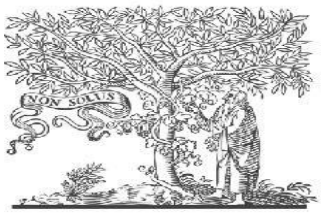
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT



ELSEVIER
SSRN

IJIEMR Transactions, online available on 04th Dec 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-12

DOI: 10.48047/IJIEMR/V10/I12/01

Title **Real-Time Maintaining of Social Distance in Covid-19 Environment using Image Processing and Big Data**

Volume 10, Issue 12, Pages: 1-6

Paper Authors
P BUSHRA ANJUM, V KRISHNA PRATAP

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Real-Time Maintaining of Social Distance in Covid-19 Environment using Image Processing and Big Data

[1]P BUSHRA ANJUM, [2]V KRISHNA PRATAP

[1]M. Tech, COMPUTER SCIENCE AND ENGINEERING,DEPARTMENT OF CSE,NRI INSTITUTE OF TECHNOLOGY(NRIIT),VISADALA ROAD PERACHARLA,MEDIKONDUR(M),GUNTUR-522438,ANDHRA PRADESH.
[2]GUIDE,ASSOCIATE PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,NRI INSTITUTE OF TECHNOLOGY(NRIIT),VISADALA ROAD PERACHARLA,MEDIKONDUR(M),GUNTUR-522438,ANDHRA PRADESH.

Abstract— This article provides an advanced technique to detect objects in video streams in real time and calculate their distance. The system composed analysis and developments to perform a stream from the camera, like video stream, distance and object detection model, incoming data stream, data stream collection and report generation. The video stream from the camera is processed with GStream. The frames from the video stream are taken by OpenCV, YOLOV3 is trained by object detection and distance model and developed by Python. Video streaming data trained with Kubeflow is published with Apache Spark and Apache Kafka. It uses HDFS used to store published data. It is used to query and analyse data in Hive, Impala, Hbase HDFS. After that Analytical reports are created. E-mail notifications can be created according to the data in the database using by Apache Oozie

Keywords—***Computer Vision, Image Processing, Big Data, Stream, Detection***

I. INTRODUCTION

In employee computer the research topic is Video surveillance that recognizes vision, detect and track objects on a range of images. Detecting and finding objects within a range of video is involved in Object detection. An object detection mechanism when the object is first displayed in the video is required in each detection method. The availability of cheap video cameras and the need for powerful computers increase the interest in object detection algorithms and interest in video analysis has also increased. Image processing refers to processing within the frame of the image or video received as input, and the resulting processing set may be the image set of the relevant parameters. Image processing purpose is to detect visible projects or to observe invisible projects. Analysis of the human movement is one of the newest and most active research topics in image processing. An important part of human perception is human movement, and to detect human's movements within the range of video is motion analysis. The vision of computers is an important research topic, which has gained great importance in the last few years, is the perception of human activities with a video. [1]

II. ARCHITECTURAL DESIGN AND METHODOLOGY

A. Architecture

The existing system was with only with open cv,to detect the image in the video they did not use any real time model.WithGStream the video stream from the camera is processed.In the video stream the frames are taken by OpenCV and YOLOV3 is trained by the distance and object detection model and developed by Python. The distance between the objects in the video stream are detected and calculated.

The streaming video is trained on the object and by OpenCV, one of the ML libraries a detection model developed. Intel created an open-source programming library called OpenCV. For PC vision applications OpenCV can be used and with the help of OpenCV an interface is made for AI.

Video streaming data trained with Kubeflow is published with Apache Kafka and Apache Spark. To store published data it uses HDFS. It is used foranalysing the data in Hive, Impala, Hbase HDFS and to query.After that Analytical reports are created. Apache can create E-mail notifications and Oozie based on data in the database. Following Figure 1 shows the architecture

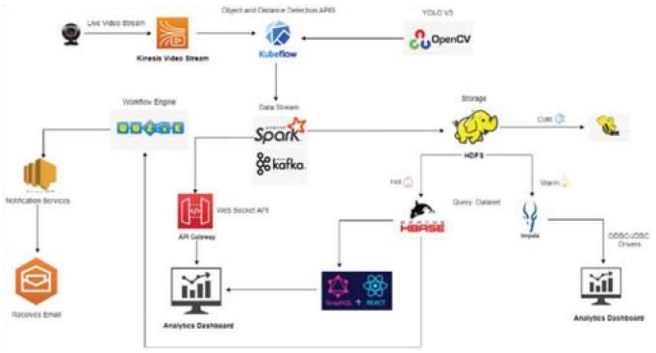


Fig. 1. System Architecture

B. Video Stream

With GStream the video stream from the camera is processed. The frames from the video stream are taken by OpenCV, YOLOV3 is trained by the distance and object detection model and developed by Python. The distance

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

between the objects in the video stream are detected and calculated. Video streaming data trained with Kubeflow is published with Apache Kafka and Apache Spark. To store published data it uses HDFS. It is used for analysing the data in Hive, Impala, Hbase HDFS and to query.

For processing video streams Gstreamer is a tool. It is both the library called from the command line tool and the software library. To provide real-time video streaming over the camera IP network is its purpose in the study. A pipeline-based open source cross-platform multimedia framework isGStreamer that connects a wide variety of media processing systems to complete complex workflows. Usually, for video streaming the HTTP or protocol is used. [3]

C. KubeFlow Pipelines

The streaming video is trained on the object andby OpenCV, one of the ML libraries a detection model developed .Intel created an open-source programming library called OpenCV. For numerical calculations OpenCV is used. For PC vision applications OpenCV can be used and with the help of OpenCV an interface is made for AI calculations.[4]

For object and distance detection with OpenCV libraryML (Machine Learning) model was created. To develop and deploy a machine learning system we use KubeFlow. Kubeflow ML provides a pipelines creation platform. A workload agnostic pipeline execution framework is KubeFlow. This means, withinKubeFlow every application which can be packaged as a container can be run. Acquisition using machine learning pipelines to orchestrate complicated workflows running on Kubernetes, Kubeflow is specially designed. Individual steps of a data processing pipeline are divided into tasks which KubeFlow provisions individually on Kubernetes. This mitigates a common resource assignment problem for high variational resource demands between intermediate pipeline tasks.[5] As seen in Figure 2, the video stream to Kubeflow is processed using the object and distance detectionmodel. The distance between two people is detectedandcalculated

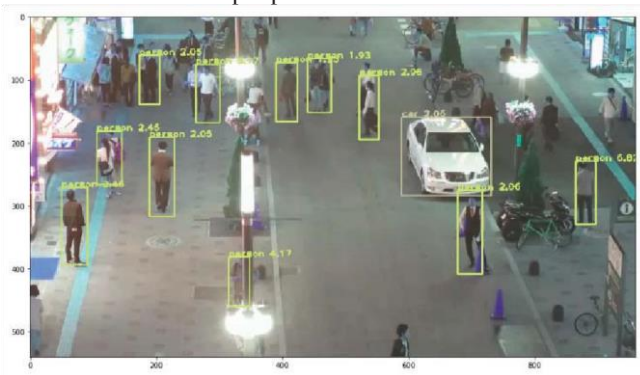


Fig. 2. Object and Distance Detection

D. Data Stream

Video streaming data trained with Kubeflow is streamed with Apache Kafka and Apache Spark. Data produced continuously by various sources and sent in the simultaneously is Data Stream. Never-ending data streams are provided. In real time the data can be analysed. A wide variety of data is included Stream data. Data such as location data, social media data, banking transactions, mobile and web log files provide this variety. [6]

Normal ML/DL problems like pandas, numpy, matplotlib, sklearn, Computer Vision, Image Processing, Big Data, Stream, Detection etc are almost all of the same libraries which we have used. To collect and analyse big data quickly and accurately is very important. The real time data ingestion and data streaming is enabled with Apache Kafka whichis a MQ (Message Queuing) system. RabbitMQ can be characterized as ActiveMQ. Apache Kafka has a queue structure. The data is sent from the source to Kafka, not directly to the analysis tool. Data is kept distributed in Apache Kafka servers,the data to be processed is then withdrawn from Kafka servers. Kafka has a cluster in itself and each node in the Kafka cluster is called a broker. Kafka cluster is the structure created by these brokers by other services.Kafka process has 3 basic building blocks. These are Producer, Topic and Consumer. The part that sends the data is the producer. The part that sends the data to the analysis toolis theConsumer.The data is kept in the publish subscriber structure is the Topic. There can be more than one topic here. In a sample system, , the analysis tool may cause data loss and the result may be manipulated when data is sent directly to the analysis tool. Also, the data will not be processed and the source will not be able to transfer when the analysis tool is broken. When Kafka and Zookeeper are in between this situation is solved. The data is kept temporarily in Kafka,if there is a problem in the analysis machine, thus preventing the problem.[7]

Apache Spark is low volume, high processing technology. The technology it is based on is Apache MapReduce. For various data processing methods such as flow processing and interactive querying Apache MapReduce has been used. In many business methods such as interactive algorithms, batch applications, interactive queries, and streaming it can be used. 10 times faster performance while working on Spark disk and 100 times faster in memory is provided. This allows the read / write operations on the disk to be reduced. Intermediate data in memory is also stored in it. Allows API creation with Java, Scala, Python languages. For interactive querying 80 top-level operators are included in Spark. Spark supports not only SQL queries, Flow data, ML and graph algorithms, but also 'Map' and 'Reduce' operations. Spark Streaming is the component that uses the fast performance capability of Spark Core for streaming. It holds data in mini-batches and applies RDD (Resilient Distributed Datasets) transforms to the data held in these mini-batches. The basis of Spark's data structure is formed by RDD. It is to keep these objects irreversibly distributed. Logical partitions are divided by each dataset in RDD so that different nodes in the cluster can handle. Python, Java, Scala objects and user-definedclasses can be hold by RDD. To make faster access and effective MapReduce Spark's RDD is used. [8]

There are two models, direct mode and receiver mode, to transfer Kafka data to Spark. Receiver mode, has a receiver.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

To receive data in the executor this receiver is used. Therefore, it covers the nucleus. In direct mode, the kernel is not used because there is no receiver. In receiver mode, zookeeper is maintaining the offset of consumer. The direct mod has to protect the offset by itself. It is seen figure 3 the working architecture of both models.

Fig. 3. Kafka Spark Streaming[9]

E. Data Storage

To store data streamed using Kafka and Spark HDFS is used. Hadoop provides safe, effective and scalable processing of data in terms of its many features. Assuming that the calculated elements and storage may fail, it works with a backup copy of the data,.[10] If an error occurs in one node, it will copy a solid copy located in another node. It starts copying again, thus ensuring that data is kept securely. Hadoop paralellime works with the principles, processing the data in parallel, thus protecting its effectiveness. Scalable, petabyte. Allows to process data of size. HDFS (Hadoop Distributed File System) is the lower layer of the Hadoop architecture [10] file system. The HDFS file system saves data across nodes in the Hadoop cluster. From outside HDFS to a client; it looks like a traditional hierarchical file system. CRUD [11] transactions can be made. In other words, files can be created, deleted, and known file operations can be performed. HDFS architecture from private nodes it is formed. These;

1. Name Node: Provides metadata service in HDFS. 2. Data Node: Provides storage blocks for HDFS

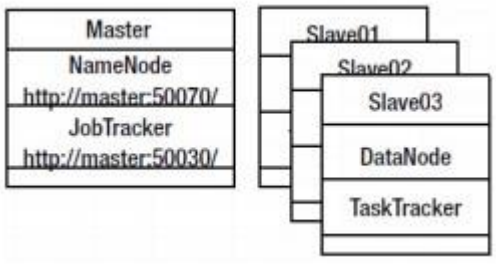


Fig .4. A master 3 slave node model[12]

In the upper layer of HDFS there is MapReduce engine. This engine is from Job Tracker and It consists of Task Tracker. It is seen figure 4.

In HDFS files are divided into blocks. These blocks are copied to computers (replicate). Block their size is usually 64 MB. The number of copies of the data and block size can be determined by the user. All file operations are managed with Name Node. All communications in HDFS are with TCP / IP protocol it realized. It is seen figure 5.

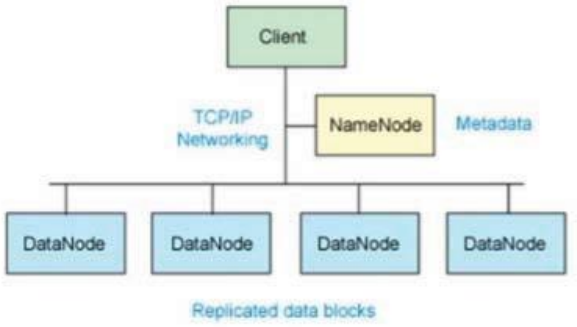


Fig .5. Simplified view of the hadoop cluster [13]

F.Query Dataset

In HDFS Hive, Impala and Hbase databases were used to query and analyse data. According to the temperature of the data databases were selected. Hot data are data that are frequently requested to be accessed. Delay rate is too low. MB-GB range is the data size. Hot data are data that are less accessible. Latency rate is higher than hot data. The data size is in the GB-TB range. Rarely requested data is cold data. The delay rate is in the minutes-hours range. The data size is in the PB-TB range. With the data temperature and database performance evaluations, hot data in the Hbase database, warm data in the Impala database and cold data in the Hive database are accessed. It is seen Figure 6 the properties of Hot, Warm and Cold data.

	Hot	Warm	Cold
Volume	MB-GB	GB-TB	PB-EB
Latency	us,ms	ms-sec	min-hrs
Durability	Low-High	High	Very High
Request Rate	Very High	High	Low
Cost / GB	$$-$	$-cc	c

Fig .6. Data Temperature

To develop SQL type scripts to do MapReduce operations Hive is a platform used.Hive language is promoted by Facebook. Hive because of its SQL like query language is often used as the interface to an Apache Hadoop based data warehouse. [14]

An opensource for 'interactive' SQL query engine for Hadoop is Impala. The interactive SQL it provides is 4-35 X faster than Hive. It provides a to write SQL queries against your Hadoop data. It is different from Hive in that it uses its own daemons to execute the queries instead of Map-Reduce. These daemons have to be installed alongside data nodes. Impala supports HIVE-QL support (ANSI-- 92 standard SQL queries with HiveQL) and ODBC drivers.[15]

Hadoop works exclusively with batch processing, which means that we can access existing data at regular intervals. For this reason, new technologies have emerged where we can access data randomly. Databases such as Hbase, Cassandra, CouchDB, Dynamo and MongoDB are databases

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

that contain large amounts of data that we can access randomly. Hbase is a column-based distributed database built on HDFS. It is a non-relational database. Its structure is very similar to Google's Big Chart. It allows you to store data with a high level of fault tolerance. They can perform real-time data processing and random read / write operations for very large data sets. The Hbase system is designed to be linear scalable. Many columns and rows contain tables together. Tables should contain items defined as primary keys. It can work with Zookeeper for high performance coordination.[16]

G.Front-End

Data streamed in Apache Kafka is transferred to React via API Gateway, for real-time reporting. Real-time reporting is done in React. An API gateway is an API management tool that intersects between a user and a collection of backend services. Using the API Gateway, RESTful APIs and WebSocket APIs are also created that enable real-time bi-directional communication applications. [17]

User interface creation is enabled by ReactJS which is a component-based JavaScript library. Notification views are available that facilitate the reading and debugging of the code in ReactJS. ReactJS can help the user create some of the easiest or most complex user interfaces and manage their own situation.[18] It is seen figure 7 the user interface and dashboards made within the scope of the study.

```
import argparse
import imutils
import cv2
import os

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", type=str, default="",
    help="path to (optional) input video file")
ap.add_argument("-o", "--output", type=str, default="",
    help="path to (optional) output video file")
ap.add_argument("-d", "--display", type=int, default=1,
    help="whether or not output frame should be displayed")
args = vars(ap.parse_args())

# load the COCO class labels our YOLO model was trained on
labelsPath = os.path.sep.join([config.MODEL_PATH, "coco.names"])
LABELS = open(labelsPath).read().strip().split("\n")

# derive the paths to the YOLO weights and model configuration
weightsPath = os.path.sep.join([config.MODEL_PATH, "yolov3.weights"])
configPath = os.path.sep.join([config.MODEL_PATH, "yolov3.cfg"])

# load our YOLO object detector trained on COCO dataset (80 classes)
print("[INFO] loading YOLO from disk...")
net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)
```

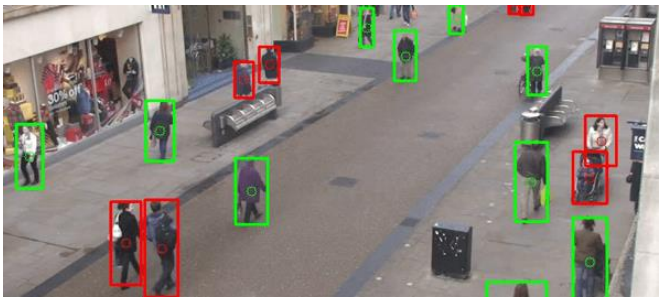
Fig .7. Social Distancing Dashboard

```
if config.USE_GPU:
    # set CUDA as the preferable backend and target
    print("[INFO] setting preferable backend and target to CUDA...")
    net.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
    net.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)

# determine only the *output* layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

# initialize the video stream and pointer to output video file
print("[INFO] accessing video stream...")
vs = cv2.VideoCapture(args["input"] if args["input"] else 0)
writer = None

# loop over the frames from the video stream
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break
```

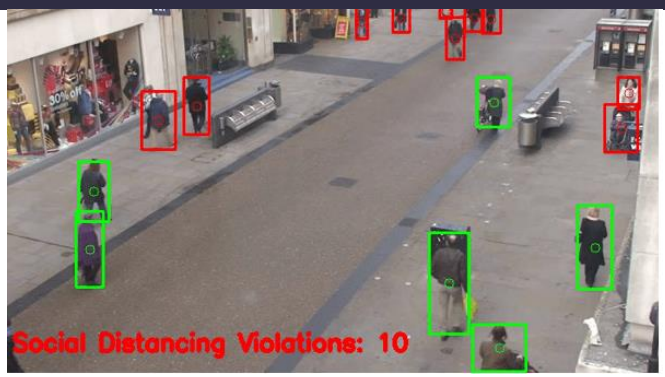
```
while True:
    # read the next frame from the file
    (grabbed, frame) = vs.read()

    # if the frame was not grabbed, then we have reached the end
    # of the stream
    if not grabbed:
        break

    # resize the frame and then detect people (and only people) in it
```

```
    # distance
    violate = set()

    # ensure there are *at least* two people detections (required in
    # order to compute our pairwise distance maps)
    if len(results) >= 2:
        # extract all centroids from the results and compute the
        # Euclidean distances between all pairs of the centroids
        centroids = np.array([r[2] for r in results])
        D = dist.cdist(centroids, centroids, metric="euclidean")

        # loop over the upper triangular of the distance matrix
        for i in range(0, D.shape[0]):
            for j in range(i + 1, D.shape[1]):
                # check to see if the distance between any two
                # centroid pairs is less than the configured number
                # of pixels
                if D[i, j] < config.MIN_DISTANCE:
                    # update our violation set with the indexes of
                    # the centroid pairs
                    violate.add(i)
                    violate.add(j)
```

```
    # writer has not been initialized, do so now
    if args["output"] != "" and writer is None:
        # initialize our video writer
        fourcc = cv2.VideoWriter_fourcc(*"MJPG")
        writer = cv2.VideoWriter(args["output"], fourcc, 25,
            (frame.shape[1], frame.shape[0]), True)

    # if the video writer is not None, write the frame to the outp
    # video file
    if writer is not None:
        writer.write(frame)
```



International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org



H. Notification

With Oozie workflows E-mail notifications are sent. Oozie is a workflow scheduler system for managing Apache Hadoop jobs. Hadoop stack that supports various Hadoop jobs (such as Java map reduction, Flow map reduction, Pig, Hive, Sqoop and Distcp) and system-specific jobs (such as Java programs and shell scripts) out of the box. Oozie is a scalable, reliable and extensible system. [19] The email action requires SMTP server configuration to be present.

III. CONCLUSION

In this research, the flow data was obtained by trained with the object and distance detection model taken from the camera video. The distance between people with the social distance and object detection model has been found in closed areas. Detected people, distance information, coordinates are sent to the stream. The data coming from the stream to the database are shown in analytical reports in historical and real-time. In cases of extreme social distance, violation densities can be sent via the notification system to inform the user. In this way, the user is provided to report the regions where the violation is density in real-time. In addition, a user interface has been created to enable the user to define cameras, generate notifications, or access real-time dashboards. Thus, it was ensured that the violation cases on the data were followed.

As a future study, different Artificial Intelligence algorithms can be used to improve performance. In addition to person detection, research and developments on mask detection algorithms will continue for analyse whether a person is wearing a mask.

ACKNOWLEDGMENT

We would like to thank Doç.Dr. Ali Hakan I? @k who supported this study. We would also like to thank Intelligence TR establishment and our analytical team for enabling us to realize this study.

REFERENCES

References

[1] Payal Panchal, Gaurav Prajapati, Savan Patel, Hinal Shah and Jitendra Nasriwala, " A Review on Object Detection and Tracking methods", International Journal for Research in Emerging Science and Technology, Volume - 2, Issue-I, January 2015.

[2] Video streaming with Gstreamer https://z25.org/static/_rd_/videostreaming_intro_plab/index.html

[3] Gobject https://en.wikipedia.org/wiki/GObject#Relation_to_GLib

[4] Mittal, Naman, A. Vaidya, and S. Kapoor. "Object detection and classification using Yolo." Int. J. Sci. Res. Eng. Trends 5 (2019): 562-565

[5] Popp, Matthias. Comprehensive Support of the Lifecycle of Machine Learning Models in Model Management Systems. MS thesis. 2019

[6] Internet: https://aws.amazon.com/tr/streaming-data/

[7] Kreps, J., Narkhede, N. and Rao, J., "Kafka: A distributed messaging system for log processing", Proceedings of 6th International Workshop on Networking Meets Databases, 2011

[8] Zaharia, M., Das, T., Li, H., Shenker, S. and Stoica, I., "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters", 4th USENIX conference on Hot Topics in Cloud Computing, 2012

[9] Internet: https://developpaper.com/optimization-andcomparison-of-reading-kafka-data-by-spark-streaming/

[10] K. Shvachko, K. Hairong, S. Radia,R.Chansle, "The Hadoop Distributed File System", Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium, Incline Village, NV, USA (2010)

[11] Internet: Create,read,update and delete http://en.wikipedia.org/wiki/Create,_read,_update_and_delete (2010)

[12] J. Venner, "Pro Hadoop", Apress, 1st ed. (2009)

[13] Internet: Ken Mann,M. Tim Jones Distributed computing with Linux and Hadoop, http://www.ibm.com/developerworks/linux/library/l-hadoop/ (2010)

[14] Pol, Urmila. (2016). Big Data Analysis: Comparision of Hadoop MapReduce, Pig and Hive. International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization). 5. 9687-9693. 10.15680/IJIRSET.2015.0506026.

[15] Maposa, Tererai& Sethi, Manoj. (2018). SQL-on-Hadoop: The Most Probable Future in Big Data Analytics.

[16] Rick Cattell. 2011. Scalable SQL and NoSQL data stores. SIGMOD Rec. 39, 4 (December 2010), 12–27.

DOI: https://doi.org/10.1145/1978915.1978919

[17] Internet: https://www.nginx.com/learn/api-gateway/

[18] Internet: HTTPs://2019-spring-web-dev.readthedocs.io/en/latest/final/taylor/index.html

[19] Internet: https://oozie.apache.org/