Fully Qualified Domain Name(FQDN) as Indexing Parameter for deduplicating the data in Network Devices.

MSc Industrial Internship Report
Cyber Security

Sudha Gaurinath Koride
Student ID: 20196083

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni.

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sudha Gaurinath Koride

Student ID: 20196083

Programme: M.Sc., Cyber Security **Year:** Jan 2021-Jan2022**.**

Module: Research Internship Project

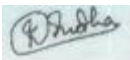
Lecturer: Mr. Vikas Sahni

Submission Due Date: 7th Jan 2022

Project Title: Fully Qualified Domain Name(FQDN) as Indexing Parameter for deduplicating the data in Network Devices.

Page count: 20 **Word Count:** 5404

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 7th Jan 2022.

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Fully Qualified Domain Name(FQDN) as indexing parameter for deduplicating the data in Network Devices.

Sudha Koride
20196083

Abstract

The complexity of network devices has increased tremendously due to the demand of accessibility and availability of the network through multiple devices from any location and any time. This has, in turn, increased amount of data that is being generated and stored. One of the factors accounting to it is duplicate or redundant data. These duplicate records can act as security gaps if not handled properly because cyber criminals may use a duplicate record of decommissioned device which is still showing in the Asset Registry as active due to duplication. FQDN is used as indexing parameter to remove the duplicate data from the data obtained through two different network resources: Configuration Management Database(CMDB) & Unknown Device Detection & Response (UDDR). This data is being compared to the data coming from another network resource SolarWinds(SW). This is considered as source-of-truth as it has only unique records in it. Two of the most efficient similarity matching and data deduplicating algorithms AdaBoost and k-Nearest neighbour have been implemented in this research attempt. AdaBoost proved to be a better data deduplication model with an accuracy of 87% while k-Nearest Neighbour model achieved an accuracy of 80%. Ultimately, the efficiency of FQDN to be used as a unique attribute for data deduplication in Network devices was proved.

1 Introduction

As it is quite evident in the recent times, the evolution of online platforms for businesses has reached new heights. Factors like advancement in technology, ease of use, consumer awareness and the current pandemic situation of Covid-19 have collectively forced most of the business to choose online platforms for their operation. Irrespective of the type of industry, be it commercial, healthcare, travel, banking etc, all industries are operating online now. This gives the owners more options on how they can interact with their clients, operate their businesses and store and analyse the data efficiently. This means that the complexity of the networks and the amount of data to be handled has also increased tremendously[1]. But this also opens the door for critical security loopholes. Even though individual device in network is secure, the network as a whole may be vulnerable to security attacks due to the option of

[1] https://www.fortinet.com/blog/business-and-technology/decrease-your-customers--network-complexity-for-increased-securi

allowing connectivity from multiple locations and multiple devices. This provides the entry point for cyber criminals to carry out undesirable activities[2].

1.1 Background

Although the connectivity and accessibility of devices from any location and any given time is the need of the hour, the ever-increasing data generated due to this is a big matter of concern. As suggested by Pritish et al. (2021), a report from International Data Corporation indicates that by 2020, the amount of data would mount up to 40 Zeta Bytes. The prime factors that stress the necessity of data deduplication are explosion of data caused by duplicate records created by users, communication equipment, applications etc, & the redundancy of data due to various reasons like creating and storing multiple copies of same data, again circulating them. The presence of duplicate data may not only reduce the performance of the network but may also lead to security gaps as multiple records for same person may be used by cyber criminals to mislead and perform cyber offences.

It is a well-known fact that data forms the basic building block of any organization. There could be various reasons for vulnerabilities in security of network devices like wider accessibility, availability, use of outdated versions of software etc, But handling the data flowing through the network devices is one of the initial parameters to be considered while mitigating the vulnerabilities of the network devices. The presence of duplicate data not only slows down the transfer of data across the network, but it also contributes to incorrect reports and eventually loss of business[3]. Above all, it may also be the reason for attacks by cyber criminals as they can use the duplicate/incorrect data as entry point to access the network of organizations.

1.2 Motivation

After getting the insight of how duplicate data can cause severe vulnerability issues in the security of Network devices in an organization, it is a matter of utmost importance that the data collected by various devices over the network should be accurately stored and processed. For example, there could be devices which have been decommissioned, but the devices associated to it can still be active in the network, this may open the doors for severe vulnerabilities and may allow attackers to use these as rogue devices to carry out ransomware attacks.
So, in an attempt to overcome this, the accurate data of assets collected by network devices is being consolidated and stored at one place. But the challenge faced here was that each network device uses a different parameter to fetch the data from the database. This may lead to various discrepancies like having duplicate records, or wrong hostnames to name a few. This research proposal is supposed to find a way by which FQDN of the devices can be used as a unique attribute which would fetch the exact information needed and not populate the duplicate records or incorrect hostnames as final outcome.

1.3 Research Question

How can Fully Qualified Domain Name(FQDN) be considered as a unique attribute associated with assets based on the communication of the devices in network?

---

[2]https://www.darkreading.com/vulnerabilities-threats/rethinking-vulnerabilities-network-infrastructure-as-a-software-system
[3] https://www.qgate.co.uk/blog/crm/10-reasons-why-duplicate-data-is-harming-your-business/

This research question is an attempt to prove the efficiency of FQDN as indexing parameter for deduplicating the data in network devices so that only unique records are obtained and stored in Asset Registry database.

1.4 Research Objectives

By the end of this research, the following objectives are targeted to be achieved:

- Thorough literature survey of the work done in the field of data duplication and the machine learning techniques used for it.
- Successful implementation of Record Linkage technique for data deduplication.
- Proving the efficiency of FQDN as a suitable indexing parameter for data deduplication in Network Devices using the algorithms k-Nearest Neighbour and AdaBoost Algorithms.
- Comparing the evaluation metrics of k-NN & AdaBoost machine learning models.

Record Linkage is an efficient data matching technique which is used when it is required to consolidate the data from various resources together based on certain similarity pattern between them[4]. Thus, Record Linkage has been considered here for deduplication of data as it will facilitate in the identification of similar data amongst multiple datasets that would be used. The following block diagram describes the basic the steps involved in deduplication approach used:

Figure 1: Deduplication of data using Record Linkage.

The data obtained from various network resources is initially pre-processed and cleaned. FQDN is used as the Indexing Parameter to compare the records and determine the similarity. The record pairs can then be classified as duplicates or non-duplicates.

2 Related Work

2.1 Background Of Data Deduplication:

A broad classification of various data deduplication methods suggested by Sulabh et al. (2021) surveyed each technique in very detail. The methods were classified based on factors like granularity, the technique of handling the duplicate records, location (server, client, network), format of data, Indexing methods, type of storage where the records are stored etc. Each technique was described in a comprehensive way to understand how the deduplication could be carried out in algorithmic perspective.

2.2 Data Deduplication using Indexing Parameter

Levy et al. (2018) proposed the automatic detection of best possible attributes for indexing in deduplication of the data. It was stated that the important steps involved in determining and removing duplicate records are indexing which is used for assigning one key for each record ,

[4]https://textbook.coleridgeinitiative.org/chap-link.html#motivation

comparing the records as per the value of key, and classification which is used to segregate the records after comparing.
While doing the experiment, a set of real and synthetic datasets were considered, and the stress was given on the initial indexing step. Both the datasets were assessed for deduplication by using the blocking method. The result was that the best possible attributes were automatically selected for indexing step. The main target was to grade the given attributes of the datasets for their efficiency and effectiveness of removing the duplicate records. The blocking and sorted neighborhood techniques of Indexing, the agnostic and configuration based functions were also addressed. The combination of various Indexing parameters was studied to achieve the target.

2.3 Record Linkage technique for deduplication

The important techniques of data duplications using record linkage were elaborately described by Shivani et al. (2021). The record linkage is a technique in which the data from multiple databases is linked on the basis of similarity among them. It is used to consolidate the data that belongs to the same category. Thus, it is very useful in finding the duplicate records. Various examples on how to use record linkage for comparing and assessing the duplicate records were discussed which can be useful while considering the machine learning models for the research question.

2.4 Machine Learning Models

Verschuurena et al. (2021) discussed various Supervised machine learning techniques for comparing the data using the similarity elements in the records.
The combination of machine learning models with string matching functions was applied in the experiment. The dataset chosen consisted of invoices from various kinds of organizations. The invoices were grouped together and were compared on the basis of similarity functions. The invoice dataset obtained in such way was used to train the machine learning models: Neural Network and Boosted Decision Tree to check the efficiency of algorithms.
Boosted Decision Tree: It boosts the efficiency of algorithm by combining various weak learners of decision trees to form a strong learner. The training of weak learners is carried out on the basis of the accuracy obtained in the previous iteration. In this experiment, the gradient boosting method was implemented and higher accuracy was achieved for removing the duplicate invoices from the datasets.
Neural Network: The Adam optimizer and binary cross-entropy function were used to train the dataset.
A further five-fold cross validation was performed on both the machine learning models for better accuracy. The results were then compared with framework developed by FISCAL and both Neural Network & Boosted Decision Tree performed well against it.
Chunbo et al. (2020) implemented the ensemble learning method called AdaBoost technique used to detect the overlapping data/similar data. Ensemble method was used to combine multiple learning algorithms to obtain a better predictive performance and accuracy as compared to using only a single machine learning model.
The three conventional machine learning methods: Logistic Regression, Decision Tree and Naive Bayes were used to detect the overlapping data. Then, the AdaBoost method was implemented, and result was compared. It was found that as compared to the conventional algorithms, AdaBoost was more effective in classifying the overlapping/similar parts of the data.
A new technique of removing redundant data was developed by Lakshmidevi et al. (2017). This was termed as two-level organized sampling selection method.

This method worked efficiently on larger datasets for removing the redundant data. The procedure followed for removing the duplicate data was blocking, comparison and classification. Initially, the blocking threshold was determined to form the record pairs. The two-level sampling method was uses for comparing the records for similarity. The AdaBoost technique was used was faster and more accurate than Support Vector Machine method.
Vincent et al. (2017) suggested a method in processing data for similarity using K-Nearest Neighbors algorithm. The data search for similarity is needed in many applications and deduplication is one of them. kNN algorithm was used in finding the similar content as it helps in finding the most similar content as per the query. kNN utilizes methods to find the distance between the similar records and implements indexing so that the volume of the data to be analyzed reduces drastically and thus the desired results were obtained with more accuracy and lesser time.
Abinaya et al. (2019) designed a kNN based web page crawler to remove duplicate links/records associated with it. This helped in identifying whether the data present in the forums is actually related to the original source or not. This model This model achieved an accuracy of 98% and was proved to be efficient.

2.5 Justification for need of the research question

The proposed methods have been applied on a comparatively smaller group of datasets. The research question of using FQDN as indexing parameter for deduplication of data requires to be performed on larger and more complex datasets of organization. Thus, it is very important to observe the limitations of the previous work done and improve it to make a better model for deduplication of data.
Also, in the previous work done, the removal of duplicate records is considered for improving the speed or using the storage space efficiently. There is not enough work done on deduplication of data in Network devices. Thus, Infrastructure Asset Management is a domain which requires a better research and development of efficient models for deduplication.

3 Research Methodology

The research methodology proposed here follows a Machine Learning workflow as suggested by Naman et al. (2019). This research methodology has been selected as it is lucid and effective at the same time. The implementation is very practical and flexible which would allow to achieve the desired outcome as per the research question. The stepwise block diagram is as shown below:

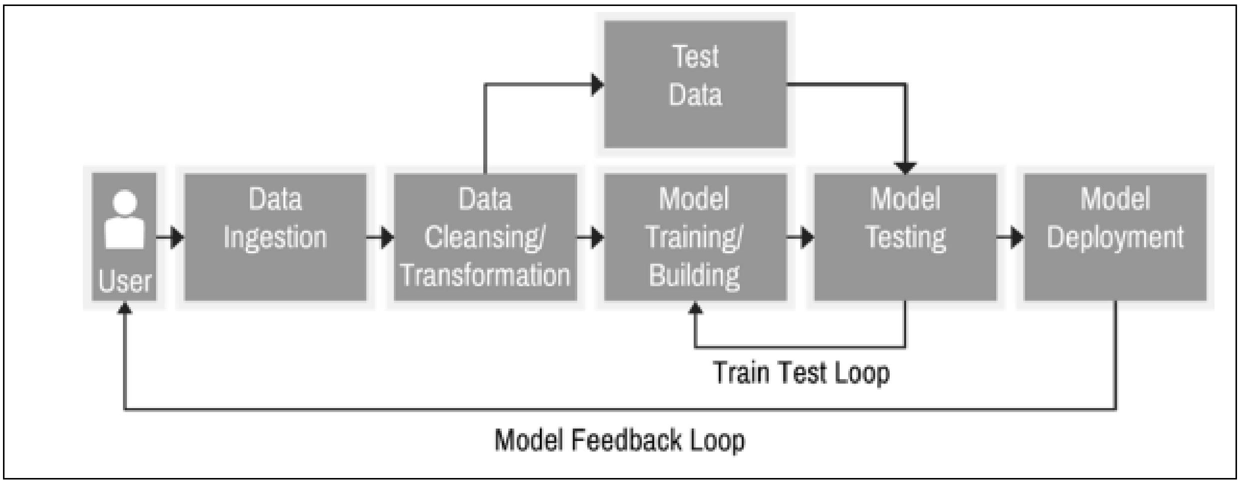


Figure 2: Machine Learning Methodology implemented.

The steps followed for implementing this methodology are as described below:

- **Data Extraction:**

  The data was obtained from 3 different network resources.

  The data obtained from one source SW (SolarWinds) has no duplicate records in it and only unique records are present. Thus, it has been used as the source of truth and the records present in CMDB(Configuration Management Database) and UDDR(Unknown Device Detection & Response) are compared with it to get the unique records.

  The data obtained from CMDB is comparatively better. But the data obtained from the third resource (UDDR) has the greatest number of duplicate records.

  Each and every record is being checked for the reason of having the duplicate records. While doing this, it was observed that there are broadly it can be classified into two categories: multiple devices having same IPs/FQDN and one device having many IPs and hence conflict with FQDN too.

  Also, the other reasons for duplicate records found till now were decommissioned devices still being scanned and populated by network devices, not cleaning the records after re-using the old IP for new device, spelling mistakes in FQDN.

  A thorough research was performed to check the reasons duplication of records, so that finally a dataset is obtained which has genuine records and do not miss even a single record when the deduplication algorithm is applied due to misunderstanding of the cause of duplication of records. For example, for some devices it is valid to have multiple IPs for single device because it has many interfaces and different IP is assigned for each interface. In such case, we need to check which is the unique FQDN name for that device and only that record should be populated.

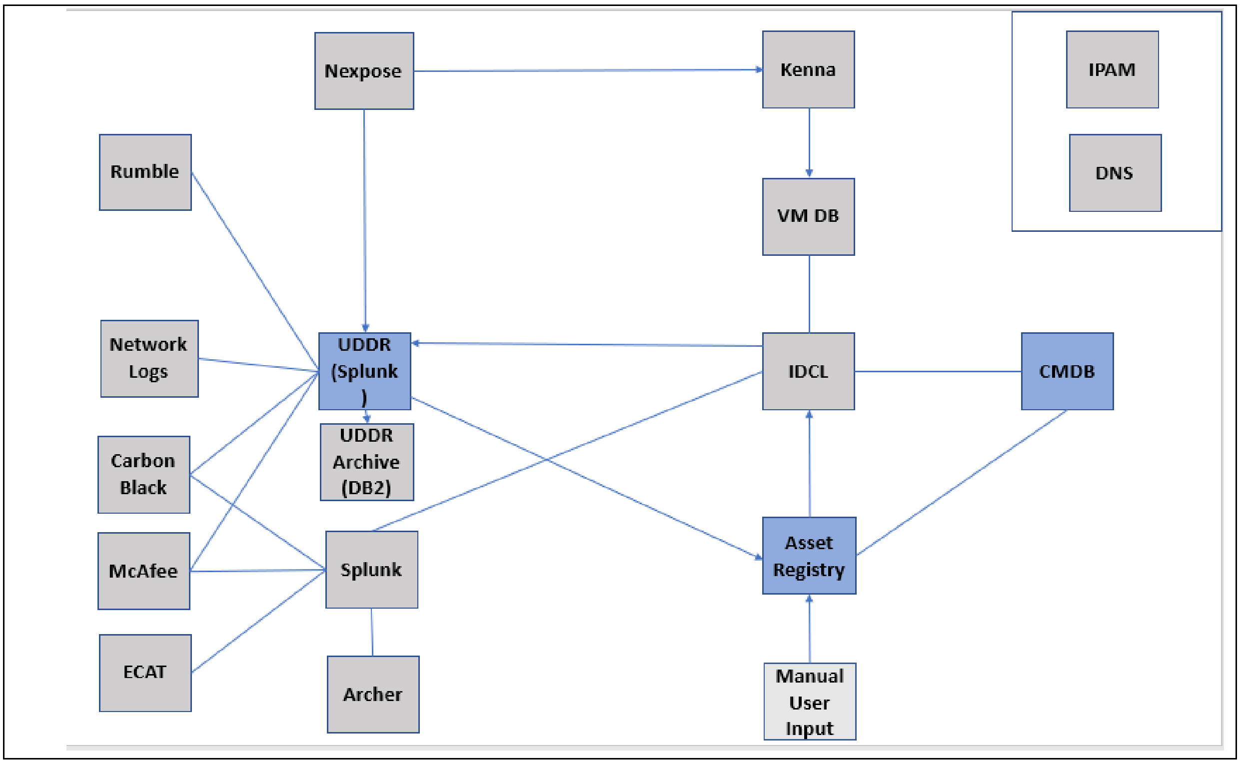


Figure 3: Block Diagram of the data flow in Network Devices.

The diagram above depicts the sources of data from various network devices. The combined data from CMDB and UDDR is stored in Asset Registry.

- **Data Pre-Processing:**
  The data obtained from the above-mentioned network resources CMDB & UDDR had many columns included. Only the columns which were relevant for research question were included. The data was then anonymised using character substitution and shuffling. It was made sure that the authenticity of the data is not tampered at all due to the anonymization.
- **Data cleaning:** The null values were checked and ffill method was used to replace the null values. The invalid values of IP address field of the dataset were removed. All the text fields were converted into Upper Case for uniformity of data.
- **Model Training:** The data thus obtained was trained for the K- Nearest Neighbour (KNN) and Adaboosting models respectively.
- **Model Testing:** The trained model was then iteratively tested for accuracy against test data which is different set of data each time the model is tested.
- **Model Deployment:** After meeting the desired accuracy, the model was then deployed so that the application can use the model.

4 Design Specification

After doing a thorough research on the previous work done, kNN and AdaBoost were found to be the most effective machine learning models for similarity matching and duplicate data removal. The detailed architecture of each machine learning model implemented has been described as following:

4.1 K-Nearest Neighbour Workflow:

Cunningham et al. (2021) describes kNN algorithm as a memory-based algorithm as the data training is done at the runtime and the data is needed to be in the memory during runtime. It is also known as Lazy Learning method. It classifies on the basis of training examples and thus is also called as example-based classifier. It is one of the simplest machine algorithms and it uses k nearest neighbours to determine the class.

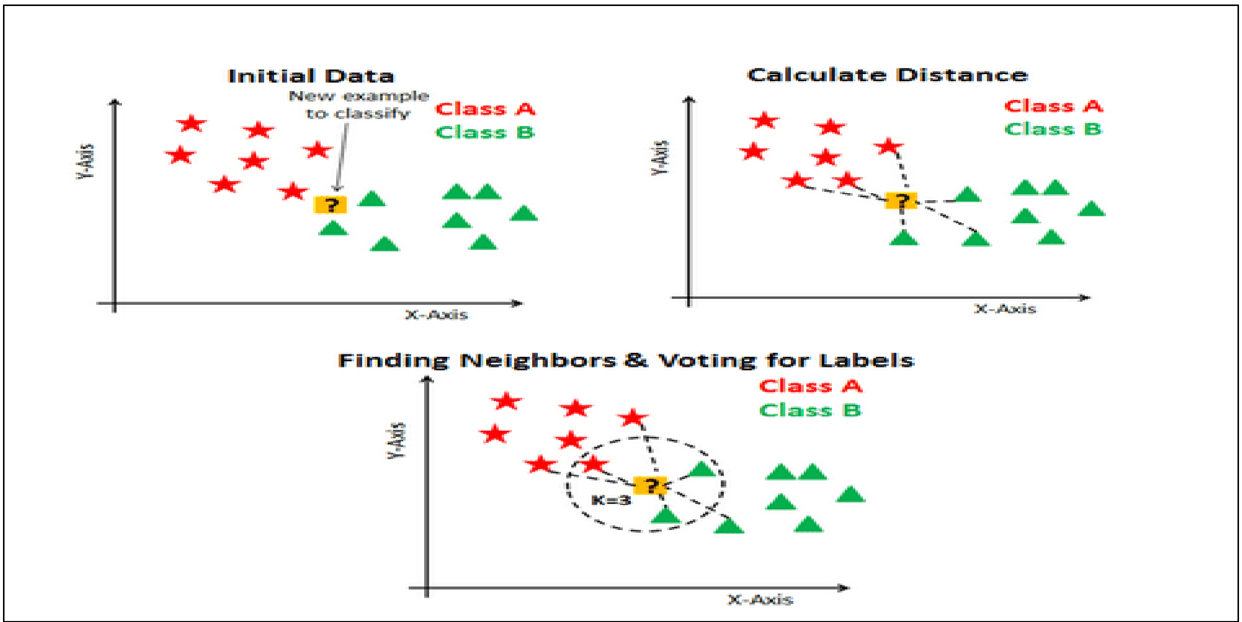


Figure 4: K-Nearest Neighbour Model's workflow.

The k-Nearest Neighbour algorithm uses the following workflow[5]:

- Initially, the data is divided into training data and test data.
- The value of k is selected.
- The function to calculate the distance is determined.
- A sample that is required to be classified is chosen from the test data and the distance is calculated to n number of training data samples.
- The distances thus obtained are sorted and the k-nearest data is chosen.
- The test class is then assigned to the class which has major number of k neighbours.

The data classification related issues can easily be resolved by implementation of distance-based machine learning models like k-NN as suggested by Najat et al. (2019). It has proven to be efficient on datasets which contain the combination of numbers as well as strings. Thus, it could be considered as a perfect algorithm for implementing the research question.

4.2 AdaBoost Architecture:

AdaBoost or Adaptive Boosting algorithm is one of the most efficient methods which improves predictive ability of a machine learning model according to Chengsheng et al.

[5] https://medium.datadriveninvestor.com/k-nearest-neighbors-knn-algorithm-bd375d14eec7

(2017). The Adaptive Boosting was designed to overcome the problems of tweaking the training data to enhance the weak classifier & to build a strong classifier after training the subsequent weak classifiers.

For this research proposal, AdaBoost algorithm is developed on Decision Tree classifiers.

It implements the combination of various weak learners to develop a strong classifier and obtain higher accuracy[6]. The weights of base classifiers are determined, and the data is trained in each iteration to ultimately form a strong classifier. The base classifier can be selected as any machine learning algorithm.

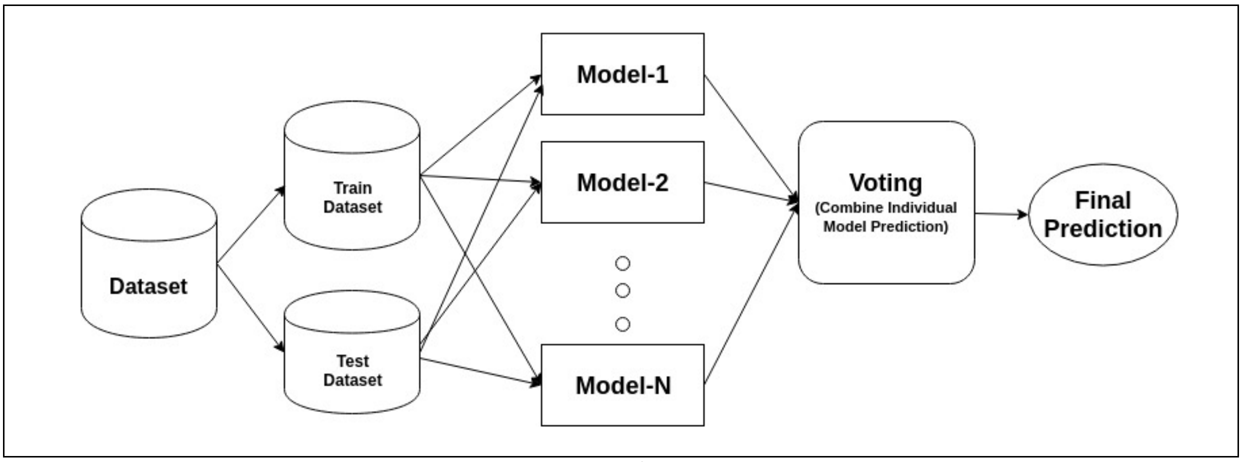


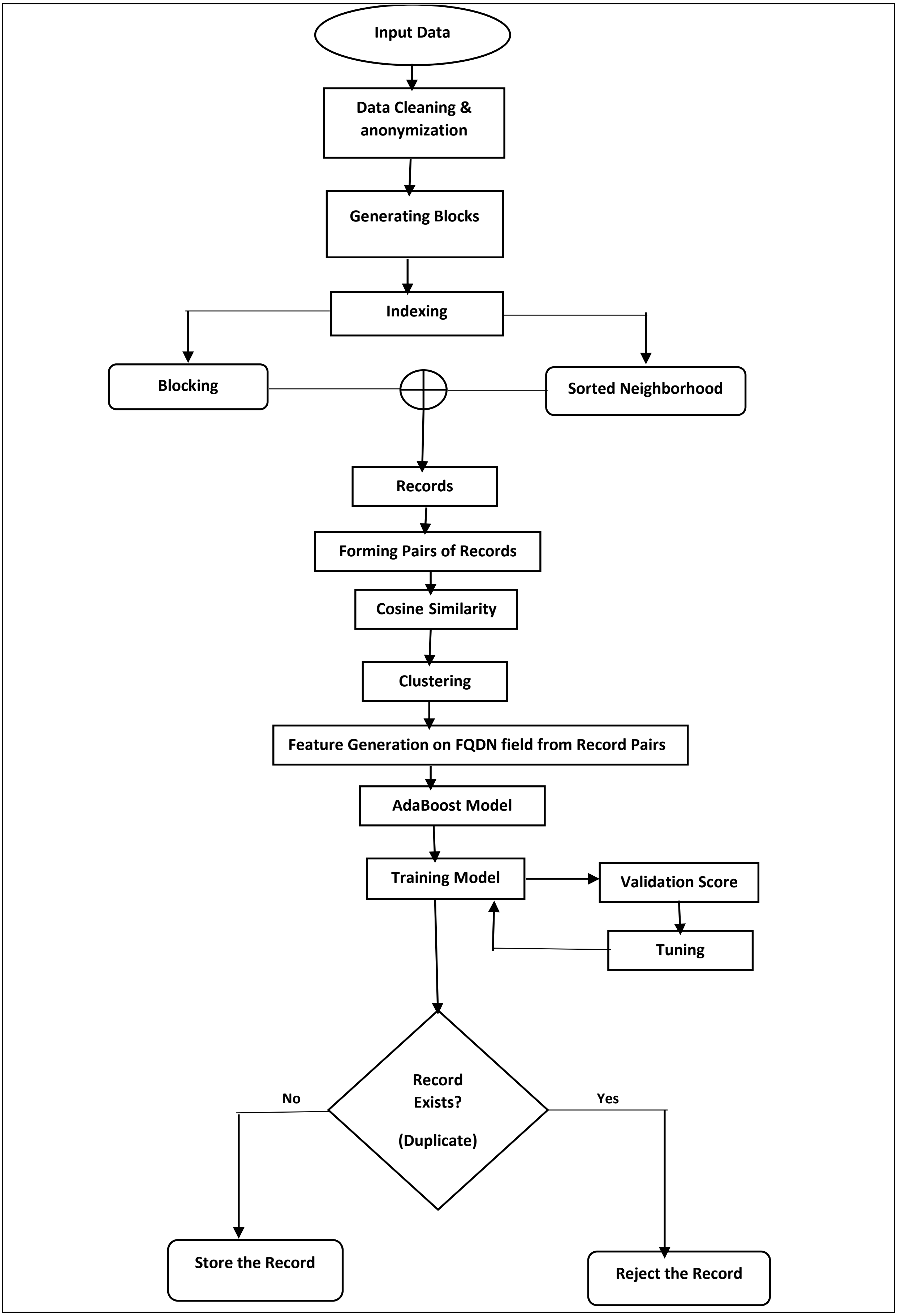
Figure 5: Architecture of Adaboosting Model

The important steps involved in AdaBoost are Sampling the data, training the data until the desired strong classifier is obtained and finally combining the prediction from individual model to get the best accuracy (Tharwat,2018).

4.3 Flowchart of the AdaBoost Algorithm:

The following diagram shows flowchart of the AdaBoost algorithm which was implemented in this research proposal:

[6] https://www.datacamp.com/community/tutorials/adaboost-classifier-python


Input Data
Data Cleaning & anonymization
Generating Blocks
Indexing
Blocking
Sorted Neighborhood
Records
Forming Pairs of Records
Cosine Similarity
Clustering
Feature Generation on FQDN field from Record Pairs
AdaBoost Model
Training Model
Validation Score
Tuning
Record Exists?
(Duplicate)
No
Yes
Store the Record
Reject the Record

4.4 The algorithm steps implemented for Adaboosting Algorithm:

Step 1: **START**

Step 2: Data loading as data frame.

Step 3: Generate the block of FQDN data.

Step 4: Create indexing based on FQDN.

Step 5: Create index based on Block method and Sorted Neighborhood method.

Step 6: Combine the records.

Step 7: Create the record pairs for clustering.

Step 8: Using Cosine Similarity to identify duplicate pairs.

Step 9: Group the pairs based on the Euclidean distance.

Step 10: Feature Generation on the field of FQDN from record pairs.

Step 11: Create an AdaBoost model as classifier.

Step 12: Train the model with the record pairs.

Step 13: Check the validation metrics for performance.

Step 14: Tune the model parameters for the better performance.

Step 15: Get the new data after tuning.

Step 16: Determine whether the duplicate record exists or not.

Step 17: If Yes → Reject the Record.

Step 18: If No → Store the Record.

Step 19: **END.**

5 Implementation

5.1 Record Linkage method implementation:

Record linkage method when used with traditional blocking methods produces faster and more accurate results for deduplication of data (Somasekhar,2018).

The main technique implemented for data deduplication in Network devices using FQDN as indexing parameter is the Record Linkage method. The steps followed for implementation of Record Linkage method for the machine learning models KNN & Adaboost are described in detail as below:

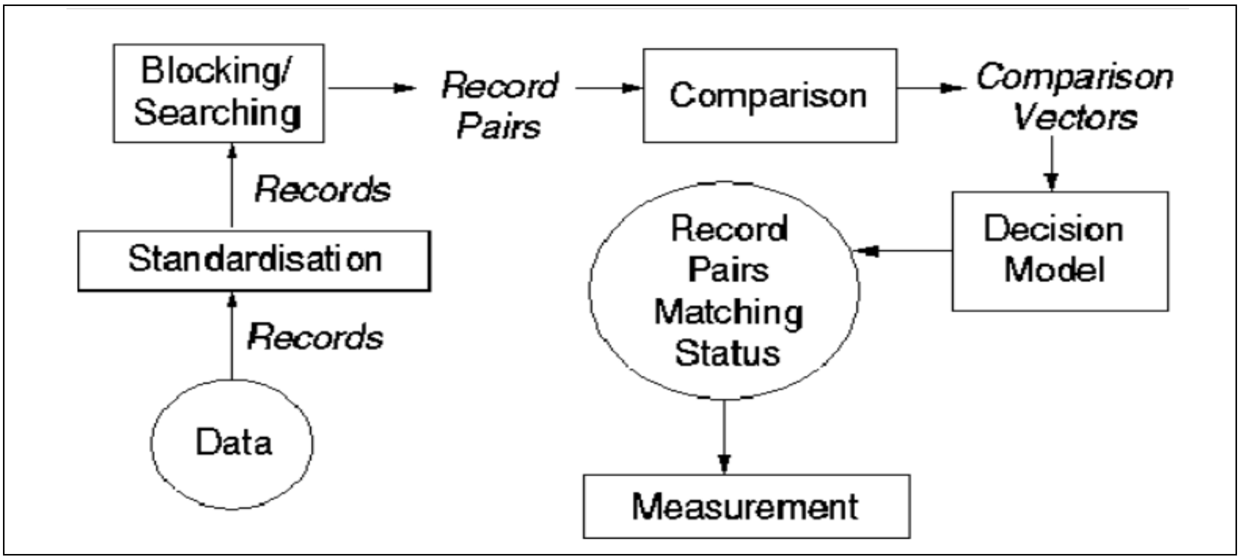


Figure 6: The Information flow of Record Linkage Technique.

The record linkage technique is used to search for similar records and combines them together so that they can be treated as one record [7]. This reduces the possibility of redundant data. The following steps are performed to apply record linkage method:

- **Pre-Processing:** It is very important to standardize the data into common format for uniformity because it will improve the chances of recognizing the duplicate records. All the text fields were converted into upper cases while implementing both the KNN & AdaBoost algorithm.
- **Indexing:** Once the data has been cleaned, the pair of records are created. Then, in order to check whether the records pairs are duplicates or not, the similarities between them are determined. The Indexing methods are provided by the record linkage toolkit in Python. While performing this research, Index by Blocking and Sorted Neighborhood techniques were used. The number of record pairs can be minimized to large extent by implementing blocking on a desired field of the dataset. Sorted Neighborhood is one more method which can generate pairs which have same values depending on the similarities of the field chosen.
- **Comparison & Similarity:** The next step after creating the record pairs is to form a comparison vector which will calculate the similarity scores between the pairs. The criteria of comparison could be any factor like string, numbers etc. The similarity score for string values needs to be calculated here and thus JaroWinkler was used. In order to calculate the similarity score using JaroWinkler, more emphasis is given on the initial part of the string and is very efficient in in identifying the pattern matching. Therefore, this algorithm is used to calculate the similarity score for text fields like FQDN in this case.
- **Classification:** Based on the dataset provided, the machine learning models are now trained to distinguish between duplicate and non-duplicate records. Prior to training the model, having a label column is mandatory in the data set do that the model would be able to differentiate between duplicate and non-duplicate records. To have an idea similarity score, the comparison vector for true duplicate pairs is created. The pairs are then converted into data frame format. The machine learning models kNN & AdaBoost were then trained with labeled data so that they could classify the data as duplicate or non-

[7] https://towardsdatascience.com/performing-deduplication-with-record-linkage-and-supervised-learning-b01a66cc6882

duplicate. The procedure was repeated till the desired accuracy for both the models was obtained.

5.2 Tools/ Languages Used for Implementation:

- **Anaconda Navigator 3:** It is a graphical user interface provided by Anaconda. The applications, packages can be easily launched and managed without the need of commands through command-line[8].
- **Jupyter Notebook 3:** It is a browser-based, interactive tool for running and editing the code. It executes and shows the output of each line just after the execution, which makes the task of troubleshooting easier.
- **Python 3.7:** It is an interpreted, object-oriented programming language[9] It is the most preferred language for making machine learning models, due to the ease of use and clarity.
- **Sklearn:** It is an open-source tool. It is the most important library for machine learning in Python [10]. It has numerous tools for performing predictive analysis like Regression, Classification, Clustering.
- **Data Linkage Library:** This toolkit is a library in Python which is used to create link between multiple sources of data on the basis of similarity parameter[11]. It consists of tools for Indexing, comparing records, classifiers etc.
- **Pandas:** It is a very powerful tool in Python used for data manipulation and analysis [12].
- **Numpy:** This library function is used in Python when there is a need to work with arrays [13]. It also supports many mathematical functions like algebra, matrices etc.
- **Matplotlib:** It is an exhaustive library in python which helps in creating visual plots, graphs for the results obtained in the implementation of machine learning models.[14]

6 Evaluation

After implementation of the machine learning models, and obtaining the output, its essential to compute the evaluation metrics as they determine the efficiency of the model and reveals how successful were the models in achieving the desired target. In order to calculate the Evaluation Metrics, the Confusion Matrix was plotted. It is a matrix with binary values and actual values are placed on one side while the estimated values are placed on other side. The parameters to be considered in confusion matrix[15] are:

- **True Positive (TP) :** When the estimated value and true value both are positive.
- **True Negative (TN):** When the estimated value and true value both are negative.
- **False Positive (FP):** When the estimated value is positive and true value is negative.
- **False Negative (FN):** When the estimated value is negative and true value is positive.

---

[8] https://docs.anaconda.com/anaconda/navigator/index.html
[9] https://www.python.org/doc/essays/blurb/
[10] https://scikit-learn.org/stable/
[11] https://recordlinkage.readthedocs.io/en/latest/about.html
[12] https://pandas.pydata.org/
[13] https://www.w3schools.com/python/numpy/numpy_intro.asp
[14] https://matplotlib.org/
[15] https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262

Based on the above parameters, the evaluation metrics can be calculated as follows:

- **Precision:** It depicts the actual positive values detected from all the positives estimated.

  Precision = TP / TP+FP

- **Recall** : This value shows the estimated positive from the total positive value.

  Recall = TP / TP+FN

- **F1-Score**: This score is calculated by taking the mean of Precision and Recall values.

  F1 score = 2*(Precision*Recall)/ (Precision + Recall)

6.1 Experiment 1

KNN Algorithm: The accuracy plot and confusion matrix for the KNN algorithm implementation is as shown below:

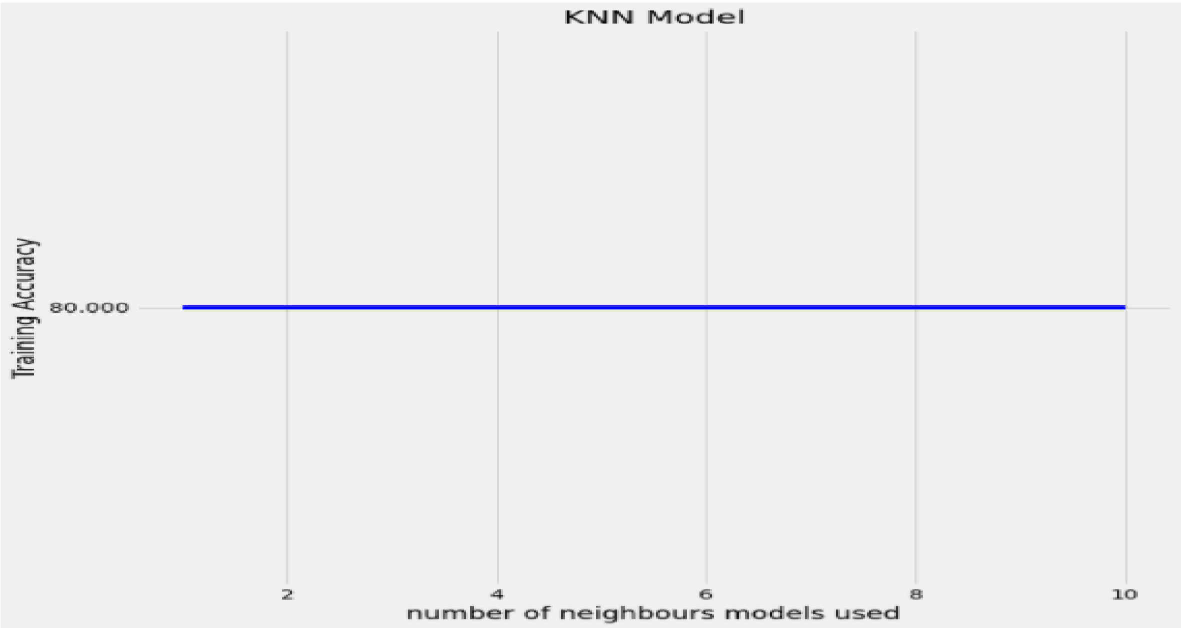


Figure 7: Accuracy plot for KNN Algorithm implementation.

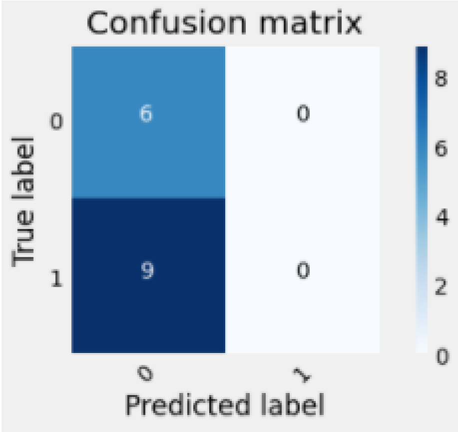


Figure 8: Confusion Matrix for KNN Algorithm implementation.

6.2 Experiment 2

Adaboosting Algorithm:

The accuracy plot and confusion matrix for Adaboosting algorithm implementation is as shown below:

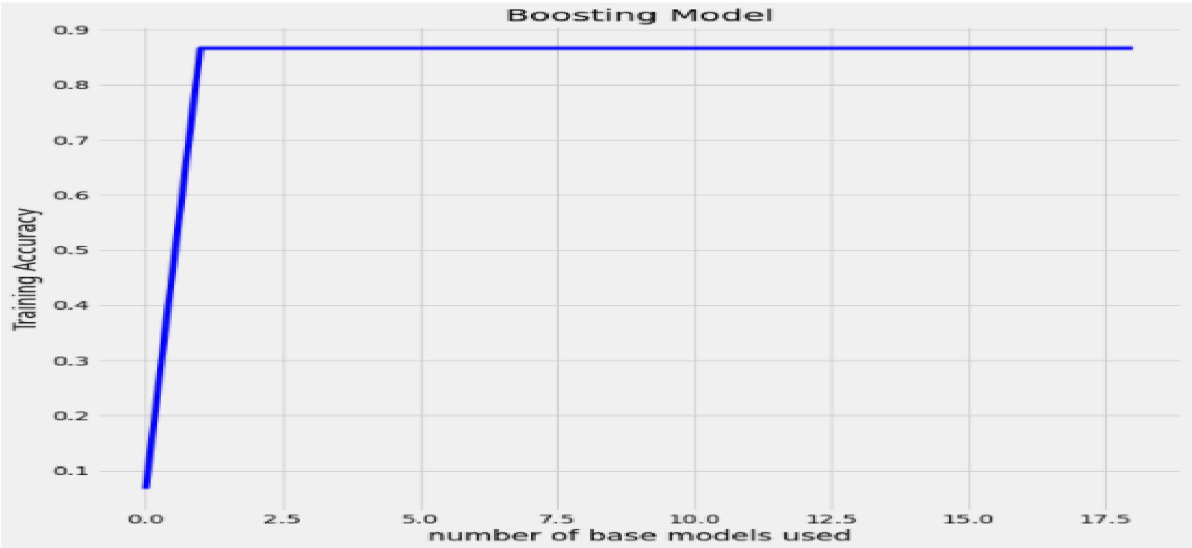


Figure 9: Accuracy plot for Adaboosting Algorithm implementation.

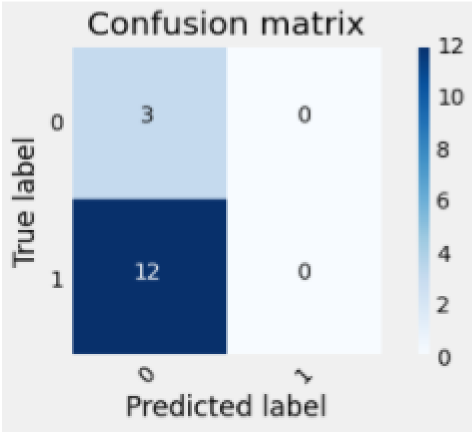


Figure 10: Confusion Matrix for Adaboosting Algorithm implementation.

6.3 Discussion

As we look into the accuracy graph and confusion matrix plots of both kNN and AdaBoost models, it can be concluded that the Adaboost Algorithm is a better technique than K-Nearest Neighbour Algorithm with an accuracy of 87%. Whereas the accuracy of kNN Model is 80%. Thus, the research question of proving the efficiency of FQDN as indexing parameter for data deduplication has been justified. All the research objectives mentioned have also been successfully achieved.

6.3.1 Comparison of Machine Learning Models implemented:

Model Implemented	Accuracy	Precision	Recall	F1 Score
KNN Algorithm	0.80	0.16	0.40	0.23
Adaboosting Algorithm	0.87	1	0.86	0.92

The comparison of KNN & AdaBoost Algorithm shows that both the models performed up to the mark. But the Accuracy, Precision, Recall & F1-Score for AdaBoosting Algorithm were better as compared to the KNN Algorithm.
The accuracy of kNN was 80% which is a good value and proves that the model is efficient. But the values of precision, recall and f1 score were not up to the mark. The possible reason for this could be that there were very few numbers of duplicates in the dataset chosen. This model would be efficient for larger datasets where the possibility of having duplicate values of FQDN is more.

7 Conclusion and Future Work

7.1 Conclusion

The Evaluation Metrics calculated in the previous section were satisfactory and AdaBoost algorithm proved to be a better model for removing duplicate records. The overall research proves that the FQDN can be considered as indexing parameter for removing duplicate data in Network devices. Thus, the purpose of research is fulfilled.

7.2 Limitations

Due to the restricted access to network devices, time constraint, and complexity of Infrastructure of the Network Devices, just data from selected network devices was considered.

7.3 Future Work

This research paper successfully implemented the KNN & Adaboost algorithms to remove the duplicate data and show the unique record pairs. In future, we can extend its functionality to printing the output of unique records in a consolidated form in a separate output file which can be handy to refer and add to the Asset Registry.

If given permission to access the Network devices, larger datasets can be considered from various network resources and better accuracy for deduplication of data can be obtained.

Also, a greater number of algorithms can be tested and compared for accuracy.

References

Pritish A. Tijare,2 - Data deduplication concepts, Editor(s): Tin Thein Thwel, G.R. Sinha, Data Deduplication Approaches, Academic Press,2021,Pages 17-35,ISBN 9780128233955, Available at : https://doi.org/10.1016/B978-0-12-823395-5.00015-X.

Sulabh Bansal, Prakash Chandra Sharma, 5 - Classification criteria for data deduplication methods, Editor(s): Tin Thein Thwel, G.R. Sinha, Data Deduplication Approaches, Academic Press, 2021, Pages 69-96, ISBN 9780128233955, https://doi.org/10.1016/B978-0-12-823395-5.00011-2.

Levy Souza, Fabricio Murai, Ana Paula C. da Silva and Mirella M. Moro. (2018). “Automatic Identification of Best Attributes for Indexing in Data Deduplication” [Online]. Available: https://homepages.dcc.ufmg.br/~mirella/projs/deduplica/paper14.pdf

Shivani Girish Dhok, Ankit A. Bhurane, 8 - Essentials of data deduplication using open-source toolkit,Editor(s): Tin Thein Thwel, G.R. Sinha,Data Deduplication Approaches, Academic Press,2021,Pages 125-151,ISBN 9780128233955. Available: https://www.sciencedirect.com/science/article/pii/B9780128233955000173

Pim Verschuurena, Serena Palazzo, Tom Powell, Steve Sutton, Alfred Pilgrim, Michele Faucci Giannelli. (Sept 2021). “Supervised machine learning techniques for data matching based on similarity metrics” [Online]. Available: https://arxiv.org/pdf/2007.04001.pdf.

Chunbo Liu, Yitong Ren, Mengmeng Liang, Zhaojun Gu, Jialiang Wang, Lanlan Pan, Zhi Wang, "Detecting Overlapping Data in System Logs Based on Ensemble Learning Method", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8853971, 8 pages, 2020. https://doi.org/10.1155/2020/8853971

LAKSHMIDEVI, R. and BABU, D., 2017. “A Novel Approach to New Two Level of Organize Sampling Selection for Reducing the Redundancy”. Available: http://ijsetr.com/uploads/216534IJSETR14136-435.pdf

Vincent T. Lee, Amrita Mazumdar, Carlo C. del Mundo, Armin Alaghi, Luis Ceze, Mark Oskin. arXiv:1606.03742v2 [cs.DC] 10 Jul 2017. "Application-Driven Near-Data Processing for Similarity Search" Available: https://arxiv.org/pdf/1606.03742.pdf

Abinaya, P. and Jayavadivel, R., 2019. Normalization Techniques for Identifying Duplicate Records from Multiple Data Sources. *ICTACT Journal on Soft Computing*, *10*(1), pp.1994-1998. Available: http://ictactjournals.in/paper/IJSC_Vol_10_Iss_1_Paper_2_1994_1998.pdf

Sodhi, Pinky and Awasthi, Naman and Sharma, Vishal, Introduction to Machine Learning and Its Basic Application in Python (January 6, 2019). Proceedings of 10th International Conference on Digital Strategies for Organizational Success, Available at SSRN: https://ssrn.com/abstract=3323796 or http://dx.doi.org/10.2139/ssrn.3323796

Pádraig Cunningham and Sarah Jane Delany. 2021. K-Nearest Neighbour Classifiers - A Tutorial. ACM Comput. Surv. 54, 6, Article 128 (July 2022), 25 pages. DOI:https://doi.org/10.1145/3459665

Ali, N., Neagu, D. & Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Appl. Sci.* **1,** 1559 (2019). https://doi.org/10.1007/s42452-019-1356-9

Chengsheng, Tu & Huacheng, Liu & Bing, Xu. (2017). AdaBoost typical Algorithm and its application research. MATEC Web of Conferences. 139. 00222. 10.1051/matecconf/201713900222.

Tharwat, Alaa. (2018). AdaBoost classifier: an overview. 10.13140/RG.2.2.19929.01122. Available:https://www.researchgate.net/publication/323119678_AdaBoost_classifier_an_overview

Somasekhar, G., K, S., P, K., & Sandeep G, S. (2017). Record linkage and deduplication using traditional blocking. *International Journal of Engineering & Technology*, *7*(1.1), 294-296. Doi :http://dx.doi.org/10.14419/ijet.v7i1.1.9705