-12-29|Brooke Bartell |fae60722-56e2-4612-824b-b693f11504d5|8331-1 |Or
al temperature |37.0 |Cel |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2015-09-04|Brooke Bartell |b84b0442-1b71-4c9a-b765-2bfb7825c095|8302-2 |Bo
dy Height |151.11|cm |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2015-09-04|Brooke Bartell |b84b0442-1b71-4c9a-b765-2bfb7825c095|29463-7|Bo
dy Weight |107.14|kg |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2015-09-04|Brooke Bartell |b84b0442-1b71-4c9a-b765-2bfb7825c095|39156-5|Bo
dy Mass Index |46.92 |kg/m2|81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
```

```

|2015-09-04|Brooke Bartell      |b84b0442-1b71-4c9a-b765-2bfb7825c095|8480-6 |Sy
stolic Blood Pressure           |103.0 |mmHg |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2015-09-04|Brooke Bartell      |b84b0442-1b71-4c9a-b765-2bfb7825c095|8462-4 |Di
astolic Blood Pressure          |80.0  |mmHg |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2016-10-06|Brooke Bartell      |c92db87f-c434-4ab0-8091-7341f107d011|8302-2 |Bo
dy Height                       |151.11|cm   |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2016-10-06|Brooke Bartell      |c92db87f-c434-4ab0-8091-7341f107d011|29463-7|Bo
dy Weight                       |105.82|kg   |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2016-10-06|Brooke Bartell      |c92db87f-c434-4ab0-8091-7341f107d011|39156-5|Bo
dy Mass Index                   |46.34 |kg/m2|81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2016-10-06|Brooke Bartell      |c92db87f-c434-4ab0-8091-7341f107d011|8480-6 |Sy
stolic Blood Pressure           |131.0 |mmHg |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2016-10-06|Brooke Bartell      |c92db87f-c434-4ab0-8091-7341f107d011|8462-4 |Di
astolic Blood Pressure          |76.0  |mmHg |81d21ad8-66c7-4878-a675-c
ecc2e8ccffc|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|4548-4 |He
moglobin Alc/Hemoglobin.total in Blood|5.8   |%      |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|8302-2 |Bo
dy Height                       |184.79|cm   |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|29463-7|Bo
dy Weight                       |113.11|kg   |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|39156-5|Bo
dy Mass Index                   |33.12 |kg/m2|ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|8480-6 |Sy
stolic Blood Pressure           |176.0 |mmHg |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|8462-4 |Di
astolic Blood Pressure          |103.0 |mmHg |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|4548-4 |He
moglobin Alc/Hemoglobin.total in Blood|5.8   |%      |ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
|2010-07-11|Brooke Hodkiewicz|58a3bc88-48ad-4296-851e-ba4ad1d26883|2339-0 |Gl
ucose                           |66.0  |mg/dL|ef5d3b50-a7f6-4a4d-a3b2-a
6cc8816b5da|
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+
-----+
only showing top 20 rows

```

In []:

Creating a data analysis on particular patient about Triglycerides

In [11]:

```
prod1=prod1.filter(prod1.DESRIPTION=='Triglycerides')
prod1=prod1.filter(prod1.PATIENT_ID=='ef5d3b50-a7f6-4a4d-a3b2-a6cc8816b5da')
prod1=prod1.select(prod1.DATE,prod1.VALUE)
prod1.show(30,truncate=False)
```

```
+-----+-----+
|DATE      |VALUE|
+-----+-----+
|2012-05-28|118.0|
|2015-07-12|133.0|
+-----+-----+
```

In [36]:

```
test=df.join(prod,df.ID==prod.PATIENT,"INNER")
test=test.select("DATE","FIRST","LAST","ID","ENCOUNTER","CODE","DESCRIPTION",
"VALUE","UNITS")
test.show()
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      DATE| FIRST|      LAST|      ID|      ENCOUNTER|  CO
DE|      DESCRIPTION| VALUE|      UNITS|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|2016-10-06|Brooke|  Bartell|81d21ad8-66c7-487...|c92db87f-c434-4ab...| 8462
-4|Diastolic Blood P...| 76.0|      mmHg|
|2016-10-06|Brooke|  Bartell|81d21ad8-66c7-487...|c92db87f-c434-4ab...| 8480
-6|Systolic Blood Pr...| 131.0|      mmHg|
|2016-10-06|Brooke|  Bartell|81d21ad8-66c7-487...|c92db87f-c434-4ab...|39156
-5|      Body Mass Index| 46.34|      kg/m2|
|2016-10-06|Brooke|  Bartell|81d21ad8-66c7-487...|c92db87f-c434-4ab...|29463
-7|      Body Weight|105.82|      kg|
|2016-10-06|Brooke|  Bartell|81d21ad8-66c7-487...|c92db87f-c434-4ab...| 8302
-2|      Body Height|151.11|      cm|
|2015-09-04|Brooke|  Bartell|81d21ad8-66c7-487...|b84b0442-1b71-4c9...| 8462
-4|Diastolic Blood P...| 80.0|      mmHg|
|2015-09-04|Brooke|  Bartell|81d21ad8-66c7-487...|b84b0442-1b71-4c9...| 8480
-6|Systolic Blood Pr...| 103.0|      mmHg|
|2015-09-04|Brooke|  Bartell|81d21ad8-66c7-487...|b84b0442-1b71-4c9...|39156
-5|      Body Mass Index| 46.92|      kg/m2|
|2015-09-04|Brooke|  Bartell|81d21ad8-66c7-487...|b84b0442-1b71-4c9...|29463
-7|      Body Weight|107.14|      kg|
|2015-09-04|Brooke|  Bartell|81d21ad8-66c7-487...|b84b0442-1b71-4c9...| 8302
-2|      Body Height|151.11|      cm|
|2014-12-29|Brooke|  Bartell|81d21ad8-66c7-487...|fae60722-56e2-461...| 8331
-1|      Oral temperature| 37.0|      Cel|
|2014-08-17|Brooke|  Bartell|81d21ad8-66c7-487...|a7e46792-5260-4c0...| 2085
-9|High Density Lipo...| 80.0|      mg/dL|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...|33914
-3|Estimated Glomeru...| 60.0|mL/min/{1.73_m2}|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...|20565
-8|      Carbon Dioxide| 25.0|      mmol/L|
```

```
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...| 2069
-3|          Chloride| 102.0|          mmol/L|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...| 6298
-4|          Potassium|  4.29|          mmol/L|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...| 2947
-0|          Sodium| 141.0|          mmol/L|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...|49765
-1|          Calcium|   9.8|          mg/dL|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...|38483
-4|          Creatinine|  1.0|          mg/dL|
|2016-06-11|Brooke|Hodkiewicz|ef5d3b50-a7f6-4a4...|3e4d7160-c1e9-4db...| 6299
-2|          Urea Nitrogen| 17.0|          mg/dL|
+-----+-----+-----+-----+-----+-----+-----+-----+
--+-----+-----+-----+-----+-----+
only showing top 20 rows
```

In [40]:

```
test=test.filter(test.DESCRPTION=='Triglycerides')
test=test.filter(test.ID=='ef5d3b50-a7f6-4a4d-a3b2-a6cc8816b5da')
test=test.select(test.DATE,test.VALUE)
test.show(30,truncate=False)

+-----+-----+
|DATE      |VALUE|
+-----+-----+
|2015-07-12|133.0|
|2012-05-28|118.0|
+-----+-----+
```

Finding out the observations collected about a particular patient

In [64]:

```
#df.show()
prod=spark.read.csv('hdfs://amazon-
1:9000/data/csv/observations.csv',header=True)
df1=prod.filter(prod.PATIENT=='0a15b603-a323-49b6-9243-150d414ffc9c')
df1.show()

+-----+-----+-----+-----+-----+-----+
--+-----+-----+
|      DATE|          PATIENT|          ENCOUNTER|  CODE|  DESCRI
PTION|  VALUE|  UNITS|
+-----+-----+-----+-----+-----+-----+
--+-----+-----+
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 4548-4|Hemoglobin A1c/
He...|  5.8|    %|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 8302-2|          Body H
eight|175.46|   cm|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...|29463-7|          Body W
eight| 98.35|   kg|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...|39156-5|          Body Mass
Index| 31.94| kg/m2|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 8480-6|Systolic Blood
Pr...|125.0| mmHg|
```

```
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 8462-4|Diastolic Blood
P...| 74.0| mmHg|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 4548-4|Hemoglobin A1c/
He...| 5.8| %|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 2339-0|          Gl
ucose| 82.0| mg/dL|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 6299-2|          Urea Nit
rogen| 20.0| mg/dL|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...|38483-4|          Creat
inine| 1.0| mg/dL|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...|49765-1|          Ca
lcium| 9.24| mg/dL|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 2947-0|          S
odium| 138.0|mmol/L|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 6298-4|          Pota
ssium| 5.08|mmol/L|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...| 2069-3|          Chl
oride| 106.0|mmol/L|
|2011-09-07|0a15b603-a323-49b...|2d832441-0ec8-4c0...|20565-8|          Carbon Di
oxide| 21.0|mmol/L|
|2013-07-21|0a15b603-a323-49b...|18c41494-d0f3-492...| 4548-4|Hemoglobin A1c/
He...| 6.3| %|
|2013-07-21|0a15b603-a323-49b...|18c41494-d0f3-492...| 8302-2|          Body H
eight|175.46| cm|
|2013-07-21|0a15b603-a323-49b...|18c41494-d0f3-492...|29463-7|          Body W
eight|101.83| kg|
|2013-07-21|0a15b603-a323-49b...|18c41494-d0f3-492...|39156-5|          Body Mass
Index| 33.07| kg/m2|
|2013-07-21|0a15b603-a323-49b...|18c41494-d0f3-492...| 8480-6|Systolic Blood
Pr...| 100.0| mmHg|
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+
only showing top 20 rows
```

In []:

```
prod=spark.read.csv('hdfs://amazon-
1:9000/data/csv/observations.csv',header=True)
prod=prod.withColumnRenamed("PATIENT","ID")
obs=prod.select(prod.ID).distinct().collect()
rc = spark.read.csv('hdfs://amazon-1:9000/data/csv/patients.csv',header=True)
pat=rc.select(rc.ID).collect()
filtered = [i for i in obs if not i in pat]
fildf=spark.createDataFrame(filtered)
from pyspark.sql.functions import to_timestamp,col,lit

fildf1=fildf.select(fildf.ID,lit("null").alias("NAME"))
fil=fildf1.collect()
fil={row['ID']: row['NAME'] for row in fil}
fil
```

JOIN function

In [47]:

```

rc = spark.read.csv('hdfs://amazon-
1:9000/data/csv/patients.csv', schema=schema, header=True)
rc=rc.where("ID!='64066 Johns Bridge Suite 317 Eastham MA 02642 US'")
split_col = pyspark.sql.functions.split(rc['FIRST'], '[0-9]+')
df = rc.withColumn('FIRST', split_col.getItem(0))
split_col = pyspark.sql.functions.split(df['LAST'], '[0-9]+')
df = df.withColumn('LAST', split_col.getItem(0))
split_col = pyspark.sql.functions.split(df['MAIDEN'], '[0-9]+')
df = df.withColumn('MAIDEN', split_col.getItem(0))
prod=spark.read.csv('hdfs://amazon-
1:9000/data/csv/observations.csv', header=True)
prod=prod.where("PATIENT !='db04ab6a-83d4-4b7a-a492-050133c04662'")
test=df.join(prod, df.ID==prod.PATIENT, "INNER")
test=test.select("DATE", "FIRST", "LAST", "ID", "ENCOUNTER", "CODE", "DESCRIPTION",
"VALUE", "UNITS")
test=test.filter(test.DESCRPTION=='Triglycerides')
test=test.filter(test.ID=='81d21ad8-66c7-4878-a675-cecc2e8ccffc')
test=test.select(test.DATE, test.VALUE)
test.show(30, truncate=False)

```

```

+-----+-----+
| DATE      | VALUE |
+-----+-----+
| 2011-06-22 | 104.0 |
| 2014-08-17 | 141.0 |
+-----+-----+

```

COUNT FUNCTION

```

prod.groupby('PATIENT').count().orderBy('count', ascending=False).show(10, truncate=False)

```

ORDER FUNCTION

```

from pyspark.sql.functions import substring
encount=spark.read.csv("hdfs://amazon-
1:9000/data/csv/encounters.csv", header=True)
encount=encount.filter(encount.REASONDESCRIPTION=='Viral sinusitis
(disorder)')
encount=encount.select('DATE', substring('DATE', 1, 4).alias('YEAR'))
encount=encount.groupby('YEAR').count().orderBy('count', ascending=False)
encount.show(5)

```


Apache Drill Commands

```
In [ ]: M Selecting the encounters table

In [0]: M SELECT *
        FROM
        hdfs.`/data/csv/encounters.csv`

In [ ]: M COUNT Function

In [1]: M SELECT YEAR,COUNT(*) as COUNT FROM
        (SELECT LEFT(`DATE`,4) YEAR, REASONDESCRIPTION
        from hdfs.`/data/csv/encounters.csv` AS O1
        WHERE O1.REASONDESCRIPTION='Viral sinusitis (disorder)')
        GROUP BY YEAR HAVING COUNT(*) > 1
        ORDER BY COUNT DESC

In [ ]: M

In [2]: M

In [3]: M

In [ ]: M ORDER Function

In [4]: M SELECT DESCRIPTION, COUNT(*) AS NumOfTimes
        FROM hdfs.`/data/csv/observations.csv` AS OBSERVATIONS
        GROUP BY DESCRIPTION HAVING COUNT(*) > 10
        ORDER BY NumOfTimes DESC

In [ ]: M JOIN Function

In [5]: M SELECT `DATE` , VALUE
        FROM
        (SELECT OBSERVATIONS.patient ID,OBSERVATIONS.`DATE` `DATE`,concat(substr(patients.FIRST, '[A-z]*'),' ',substr(patients.LAST,
        FROM hdfs.`/data/csv/observations.csv` AS OBSERVATIONS
        JOIN
        hdfs.`/data/csv/patients.csv` AS patients
        ON OBSERVATIONS.PATIENT=patients.Id)
        WHERE ID='b0305ffc-4eae-4f95-ae42-1d818dc60baa' AND DESCRIPTION='Triglycerides'
```

Apache Spark with Deep Learning for Pneumonia Detection

Initializing the Spark Context

SC

In [1]:

Out[1]:

SparkContext

[Spark UI](#)

Version

v2.4.8

Master

spark://amazon-1:7077

AppName

PySparkShell

In [2]:

```
import tensorflow
import keras
import h5py
import sparkdl
```

In [4]:

```
!ls
ChestXrayOld.ipynb                                hadoop-3.1.1.tar.gz
'KOA_Nassau_2697x1517.jpg?itok=iQEwihUn'         spark-2.4.8-bin-hadoop2.7.tgz
cst4570_suhail.pem
```

In []:

Initializing Spark deep learning library

In [7]:

```
import pyspark.sql.functions as f
import sparkdl as dl
```

In []:

Downloading and unzipping the lungs xray dataset from the web

In [27]:

```
!wget https://md-datasets-public-files-prod.s3.eu-west-
1.amazonaws.com/31ab5ede-ed34-46d4-b1bf-c63d70411497
```

In [28]:

```
!mv 31ab5ede-ed34-46d4-b1bf-c63d70411497 lungsXray.zip
```

In [30]:

```
!sudo apt install unzip
!unzip lungsXray
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 15 not upgraded.
Need to get 168 kB of archives.
After this operation, 567 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd6
4 unzip amd64 6.0-21ubuntu1.1 [168 kB]
Fetched 168 kB in 0s (10.6 MB/s) 0m
```

```
78Selecting previously unselected package unzip.
(Reading database ... 74616 files and directories currently installed.)
Preparing to unpack ../unzip_6.0-21ubuntu1.1_amd64.deb ...
7Progress: [ 0%] [.....]
.] 87Progress: [ 17%] [#####.....]
.....] 8Unpacking unzip (6.0-21ubuntu1.1) ...
```

```

7Progress: [ 33%] [#####.....
.] 87Progress: [ 50%] [#####.....
.....] 8Setting up unzip (6.0-21ubuntu1.1) ...
7Progress: [ 67%] [#####.....
.] 87Progress: [ 83%] [#####.....
.....] 8Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...

78Archive:  lungsXray.zip
  creating: chest_xray/
 inflating: chest_xray/.DS_Store
  creating: __MACOSX/
  creating: __MACOSX/chest_xray/
 inflating: __MACOSX/chest_xray/._.DS_Store
  creating: chest_xray/test/
 inflating: chest_xray/test/.DS_Store
  creating: __MACOSX/chest_xray/test/
 inflating: __MACOSX/chest_xray/test/._.DS_Store
  creating: chest_xray/test/PNEUMONIA/
 inflating: chest_xray/test/PNEUMONIA/person147_bacteria_706.jpeg
 inflating: chest_xray/test/PNEUMONIA/person100_bacteria_482.jpeg
 inflating: chest_xray/test/PNEUMONIA/person78_bacteria_382.jpeg
 inflating: chest_xray/test/PNEUMONIA/person124_bacteria_589.jpeg
 inflating: chest_xray/test/PNEUMONIA/person1647_virus_2848.jpeg
 inflating: chest_xray/test/PNEUMONIA/person1675_virus_2891.jpeg
 inflating: chest_xray/test/PNEUMONIA/person89_bacteria_440.jpeg
 inflating: chest_xray/test/PNEUMONIA/person35_virus_80.jpeg
 inflating: chest_xray/test/PNEUMONIA/person122_bacteria_582.jpeg
 inflating: chest_xray/test/PNEUMONIA/person119_bacteria_565.jpeg
 inflating: chest_xray/test/PNEUMONIA/person1662_virus_2875.jpeg
 inflating: chest_xray/test/PNEUMONIA/person85_bacteria_422.jpeg
 inflating: chest_xray/test/PNEUMONIA/person1669_virus_2884.jpeg
 inflating: chest_xray/test/PNEUMONIA/person39_virus_85.jpeg
 inflating: chest_xray/test/PNEUMONIA/person36_virus_81.jpeg
  creating: __MACOSX/chest_xray/test/PNEUMONIA/
 inflating: __MACOSX/chest_xray/test/PNEUMONIA/._person36_virus_81.jpeg

```

Inserting the chest xray file into the hdfs system

In [31]:

```
!hadoop fs -put chest_xray /user/input/lungs
```

```

imgNorm='hdfs:/user/input/lungs/train/NORMAL'
imgPne='hdfs:/user/input/lungs/train/PNEUMONIA'
imgNormtest='hdfs:/user/input/lungs/test/NORMAL'
imgPnetest='hdfs:/user/input/lungs/test/PNEUMONIA'

```

In [5]:

```
from pyspark.ml.image import ImageSchema
```

In []:

Using the training and testing dataset and splitting them into half

In [8]:

```
traindfNorm = ImageSchema.readImages(imgNorm).withColumn('label', f.lit(0))
traindfPne = ImageSchema.readImages(imgPne).withColumn('label', f.lit(1))
```

In [9]:

```
testdfNorm = ImageSchema.readImages(imgNormtest).withColumn('label',
f.lit(0))
testdfPne = ImageSchema.readImages(imgPnetest).withColumn('label', f.lit(1))
```

In [10]:

```
trainDFNorm1, trainDFNorm2 = traindfNorm.randomSplit([0.5, 0.5],24)
```

In [11]:

```
trainDFPne1, trainDFNPne2 = traindfPne.randomSplit([0.5, 0.5],24)
```

In [12]:

```
trainDF=trainDFNorm1.union(trainDFPne1)
```

```
testDF=testdfNorm.union(testdfPne)
```

In [73]:

```
trainDF.show(20)
```

```
+-----+-----+
|          image|label|
+-----+-----+
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
|[hdfs://amazon-1:...|  0|
+-----+-----+
only showing top 20 rows
```

In [12]:

```
trainDF.count()
```

Number of images in training dataset are 5232

Out[12]:

5232

In [81]:

```
#dfMessi = dl.readImages('football/messi/').withColumn('label', f.lit(0))
#dfRonaldo = dl.readImages('football/ronaldo/').withColumn('label', f.lit(1))
testDF.show(20)
```

```
+-----+-----+
|          image|label|
+-----+-----+
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
|[hdfs://amazon-1:...| 0|
+-----+-----+
only showing top 20 rows
```

In []:

Initializing the Machine Classification and DeepImageFeaturizer function with model Inception V3

In [13]:

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline
vectorizer = dl.DeepImageFeaturizer(inputCol="image",
outputCol="features",
modelName="InceptionV3")
```

In []:

Starting the training of the model with 20 iterations

In [14]:

```
logreg = LogisticRegression(maxIter=20,
labelCol="label")
pipeline = Pipeline(stages=[vectorizer, logreg])
pipeline_model = pipeline.fit(trainDF)
```

In [15]:

```
predictDF = pipeline_model.transform(testDF)
#predictDF.select('prediction', 'label').show(truncate=False)
```

In [16]:

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
scoring = predictDF.select("prediction", "label")
accuracy_score = MulticlassClassificationEvaluator(metricName="accuracy")
rate = accuracy_score.evaluate(scoring)*100
print("accuracy: {}%" .format(round(rate,2)))
accuracy: 78.37%
```

In []:

12 ETHICS FORM

Application for Ethical Approval for Research Projects

This is an application form for ethical approval for research undertaken by any Middlesex University Dubai staff and students. The person who completes this form should be the principal (or sole) Middlesex University Dubai researcher on the proposed study. After completion, this form (along with accompanying documents) should be submitted to the Research Ethics Committee (REC) for review. Student Researchers should submit to their Supervisor.

Section 1 – Applicant details

1.1 Details of Applicant (Principal Investigator or Student Researcher)		
Name: Shaik Mohammed Suhail	Department/Position: CEI	
Qualifications: M.Sc NMCC	Email: ss4209@live.mdx.ac.uk	Tel: +97450481697
1.2 Details of Supervisor for student applicants (if applicable)		
Name: Jaspreeth Singh	Programme of study/module: CST4599	
Qualifications: Associate Professor	Email: j.Sethi@mdx.ac.ae	Tel: 4330488
1.3 Details of any co-investigators (if applicable)		
Name:	Organisation:	Email:
Name:	Organisation:	Email:
Name:	Organisation:	Email:
1.4 Details of External Funding (if applicable)		

Section 2 – Details of the proposed study

2.1 Research project title	Performance evaluation of Apache Hadoop and Apache Spark using Electronic Health Records Dataset and X-Ray Images dataset to detect Pneumonia.		
2.2 Proposed start date	1/3/2022	2.3 Proposed end date	1/8/2022
2.4 Describe the aim and rationale of this study?			
Apache Spark Standalone and Apache Spark with YARN excel in different applications. Apache Spark excels when there is the excess main memory in the cluster assigned from YARN, while Spark Standalone acquires memory through its own application. This project will find what configuration one can perform better than the other data framework where the server's configuration resources are limited.			
2.5. Discuss the research questions and/or hypotheses of this study?			
Which of the following, Apache Spark or Apache Spark with YARN, can perform better when present with EHR Data?			
2.6 Details of study design, data collection methods to achieve the research aims (e.g., interviews, questionnaire, observation etc.) and/or secondary data sources (e.g., National			

Statistics) to be used in the research, proposed hypotheses, data analysis, with references and citations (where applicable). Include details of any online data collection (ie online survey, via social media).

- Developing a cluster of Virtual Machines
- Using Amazon Web Services and local Virtual Machines for this cluster
- Deploying Hadoop and integrating with YARN, HDFS, and Apache Spark
- Literature Review
- Comparing the performance between two data frameworks under different configurations with the given EHR dataset
- The EHR dataset being considered are
 - Using of Lungs X-Ray data to identify Pneumonia
 - Parsing of FHIR format Electronic Health Records

Section 3 – Initial Checklist to be completed by the applicant

3.1 Does this research involve human participants	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
If yes, please provide the following details:		
Who are your participants? Please specify any specific groups of human participants: (e.g., students, general public, specific groups etc.) The EHR data produced is artificially generated.		
How many participants will you have? (Under each category) Not Applicable		
How will participants be recruited and approached? Not Applicable		
Do you need access to groups of participants (e.g., through gatekeepers, e.g., organisations, managers, parents, schools etc.)	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
If yes, please provide details including no objection certificate(s):		
3.2 Does this research involve secondary data collection?	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
If yes, please Indicate your response below:		
3.2.1 Do you have the necessary approval to access the data*? (*If yes, please provide evidence of approval) The data provided for lungs dataset is in mendely data reference. It is indicated that the data processed is available for research purpose[1][2]. The EHR data present is aritificially generated from synthea[3] [1] https://data.mendeley.com/datasets/rscbjbr9sj/2	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No

<p>[2] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia</p> <p>[3] https://synthea.mitre.org/downloads</p> <p>[4] https://doi.org/10.1093/jamia/ocx079</p> <p>(If no, please provide details and plan of action)</p>		
<p>3.3 The outputs from research (e.g., products, reports, publications, etc.) are not likely to cause harm to others and are in-line with the local legislation</p>	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<p>If no, please explain how this can be avoided or managed:</p>		
<p>3.4 Will the study require data collection by proxy (someone else doing part of or all of your data collection)</p>	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
<p>If yes, please provide details including its rationale:</p> <p><i>Note: When collecting data by proxy, you need to ensure that the highest ethical standards and procedures are adopted by all research partners/fieldworkers</i></p>		

Section 4 – Anonymity, confidentiality, and consent for primary and secondary research

<p>4.1 Will the research involve collecting or analysing personal data or sensitive personal data? or involve sharing of confidential information beyond the initial consent given (i.e., personal data refers to information that may identify individuals e.g., name, address, date of birth, opinion, specific event, set of characteristics that would clearly identify individuals or very small groups. Sensitive personal data refers to racial or ethnic origin, political opinion, religious beliefs, trade union membership, sexual life, physical or mental health, criminal matters.)</p>	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No	<input type="checkbox"/> NA
<p>If yes, please provide details: (e.g., Justification for use personal data or sensitive personal data? How you plan to anonymise the data? Where the data will be kept and care/storage facilities etc.)</p> <p>The EHR data produced is synthetic data i.e artificially generated. Lungs X Ray has no names or any other details attached to it.</p> <p><i>Alternatively, if personal or sensitive personal data is required for the research, you must comply with the GDPR act and understand your responsibilities under the GDPR and have received data protection training. Please complete the Data Protection Checklist for Researchers</i></p>			
<p>4.2 Will lists of identity numbers/codes or pseudonyms for individuals and/or organisations (i.e., linking keys to personal identifiers) be stored securely and separately from the research data and destroyed after the study to avoid any risk of confidentiality being compromised?</p>	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
<p>If no, please provide details on how this can be avoided or managed:</p>			
<p>4.3 Will you tell participants that their data will be treated confidentially and the limits of anonymity will be made clear in your <u>Participant Information Sheet</u>? (e.g., their identities as participants will be concealed unless prior</p>	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA

consent is given to include the name of the participant in any documents resulting from the research. Consider how participants' narratives, quotes or involvement in specific events may make anonymity difficult to maintain.) Attach: Participant information sheet			
If yes, provide details on how you will ensure this:			
4.4 Will you obtain <u>Written Informed Consent</u> directly from research participants (if applicable)? Attach: Informed Consent sheet	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			
If yes, please specify how and when this will be achieved?			
4.5 Will you obtain <u>Written Informed Consent</u> directly from gatekeepers (if applicable)? Attach: Informed Consent sheet	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			
If yes, please specify how and when this will be achieved?			
4.6 Will you inform participants that their participation is <u>voluntary</u> and that they have a <u>right to withdraw</u> from the research at any time?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			
4.7 Will you have a process for managing <u>withdrawal of consent</u>? Please provide details:	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			
If yes, please provide details on how this will be managed?			
4.8 Will it be necessary for <u>participants to take part in the study without their knowledge and consent at the time, or by deception e.g., covert observation</u>?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide justification and details of how this will be managed to respect the participants/third parties involved to respect their privacy, values, and to minimise any risk of harmful consequences:			
4.9 Will you provide a <u>Written Debriefing Sheet</u>? (if applicable, also attach)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			
4.10 Will you need <u>consent from people who appear in visual data (e.g., photos or films or social media)</u>?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details on how this will be managed:			

If no, please explain why?			
4.11 Will you <u>audio or video record</u> interviews and/or observations?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details on how participants' anonymity will be maintained:			
4.12 Will your research involve <u>participants responding to internet surveys, emails, chatroom discussions, blogs, interactive games, social media and networking sites etc,</u>	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If 'yes', please explain how will you obtain permission from the website authors, or informed consent from participants, and ensure anonymity and protect confidentiality in an environment that generates significant amounts of background information e.g., data logs, IP addresses, cookies and caches and/or with low levels of system security?			
4.13 Do you have a Data Management Plan? (E.g.: Where the data will be stored, who will have access to data, how will the data be shared, how long the data will be stored, how it will be deleted/destroyed after your research completion etc.)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If no, please explain why?			

Section 5 – Avoiding harm: risk assessment and management, safety and legal issues

5.1 Will you use an <u>experimental research design</u> (ie., implement a specific plan for assigning participants to conditions and noting consequent changes?)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details of treatment/intervention (and specify if these are intrusive interventions e.g., hypnosis or physical exercise, or include the use of drugs, placebos or other substances e.g., vitamins, food substances etc.) and provide details of required resources for this study:			
5.2 Will the research involve <u>discussion of sensitive topics</u>? (e.g., sexual activity, drug use, national security etc.)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details of how possible adverse reactions will be avoided and what support will be in place to manage any adverse consequences:			
5.3 Is <u>pain or more than mild discomfort</u> likely to result from the study?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details on how this can be avoided or managed:			

5.4 Could the study induce psychological stress or anxiety or cause harm or negative consequences beyond the risks encountered in normal life?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details and state how participants will be supported:			
5.5 Will the study involve prolonged and repetitive testing ?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details, justification and state how participants will be supported and length of each data collection session, number of sessions and location of data collection:			
5.6 Will this research be conducted off-site (i.e., not on Middlesex University Dubai premises)?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details of other locations and explain how you will minimise any risks to your own health while off-site.			
5.7 Will you being alone with individual participants or group of participants place you at risk?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please state how this can be avoided or managed?			
5.8 Are there any adverse risks or safety issues (e.g., from potential hazards) that your methodology raises for you and/or for your participants or others?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please specify and provide details of mitigating actions that will be taken (e.g., travelling alone, working in hazardous conditions, discussing illegal activities on-line etc.) and how you, and your participants/third parties will be supported?			
5.9 Is the research or outputs from the research likely to cause harm to others (e.g., to their physical well-being, mental health, dignity or personal values) to an extent greater than that encountered in ordinary life?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please state how this can be avoided or managed?			

Section 6 – Research Sponsorship and/or Collaboration (if applicable)

6.1 Does the research have a sponsor	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
---	------------------------------	-----------------------------	--

(i.e., any person or organisation who provides support for the research in the form of income, use of data, facilities, materials, assistance with data collection etc.) that may have ethical implications for the research?			
If 'yes' please provide details of the role of the funder and issues:			
6.2 Does the research involve an international collaborator or research conducted overseas?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If 'yes', what ethical review procedures must this research comply with for that country, and what steps have been taken to comply with these: (e.g., Do you need local permission/approval? Are there any country specific cultural social or legal considerations that need to be taken into account? Who will be collecting the data overseas? Have you considered intellectual property issues?)			
6.3 Does this research already have or require Approval from an External Research Ethics Committee?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If 'yes' please provide details:			
6.4 Will this research or part of it be conducted in a language other than English?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If 'yes', full translations of all non-English materials will need to be submitted.			

Section 7 – Other Issues

7.1 Does the research involve any ethical and/or legal issues not already covered that should be taken into consideration?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please give details:			
7.2 Do you require training on the requirements of GDPR for researchers?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please give details:			
7.3 Does the research raise any other risks to safety for you or others that would be greater than in normal life?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details and state how this can be avoided or managed? If appropriate, complete a separate state <u>Risk Assessment Form</u> along with this application			
7.4 Will participants receive any reimbursements or payments for participating?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please provide details and justification:			
7.5 Are there any conflict of interests to be declared in relation to this research?	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> NA
If yes, please complete and attach the "Disclosure of Potential Conflict of Interest Form" along with this application			

--

Section 8 – Pre-Submission Checklist

Please mention the documents (where applicable) you will be attaching with this application:

Please check and attach the following documents where applicable:			
1. Participant Information Sheet	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
2. Informed Consent Sheet	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
3. Debriefing Sheet	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
4. Copy of questionnaire/interview guide/details of materials for data collection (including translations, visual images etc.)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
5. Letter of permission (if required from organisation where research is to be conducted)	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
6. Evidence of external approval – for access to secondary data	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
7. Completed Risk Assessment Form	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
8. Data Protection Checklist for Researchers			N/A
9. Disclosure of Conflict of Interests	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
10. Evidence of external approval – from external ethics body	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A

11. Evidence of relevant licence for research with animals/animal by-products	<input type="checkbox"/> Yes	<input type="checkbox"/> No	<input checked="" type="checkbox"/> N/A
12. If you are attaching any other documents, please provide details below:	<input checked="" type="checkbox"/> NA		

Section 9: Declaration – to be completed by student, supervisor and reviewers

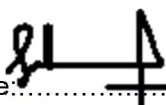
As principal investigator or student researcher I confirm that:

1. I have read and agree to abide by the relevant Code(s) of Ethics appropriate to my research field and topic.
2. I have reviewed the information provided in this form and believe it accurately represents the proposed research.
3. I have read and agree to abide by the University's Code of Practice for Research: Principles and Procedures.
4. I agree to inform my Supervisor of any adverse effects or changes to the research procedures.
5. I understand that research/data may be subject to inspection for audit purposes and I agree to participate in any audit procedures required by the Research Ethics Committee (REC) if requested.
6. I have completed and signed a risk assessment for this research study (if applicable).
7. I understand that it is my own responsibility and not that of Middlesex University Dubai to assess the personal risks involved with undertaking this research and to do my best to limit them.
8. I understand that Middlesex University Dubai is not accountable or liable for any adverse personal circumstances I may encounter as a result of the risk factors involved with undertaking this research.
9. No data collection will be undertaken before receiving approval for this application. If there is any alteration in the research methodology after approval, then submission of a Change in Ethics Approval form is required.
10. I understand that the owner of the data from this research will be the supervisor for undergraduate and master's level students' projects.

Principal Investigator or Student Researcher

Shaik Mohammed Suhail

Name:..... Signature:..... Date:.....



17/6/2022

As Supervisor, I confirm that (Student Applicant only):

1. I have reviewed all the information submitted with this research ethics application and believe it accurately represents the proposed research.
2. I accept responsibility for guiding the applicant so as to ensure compliance with the terms of the protocol and with any applicable Code(s) of Ethics.
3. I understand that research/data may be subject to inspection for audit purposes and I agree to participate in any audit procedures required by the Research Ethics Committee (REC) if requested.
4. I confirm that it is my responsibility to ensure that students under my supervision undertake a risk assessment to ensure that health and safety of themselves, participants and others is not jeopardised during the course of this study.
5. I understand that personal data about me contained in this form will be managed in accordance with the GDPR Act.
6. I have seen and signed a risk assessment for this research study (if applicable).

Supervisor's recommendation to the REC		
This is a low risk project and all ethical, legal and safety issues have been sufficiently addressed	<input type="checkbox"/> Yes	<input type="checkbox"/> No

Supervisor Name :..... Signature:..... Date:.....

As peer-reviewer I confirm that (Student Research Applications only):

1. I have carefully reviewed the ethics application
2. I have relevant knowledge of the research topic
3. I have no involvement in the study
4. Declare any conflicts of interest which may influence the peer review process
5. Act in confidence and not disclose the content or outcome of the process to anyone other than to REC and those responsible in research supervision)

Peer Reviewer Assessment	
This is a low risk project	<input type="checkbox"/>
This is a high risk project and therefore recommends full review by the University REC	<input type="checkbox"/>

Peer Reviewer Decision (For Low Risk Projects) (Please select one)	
1. Approved	<input type="checkbox"/>
2. Approved with minor amendments (Please provide details):	<input type="checkbox"/>
3. Revisions and further information required (Please provide details):	<input type="checkbox"/>
4. Not Approved for the following reasons:	<input type="checkbox"/>

Peer Reviewer Name:.....Signature:..... Date:.....

FOR RESEARCH ETHICS COMMITTEE (REC) USE ONLY

Research Committee Decision (For Staff Application or High Risk Student Projects) (Please select one)	
1. Approved	<input type="checkbox"/>
2. Approved with minor amendments (Please provide details):	<input type="checkbox"/>
3. Revisions and further information required (Please provide details):	<input type="checkbox"/>
4. Not Approved for the following reasons:	<input type="checkbox"/>

Name of the Chair of the Research Ethics Committee or nominee (If applicable):.....

Signature:..... Date:.....

ADDITIONAL NOTES FOR COMPLETING THIS FORM

- 1) Refer to Middlesex University Research Ethics section on the University intranet

- 2) Please read Middlesex University's Code of Practice for Research: Principles and Procedures available on the University intranet
- 3) Please read and ensure compliance with Data Protection under the General Data Protection Regulation (GDPR)
- 4) Please note that a student (UG, PG taught or research) cannot be the Principal Investigator for ethics purposes
- 5) External ethics approval is required from some organisations, agencies and local authorities that have their own ethics processes and require completion of additional ethical approval forms and processing in addition to the MU process. It is your responsibility to check whether additional permissions apply to you.
- 6) Accompanying forms and checklists are available on the University Intranet. This include but not limited to:
 - The Middlesex University Risk Assessment Form is available on the University intranet
 - Disclosure of Potential Conflict of Interest form is available on the University intranet
 - Data Protection Act Checklist for Researchers is available on the University intranet
 - Child Parent Consent Form
 - Gate Keeper Letter
- 7) Templates for Participant information sheet, Informed consent sheet, Debriefing guide and other related materials are available on the University intranet

Appendix: Data protection

As stated in the privacy policy, Middlesex University is required by law to comply with the Data Protection Act, 1998 (the 1998 Act). To comply with the law, information is collected and used fairly, stored safely and not disclosed to any other person unlawfully. To do this Middlesex University complies with the Data Protection Principles which are set out in the 1998 Act. In summary these state that personal data shall be:

- Processed fairly and lawfully and shall not be processed unless certain conditions are met.
- Obtained for specified and lawful purposes and not further processed in a manner incompatible with that purpose.
- Adequate, relevant and not excessive.
- Accurate and where necessary up to date.
- Kept for no longer than necessary.
- Processed in accordance with data subjects' rights.
- Protected by appropriate security.
- Not transferred without adequate protection. The university is committed to ensuring that current employees comply with this act regarding the confidentiality of any personal data held by the university, in whatever medium.

In addition, people whose data is recorded have the right to view that data ('right of subject access'), make corrections or have it deleted.