Strathmore UNIVERSITY

SU+ @ Strathmore University Library

Electronic Theses and Dissertations

2022

A HDF5 data compression model for IoT applications.

Chabari, Risper Nkatha
School of Computing and Engineering Sciences
Strathmore University

Recommended Citation

Chabari, R. N. (2022). *A HDF5 data compression model for IoT applications* [Strathmore University].

http://hdl.handle.net/11071/13177

Follow this and additional works at: http://hdl.handle.net/11071/13177

This work is availed for free and open access by Strathmore University Library.
It has been accepted for digital distribution by an authorized administrator of SU+ @Strathmore University.
For more information, please contact library@strathmore.edu

A HDF5 Data Compression Model for IoT Applications

RISPER NKATHA CHABARI

135282

Master of Science in Information Technology

2022

A HDF5 Data Compression Model for IoT Applications

RISPER NKATHA CHABARI

135282

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

October, 2022

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University

Risper Nkatha Chabari

Signature: .....

Date: 25th July, 2022

Approval

The thesis of Risper Nkatha was reviewed and approved by the following:

Dr. Dickson Odhiambo Owuor

School of Computing and Engineering Sciences

Strathmore University

Dr. Julius Butime,

Dean, School of Computing & Engineering Sciences,

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Internet of things has become an integral part of the modern digital ecosystem. According to current reports, more than 13.8 billion devices are connected as of 2021 and this massive adoption will surpass 30.9 billion devices by 2025. This means that IoT devices will become more prevalent and significant in our daily lives. Miniaturization in form factor chipsets and modules has contributed to cost-effective and faster running computer components. As a result of these technological advancements and mass adoption, the number of connected devices to the internet has been on the rise, leading to the generation of data, in high volumes, velocity, veracity, and variety. The major challenge is the data deluge experienced which in turn makes it challenging to visualize, store and analyse data generated in various formats. The adoption of relational databases like MySQL has been majorly used to store IoT data. However, it can only handle structured data because data is organized in tables with high consistency. On the other hand, NoSQL has also been adopted because of its capabilities of storing large volumes of data and has no reliance on a relational schema or any consistency requirements. This makes it suitable for only unstructured data. This outlines a clear need of adopting an effective way of storing and data managing IoT heterogeneous data in a compressed and self-describing format. Furthermore, there is no one- size all approach of managing heterogeneous data in IoT architecture. It is in the paradigm that this research solved this challenge by creating a tool that compresses heterogeneous data while saving it in a HDF5 format. The format of the data used was in .csv datasets. These data was parsed in the storage tool and data tool of the HDF5 for compression and conversion. The tool managed to achieve a good compression ratio percentage of 89.34% decrease from the original file. The output of the compressed file was represented on an external interactor called hdfview to validate that the algorithm used was lossless.

Keywords: Hierarchical Data Format version 5(HDF5), Internet of Things, Heterogeneous

Table of Contents

List of Figures

List of tables

Abbreviations and Symbols

CPU – Central Processing Unit

CSV - Comma Separated Value

DB - Database

GPS - Global Positioning System

HDF5 - Hierarchical Data Format version 5

Hdfview - Hierarchical Data Format Viewer

I/O – Input Output

IDC - International Data Corporation

IDE - Integrated Development Environment

IoT - Internet of Things

JPSS - Joint Polar Satellite System

NASA - National Aeronautics and Space Administration

NCSA - National Center for Supercomputing Applications

SQL - Structured Query Language,

QET – Query Entry Time

RDF - Resource Description Framework

RFID - Radio-frequency identification

Acknowledgments

I would like to extend my sincere gratitude to my supervisor Dr.Dickson Owour for the guidance and support. Thank you for encouraging and inspiring me in this pursuit. To my family, thank you for your enduring support during this course. Special thanks to my parents Edward Chabari and Hellen Chabari for your prayers, support and encouragement. In a special way, I would like to thank my friends Mariamu, Patrick, Evans, Mercy and Jebet for their constant encouragement, reminders and support they accorded me during this journey.

Chapter 1: Introduction

1.1 Background of the study

The Internet of Things is a system of interrelated and connected computing devices with the ability to interact with each other (In & Kyoochun, 2015). The rapid increase in the advancement of sensor data collection has enabled many devices and computational systems to be intelligent, enabling the IoT vision application (Mahmoud, Yousuf, Aloul, & Zualkernan, 2016). These entities are capable of transferring data across networks, without requiring human-to-computer or human-to-human interaction. The IoT applications have massively enhanced the quality of many existing systems by lowering costs, enhancing functionalities, automation, and increasing availabilities. Typical examples of IoT applications include camera networks, patient monitoring, industrial internet, smart homes, connected cars, smart retail, and smart supply chains (Adel & Rabie, 2017).

Internet of Things has been around since the start of the fourth industrial revolution. Since then, it has evolved and grown to become a dominant technological phenomenon. According to current reports, more than 13.8 billion IoT devices are connected as of 2021. Experts forecast that this number will surpass 30.9 billion devices by 2025 (Lionel, 2021).This means that IoT devices will become more prevalent and significant in our daily lives.Since technological advancements have increased the number of devices connected to the internet, huge amounts of data are being generated, transmitted, and used by these devices (Ukil, Bandyopadhyay, & Pal, IoT Data Compression: Sensor-Agnostic Approach., 2015).

The IoT paradigm has transformed many areas in human lives; however, several challenges arise because of the massive adoption. One of the main challenges is the mega–data produced by the IoT devices, which has become difficult to capture, visualize and analyse (Adel & Rabie, 2017). The proliferation of the data raises a need to have multimodal processing capabilities in the IoT data management architecture (Wei, et al., 2020). This will mean that data coming in form of text, video and audio will be stored in a self-describing hierarchical structure.

Dealing with the increasing volume of data is not the only concern of managing stored IoT data. The data captured by IoT devices is either structured, semi-structured, or unstructured data. Since

the data is not uniform, there is no one-size-fits-all approach for storing IoT data. Transforming, aggregating and integrating IoT data to prepare it for analytics while keeping track of data provenance will help solve this problem (Anna & Satwik, 2017).Lastly, IoT data is increasing very fast and needs to be managed more efficiently requiring a more effective storage management structure.

The Hierarchical Data Format version 5 (HDF5) is a simple file format that has a powerful way of organizing information. It is used for handling large amounts of data and includes metadata that provides context and makes the data understandable. It was developed to solve the need of managing the diverse collection of data objects. It also supports data slicing, which extracts a portion of data that is required from a large dataset, without reading the entire file into memory. HDF5 has been widely used by the National Aeronautics and Space Administration (NASA), Joint Polar Satellite System (JPSS) amongst others, this is because it has user-defined data types, multiple unlimited dimensions and per variable data compression, which makes it suitable for data management and data analysis (Berrick, 2022). This makes HDF5 a suitable tool for data management and compression of voluminous IoT data. Hence, this research will focus on developing a HDF5 data compression tool for IoT applications.

1.2 Problem Statement

Internet of things (IoT) has become an integral part of the modern digital ecosystem. Numerous and diverse types of sensors generate a plethora of data that needs to be stored and processed (Bose, Bandyopadhyay, Kumar, Bhattacharyya & Pal, 2016). The collection, storage and usage of IoT data has affected the utmost aspects of the IT infrastructure. This is because most of the data that is collected and stored, not all of it that is used for data analysis hence taking up a lot of storage space.

In data storage, many traditional data storage platforms were based on relational databases. Although relational databases are still prominent for data storage, they could hardly provide sufficient performance in big data environment. As complements to relational databases, those tools that can efficiently process massive data in distributed environment, such as Hadoop, and NoSQL

database are attracting increasing attention. NoSQL databases provide a series of features that relational databases cannot provide, such as horizontal scalability, memory and distributed index, dynamically modifying data schema.On the other hand, NoSQL database lacks completed atomicity, consistency, isolation, durability (ACID) constraint and support for some complicated queries.

This limitations reduce the applicability of the databases to store increasing heterogeneous data in IoT. The hierarchical data file format version 5 (HDF5) created, compressed, stored and shared datasets in the same structure (Heber, 2011). This gave it a fast write and read access time since you could read partial datasets that you want to share without reading the entire file. Additionally, only a single file containing all compressed datasets could be shared and viewed. This made it suitable to support massive heterogeneous data in this case IoT data. Therefore, this research will focus on developing a HDF5 data compression tool for IoT applications.

1.3 Objectives

1.3.1 *General Objectives*

The purpose of this study was to develop a HDF5 data compression tool for IoT applications.

1.3.2 *Specific Objectives*

This study was guided by the following objectives;

1. To investigate the impact of increasing data in IoT devices.
2. To review existing data management tools for in storing data in IoT applications.
3. To develop a HDF5 tool for compressing IoT data.
4. To validate the developed HDF5 tool.

1.4 Research Questions

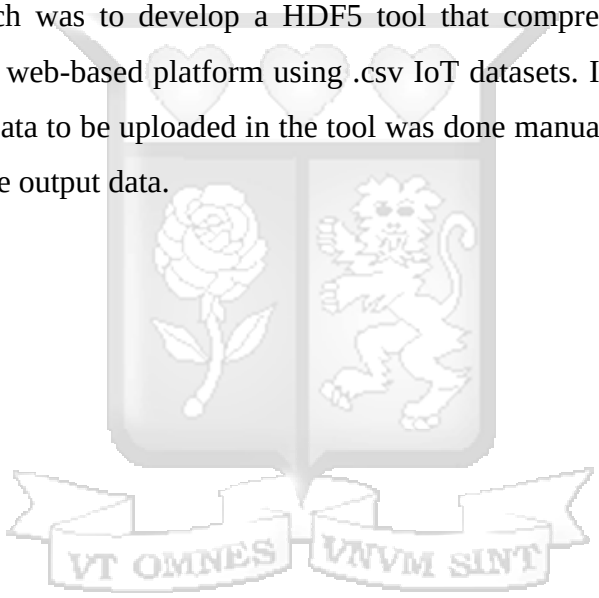
1. How is increasing data affecting IoT devices?
2. Which existing data management tools have been used to store IoT data?
3. How will the proposed HDF5 tool for compressing IoT data be developed?
4. How will the HDF5 tool be validated?

1.5 Justification

This research created a data compression tool for IoT applications while storing data in a .hdf5 file format, this would be beneficial to data analysts who may wish to analyse large IoT datasets. This was made possible since, the HDF5 file format had a simple file structure that had wide support and compatibility. Additionally, you can append datasets dynamically to an existing group without having to create an entirely new group. This made it suitable for data sharing since the tools combined all compressed files into one single file.

1.6 Scope and Limitation

The focus of this research was to develop a HDF5 tool that compresses IoT data. This was implemented on the local web-based platform using .csv IoT datasets. It was not applicable as a user-facing application. Data to be uploaded in the tool was done manually and data analysis was not to be carried out on the output data.



Chapter 2: Literature Review

2.1 Introduction

This chapter is divided into sections motivated by the objectives of the research listed in chapter one. The impact of increasing data in IoT devices is discussed, a brief introduction of features, data life cycle and how data is stored in IoT devices. Tools of interest for this research and their applications in data storage are also discussed with their objectives, weaknesses and how they relate to the objectives of this research.

2.2 Theoretical Framework

Wei et al., (2020) stated that the proliferation of data raised a need of having multimodal processing capabilities in the IoT data management architecture. As a result, this would mean that data would be stored in a self-describing format that was not limited to any storage structure. This would curb the challenge of storing large amounts of heterogeneous data. Since the HDF5 tool supported data slicing and chunking it reduced the operational and time complexity of the tool. As a result, reviewing the impact of increasing data in IoT devices, the current data storage tools and the HDF5 data model guided me in the implementation the HDF5 tool that compressed datasets into one dataset.

2.3 Impact of Increasing Data in IoT Devices

According to Harjinder and Ajay (2018), the volume of data captured by IoT devices has been rising and this is opening up new sources of data, which has affected big data analytics. Because of the voluminous data, it has become difficult to store, process and analyse data using traditional management technologies. Hence, requiring new technologies and techniques for data management to be adopted (Somayya, R, & Tripathi, 2015).Mohsen et al., also found that integrating large IoT data was a complex and challenging issue and also suggested that technologies that could aid in the integration should be proposed.

The collection, storage and usage of IoT data has affected the utmost aspects of the IT infrastructure. This is because most of the data that is collected and stored, not all of it that is used for data analysis. According to Veritas, 2019 dark data that contains valuable consumer information that is of importance is not analysed. This is because of the incompatibility between the analytical

tools and the data types and formats used to store the data (Riley, 2019). Paul et al., (2020) mentions that linking heterogeneous data from heterogeneous data sources in IoT can create data quality issues, which results in misleading information.Rajeshwari, 2015 also mentioned that data produced would be of no use if there were no methods to properly analyse and make use of data. This was influenced by the lack of a good storage structure that can store data in a hierarchical and compressed format minimizing the storage infrastructure required to support large and complex data.

2.3.1 *Features of IoT data*

A major requirement of IoT is that all devices must be interconnected. This interconnection enables the exchange of information and generation of data between the IoT devices (Tingli, Yang, Ye, Shuo, & Wei, 2012). IoT data comes from different physical sensing devices therefore; the data comes in varying formats and attributes (Shanshan, Liang, Zisheng, Fan, & Weizhao, 2017). IoT data has the following features;

i. The data collected by RFID sensors, camera and temperature sensors, etc. shows that the data is multisource and heterogeneous because it comes with different semantics and structures.
ii. The data that is collected automatically by numerous perception devices shows that it is huge scale data.
iii. The data that has time and space attributes shows that data has a temporal-spatial association.
iv. The data has interoperability features because it facilitates data sharing and collaborative work between various IoT devices.
v. The data is multidimensional because the sensed data is included in multiple attributes and events.

2.3.2 *IoT Data Life Cycle*

Figure 2.1 illustrates the data life cycle in IoT applications. It begins with data production to data collection, filtering, aggregation, and pre-processing, and finally, storage and data archiving. Data

querying and data analysis are the endpoints that initialize requests (IoT-A.Converged Architectural Reference Model for the IoT v2.0, 2012).

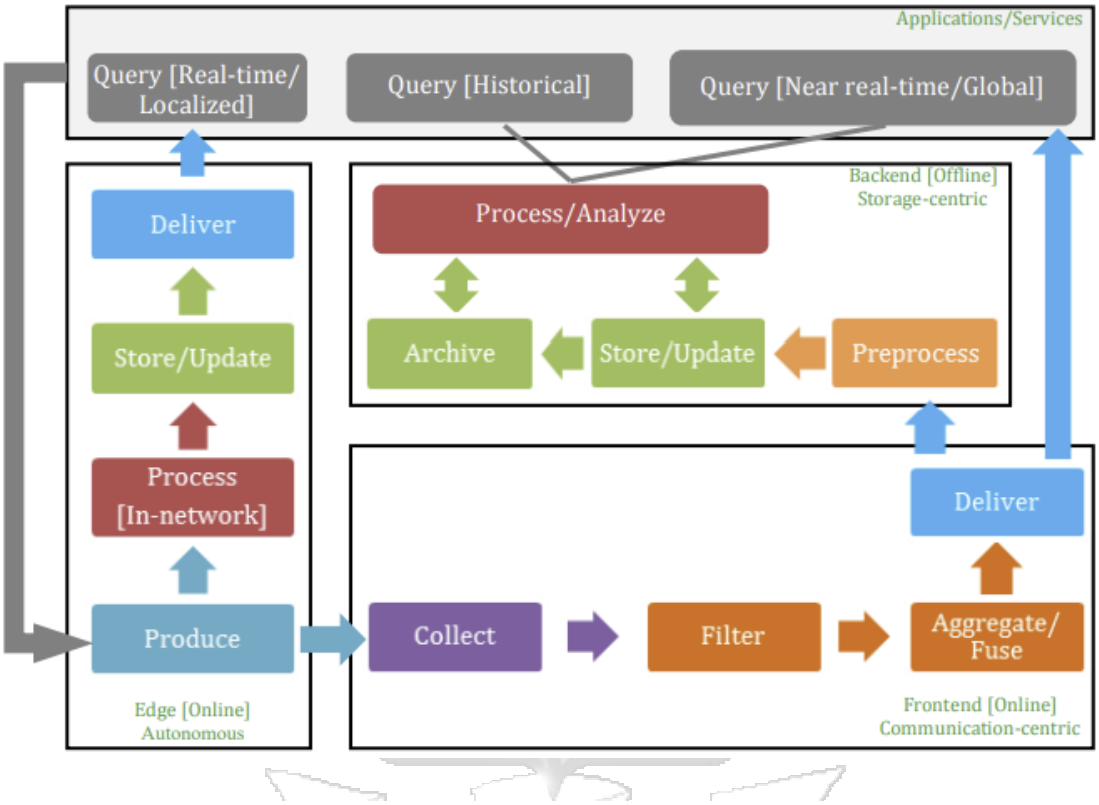


Figure 2.1 Data Management Life Cycle in IoT Applications source (Abu-Elkheir, Hayajneh, & Ali, 2013)

IoT systems may generate, process and store data in-network for local and real-time services without the need of propagating data up to the concentration points in the system that combine storage and processing elements. In the subsequent paragraphs, each component in the data management life cycle was explained.

Querying: IoT systems rely on this process to access and retrieve data this can be either real-time data collected for localized monitoring purposes or retrieving a certain view of data stored in the system for in-depth analysis.

Production: This involves the transfer and sensing of data by the IoT systems, which are used for periodic reporting. This date is usually time-stamped geo stamped and may contain audio, video and image content with varying degrees of complexity.

Collection: Sensors within the IoT system store data intervals or report it to respective components. Data collected within the network is filtered, processed and fused into compact forms for efficient transmission. Technologies such as ZigBee, WI-FI, and cellular are used by smart objects to send data.

Aggregation: This technique deploys summarization and merging operations in real-time to compress the volume of data that should be stored and transmitted (Luo, Wu, Sun, & Chen, 2009).

Delivery: When data is being filtered, fused and being processed within the IoT system the data may need to be sent further up in the system for storage or analysis. To transfer data permanently to the data stores wired and wireless broadband communications are used.

Pre-processing: IoT data comes with multi-dimensional features and formats and data cleaning techniques need to be applied in this stage before committing the data for storage (Chen, Tseng, & Lian, 2010).

Storage: This phase handles continuous updates and storage of data as it becomes available. Archiving of data is also done in this phase, which is the offline long-term storage of data that is not needed for ongoing operations.

Analysis: This phase involves the retrieval and analysis operations performed on stored and archived data to gain insights into historical data and predict future trends, or detect abnormalities in the data that may trigger further investigation or action.

Following the above data management architecture, relational database management systems like SQL are the most popular storage choice for IoT data. This involves the organization of data in tables with interrelationships and metadata for efficient retrieval at later stages (Ramakrishnan & Gehrke, 2011). Additionally, NoSQL DB like MongoDB or SOLR is also used since; they have capabilities of storing large volumes of IoT data with no reliance on a relational schema or any consistency requirements of a typical relational database system (Cattell, 2010). Storage can also

be decentralized for autonomous IoT systems, where data is kept at the objects that generate it and is not sent up the system. However, due to the limited capabilities of IoT architecture, storage capacity on-device remains limited in comparison to the centralized storage that sends data to the specified data center for storage.

2.4 How is Data stored in IoT

There are three main forms of data storage local, distributed, and centralized (Tsiftes & Dunkels, 2011). The local form indicates that the data sampled by sensors like RFID readers and GPS sensors are stored in the local storage unit. The distributed form means that the data is stored in some nodes in a network through distributed technologies. Intermediary mechanisms are used to ensure that there is access to the data. The centralized form refers to the data that is collected by a node and then is sent to and stored in a data center. Because of the limited storage capacity and the constrained power of the sensors, local and distributed forms are not suitable for IoT applications that need huge-scale data and intensive queries (He & He, 2011). Therefore, due to the aforementioned limitation, IoT data statically resides in fixed or flexible databases and roams the network from dynamic and mobile objects to concentration storage points until it reaches the centralized data stores.

2.5 Current Data Storage Tools for IoT

In the area of IoT data storage and processing, numerous efforts have been made. Many storage platforms are based on relational databases and non-relational databases. Gizen et al. (2017) stated that the most commonly used relational DBMS were MySQL and NoSQL. They stored data rows and columns in tables with high data consistency, and would be resource intensive if large data was to be stored. However, NoSQL was not dependent on table definitions and rigid schemas. Since some relational database system could hardly provide sufficient performance in big data environment, Lihong, et al.(2014) stated that NoSQL database were attracting increasing attention since they could provide additional features that some relational databases could not provide. This included horizontal scalability, distributed index and dynamic modification of data (J, Xu, G, & Z, 2012).

In the recent past, data processing was done by scanning the entire file to answer the queries each time. Oracle & MySQL allowed querying of CSV data files directly without loading the data, but it scanned the entire file each time a query was being processed (Patel & Bhise, 2019). NoSQL was the first mature raw data processing system to be used on IoT data from a research group at EPFL. The core idea behind this work was to remove the major bottleneck of data-to-query time, which is data loading. The work proposed extending the traditional query processing architecture to work directly on raw data files to answer the queries.

The most recent work of the group on data tries to choose the optimal data-caching mode automatically based on query history, cache behaviour and cache size for faster query execution (Azim, 2018). SCANRAW is another processing engine for raw files. This work focused on making query execution faster by giving priority to query execution and load data only when the I/O bandwidth or CPU is available (Cheng, Y. and Rusu, F, 2015). In contrast to Invisible loading, it may not load data with query processing, because if the I/O bandwidth is fully utilized the data loading can slow down the query execution performance (Abouzied, Abadi, & Silberschatz, 2013). Many data partitioning techniques have been used on RDF & relational data with traditional databases to improve the QET for semantic web data analysis (Vasani, 2013). Modern techniques used to process streaming data include bulk loading, in-memory databases, or cloud.

Lihong, et al.(2014) proposed a data storage framework for storing massive IoT by integrating both structured and unstructured data. It combined file repository tool and database management tool for unstructured and structured data respectively. This file repository was based on Hadoop which was an open source implementation of Map reduce. This solution had several advantages, it combined different types of databases and operated them on one interface, and it was able to handle voluminous data. It was better than the previous tools of raw data processing because it combined relational databases and No SQL, which solved their overheads. However, it was not suitable for storing huge numbers of small files since it had a block of 128 MB that would cause an overload.

According to Tonglin et al.,(2019) they showed that HDF5 achieved good perfomance while performing I/O operations in the perfomance analysis of applications. Since many storage technologies have shown not to achieve acceptable perfomance when accessing storage systems

with standard operations I/O operations (Tonglin, Quincey, Houjun, Jialin, & Suren, 2019).The curent HDF5 format that has evolved into a parrallel data management system.This bridges the gap between data access time and various storage systems.This is because it supports parrel I/O ,chunking and compression which maintains acceptable write perfomance while limiting overheads (Oliver & Ben, 2015).

All discussed related works as shown in table 2.1 have some advantages and limitations. The HDF5 application has shown to have potential advantages amongst the discussed applications and it has shown to be suitable for implementation in this research.

Table 2. 1 Comparison table of the reviewed data storage tools

Tool	Advantages	Disadvantages	Author
Data storage framework based on Hadoop	• Store structured & unstructured data	• Not suitable for large no of small files	Lihong, et al.(2014)
No SQL	• Dynamic Schema • Horizontal Scaling • Distributed Indexing	• Not suitable for storing structured data	J, Xu, G, & Z (2012)
Relational DBMS	• High data consistency in structured data	• Not suitable for large datasets • Not suitable for unstructured data.	Gizen et al. (2017)

2.6 Application of Using Hierarchical Data Format Version 5

2.6.1 *Simulation of Using Hierarchical Data Format Version 5*

The National Center of Super Computing Applications (NCSA) developed the last version five of its Hierarchical Data Format (HDF5) library. This had improved features from the version 4 hierarchical data format like; supporting large files of s more than two gigabytes, a large number of objects and compression packages (Yang, 2008).Ertl, Frisch, & Mundani 2016 developed an

input and output kernel using HDF5 for massive fluid flow. For them to be able to handle large datasets a dedicated approach to the code input and an output routine was employed. The major building block for this approach used the high-level HDF5 format.This is tailored to store array-based data in a database-like view. The key concept behind the HDF5 data tool was based on an abstract data tool, abstract storage tool and libraries that implemented the data tool and mapped the storage tool to different storage mechanisms. Datasets that contained the actual data and groups made up the general structure. Starting from a root group, groups contained additional groups or datasets themselves, which resulted in a hierarchical tree-like structure resembling a UNIX file system (Ertl, Frisch, & Mundani, 2016).

2.6.2 *HDF5 Data Tool*

HDF5 consists of a data tool for organizing data, the file format for storing data and a software that contains libraries, interfaces, tools and languages for working with this format. The data tool contains groups that have instances of zero to many groups or datasets with the supporting metadata that has dataspaces, properties and attributes. The programming tool instantiates populates and retrains objects from the data tool, while the stored data is the implementation of the storage tool (TheHDF5Group, 2015).

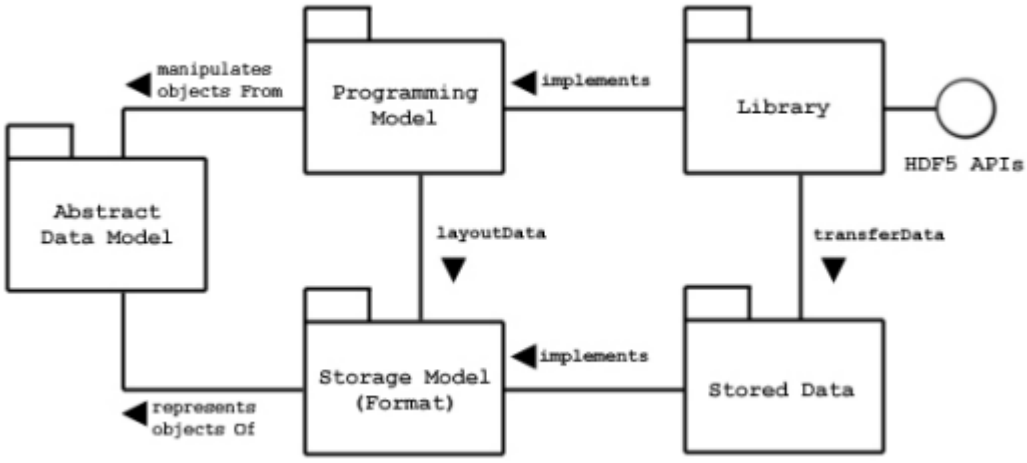


Figure 2.2 HDF5 Data Tool Implementation source (TheHDF5Group, 2015)

The most important issues that were considered while developing the I/O kernel were fixing the desired functions and determining what the HDF5 tree structure would look like. Since the

structure, the functionality, and the consideration concerning the implementation strongly influenced each other they had to be reviewed throughout the entire development.

2.6.3 *HDF5 Tree Structure*

Shown in Figure 2.2 is the tree structure of a file stored in.hdf5 format. It is organized like a directed graph. It has a root group denoted with a / which may not be a member of any other group. It has three classes of named objects a group, dataset and attributes.

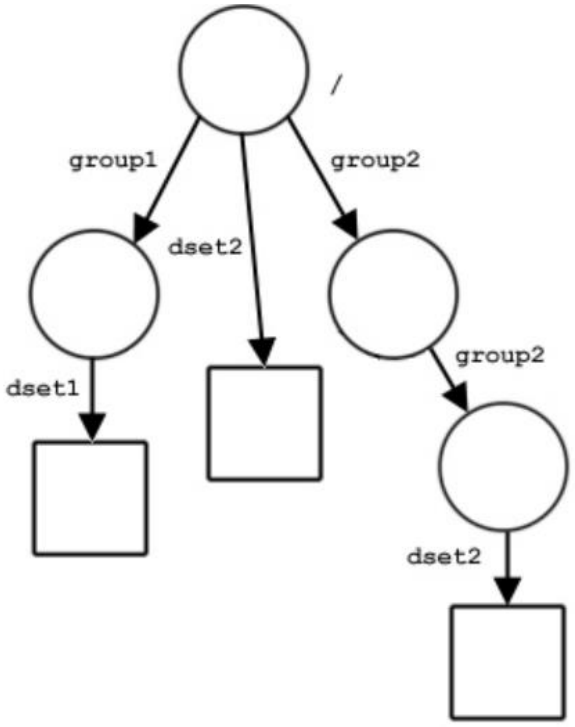


Figure 2.3 The HDF5 File source (TheHDF5Group, 2015)

2.7 Research Gap

The increase in data produced by IoT devices has proved to have challenges in terms of the read and write access time and the data integration capabilities of the storage system. Most of the work reviewed, focused on using raw data as input in the relational database systems and No SQL databases. This raw data does not have any specified format in terms of how it is stored or accessed. It has sharing issues because the data needs to be pre-processed and converted requiring more storage space. Converting and compressing raw IoT datasets into a .hdf5 format gives a one size fit approach for the IoT data. This makes it easy to share files because of the compatibility of the

file system and the tree structure. HDF5 has capabilities of supporting partial I/O and indexing, it supports a chunked storage layout that can be compressed to reduce the storage size and the read and write time of data in the root file. Therefore, this research focused on developing a HDF5 data compression tool for IoT applications.

2.8 Conceptual Framework

The HDF5 data compression tool for IoT applications was implemented as shown in figure 2.4. It had its inputs as the raw data collected by various IoT sensors. These inputs were retrieved from a publically available repository called data.world as .csv datasets. The datasets were then populated in a local database and parsed to the tool for compression and conversion. The output was a .hdf5 dataset that was viewed on a hdfview external interactor. This .hdf5 dataset would then be uploaded to the centralized gateway for purposes of data analysis in the future.

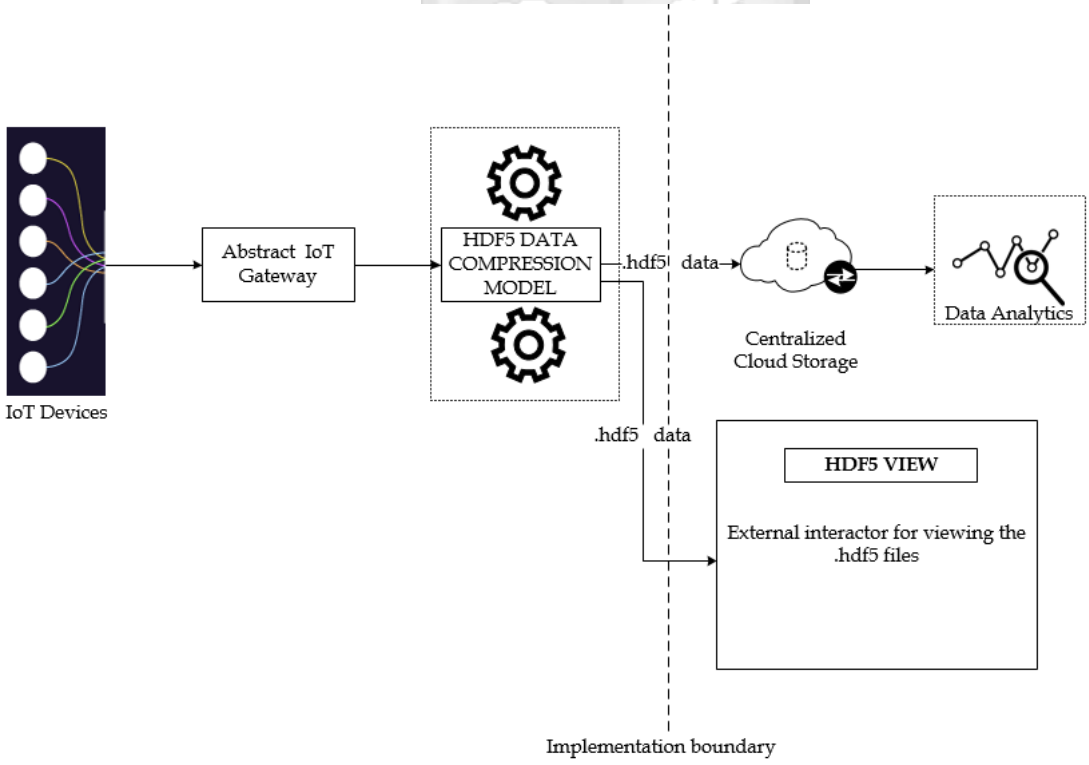


Figure 2.4 Conceptual Framework for a HDF5 Compression Tool for IoT data source(Authors work)

Chapter 3: Research Methodology

3.1 Introduction

This chapter presents a framework of the research approach that was taken. It highlights the research design and how research planning was carried out. The research was guided by the objectives intended to be met. Furthermore, the analysis of the data and the methods used in the study are also introduced. Lastly, ethical issues to bear in mind during this research are also summarized in this chapter.

3.2 Research Design and System Development Methodology

Research design is the format used in the research in terms of collection of data, measurement and analysis in an effort to get answers to research questions. Wanjugu further reiterates that the research design should bring out the relevance of the research to the economy in the procedure (Wanjugu, 2020). The selected approach was partly theoretical in nature through literature review and the development of a tool used to validate the proposed solution.

The objectives of this research were to develop a HDF5 data compression tool for IoT applications, investigate the impact of increasing data in IoT devices and review existing tools that have been used in compressing data in IoT. The research was qualitative in nature and it sought to compress IoT data and store it in a .hdf5 format. The tool used information from the data.world community repository which was collected from IoT devices. The proposed tool was implemented using the iterative methodology. This methodology focused on an initial, simplified implementation, which progressively gained more complexity and a broader feature set until the final tool was completed.

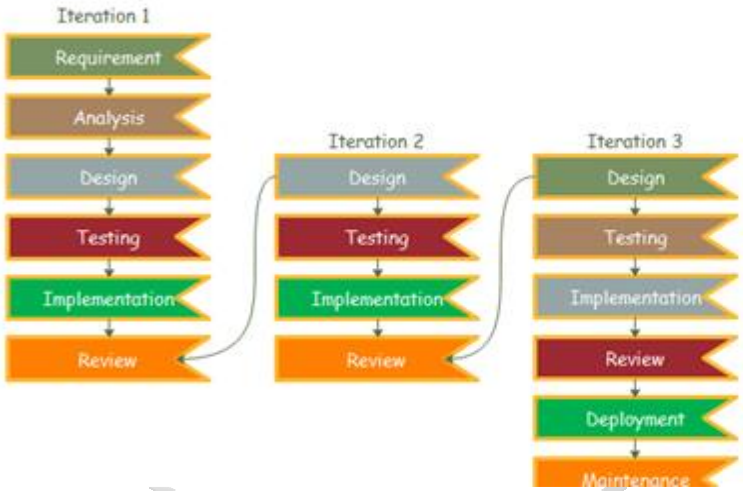


Figure 3. 1 Iterative Methodology source (Ruparelia, 2010)

3.3 Tool Development

Step followed to develop the HDF5 data compression tool for IoT applications;

i) Obtain data
ii) Pre-process the data
iii) Develop the tool
iv) Validate the tool

3.3.1 *Obtaining Data*

The data used in the developing of the HDF5 data compression tool was obtained from data.world community https://data.world. This is a free public benefit corporation that has extensible data, a simple interface, and an automated cloud-native repository. The data was obtained in form of .csv files.

3.3.2 *Data pre-processing*

The data obtained for this study contained redundant data, conflicting data, incomplete data and invalid data. Since this data was not required in the HDF5 tool, the data was checked for correctness based on the set standards for the HDF5 library tool. The .csv file was imported in Jupyter lab and checked for correct values, correct data types in the data rows, data uniformity and the changes

made uniformly. The file was exported back to a renamed .csv file to be used as input in the HDF5 data compression tool.

3.3.3 *Develop of the Tool*

The tool was developed using visual studio IDE. This used an open-source Python 3 library that contained the H5py package. The package acted as an interface to HDF5 binary data. Flask web framework was used to develop lightweight web applications and lastly, pandas was used for data storage in the HDF store. Execution was done on a local machine.

3.3.4 *Validation of the Tool*

The tool was validated using the compression ratio. This was measured by using the original data file size against the compressed .hdf5 data file size. Additionally, the external interactor hdfview validated the contents of data stored in the HDF5 file were the same as the original dataset hence, proving that the compression algorithm used in the data tool was lossless.

3.4 System Validation

To validate the system, pre-processed test IoT datasets were used as inputs to check if the tool could compress and store datasets in .hdf5 format. The output was displayed on an external interactor called hdfview. The interactor showed a file structure, compression algorithm, chunked layout and the storage layout of the compressed dataset in a simple format.

3.5 Location of the study

The location of the study was https://data.world data.world community.

3.6 Target Population and Sampling

For the purposes of this study, the target population was publicly available data from data.world repository specifically information that applied to IoT sensor data. The sampling technique used was purposive sampling where a sample csv dataset was selected.

3.7 Data Collection

Data Collection explains and demonstrates the methods, techniques and approaches adopted in social research projects (Denscombe, 2014). Datasets from the data.world community were the

main source of data for this research. The use of randomized datasets was the major method employed in the data collection phase.

3.8 Data Analysis

This involved analysing the file size of the IoT datasets before and after compression. The datasets served as the inputs to the HDF5 data compression tool. The output was then viewed using an external interactor for interpretation and analysis.

3.8.1 *Data Cleaning*

Since the data obtained from the data.world community contained attributes that were not necessary for the development of the tool, data cleaning was done using the Jupyter lab application based on the existing numerical dataset principles. Where there was missing data, the mean or mode of the dataset was used to in replacement or a constant value. This ensured that a balanced data set was used in the study.

3.9 Research Quality

This was a very important aspect of maintaining accuracy in this research. Therefore, validity, objectivity and reliability of the research were important. This research achieved this by limiting bias in the data collected. The research minimized the effects of extraneous independent variables that were undesirable by cleaning the data.

3.10 Ethical Approval

The necessary approval to conduct this research was acquired from the institution's Ethical Committee at Strathmore University. This was to ensure that all processes regarding data collection were done and followed the stipulated guidelines. This research upheld honesty, truthfulness and avoided any form of plagiarism, falsification and fabrication of data as well as adhered to the Ethical code of conduct.

Chapter 4: System Analysis Design and Architecture

4.1 Introduction

This section sought to develop a tool for compressing IoT data. The objectives of this chapter were to outline the system design, architecture, and functional and non-functional requirements that have been used in the algorithm. This included the diagrammatic representation of the entity-relationship diagram, context diagram, data flow diagram and database schema.

4.2 Requirement Gathering and Analysis

Data gathered for this study was obtained from the data world community https://data.world.The data collected from IoT devices was largely numerical data stored in .csv format. To analyse the data descriptive-analytical method were used, this included mean, median and variance. Exploratory data analysis was also done on the data sets.

4.2.1 *Functional Requirements*

For the system to achieve the objectives of this study, it needed to perform certain functions and processes:

i. The user should be able to select and upload IoT dataset stored in .csv format.
ii. The user should be able to initiate compression of the uploaded dataset.
iii. The system should be able to compress the dataset and store it in a .hdf5 format.
iv. The user should be able to add new IoT datasets in the root file of the tool.
v. The user should be able to view the file structure of the data using hdfview.

4.2.2 *Non-Functional Requirements*

Non-functional requirements were used to define the system behaviours, features, and characteristics that affect the system's usability. The non-functional requirements were well defined in order to have good user experience by meeting the user expectations. The system meet these requirements, which ensured its effectiveness and usability:

- Storage: The system should be able to store original datasets and compressed data sets after compression.
- Performance: The response and retrieval time of the compression should be acceptable.

- Scalability: The system should gradually improve the compression ratio of the datasets. This would ensure a good application response on data.
- Usability: The users of the system should be able to update, view, and select datasets that need compression.

4.3 System Design

System design is the process of defining elements of a system like architecture components and their interfaces and data based on specified system requirements (Wasson, 2005). This used the iterative methodology, which was progressive and gradual in order to refine the tool accordingly.

4.3.1 *System Architecture*

This tool was populated with cleaned IoT data. This data was in a .csv format. The web application provided an interface that was used by the users to select, upload and compress data.

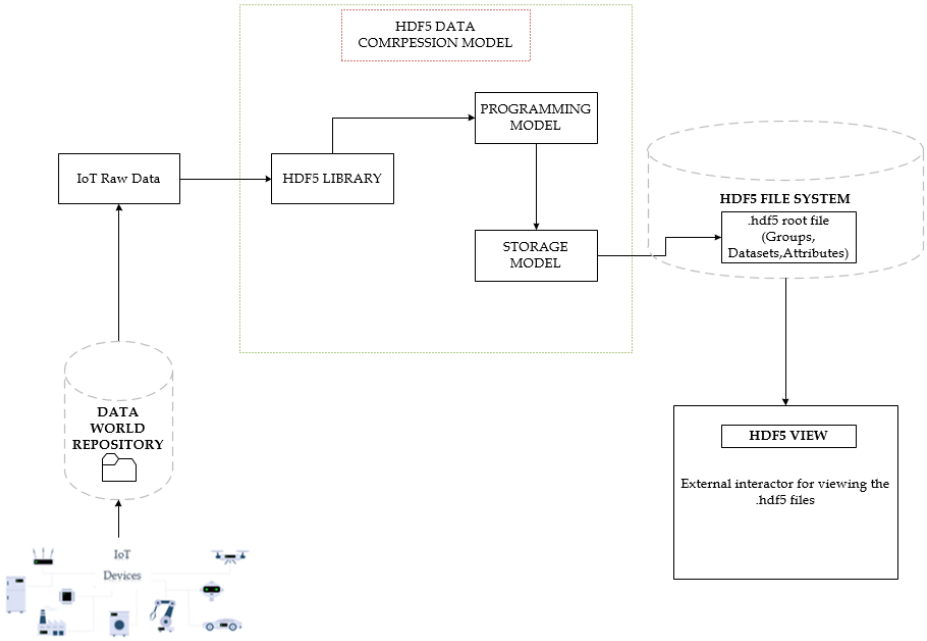


Figure 4.1 System Architecture

VT OMNES VNVM SINT

4.3.2 *Use case Diagram*

Use cases diagrams were used to illustrate a collection of related success and failure scenarios during the interaction of the actors and the system processes. Fig 4.2 illustrates how the actors will interact with the HDF5 data compression tool

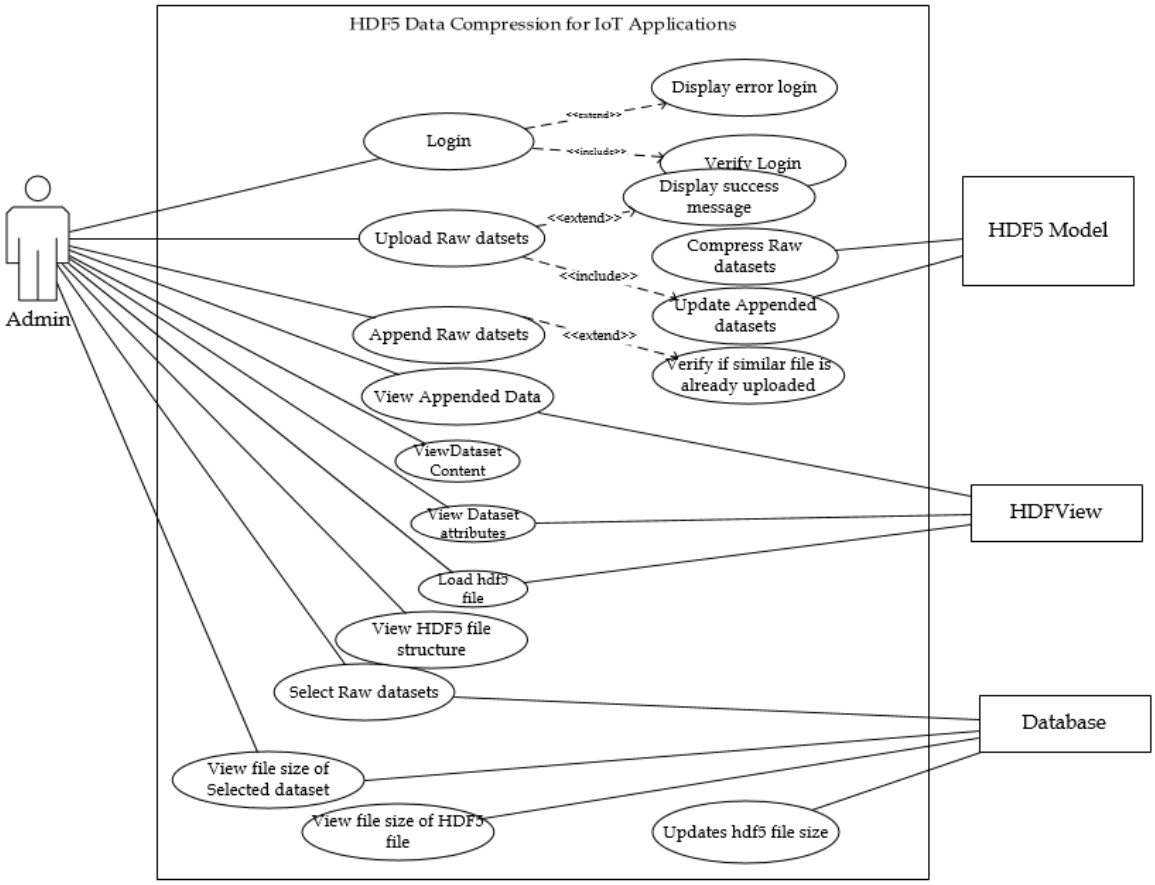


Figure 4. 2 Usecase Diagram

4.3.3 *Detailed Use case Description*

This section gave an elaborate description of the use cases In Fig 4.2 written in a fully dressed format.

Use Case UC1: Compress Dataset

Preconditions: Web application running

Main Success Scenario

1. User logs in to the web application
2. User selects a CSV dataset
3. User upload datasets and initiates compression
4. System compresses the dataset
5. System presents the user with a HDF5 file size summary

Extensions

At any time, the system fails to compress:

1. The user reloads the system.
2. System initializes.
3. The user starts the process

2a. System detects incorrect data entry

1. System signals error to the user and enters a clean state
2. Patient starts the process

Use case 2: Append Datasets

Preconditions: The user should have compressed a dataset with a similar file structure.

Main Success Scenario

1. User logs into the system
2. User selects CSV dataset
3. User uploads dataset
    - User checks the append checkbox
    - System appends the dataset

4.3.4 *Sequence Diagram*

The system sequence diagram showed the interactions between the main entities in the tool. Figure 4.3 below showed the flow of activities in sequence. The user selected a CSV dataset for compression. The user got the file size of the selected dataset , uploaded and compressesed the dataset. The file size of the result was displayed. The user loaded the HDF5 file to the hdfview application and viewed the file structure, data attributes dataset content and appended datasets.

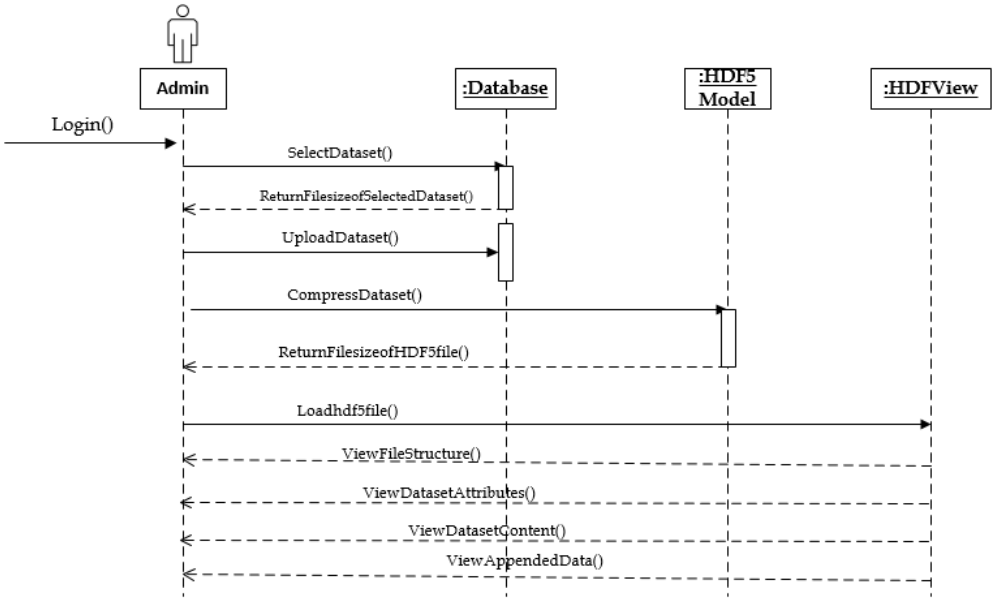


Figure 4. 3 Sequence Diagram

4.3.5 *Database Schema*

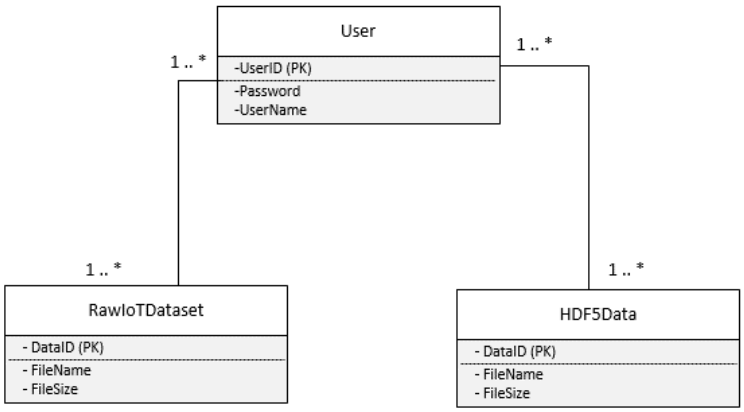


Figure 4. 4 Database schema

4.3.6 *Entity Relationship Diagram*

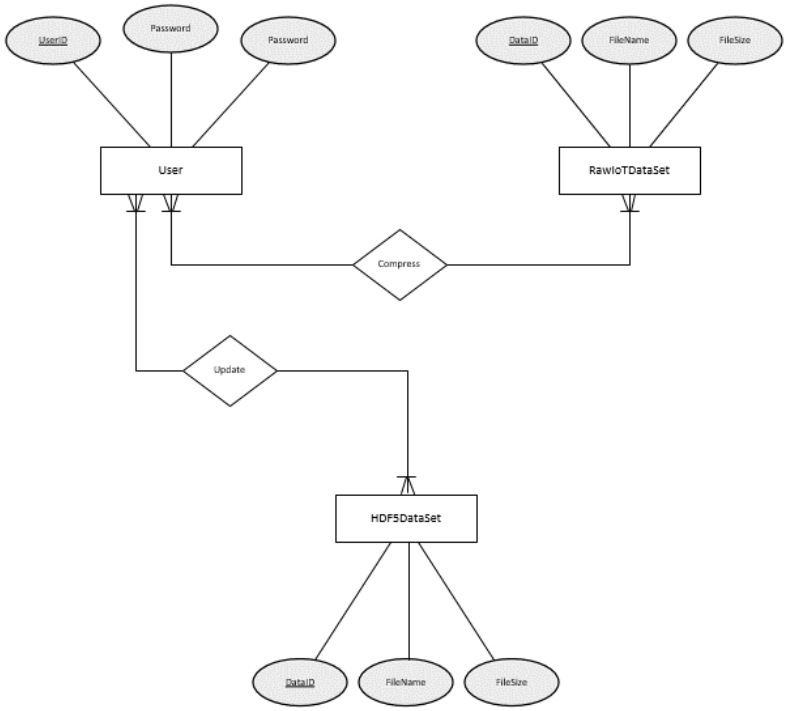


Figure 4. 5 Entity Relationship Diagram

4.3.7 *Web Application Wireframes*

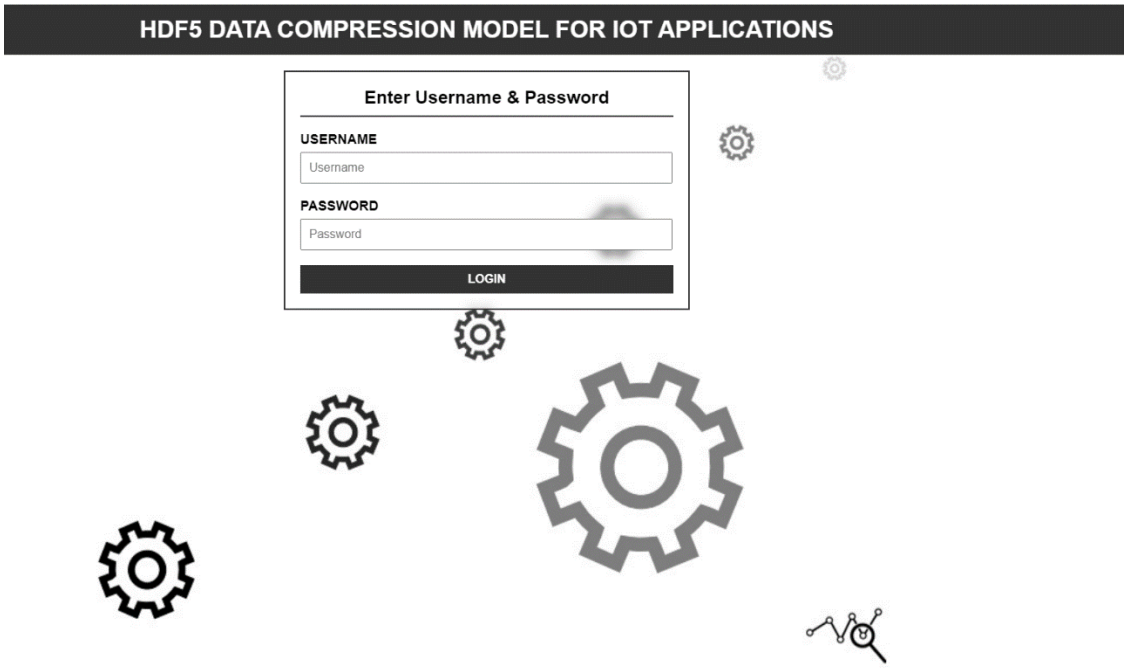


Figure 4. 6 Login page

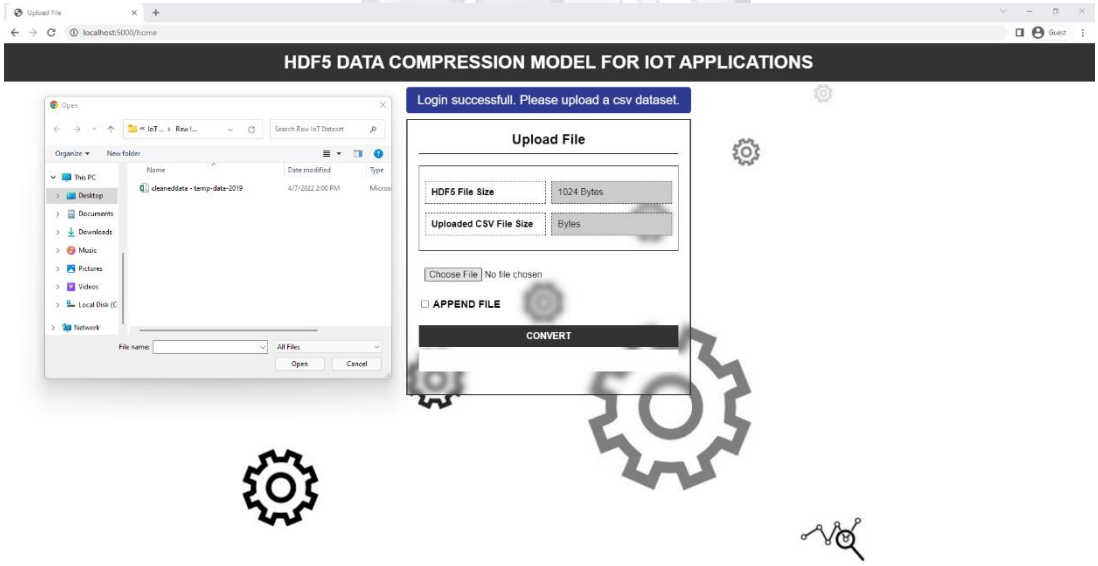


Figure 4. 7 Upload file page

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter was centered on how the tool was built, tested and validated. The implementations phase involved exploring the various modules of the tool, how development was done, how the tool worked and how testing was done. In addition, the programming tools and the viewing environment were also discussed in the chapter. As described in the conceptual tool in chapter 3, the tool entailed selecting, uploading, and compressing data sets.

5.2 System Development

5.2.1 *Development Platform*

The HDF5 data compression tool was developed using python for the backend because of its massive libraries and frameworks and good applicability in HDF5 tool develop. The user interface was developed using HTML, JavaScript and CSS because of its incrementally adaptable and progressive framework.

5.2.2 *Hardware requirements*

The minimum system requirements of the lossless compression algorithm will be:

i. A laptop computer with Windows 11 operating system.
ii. A Lenovo core i7 laptop with a RAM of 8 GB, and a solid-state drive space of 512 GB.

5.2.3 *Software Requirements*

Table 5.1 Requirements table

Software	Details
Visual Studio IDE	
Python 3.10	• **Flask**
	• **Flask –login**

	• **H5py**
	• **Pandas**
	• **Numpy**
	• **Pillow**
	• **Tables**
HyperText Markup Language (HTML)	
Cascading Style Sheets (CSS)	
Java Script	

5.3 Tool Implementation

Agile software development methodology was used in this study during the development phase of the tool. Its continuous iteration ability helped in the modifications of different versions of the system to meet the study's objectives. The HDF5 data compression tool was a web-based application that was developed using HTML and Python. The data was stored in the root directory of the main code on the local machine. In order for users to understand the workability of the tool an external interactor called hdfview was used. The web application enabled user interaction with the tool in terms of getting access to the main page of compressing IoT datasets by logging in. The user then viewed the current file size of the hdf5 file, which was initialized as 1kb or 1024 bytes, then upload a dataset from the Raw IoT dataset folder, then view the file size of the uncompressed file and lastly initiate compression. After compression, the updated compressed hdf5 file size was automatically updated on the page. The user then accessed the root folder of the tool and opened the updated hdf5 file using hdfview. This gave the user visibility of the file structure of the hdf5 file, the compression, storage layout, dimension size, datatype and the datasets members.

Data pre-processing

```
# Pandas is used for data manipulation
import pandas as pd

# Read in data as pandas dataframe and display first 5 rows
features = pd.read_csv('temp-data-2019.csv')
features.head(5)
```

```
# drop all rows with any NaN and NaT values
df = features.dropna()
print(df)
```

```
features.isnull().sum() # - Checking for null values
```

Figure 5. 1 Data pre-processing screenshots

Initializing the HDF5 file

```
import pandas as pd
from pandas import HDFStore

hdf = pd.HDFStore(r'C:\Users\Administrator\Desktop\IOT HDF5 MODEL\HDF5 App\H5FILE.hdf5')  # Create  a new hdf5 file
```

Figure 5. 2 Initializing HDF5 file

Loading the CSV file for compression

```
ALLOWED_EXTENSIONS = {'csv'}
hdf.put(filename.split('.')[0],pd.read_csv(filename),format='table',data_columns=True ,complevel=9,complib='zlib',append=True,index=True,dropna=False,track_times=True)
```

Figure 5. 3 Loading CSV dataset for compression

Writing into the HDF5 file

```
hdf = pd.HDFStore(save_path ,complevel=9,complib='zlib',append=True,index=True,dropna=False,track_times=True)
hdf.put(filename.split('.')[0],pd.read_csv(filename),format='table',data_columns=True ,complevel=9,complib='zlib',append=True,index=True,dropna=False,track_times=True)


hdf.close()
```

Figure 5. 4 Writing into the HDF5 file

Updating the HDF5 file size

```
return render_template ('home.html', file_size = os.path.getsize (save_path)
```

Figure 5. 5 Updating the HDF5 filesize

Appending existing HDF5 datasets

```
import pandas as pd
from pandas import HDFStore
from .app import  upload_file
from .app import allowed_file

save_path = './H5FILE.hdf5' # Path to HDF5 file
hdf = pd.HDFStore(save_path)


hdf.put(pd.read_csv(filename), hdf ,format='table', data_columns=True,append=True, index=False,dropna=False,track_times=True) #  Iterative addition of new data
```

Figure 5. 6Appending existing HDF5 datasets

5.3.1 *HDF5 Web Application*

The application was developed to run on a web application on a laptop. Figure 5.6 represents the first interface a user gets after installing the required libraries. This page run on the localhost port 5000 as shown in figure 5.7

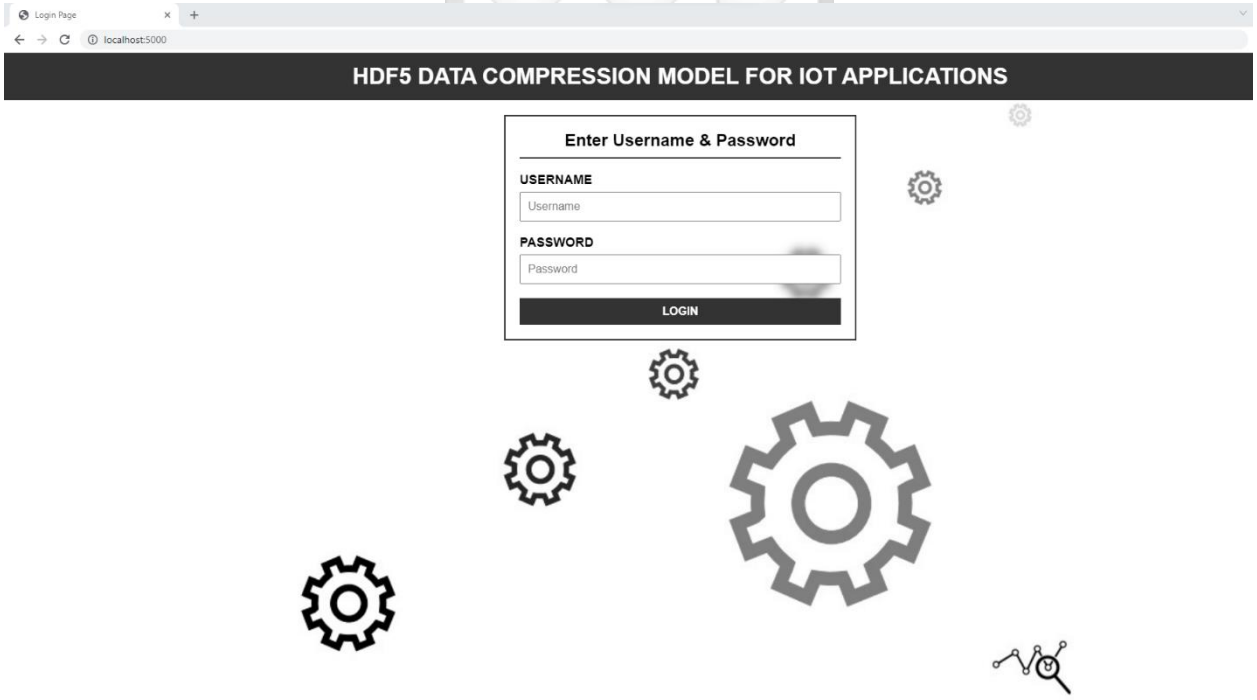


Figure 5. 7 Login Page

```
if __name__ == '__main__':
    app.run(debug=True, host='localhost')
```

Figure 5. 8 Localhost code

templates > login.html > html > body > div.container > div.form-container > form > div > input#username

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="static/style.css" />
    <title>Login Page</title>
  </head>
  <body>
    <div class="container">
      <h1 class="logo">HDF5 Data Compression Model for IOT Applications</h1>
      <div class="form-container">
        <h2>Enter Username & Password</h2>
        <form action="" method="post">
          <div>
            <label for="username">Username</label>
            <input
              type="text"
              placeholder="Username"
              name="username"
              id="username"
              value="{{
            request.form.username }}"
            />
          </div>
          <div>
            <label for="password">Password</label>
            <input
              type="password"
              placeholder="Password"
              name="password"
              id="password"
              value="{{
            request.form.password }}"
            />
          </div>
          <button type="submit">Login</button>
          {% if error %}
          <p class="error"><strong>Error:</strong>{{ error }}</p>
          {% endif %}
        </form>
      </div>
    </div>
  </body>
</html>

```

Figure 5. 9 Login code

After successful login, the user views the initialized hdf5 file and then uploads CSV data as shown in figure 5.9

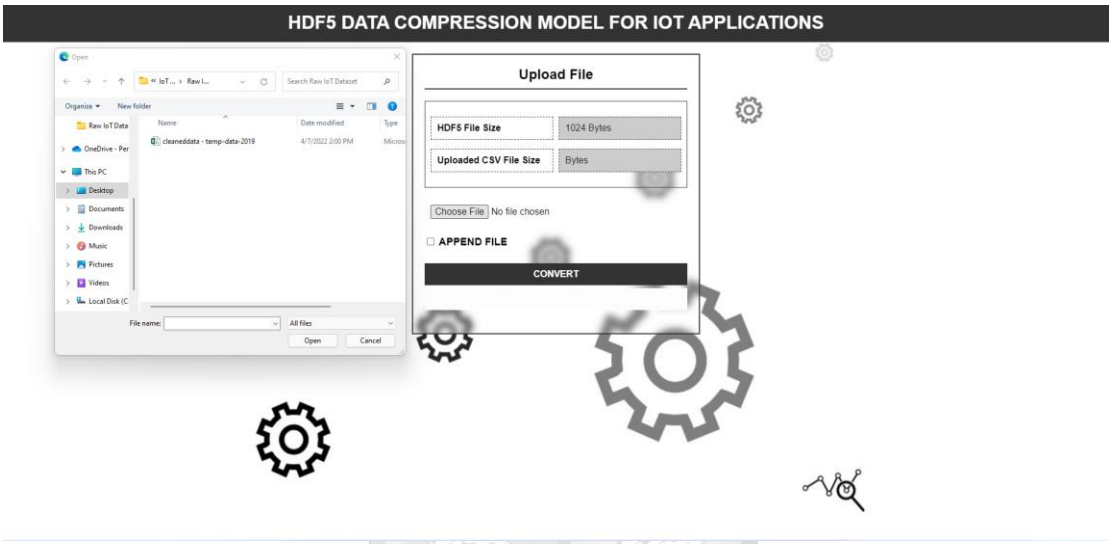


Figure 5. 10 Upload file page

The user views the file size of the uploaded file before compression as shown in figure 5.10

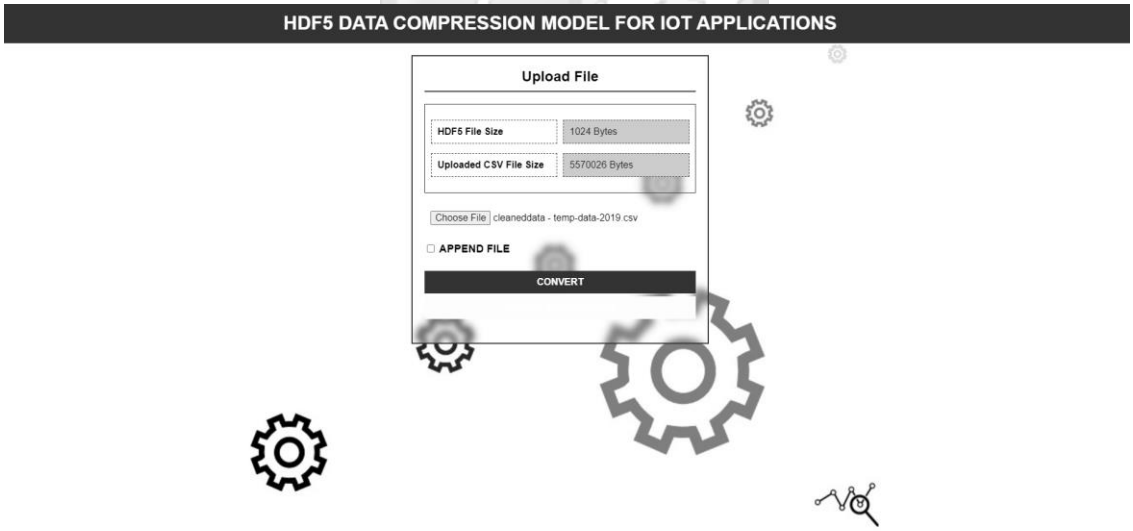


Figure 5. 11 Uploaded file size page

The user compressed the uploaded dataset and viewed the file size of the updated compressed hdf5 file as shown in figure 5.11

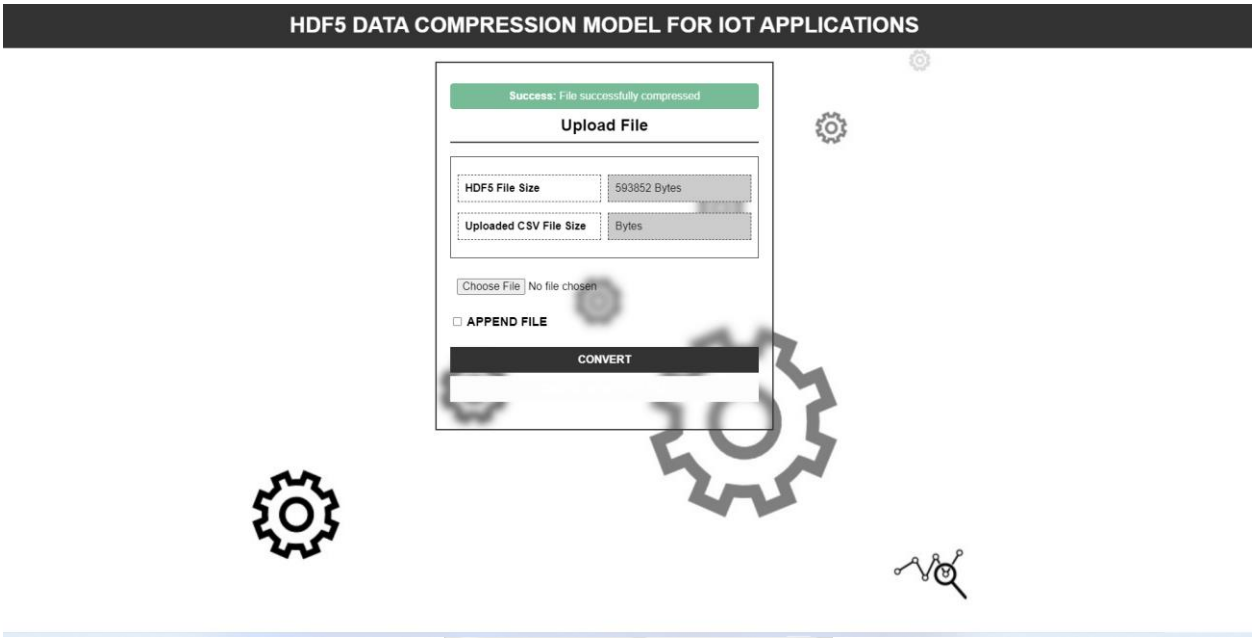


Figure 5. 12 Uploaded file success page

5.4 System Testing

The HDF5 data compression tool was developed using agile software development methodology, which allowed for continuous iterations at different phases. Continuous testing was done to test for the best compression algorithm to use in the tool. Functional testing was conducted to assess the compliance of the developed tool with the specified functional requirements.

5.4.1 *Testing of the tool usability*

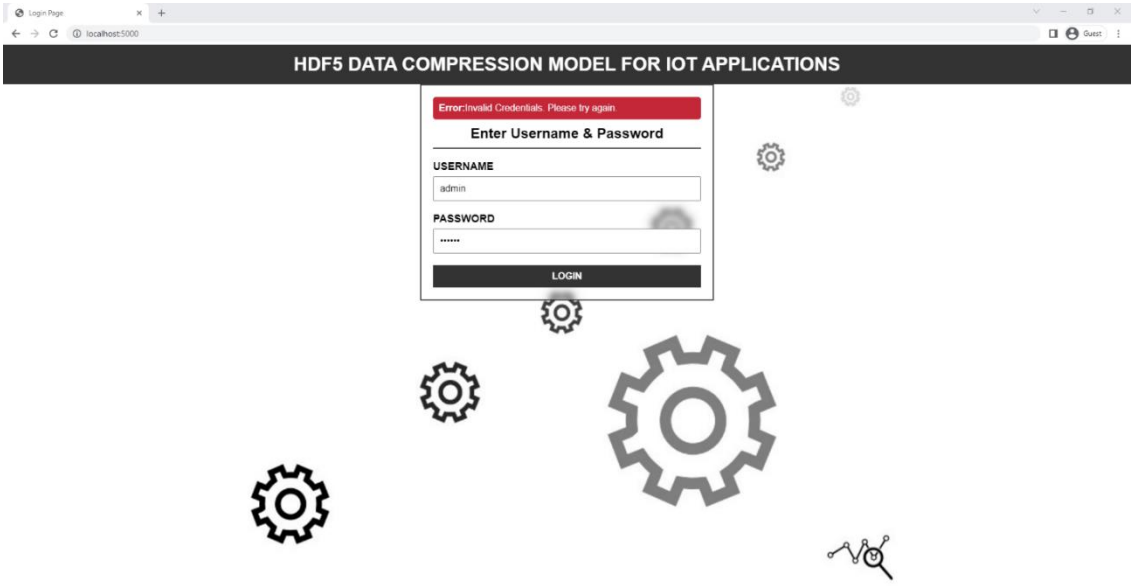


Figure 5. 13 Login test error

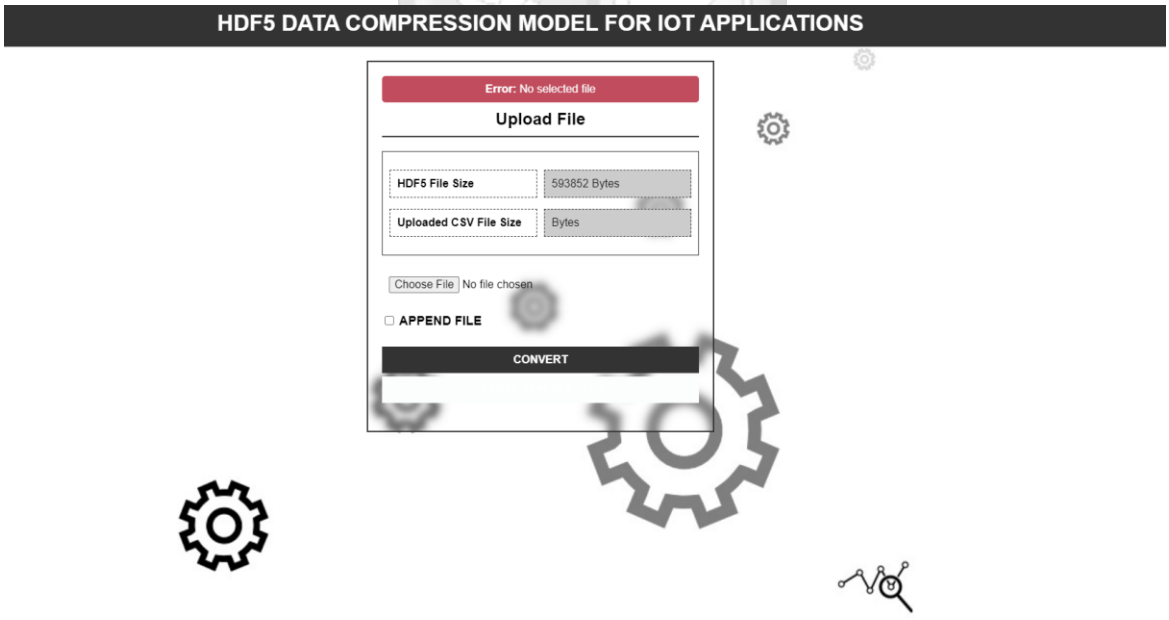


Figure 5. 14 No selected file error

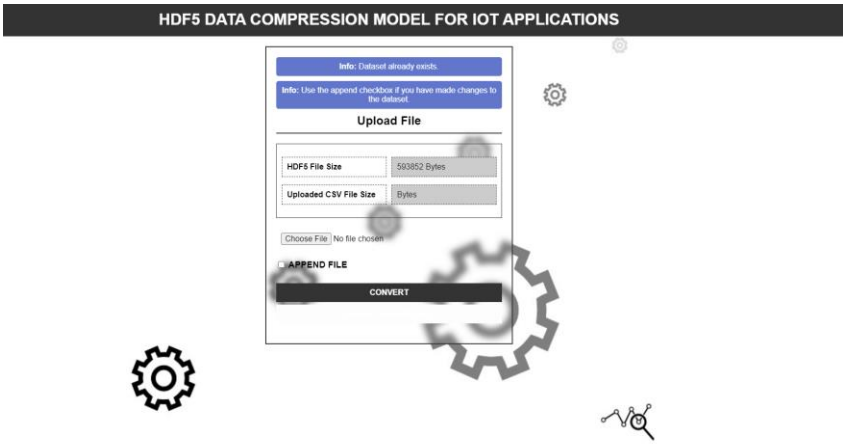


Figure 5. 15 Append file error

5.4.2 *Testing of the tool functionality*

After the user compressed the data set the file size of the updated hdf5 file was compared with the file size of the uploaded raw datasets as shown in figure 5.16 & 5.17. Further on figure 5.20 the hdf5 file content was viewed using hdfview to confirm the content and size of the dataset.

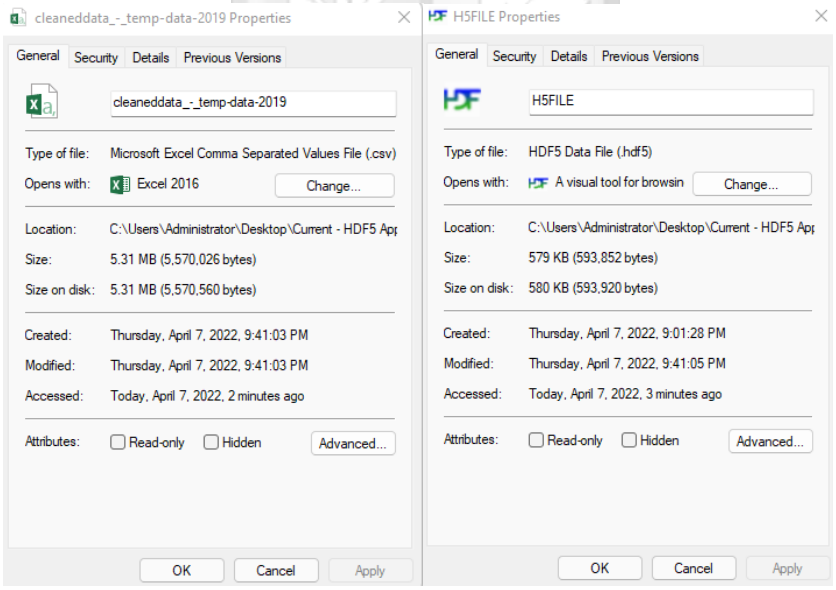


Figure 5. 16 Raw dataset before compression Figure 5. 17 Updated hdf5 file after compression

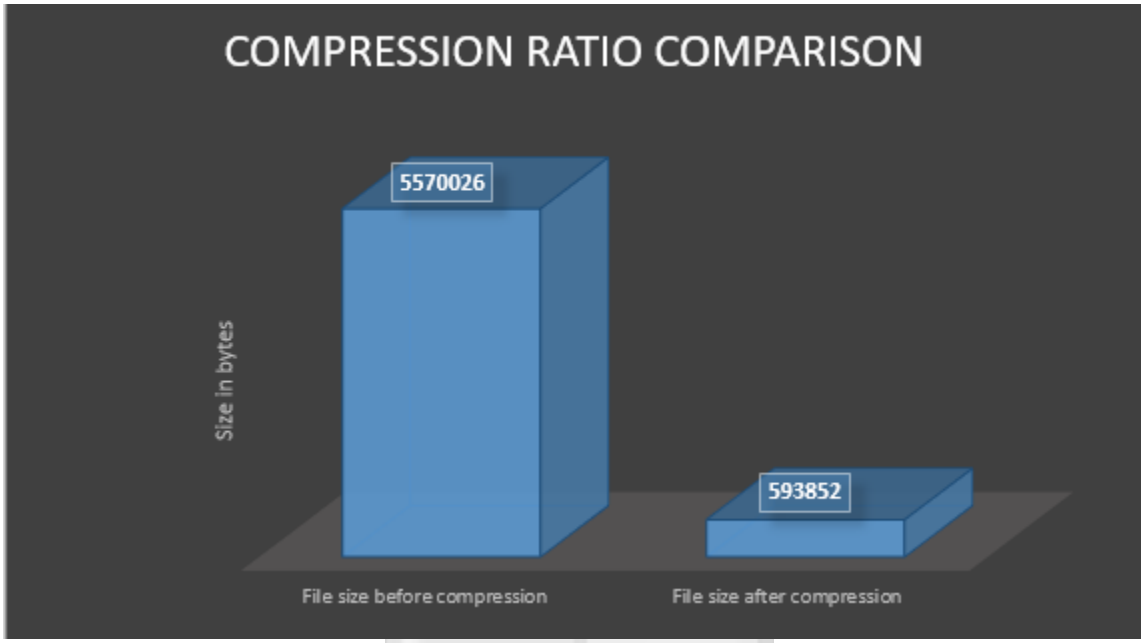


Figure 5. 18 File compression ratio Chart

=([@[File size before compression]]-[@[File size after compression]])/[@[File size before compression]]*100

B	C	D	E	F	G	H
		File size before compression	File size after compression	Percentage Difference		
cleaneddata_-_temp-data-2019		5570026	593852	89.34		

Figure 5. 19 Percentage comparison of the datasets

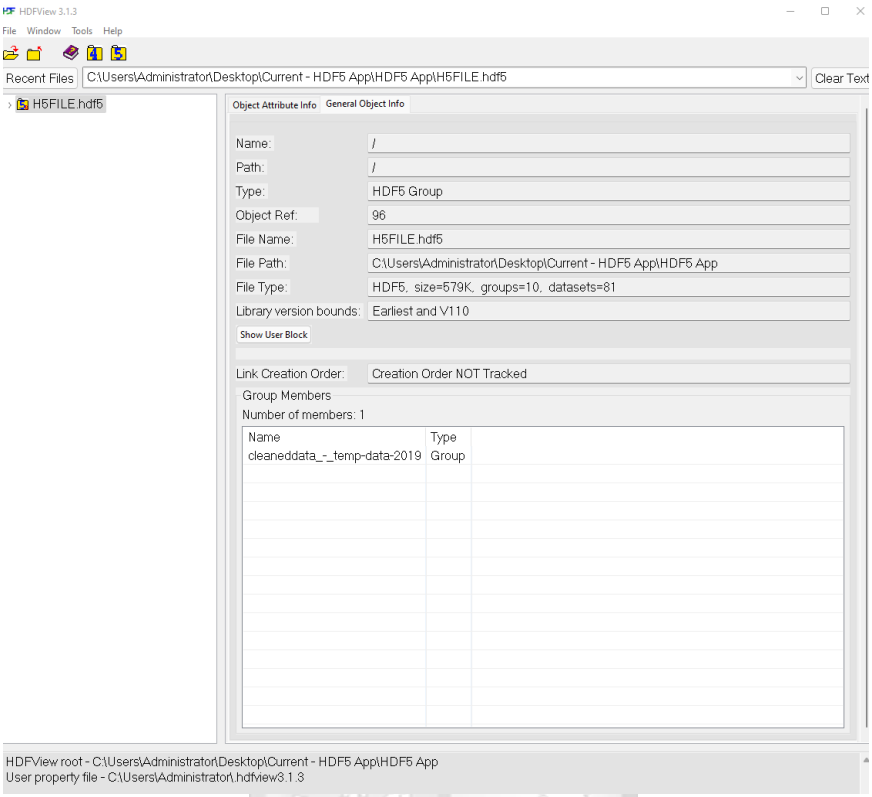


Figure 5. 20 Hdfview file contents

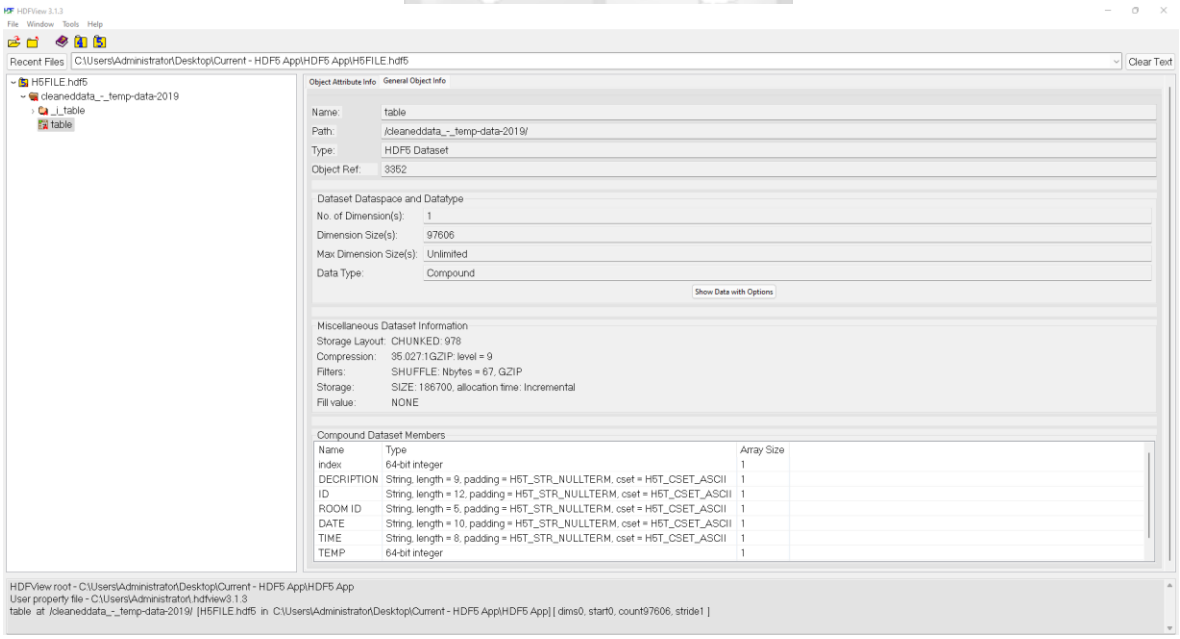
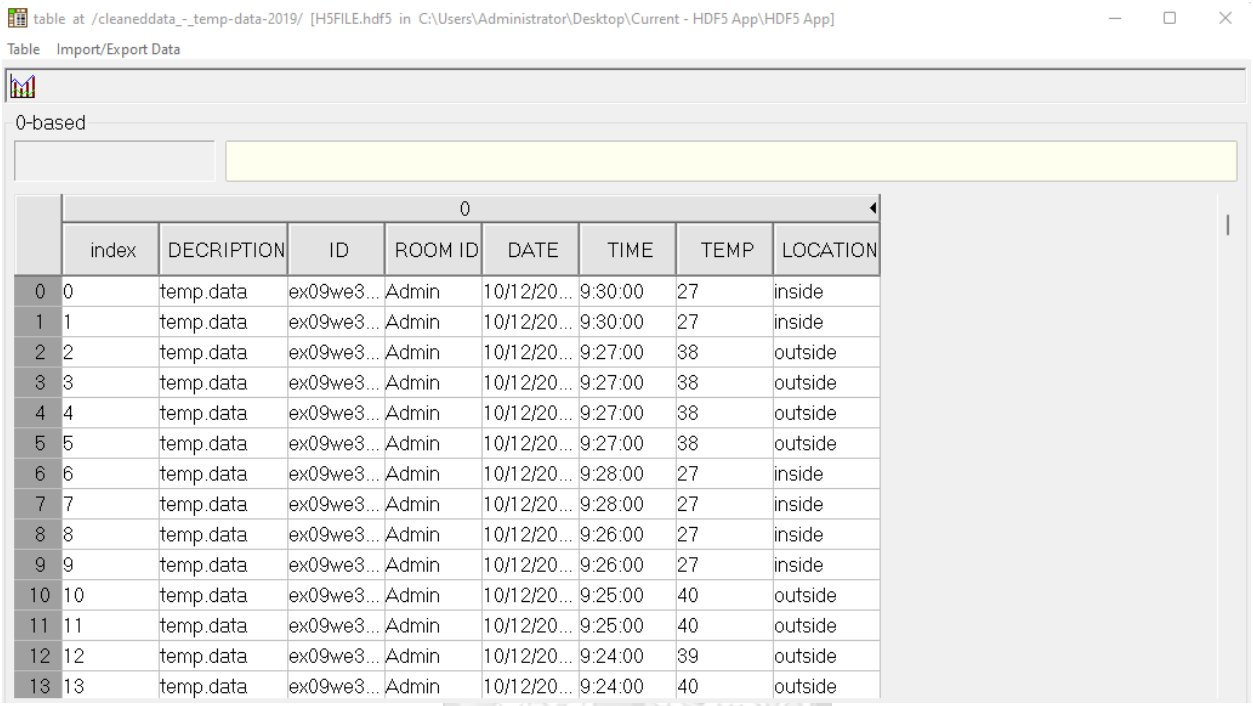
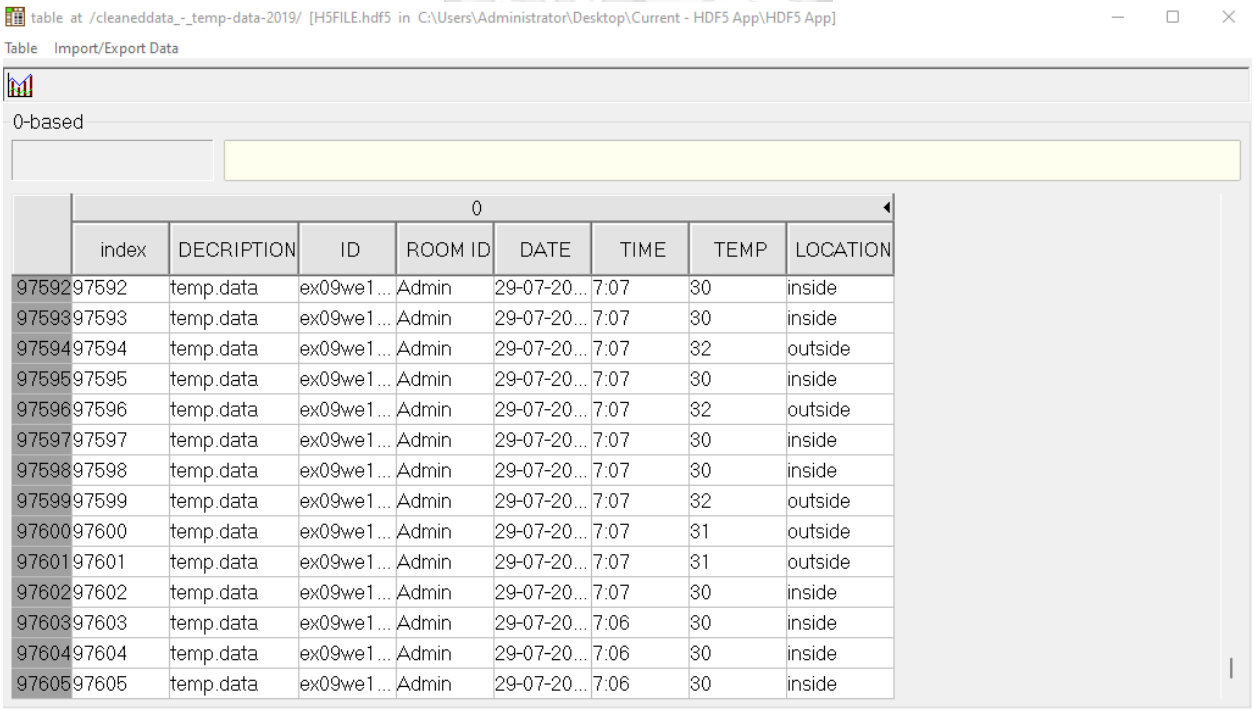


Figure 5. 21 HDF5 Dataset attributes



	0							
	index	DECRIPTION	ID	ROOM ID	DATE	TIME	TEMP	LOCATION
0	0	temp.data	ex09we3...	Admin	10/12/20...	9:30:00	27	inside
1	1	temp.data	ex09we3...	Admin	10/12/20...	9:30:00	27	inside
2	2	temp.data	ex09we3...	Admin	10/12/20...	9:27:00	38	outside
3	3	temp.data	ex09we3...	Admin	10/12/20...	9:27:00	38	outside
4	4	temp.data	ex09we3...	Admin	10/12/20...	9:27:00	38	outside
5	5	temp.data	ex09we3...	Admin	10/12/20...	9:27:00	38	outside
6	6	temp.data	ex09we3...	Admin	10/12/20...	9:28:00	27	inside
7	7	temp.data	ex09we3...	Admin	10/12/20...	9:28:00	27	inside
8	8	temp.data	ex09we3...	Admin	10/12/20...	9:26:00	27	inside
9	9	temp.data	ex09we3...	Admin	10/12/20...	9:26:00	27	inside
10	10	temp.data	ex09we3...	Admin	10/12/20...	9:25:00	40	outside
11	11	temp.data	ex09we3...	Admin	10/12/20...	9:25:00	40	outside
12	12	temp.data	ex09we3...	Admin	10/12/20...	9:24:00	39	outside
13	13	temp.data	ex09we3...	Admin	10/12/20...	9:24:00	40	outside

Figure 5. 22 HDF5 Dataset Content column 0-13



	0							
	index	DECRIPTION	ID	ROOM ID	DATE	TIME	TEMP	LOCATION
97592	97592	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97593	97593	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97594	97594	temp.data	ex09we1...	Admin	29-07-20...	7:07	32	outside
97595	97595	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97596	97596	temp.data	ex09we1...	Admin	29-07-20...	7:07	32	outside
97597	97597	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97598	97598	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97599	97599	temp.data	ex09we1...	Admin	29-07-20...	7:07	32	outside
97600	97600	temp.data	ex09we1...	Admin	29-07-20...	7:07	31	outside
97601	97601	temp.data	ex09we1...	Admin	29-07-20...	7:07	31	outside
97602	97602	temp.data	ex09we1...	Admin	29-07-20...	7:07	30	inside
97603	97603	temp.data	ex09we1...	Admin	29-07-20...	7:06	30	inside
97604	97604	temp.data	ex09we1...	Admin	29-07-20...	7:06	30	inside
97605	97605	temp.data	ex09we1...	Admin	29-07-20...	7:06	30	inside

Figure 5. 23 HDF5 Dataset content column 97592 -97605

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	DECRIPTION	ID	ROOM ID	DATE	TIME	TEMP	LOCATION													
2	temp.data	ex09we3091	Admin	10/12/2019	9:30:00	27	inside													
3	temp.data	ex09we3092	Admin	10/12/2019	9:30:00	27	inside													
4	temp.data	ex09we3093	Admin	10/12/2019	9:27:00	38	outside													
5	temp.data	ex09we3094	Admin	10/12/2019	9:27:00	38	outside													
6	temp.data	ex09we3095	Admin	10/12/2019	9:27:00	38	outside													
7	temp.data	ex09we3096	Admin	10/12/2019	9:27:00	38	outside													
8	temp.data	ex09we3097	Admin	10/12/2019	9:28:00	27	inside													
9	temp.data	ex09we3098	Admin	10/12/2019	9:28:00	27	inside													
10	temp.data	ex09we3099	Admin	10/12/2019	9:26:00	27	inside													
11	temp.data	ex09we3100	Admin	10/12/2019	9:26:00	27	inside													
12	temp.data	ex09we3101	Admin	10/12/2019	9:25:00	40	outside													
13	temp.data	ex09we3102	Admin	10/12/2019	9:25:00	40	outside													
14	temp.data	ex09we3103	Admin	10/12/2019	9:24:00	39	outside													
15	temp.data	ex09we3104	Admin	10/12/2019	9:24:00	40	outside													
16	temp.data	ex09we3105	Admin	10/12/2019	9:22:00	27	inside													
17	temp.data	ex09we3106	Admin	10/12/2019	9:22:00	27	inside													
18	temp.data	ex09we3107	Admin	10/12/2019	9:21:00	38	outside													
19	temp.data	ex09we3108	Admin	10/12/2019	9:21:00	38	outside													
20	temp.data	ex09we3109	Admin	10/12/2019	9:20:00	27	inside													

Figure 5. 24 Raw CSV Dataset Content column 1-20

	A	B	C	D	E	F	G	H	I	J	K	L
97594	temp.data	ex09we100683	Admin	29-07-2019	7:07	30	inside					
97595	temp.data	ex09we100684	Admin	29-07-2019	7:07	30	inside					
97596	temp.data	ex09we100685	Admin	29-07-2019	7:07	32	outside					
97597	temp.data	ex09we100686	Admin	29-07-2019	7:07	30	inside					
97598	temp.data	ex09we100687	Admin	29-07-2019	7:07	32	outside					
97599	temp.data	ex09we100688	Admin	29-07-2019	7:07	30	inside					
97600	temp.data	ex09we100689	Admin	29-07-2019	7:07	30	inside					
97601	temp.data	ex09we100690	Admin	29-07-2019	7:07	32	outside					
97602	temp.data	ex09we100691	Admin	29-07-2019	7:07	31	outside					
97603	temp.data	ex09we100692	Admin	29-07-2019	7:07	31	outside					
97604	temp.data	ex09we100693	Admin	29-07-2019	7:07	30	inside					
97605	temp.data	ex09we100694	Admin	29-07-2019	7:06	30	inside					
97606	temp.data	ex09we100695	Admin	29-07-2019	7:06	30	inside					
97607	temp.data	ex09we100696	Admin	29-07-2019	7:06	30	inside					

Figure 5. 25 Raw CSV Dataset Content column 97594-97607

The above figures shown the functionality of the tool and it was evident that there was significant compression achieved while the HDF5 tool was used. The percentage of the raw dataset against the compressed dataset shown 89.34% decrease from the original file size. Additionally, as shown in Figures 5.20 to 5.23 the file contents were still the same which meant that no data was lost during compression.

5.5 Conclusion

The requirements identified during requirements analysis acted as a guide during the implementation phase. The system design gave information on how the system development was done. The research objectives were the cornerstones used during the development of the system to ensure it met its functionality.

Chapter 6: Discussion

6.1 Introduction

This chapter explains the research conclusions, achievements, and reviews of how the research objectives were realized not limited to mentioning the application's advantages and limitations. The research study investigated the challenges associated with increasing data in IoT devices and the current ways of managing data. After reviewing these data management tools, the researcher chose to focus on data compression using HDF5, which was well suited for large IoT datasets. A web application for the tool was developed.

6.2 Review of Research Objectives

To reference section 1.3 the first objective was to investigate the impact of increasing data in IoT devices. According to the findings it was seen that data analysis was not being carried out on all the data collected,integration of heterogeneous data was challenging and the lack of a one size fit approach of storing IoT data are some of the factors that were being influenced by the increasing IoT data. These findings are in line with the literature reviewed in section 2.2. In the reviewed literature implementing a HDF5 data tool that can represent data in à format suitable for analysis while supporting data integration would play an important role when it came to the management of large IoT data.

The second objective was to review existing data management tools for storing data in IoT. To develop a suitable solution the research findings were used to help identify the most appropriate tool. The literature review explored the different technologies and tools that are used for storing data in IoT devices such as NoSQL, SCAN RAW, Oracle, MySQL and Hadoop.The study established that using the current version of HDF5 would play a key role in bridging the gap between the storage system and data access time of data and I/O operations. This in line with the literature discussed in section 2.3.

The third objective was to develop a HDF5 data compression tool for IoT application. According to the research findings, the tool developed can compress and convert raw datasets into a. hdf5 file format. The fourth objective was to validate the developed tool. The test validation section 5.4.2 was used to validate the tool and represent the finding in an understandable form.

6.3 Advantages of the tool

The developed tool converts and compresses raw IoT datasets into a. hdf5 format. This is suitable for data sharing since only a single compressed. Hdf5 file is shared. It also saves on storage space by compressing data and significantly reducing the storage infrastructure required for large IoT data.

6.4 Limitations of the tool

The tool requires you to have an external interactor for viewing the file structure of the compressed file. The tool only uses .csv data to do the compression.

Chapter 7: Conclusions and Recommendations

7.1 Introduction

In this chapter, conclusions, recommendations, and future work are discussed. As for the conclusions, all the objectives are reviewed by looking at how the research questions were answered. Then the researcher gives some recommendations for the tool and lastly, the future works relating to this research are proposed.

7.2 Conclusions

From the study, it was established that having a one size fit approach for handling large amounts of data would be very important in the IoT industry. This meant that data would be organized in a single file format suitable for data analysis and storage. The HDF5 data compression tool compressed and converted raw IoT data into a .hdf5 file format. Each dataset was organized in a named group for ease of user understanding of the file structure. Other file extensions were not included in the tool since they are not suitable for data analysis as much as the tool could store, compress and represent them.

7.3 Recommendations

This study showed that HDF5 could be used in compressing IoT application data. Therefore, the researcher recommends using the HDF5, Map Reduce and visualization tools to see what can be improved in terms of achieving better efficiency on IoT data and show, which would best suit the large IoT data better.

7.4 Future Works

The researcher saw the tool has the potential to be simulated in a real IoT environment in the future and proposes implementation of the same in a simulated IoT environment, which would then give data access to data analysts for a proof of concept.

References

Abouzied, A., Abadi, D. J., & Silberschatz, A. (2013). Invisible Loading: Access-Driven Data Transfer from Raw Files into Database Systems. *Proceedings of the 16th International Conference on Extending Database Technology*, *13*, pp. 1-10. doi:10.1145/2452376.2452377

Abu-Elkheir, M., Hayajneh, M., & Ali, N. A. (2013). Data management for the internet of things: design primitives and solution. Sensors. (13(11)), 15582–15612. Retrieved from https://doi.org/10.3390/s131115582

Adel, A., & Rabie, A. R. (2017). IoT data provenance implementation challenges. *The 3rd International Workshop on Tasks on High Performance Computing (THPC 2017)*.

Alex, Koohang; Carol ,Springer, Sargent; Jeretta, Horn, Nord; Joanna ,Paliszkiewicz. (2022). Internet of Things (IoT): From awareness to continued use. *Internet Journal of Information Management, 62*(102442). doi:https://doi.org/10.1016/j.ijinfomgt.2021.102442.

Anna, G., & Satwik, K. (2017, December 18). *Making sense of IoT data*. Retrieved from IBM developer: https://developer.ibm.com/

Atzori, L., Iera, A., & Morabito, G. (2010). *The internet of things: A survey', Computer networks,*.

Azim, T. (2018). Adaptive Cache Mode Selection for Queries over Raw Data. *In Ninth International Workshop on Accelerating Analytics and Data Management Systems Using Modern Processor).*

Berrick, S. (2022, February 25). *HDF5 Data tool,file format and library*. Retrieved from EARTHDATA: https://earthdata.nasa.gov/

Buckley, K. (2017). *451 Research Voice of the Enterprise: Internet of Things, Workloads and Key Projects.* Newyork: 451 Research .

Castro, D. (2008). *Digital Quality of Life: Government.* Rochester ,NY: Social Science Research Network,.

Cattell, R. (2010). Scalable SQL and NoSQL data stores. (pp. 12-27). ACM SIGMOD.

Chen, L., Tseng, M., & Lian, X. (2010). Development of foundation tools for Internet of Things. (pp. 376-385). China: Front. Comput. Sci.

Cheng, Y. and Rusu, F. (2015)., *40(3)*, pp. 19:1--19:45. doi:10.1145/2818181.

Denscombe, M. (2014). *The good research guide for small-scale social research projects.* . UK: Hill Education.

Ertl, C., Frisch, J., & Mundani, R.-P. (2016). Massive Parallel Fluid Flow Simulations using Hierarchical Data Format Version 5 (HDF5). *2016 15th International Symposium on Parallel and Distributed Computing.* Germany.

Gizem, K., & Toğay, C. (2017). IoT Data Storage: Relational & Non-Relational Database. *Billisim 2017 paper, 2045*.

Group, H. (2011). The HDF5 Data Tool and File structure.

GSMA. (2014). Understanding the Internet of Things. In G. Association, *Connected Living* (pp. 1-15). London: GSMA.

Harjinder, K., & Ajay, S. ,. (2018). A Review on Integration of Big Data and IoT. *2018 4th International Conference on Computing Sciences (ICCS).* Jalandhar, India: IEEE. doi: 10.1109/ICCS.2018.00040

He, Z. ,., & He, Y. ,. (2011). Preliminary Study on Data Management Technologies of Internet of Things. *2011 International Conference on Intelligence Science and Information Engineering*, (pp. 137-140).

Heber, G. (2011, February 17). *HDF5 Dataspaces and Partial I/O*. Retrieved from The HDF5 Group: http://davis.lbl.gov/Manuals/HDF5-1.8.7/

In, L., & Kyoochun, L. (2015). The internet of Things (IoT):Applications,investments and challenges for enterprises. *Business Horizons*, 431-440.

IoT-A.Converged Architectural Reference Tool for the IoT v2.0. (2012). Switzerland.

Ismail, N. (2017, October 2). *How do you fuel the IoT*. Retrieved from Information Age: https://www.information-age.com/fuel-iot-data-data-data-123468842/

J, G., Xu, L., G, X., & Z, G. (2012). Improving multilingual semantic interoperation in cross-organizational enterprise systems through concept disambiguation. *IEEE Trans. Ind. Informat, 8*(3), 647–658.

Levy, E. (2019, June 20). *Upsolver*. Retrieved from IoT Analytics:Challenges Apllications and Innovations: https://www.upsolver.com/

Lihong, J., Li, D. X., Hongming, C., Zuhai, J., Fenglin, B., & Boyi, X. (2014). An IoT-Oriented Data Storage Framework in Cloud Computing Platform. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, 10*(2), 1443-1450.

Lionel, S. ,. (2021, March 8). *Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025*. Retrieved from statista: https://www.statista.com/statistics/

Luo, C., Wu, F., Sun, J., & Chen, C. (2009). Compressive Data Gathering for Large-Scale Wireless Sensor Networks. *In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom 2009)* (pp. 145-156). Beijing, China: MobiCom.

Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2016). Internet of things IoT security:Current status,challenges and prospective measures. *10th International Conference Internet Technology Security* (pp. 336-341). Trans.

Oliver, R., & Ben, B. (2015). *HDF5 I/O Performance*. Retrieved from BASTest: https://biorack.github.io/BASTet/HDF5_format_performance.html?msclkid=dd43bf82b98411ecb1d61e63a52173f0#discussion

Olma, M. (2017). Slalom : Coasting Through Raw Data via Adaptive Partitioning and Indexing. *Proceedings of the VLDB Endowment, 10*, pp. 1106-1117.

Patel, M., & Bhise, M. (2019, January 7). *Raw Data Processing Framework for IoT.* Retrieved from Research Gate: https://www.researchgate.net/publication/329831507

Paul, B., Janssen, M., & Herder, P. (2020). The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations,. *International Journal of Information Management, 51*. doi:10.1016

Pius, D. (2016). *'CityPulse : Large Scale Data Analytics Framework for Smart Cities.* IEEE Access.

Rajeshwari, D. (2015). State of the art of big data analytics "A survey,". *International Journal of Computer Applications, 120*(22).

Ramakrishnan, R., & Gehrke, J. (2011). Database Management Systems, 3rd ed.;. New York: McGraw-Hill.

Riley, J. (2019, November 30). *A GUIDE TO EVERYTHING YOU NEED TO KNOW ABOUT DARK DATA.* Retrieved from INFLUENCE: https://www.influencive.com/a-guide-to-everything-you-need-to-know-about-dark-data/

Somayya, M., R, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature. *Journal of Computer and Communications*, 164-173. doi:10.4236

TheHDF5Group. (2015). *HDF5 User's Guide.* Urbana.

Tingli, L., Yang, L., Ye, T., Shuo, S., & Wei, M. (2012). A Storage Solution for Massive IoT Data Based on NoSQL. *2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conferenceon Cyber, Physical and Social Computing.* China: IEEE.

Tonglin, L., Quincey, K., Houjun, T., Jialin, L., & Suren, B. (2019). *H5Prov: I/O Performance Analysis of Science Applications Using HDF5 File-level Provenance.* Berkely, USA: Semantic Scholar.

Tsiftes, N., & Dunkels, A. (2011). "A database in every sensor. *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, (pp. 316-329). Seattle.

Ukil, A., Bandyopadhyay, S., & Pal, A. (2015). IoT Data Compression: Sensor-Agnostic Approach. *Data Compression Conference* (pp. 303-312). USA: Snowbird.

Ukil, A., Bandyopadhyay, S., Sinha, A., & Pal, A. (2015). *Adaptive Sensor Data Compression in IoT systems: Sensor data analytics based approach.* Book Adaptive Sensor Data Compression in IoT systems:Sensor data analytics based approach.

Vasani, S. (2013). *Faster Query Execution for Partitioned RDF Data.* ICDCIT2013. doi:10.1007/b104418.

Veritas. (2019). *Optimization with Dark Data and Big Data*. Retrieved from PFIESTER MARKETING ANALYTICS: https://pfisterer-marketing-analytics.com/optimization-with-dark-data-and-big-data/

Wanjugu, E. ,. (2020, February 12). *Role of information and communication technology investment on project performance of large supermarkets in Kenya*. Retrieved from wired.com: https://www.wired.com/2015/ 01/german steel-mill-hack-destruction/

Wu G., T. S. (2011). M2M: From mobile to embedded internet. *49*, 36–43. doi:10.1109/MCOM.2011.5741144

Xu, L., He, W., & S, L. (2014). *Internet of things in industries:A survey*. IEEE Transactions on Industrial Informatics.

Yang Hui, Q. Y. (2013). Online monitoring of geological CO2 storage and leakage based on wireless sensor networks. *IEEE Sensor Journal*(13(2)).

Yang, M. (2008). Parallel hdf5 tutorial. *in 14th IBM System Scientific User Group(ScicomP)*.

APPENDIX A: Similarity Report

Curiginal

Document Information

Analyzed document	135282 - RISPER NKATHA.docx (D133617042)
Submitted	2022-04-14T11:40:00.0000000
Submitted by	
Submitter email	Risper.Chabari@strathmore.edu
Similarity	8%
Analysis address	library.strath@analysis.urkund.com

Sources included in the report

SA	**masters_project_template.pdf** Document masters_project_template.pdf (D21636659)	1
SA	**Strathmore University / Risper Nkatha 135282.docx** Document Risper Nkatha 135282.docx (D130322867) Submitted by: Risper.Chabari@strathmore.edu Receiver: aomondi.strath@analysis.urkund.com	15
W	URL: https://prace-ri.eu/wp-content/uploads/Implementing_a_XDMF_HDF5_Parallel_File_System_in_Alya-2.pdf Fetched: 2022-04-14T11:41:32.7870000	2
W	URL: https://arxiv.org/pdf/1807.06534 Fetched: 2022-04-14T11:41:51.3530000	4
W	URL: http://davis.lbl.gov/Manuals/HDF5-1.8.7/UG/03_DataModel.html Fetched: 2020-11-12T03:34:04.4930000	2
SA	**48373633.pdf** Document 48373633.pdf (D124474204)	1
SA	**48511452.pdf** Document 48511452.pdf (D124482607)	1
SA	**Neha Kaliya.pdf** Document Neha Kaliya.pdf (D27776917)	1
SA	**A Comprehensive Study on Data Management solutions to Internet of Things - Final copy.doc** Document A Comprehensive Study on Data Management solutions to Internet of Things - Final copy.doc (D45592965)	1
SA	**Synopsis.doc** Document Synopsis.doc (D111410396)	1

APPENDIX B:Ethical Approval

Strathmore UNIVERSITY

11th April 2022

Ms Nkatha Risper,
risper.chabari@strathmore.edu

Dear Ms Nkatha,

RE: A HDF5 Data Compression Model for IoT Applications

This is to inform you that SU-IERC has reviewed and **approved** your above **SU Masters'** research proposal. Your application reference number is **SU-IERC1230/21.** The approval period is **11th April 2022 to 10th April 2023.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE
(SU-IERC)
11 Apr 2022
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA