International Journal of Computer Applications Technology and Research
Volume 8–Issue 03, 58-65, 2019, ISSN:-2319–8656

Application of Cortical Learning Algorithms to Movement Classification

Abdullah Alshaikh
School of Computing and Digital Technologies
Staffordshire University
Stoke-on-Trent, United Kingdom

Mohamed Sedky
School of Computing and Digital Technologies
Staffordshire University
Stoke-on-Trent, United Kingdom

Abstract: Classifying the objects' trajectories extracted from Closed-Circuit Television (CCTV) feeds is a key video analytic module to systematize or rather help to automate both the real-time monitoring and the video forensic process. Machine learning algorithms have been heavily proposed to solve the problem of movement classification. However, they still suffer from various limitations such as their limited ability to cope with multi-dimensional data streams or data with temporal behaviour. Recently, the Hierarchical Temporal Memory (HTM) and its implementation, the Cortical Learning Algorithms (CLA) have proven their success to detect temporal anomalies from a noisy data stream. In this paper, a novel CLA-based movement classification algorithm has been proposed and devised to detect abnormal movements in realistic video surveillance scenarios. Tests applied on twenty-three videos have been conducted and the proposed algorithm has been evaluated and compared against several state-of-the-art anomaly detection algorithms. Our algorithm has achieved 66.29% average F-measure, with an improvement of 15.5% compared to the k-Nearest Neighbour Global Anomaly Score (kNN-GAS) algorithm. The Independent Component Analysis-Local Outlier Probability (ICA-LoOP) scored 42.75%, the Singular Value Decomposition Influence Outlier (SVD-IO) achieved 34.82%, whilst the Connectivity Based Factor algorithm (CBOF) scored 8.72%. The proposed models have empirically portrayed positive potential and had exceeded in performance when compared to state-of-the-art algorithms.
Keywords: video analytic; movement classification; machine learning; video forensic; hierarchical temporal memory; cortical learning algorithms

1. INTRODUCTION

In recent years, the development of outdoor surveillance technologies has captured the interest of both researchers and practitioners across the globe. The objective of these technologies is to detect the presence of objects that are moving in the field of view of a CCTV camera(s) for national security, traffic monitoring in big cities, homes, banks and market safety applications or to automate the video forensic process. The use of video analytic technologies has gained much attention in the research community and the global security around the world [2].

The purpose of intelligent visual surveillance in most cases is to detect, recognise, or learn interesting events that seem to constitute some challenge to the community or area of the target [3]. These challenges posed by defining and classifying events as unusual behaviour [4], abnormal behaviour [5], anomaly [6] or irregular behavior [7].

When considering a real environment and trying to relate the way objects interact in surveillance covered area, it is not so easy interpreting every activity correctly. Cluttered environments that contain so many moving objects pose a challenge for many anomaly-detection algorithms. However, in real life cases, these are the kind of scenarios we meet when considering movement classification in video surveillance [2].

There are many hurdles faced by outdoor surveillance system designers and implementers. The first step toward automated activity detection is detection, tracking and classification of moving objects in the field of view of CCTV cameras; another challenge is that sensor resolution is finite, and it is impractical for a single camera to observe the complete area of interest. Therefore, multiple cameras need to be deployed. Also, the detected objects are context-dependent, but for a general surveillance system any independently moving object such as a vehicle, animal or a person are deemed to be interesting, but detecting and classifying these objects is a difficult problem because of the dynamic nature of object appearances and viewing conditions in practical scenarios [8].

In general, to develop a video analytic system that can detect and classify the presence of objects moving in its field of view, that system must be able to: i) detect and classify objects into various categories, ii) track the detected objects over time and iii) classify their movements. Each of the above tasks poses its challenges in term of design and implementation. However, detecting, classifying and analysing the movements of objects were traditionally a manual job performed by humans in which the guaranty of absolute attention over time by a human on duty remains small, especially in practical scenarios [9].

A video analytic system consists of many modules; e.g. change/object detection, object classification, object tracking and movement classification. One key module is the movement classification module. In this module, the movements of detected objects are recorded and compared to infer anomaly. State-of-the-art movement classification rely mainly on rule-based classification techniques, i.e. applying a set of pre-determined spatio-temporal rules, often based on statistical learning techniques, which have been found to correlate to what humans, would interpret as situations of interest, corresponding to threats. Where the abnormalities in the video are traced and reported to the user [10]. Such techniques attempt to learn normal movements to identify abnormal movements.

Machine learning techniques have been heavily proposed to solve the problem of movement classification. However, they still suffer from various limitations such as their limited

ability to learn data streams or data with temporal behaviour. In the attempt of mimicking the function of a human brain, learning models inspired by the neocortex has been proposed which offer better understating of how our brains function. Recently, new bio-inspired learning techniques have been proposed and have shown evidence of superior performance over traditional techniques. In this regard, Cortical Learning Algorithms (CLA) inspired from the neocortex are more favored. The CLA processes streams of information, classify them, learning to spot the differences, and using time-based patterns to make predictions. In humans, these capabilities are largely performed by the neocortex. Hierarchical Temporal Memory (HTM theory attempts to computationally model how the neocortex performs these functions. HTM offers the promise of building machines that approach or exceed the human level performance for many cognitive tasks [11].

Considering the need for improved video analytic systems for the detection and classification of events in video feeds, various benchmark datasets are available in public domain [12]. For example, the i-Lids, Imagery Library for Intelligent Detection Systems, datasets developed by the UK Home Office [14] and VIRAT dataset, a large-scale benchmark dataset for event recognition in surveillance video, developed by DARPA, Defence Advanced Research projects agency [15]. These datasets are captured from realistic surveillance scenarios.

This article introduces a novel CLA-based movement classification algorithm to classify the movements of moving objects in realistic video surveillance scenarios. The performance evaluation of how well the proposed algorithm can differentiate between an unusual movement and a normal movement was carried out based on the ground truth provided by the used dataset [15]. A comprehensive objective evaluation was adopted, which is targeted at comparing the output of the proposed algorithm to state-of-the-art movement classification algorithms.

2. RELATED WORK

A movement classification module attempts to understand the trajectories of tracked objects and the interactions between them. In this stage, the technique may attempt to classify the consistent and predictable object motion. Movements could be classified into two categories, stand-alone or interactive, where stand-alone movements refer to the action of an individual object, while interactive movements refer to the interaction between two or more objects. Statistical learning techniques are often utilised to classify between normal and abnormal activities, based on a priori information, and a user query. The overall aim is to produce a high level, compact, natural language description of the scene activities.

When considering movement classification, the question "what is going on in a scene" is considered [2]. In this sense, there must be a clear definition of what is considered normal/usual and abnormal/unusual. Abnormalities are defined as actions that are fundamentally different in appearance or action done at an unusual location, at an unusual time [16]. When considering anomaly detection algorithm, detecting the spot and where anomalies occur with little to no false alarm is of great emphasis.

Some researches in video surveillance focused on action recognition, body parts recognition and body configuration estimation [15][17]. There has been a recent advancement in researching about semantic descriptions of humans in challenging unconstrained environments [18].

When considering modelling scene behaviour, statistically based methods are currently used instead of rule-based methods, which use already defined rules to classify the normal behaviour from abnormal behaviour that was previously used. Statistical based methods are believed to achieve a more robust solution to get useful information in behaviours of a considered scenario [2]. This kind of method is based on either learning from normal behaviour and then using such criteria to differentiate between normal behaviour and abnormal behaviour or the process of learning and detecting normal and abnormal behaviour is done automatically. The challenge with this kind of approach is that human beings behave abnormally in different ways, and as such systems may trigger wrongly, hence a false alarm. Much work has been done on using machine learning techniques to train some algorithms on normal behaviours and abnormal behaviours so that such algorithms could effectively differentiate these two cases. Unfortunately, it is a hard task to exhaust all the abnormal behaviour to be carried out in real life scenarios [19].

Several attempts have been in place for movement classification, using different techniques such as pattern recognition [20], artificial intelligence [21], and neural network techniques [23].

There has been some research in using AI and NN techniques to solve problems of movement classification. However, AI and NN have shortcomings of classifying what is abnormal based on the training it previously acquired from the inputs (Hawkins and Blakeslee 2004). However, Bio-inspired computational models have proved to be successful over conventional methods. Although ANNs remain of the most active classification techniques in various applications and researches, it is not without flaws as mentioned earlier. One effective method is known as Back Propagation Neural Network (BPNN), which has shown more accuracy when compared with maximum likelihood method. However, this work is, therefore, focusing on the study of the CLA through the investigation of the possibility of applying it to movement classification, to develop a novel movement classification technique.

The problem with the movement classification in video analytics is discussed in detail. To understand the issue better, the literature on the subject is discussed with a further focus on the weaknesses and strengths of these research studies. Seemingly, there are challenges that are associated with video forensics as far as information analysis is concerned. The analysis of the video in different contexts depends on the method that is used. Here in, there are discussions on the different researches that have been conducted in the past seeking to shed more light on video analytic especially in its movement classification task.

Artificial Neural Networks (ANNs) has been noted to be an active research area since the 1980s and has made a huge success. It is also indicated herein that it faces several challenges such as selection of the structure and the parameters of the networks, selection of the learning samples, selection of the initial values, the convergence of the learning algorithms amongst several other challenges [24].

Despite series of effort made by AI researchers and recent effort made by Artificial neural network researchers to build viable algorithms for achieving human-like performance, they still suffer fundamental flaws, as they could not meet very three important criterion which the brain had. These are:

- In real cases, brains process a rapidly changing stream of information and not the static flow of information.
- The feedback connections which dominates most connections in the neocortex were not understood
- Any theory that wishes to imitate the brain should take the physical structure of the brain into consideration and as such neocortex is never a simple structure.

3. CORTICAL LEARNING ALGORITHMS

HTM approach is based on modelling the structure of the neocortex and how it works. However, approaches like AI is built upon the idea of a neural network, which in essence, NN does not behave in the same way as the brain thinks, and this is not what is considered intelligence. The HTM is different form NN in that there is no need to carry out a backpropagation since the HTM is usually being updated as the information flows for the first time. NNs cannot produce systems that can have intelligent behaviour; this approach is thought to be implemented using the Cortical Learning Algorithm (CLA). This approach is usually made up of six very important components which include:

- online learning from streaming data,
- hierarchy of memory regions,
- sequence memory,
- sparse distributed representations,
- all regions are sensory and motor, and
- Attention.

The CLA processes streams of information, classify them, learning to spot differences, and using time-based patterns to make predictions.

In this study, the Cortical Learning Algorithm (CLA) is applied. The choice of classification algorithm depends on functionality and the design of such algorithm [1]. The CLA processes streams of information, classify them, learning to spot differences, and using time-based patterns to make predictions [28]. Fan et al. [30] critically analysed HTM theory and concluded that the CLAs enable the development of machines that approach or surpass performance level of human for numerous cognitive tasks. The neocortex is said to control virtually most of the important activities performed by mammals including touch, movement, vision, hearing, planning and language [11]. HTM models neurons which are arranged in columns, in layers, in regions, and in a hierarchy. HTM works on the basis of a user specifying the size of a hierarchy and what to train the system on, but how the information is stored is controlled by HTM. According to [9], the CLA processes streams of information, and also classifies the information, learning to identify variations, and using time-based patterns to make predictions. However, the place of time is significant in learning, inference and prediction. The temporal sequence is learned from HTM algorithm from the stream of input data; despite the difficulty in predicting the sequence of patterns. This HTM algorithm is very important since it captures the so-called building block of the neural organisation in the neocortex **Error! Reference source not found.**.

4. PROPOSED TECHNIQUE

This work proposes the application of HTM for movement classification. The proposed algorithm, which is based on the CLA that learns to predict a sequence of movement, will learn some events, represented by a sequence of movements and then it will be tasked to differentiate between an event similar to the ones that have been learnt and events that have not been learnt. This would be a desirable property since, post-incident analysis, the detection of abnormal movement is required. A slightly erroneous copy of the learned sequences will be presented to the algorithm, which will recover quickly after any unexpected or suspicious movement patterns. In the section below, the requirements for the proposed technique are presented.

4.1 Requirements

A set of requirements for the proposed movement classification technique are:

- Spatial and temporal object movement classification input pattern (feed-forward). Feeding the HTM network stream of movement trajectories discovering the temporal statistic to predict how a specific object type moves.
- Normal trajectory patterns to be learned and efficient storage for storing a representation of learnt patterns.
- A defined inhibit radius that defines the area around a column that actively inhibit.
- Provision of scalar value which indicates the connection state of potential synapses. This value indicates the synapses are not formed if the value is below the threshold otherwise it is valid if it is above the threshold.
- A set of model parameters.

4.2 Implementation

The implementation of the proposed movement classification algorithm follows the following steps:

Model creation: the model is created by running swarm to define the model parameters and adjust for any modifications if required. The swarm starts by running the Permutations function to automatically generate multiple prediction experiments that are permutations of a base experiment via the CLA engine. The type of inference is to be specified, e.g., 'Temporal Anomaly,' the encoder settings as well as Spatial Pooler and Temporal Pooler parameters. The encoder parameters include the type of the encoder, e.g. ‘Scalar,’ ‘Adaptive Scalar,’ ‘Category’ or ‘Date’ encoders. N the number of bits to represent input and w the width or the number of ‘On’ bits (1’s). The Spatial Pooler parameters include: Column Count, number of cell columns in the cortical region, number of Active Columns Per Inhibition Area, maximum number of active columns in the SP region's output (when there are more, the weaker ones are suppressed), potential Percentage, the percent of the columns' receptive field is available for potential synapses, synapse Permanence Connected, the default connected threshold. Any synapse whose permanence value above the connected threshold is a "connected synapse," meaning it can contribute to the cell's firing. The Temporal Pooler parameters include Column Count, number of cell columns in the cortical region (same number for Spatial Pooler), cells Per Column, the number of cells (i.e., states), allocated per column, max Synapses Per

Segment, maximum number of synapses per segment, max Segments Per Cell, maximum number of segments per cell, initial Perm, initial Permanence, permanence Increment, permanence increment and permanence Decrement, permanence decrement, min Threshold, minimum number of active synapses for a segment to be considered during search for the best-matching segments, activation Threshold, Segment activation threshold, a segment is active if it is greater than this threshold.

Learning: the model starts by enabling learning, this indicates the training state. After this stage, the model has learned normal activities and is ready to perform prediction and anomaly detection.

Anomaly detection: When the preparation is finished, the learning is disabled, and the model is exchanged in the anomaly detection state. The model can perform detection of normal and abnormal activities.

5. TEST AND EVALUATION

There are two modes of evaluation commonly used for testing datasets; they include scene-independent and scene-adapted learning recognitions. According to Wan et al.[27], scene-independent involves trained event detector on the scene which is not considered in the test. In this case, the test clips are used during the test process. Meanwhile, in the scene-adapted learning recognition, the used of clips are involved in training processes. Anjum et al. [17] stated that evaluation techniques consist of multi-object tracking and functional scene recognition that is ground-based annotation giving useful basis for large-scale performance evaluations and real-life performance measures. As a result, various metrics are devised for the evaluation of movement classification algorithms

5.1 Datasets

Outdoor scenarios have been targeted in most post-incident analysis cases. Not all publicly available action recognition and movement classification datasets characterise realistic real-life surveillance scenes and/or events as they, mostly consist short clips that are not illustrative of each action performed **Error! Reference source not found.**. Some of them provide limited annotations which comprise event examples and tracks for moving objects, and hence lack a solid basis for evaluations in large-scale. Moreover, according to Khanam and Deb **Error! Reference source not found.**, performance on current datasets has been flooded, and therefore requires a new more complex and large dataset to improve development.

large-scale dataset enables the evaluation of movement classification algorithms. The dataset was designed to challenge the video surveillance fields required to its background clutter, resolution, human activity categories and diversity in scenes than existing action recognition datasets. Therefore, VIRAT video datasets are distinguished by the following characteristics; diversity, quality, realistic, natural, ground, aerial, wider range of frame and resolution **Error! Reference source not found.**.

a. Human getting into vehicle

b. Human entering a facility

c. Human carrying a bag

d. Human accessing a trunk

e. Human exiting car

f. Human entering a car

g. Not known

Figure 1 Snapshots from VIRAT video dataset

5.2 VIRAT video dataset

There are total of 11 scenes in VIRAT video that were captured by stationing high definition cameras and encoded in H.264. Individual scene consists of many video clips with various activities. The file name format is unique which makes it easier for the identification of videos that are from the same scene using the last four digits that indicate collection group ID and scene ID. As shown in figure 16-1, the datasets snapshots show the VIRAT dataset in three sample activities. In this paper, the VIRAT video dataset is used to perform the evaluation for the proposed movement classification algorithm. There are two categories in which the video dataset is divided into testing and training datasets. The latter contains video scenes with several categories of human and vehicle activities recorded by stationary cameras, in a surveillance setting, in scenes considered realistic. Six object categories are included, unknown, person, car, other vehicle, other object and bike. Seven activities are presented, unknown, loading, unloading, opening-trunk, closing-trunk, getting-into-vehicle and getting-out-of-vehicle.

5.2.1 Annotation Standards

In annotation standards, 12 events are either fully annotated or partially annotated were present. The fully annotated videos have Thirteen (13) event types labelled from 0 to 12 while the partial annotation has Seven (7) event types labelled from 0 to 6. Event, activity, is represented as the set of objects involved with the temporal interval of interest e.g. “PERSON loading an OBJECT to a VEHICLE” and “PERSON unloading an OBJECT from a VEHICLE”. All this is clearly shown in the recorded videos. A person or object are annotated if they are within the vicinity of the camera and the dataset stops recording a few seconds after the object is out of the vicinity of the camera.

The training dataset includes two sets of annotation files that describe a. the objects and b. the events depicted in the videos. Samples of the event annotation files and the object annotation files are shown in Table 1 and Table 2. These annotation files were generated manually and represent the ground truth. The training includes 66 videos representing three scenes.

5.3 Evaluation

This evaluation is basically based on the documents from VIRAT DATASET RELEASE 2.0 accessed from VIRAT DATASET . The VIRAT Video Dataset Release 2.0 is used in the analysis and evaluation of the data throughout this paper.

The results of the HTM anomaly detection algorithm is represented by an anomaly score for each field; a field represents a movement. The anomaly scores vary between Zero and One. Where Zero represents a normal movement (ideally part of an event that has been learned) and One represents an abnormal movement. Values between Zero and One represent the anomaly score, where values close to Zero represent movements closer to normal ones and values closer to One represent movements that are closer to abnormal movements, i.e. suspicious.

First, the evaluation starts with the first scenario, for each record, the Precision, Recall and F-measure are calculated by comparing the resulted anomaly score with a threshold. If the anomaly score is less than the threshold, the detection is considered correct. In the case of an event that has not been shown in the training dataset, if the resulted anomaly score is greater than the threshold the result is considered correct. The True Positive, True Negative, False Positive and False Negative are considered as below:

- TP - the number of "true positives", positive Examples that have been correctly identified
- FP - the number of "false positives", negative Examples that have been incorrectly identified
- FN - the number of "false negatives", positive Examples that have been incorrectly identified
- TN - the number of "true negatives", negative Examples that have been correctly identified

This process has been repeated for threshold values between 0.1 and 0.9 with a step of 0.05 to find the maximum accuracy and hence identify the optimum threshold.

5.4 Test Results

The table shown below explains the statistics of events presented in the seven experiments including the number of training and testing samples as well as the total number of samples for each hidden event.

Table 1 : The hidden numbers of events

Hidden event	Training samples	Testing samples	Total samples
Event 0	59309	62726	122035
Event 1	60414	61621	
Event 2	61280	60755	
Event 3	57095	64940	
Event 4	58571	63464	
Event 5	47951	74084	
Event 6	32144	89891	

In this part of the experiment, an evaluation of the proposed HTM Cortical Learning Algorithm has been tested using the same dataset, Virat, to do a comparison of the performance metrics between each output of different machine learning technique.

Several anomaly detection algorithms are evaluated using RapidMiner Studio version 8.2. Each model’s anomaly score is normalised to the range 0.0 to 1.0. The higher the value is, the higher the likelihood of an anomaly occurring.

5.4.1 k-Nears Neighbour Global Anomaly Score (kNN-GAS):

k-NN Global Anomaly Score algorithm (GAS) calculates anomaly scores using the nearest k neighbours. Ramaswamy et al. (2000) proposed that the outlier score as the average distance nearest kth neighbour. The core algorithm spends 99% of the execution computing all K Neighbours while the rest of the time is used for storage. The algorithm results are shown in Figure 2.

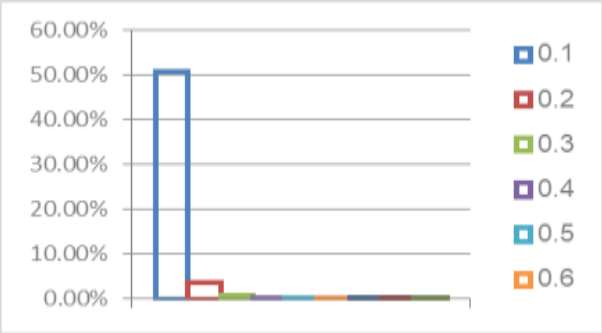

Figure 2: Average F-measure for the k-NN algorithm

5.4.2 *Connectivity-Based Outlier Factor*

Connectivity-Based Outlier Factor (CBOF) algorithm determines performance of data using queries, but its effectiveness is affected by sensor-generated time sequences. CBOF is used to determine bounds for the k-Nearest Neighbour KL-CBOF algorithm. This algorithm determines the lowest and highest bound of the multivariate data. Amongst the different models available for testing bounds for the k-Nearest Neighbour, this analysis uses only CBOF models to focus on the changes in bounds. This algorithm involves the rearrangements of the order of the N time domain samples through the counting in binaries that have been flipped from left to right. After the bit reversal sorting stage of the "CBOF" algorithm, the next step is the finding of the frequency spectra which belongs to the 1-point time domain signals at the end of the last decomposition phase. This is a very easy process since the frequency spectrum of a 1 point signals is equal to itself, and therefore there is virtually nothing to be done at this stage. Also, it should be noted that the final 1-point signals are no longer time domain signals but rather, a frequency spectrum.

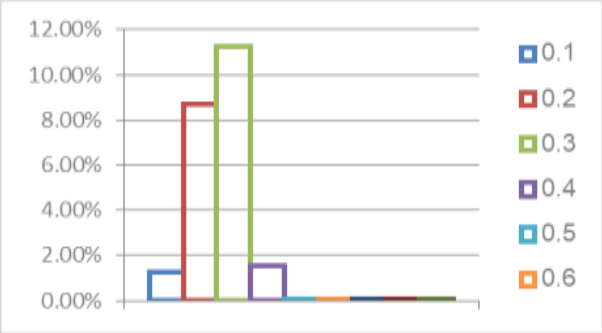

Figure 3 ∶Average F-measure for the CBOF algorithm

5.4.3 *Singular Value Decomposition Influence Outlier (SVD-IO)*

According to [31], COF is a modification of LOF algorithm is used to handle outliers that are not of low-density patterns. The outliers have an outlier score of more than 1. There is no difference between COF and LOF algorithms except density calculation. Whereas LOF, the distances are determined using a hypersphere centred on a data point. The COF calculates the distance incrementally [32].

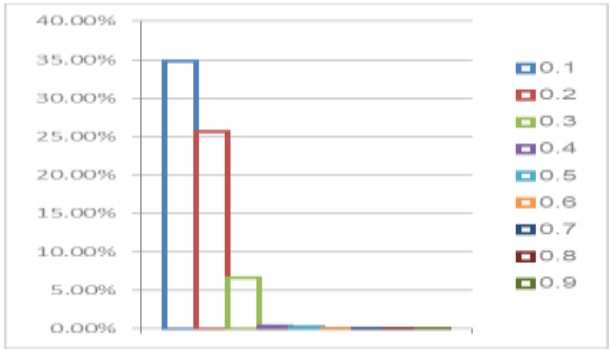

Figure 4 ∶F-measure for the SVD-IO algorithm

3.2.4 *Independent Component Analysis - Local Outlier Probability (ICA-LoOP)*

Independent Component Analysis (ICA) is an algorithm that is used to compute hidden factors within sets of statistical data. The algorithm generates a model that will be too hidden factors of multivariate data that is data from the big database. The model generated by the algorithm is assumed to linear and has unknown latent variables. This is usually applicable to Audio Processing, Array processing and medical data analysis. Most of this data is non-Gaussian and mutually independent, thus fit for ICA LOP. The following shows Independent Component Analysis of a signal.

In this section, the objective evaluation is carried out to compare and rank the proposed algorithm with the state-of-the-art anomaly detection algorithms. These results complement the illustrative visual comparison results in the previous Section 3.1 and 3.2. The evaluation of the accuracy of anomaly detection is carried out using Precision, Recall and the overall metric F-measure.

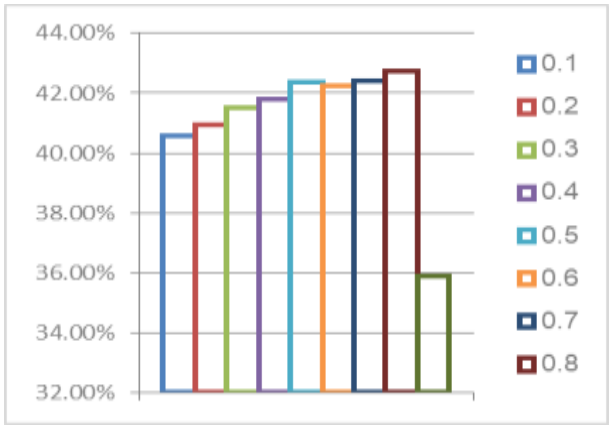

Figure 5 ∶ Average F-measure for the ICA-LoOP algorithm

5.4.4 *Proposed Algorithm*

The proposed HTM algorithm has been run with the datasets to evaluate its performance. The results are shown below:

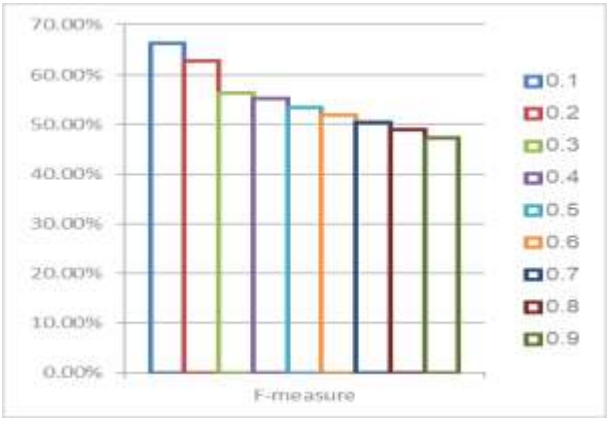

Figure 6 : Average F-measure for the proposed algorithm

5.5 Objective Evaluation

Tests applied on twenty-three videos have been conducted to detect movement anomalies in different scenarios. Additionally, in this study, the proposed algorithm has been evaluated and compared against several state-of-the-art anomaly detection algorithms.

The proposed algorithm has achieved 66.29% average F-measure, with an improvement of 15.5% compared to the k-Nearest Neighbour Global Anomaly Score (kNN-GAS) algorithm. The Independent Component Analysis-Local Outlier Probability (ICA-LoOP) scored 42.75%; the Singular Value Decomposition Influence Outlier (SVD-IO) achieved 34.82%, whilst the Connectivity Based Factor algorithm (CBOF) scored 8.72%. The proposed models, which are based on HTM, have empirically portrayed positive potential and had exceeded in performance when compared to several other algorithms.

Table 2 : Performance metrics for all algorithms

	Precision	Recall	F-measure
Proposed Algorithm	61.54%	71.84%	66.29%
KNN-GAS	72.2%	39.17%	50.79%
ICA-LOP	42.58%	42.93%	42.75%
SVD-IO	54.42%	25.60%	34.82%
CBOF	39.07%	4.91%	8.72%

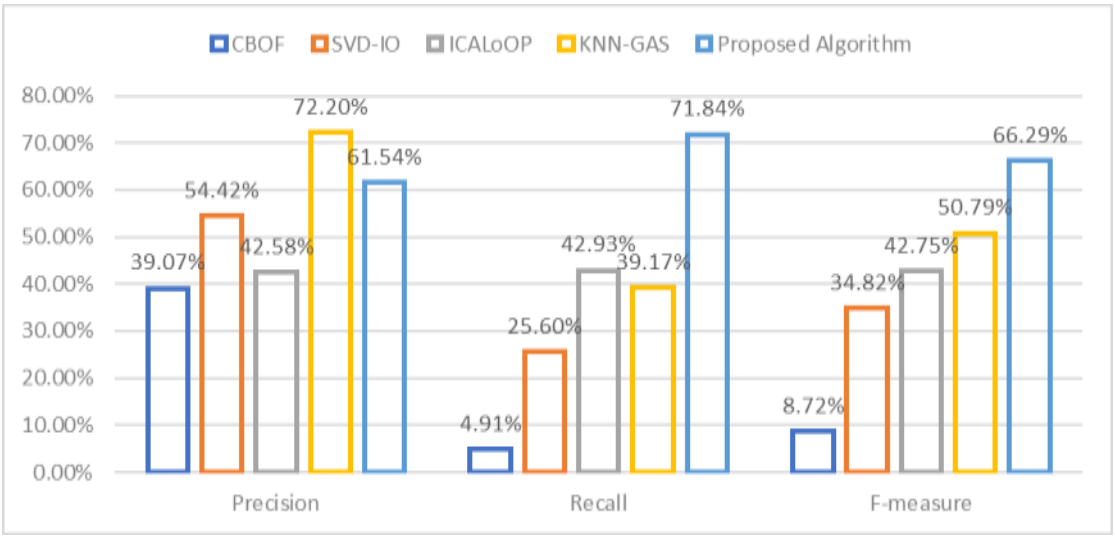

Figure 7 : Comparisons performance

6. CONCLUSION

Video analytic technologies have gained much attention especially in the context of the security of the community. The ultimate purpose of the intelligent visual surveillance is to handle different behaviours. Currently, the discovery of what is happening in a scene can be seen by automatic scrutiny of activities included in a video. Different algorithms that have been proposed to identify a solution to the movement classification problems. However, the required performance of such algorithms differs depending on the target scenario, and on the characteristics of the monitored scene.

Due to the diversity of video surveillance scenarios and the increasing development of movement classification algorithms, an automatic assessment procedure is desired to compare the results provided by different algorithms. This objective evaluation compares the output of the algorithm with the ground truth, obtained manually, and measures the differences using objective metrics. There are various datasets for activity and human action recognition, though older datasets provide limited ground truth classification to manual annotation at a simpler level, most of the modern datasets, in this case, VIRAT Video Dataset, gives high-quality ground truth. In this paper, the proposed movement classification algorithm has been tested and its accuracy evaluated. Several experiments have been carried out to calculate the optimum anomaly threshold for each algorithm. the average achieved average F-measure for the proposed algorithm was 66.29%, with an improvement of 15.5% compared to the k-Nearest Neighbour Global Anomaly Score (kNN-GAS) algorithm.

7. REFERENCES

[1] Adams, A.A. and Ferryman, J.M. 2015. The future of video analytics for surveillance and its ethical implications. Security Journal, 28(3), pp.272-289.

[2] Popoola, O.P. and Wang, K., 2012. Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), pp.865-878.

[3] Lavee, G., Khan, L. and Thuraisingham, B., 2007. A framework for a video analysis tool for suspicious event detection. *Multimedia Tools and Applications*, *35*(1), pp.109-123.

[4] Hara, K., Omori, T. and Ueno, R., 2002, September. Detection of unusual human behavior in intelligent house. In *NNSP* (pp. 697-706).

[5] Lee, C.K., Ho, M.F., Wen, W.S. and Huang, C.L., 2006, December. Abnormal event detection in video using n-cut clustering. In *International Conference on Intelligent Information Hiding and Multimedia* (pp. 407-410).

[6] Pan, F. and Wang, W., 2006, January. Anomaly detection based-on the regularity of normal behaviors. In *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics* (pp. 6-pp). IEEE.

[7] Zhang, Y. and Liu, Z.J., 2007, November. Irregular behavior recognition based on treading track. In *2007 International Conference on Wavelet Analysis and Pattern Recognition* (Vol. 3, pp. 1322-1326). IEEE.

[8] Alshaikh, A., & Sedky, M., 2016. Movement Classification Technique for Video Forensic Investigation. International Journal of Computer Applications, 135(12), pp. 1-7.

[9] Akintola, K. 2015. Real-time Object Detection and Tracking for Video Surveillance. VFAST Transactions on Software Engineering, 4(2), pp.9-20.

[10] Verma, B., Zhang, L. and Stockwell, D., 2017. *Roadside Video Data Analysis: Deep Learning* (Vol. 711). Springer.

[11] Hawkins, J., Ahmad, S. and Dubinsky, D. 2011. Hierarchical Temporal Memory Including HTM Cortical Learning Algorithms, 0.2. Technical report.

[12] Li, C.-T, and IGI Global, 2013. Emerging digital forensics applications for crime detection, prevention, and security. Hershey, PA: IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA.

[13] Chris, D. and David, D., 2012. A New Approach of Digital Forensic Model for Digital Forensic Investigation. *International Journal of Advanced Computer Science and Application*s, 2(12). doi:10.14569/ijacsa.2011.021226.

[14] Branch, H.O.S.D., 2006, June. Imagery Library for Intelligent Detection Systems (i-LIDS). In Crime and Security, 2006. *The Institution of Engineering and Technology Conference on* (pp. 445-448). IET.

[15] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.C., Lee, J.T., Mukherjee, S., Aggarwal, J.K., Lee, H., Davis, L. and Swears, E., 2011. A large-scale benchmark dataset for event recognition in surveillance video. *In Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on (pp. 3153-3160).

[16] Varadarajan, J. and Odobez, J.M., 2009. Topic models for scene analysis and abnormality detection. In Computer Vision Workshops (ICCV Workshops), *2009 IEEE 12th International Conference on* (pp. 1338-1345).

[17] Aggarwal, J.K., (2011). Motion analysis: Past, present and future. In Distributed Video Sensor Networks, pp. 27-39, Springer.

[18] Poppe, R., 2010. A survey on vision-based human action recognition. *Image and vision computing*, 28(6), pp.976-990.

[19] Amershi, S., Cakmak, M., Knox, W.B. and Kulesza, T., 2014. Power to the people: The role of humans in interactive machine learning. AI Magazine, 35(4), pp.105-120.

[20] Weiming, H. Tan, T., Wang, L. and Maybank, S. 2004 'A Survey on Visual Surveillance of Object Motion and Behaviors' Ieee Transactions On Systems, Man, And Cybernetics 34 (3).

[21] Iwashita, Y., Ryoo, M.S., Fuchs, T.J. and Padgett, C., (2013). Recognizing Humans in Motion: Trajectory-based Aerial Video Analysis. In BMVC 1 (3) p. 6.

[22] Agrawal, D. and Meena, N., 2013. Performance comparison of moving object detection techniques in video surveillance system. The International Journal of Engineering and Science (IJES), 2(01), pp.240-242.

[23] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

[24] Ding, S., Li, H., Su, C., Yu, J. and Jin, F., 2013. Evolutionary artificial neural networks: a review. Artificial Intelligence Review, 39(3), pp.251-260.

[25] Hawkins, J. and Blakeslee, S., 2004. On intelligence (Adapted ed.).

[26] Anjum, A., Abdullah, T., Tariq, M., Baltaci, Y. and Antonopoulos, N. 2016. Video stream analysis in clouds: An object detection and classification framework for high performance video analytics. IEEE Transactions on Cloud Computing.

[27] Breunig, M., Kriegel, H., Raymond T., and Sander, J. 2000. LOF: identifying density-based local outliers. In SIGMOD Record volume 29, pages 93–104. ACM, 2000.

[28] Byrne, F. 2015. Encoding reality: Prediction-assisted cortical learning algorithm in hierarchical temporal memory. arXiv preprint arXiv:1509.08255.

[29] Comon, P. 1994. Independent component analysis, A new concept? Signal Processing 36 (1994) 287-314

[30] Fan, D. Sharad, M., Sengupta, A. and Roy, K. 2016. Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing. IEEE transactions on neural networks and learning systems, 27(9), pp.1907-1919.

[31] Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. 2000. Lof: Identifying density based local outliers. SIGMOD Rec., 29(2), 93–104.

[32] Tang, J., Chen, Z., Fu, A. W.-C., & Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 535–548).: Springer.

International Journal of Computer Applications Technology and Research
Volume 8–Issue 03, 66-81, 2019, ISSN:-2319–8656

A K-Means Based Multi-level Text Clustering Algorithm for Retrieval of Research Information

Damaris Ndinda Waema
Department of Computing
Jomo Kenyatta University of Agriculture and Technology
Juja, Kenya

Petronilla Muriithi
Department of Computing
Jomo Kenyatta University of Agriculture and Technology
Juja, Kenya

George Okeyo
School of Computer Science and Informatics
De Montfort University
Leicester, United Kingdom

Abstract: Academic researchers in institutions of higher learning and research institutes use research outputs and metadata throughout their research work and to help in identifying research collaborators as well as getting to know existing research. Research outputs range from academic theses, journal and conference articles, books and book chapters, and datasets while research meta-data includes authors, affiliations, research areas, and projects, among others. However, access and retrieval of relevant research outputs and meta-data remains a major challenge. As a result there is duplication of research, fewer opportunities for networking, and difficulty in detecting scientific fraud. Efforts need to be made to make academic research outputs and meta-data readily available and easy to retrieve. The main purpose of this work is to develop a tailor-made approach to information retrieval for the retrieval of research information and related meta-data. Therefore, the paper presents a multi-level text clustering algorithm for retrieval of scholarly research outputs and metadata from a central repository through a web based interface. The algorithm first clusters SQL data records that represents meta-data at the first level, then retrieves and clusters text documents representing research outputs at the second level. The algorithm was tested on retrieving information in the areas of text clustering, cloud computing, banking, HIV/AIDS, food security and cancer. The results show that it enables researchers to retrieve relevant information according to their information needs. To enable further enhancements and improvements, the algorithm will be released to the public domain for use in similar application domains or extension by other researchers.

Keywords: Text Clustering, Multi-level, Research Metadata, Information Retrieval, SQL Data Clustering

1. INTRODUCTION

In order to fuel research activities, it is important to have research data available and easy to retrieve. In the context of this paper, research data refers to research outputs such as journal articles, research papers, and theses produced by postgraduate students. Research data also includes research metadata found in scholarly document repositories such as bio data about researchers or authors, research projects, as well as funding opportunities provided by various funding bodies to provide funds for researchers in their research endeavours. With the availability of reliable communication and networking platforms, access to this research data enables researchers to find other researchers from their research institutions or other institutions with similar research interests whom they can collaborate with. According to Muriithi [1], collaboration among researchers has various advantages such as availability of diverse range of skills, access to resources and special equipment not locally available, higher productivity, and increased visibility of the researchers' research outputs. Xia, Wang, Bekele and Liu [2] also note that it may be difficult to achieve scientific success without collaboration among scholars. Availability and easy retrieval of academic research data also helps to understand the knowledge creation process [3], reduce duplication of research in various research institutions, as well as aiding in curbing scientific fraud.

Availability and access to research data remains a challenge for Kenyan researchers. There does not exist a single national scholarly document repository that consolidates research data from scholars across Kenyan institutions of higher learning as well as research institutions and make it available for retrieval by interested researchers [4]. In a bid to solve this problem, some Kenyan universities have developed their own institutional repositories where they keep research outputs and research metadata from scholars in their universities and allow those researchers and their colleagues to have access to it when need arises. Such universities include Dedan Kimathi University of Technology [5]. However, despite these attempts by Kenyan institutions of higher learning to have their own institutional repositories, availability and retrieval of research outputs and research metadata is still a key challenge facing academic researchers in Kenya. This is because some of these institutional scholarly repositories have very low volumes of data available online [5]. Moreover, since these institutional repositories are usually the property of the particular universities, only researchers from those universities are sometimes given credentials to be able to access the repositories. This makes it hard for researchers in one institution to know what their fellow researchers in other institutions are working on. In addition, some of these repositories are also not updated as frequently as they should be [5].

In a research conducted by Erima, Masai & Wosyanju [6], with Moi University in Kenya as the case study, it was reported that unless there is a plan for continued access and use of academic research data, there is no guarantee that the research outputs generated today will be available, accessible and useful in the future. This alludes to the need of a national scholarly document repository as well as efficient information retrieval technology to enable retrieval of the consolidated data by interested researchers. The key question this paper addresses is: how can an information retrieval approach be

developed that ensures relevant research information and meta-data is obtained from research document repositories?
This paper answers the above question and addresses the challenges by making the following key contributions. Firstly, to provide the information retrieval technology required for the retrieval of academic research data from scholarly document repositories, this research, therefore, develops a multilevel text clustering algorithm capable of clustering research outputs (text documents) as well as research metadata in the form of SQL data records. Secondly, in addition to clustering data, this developed approach to text clustering also performs matching and ranking, which are important operations in information retrieval. Thirdly, this research also constructs an information retrieval model, which is composed of the developed algorithm, an identified scholarly document repository [7], as well as a web interface [7] used by researchers to access data from the repository. Finally, we evaluate the effectiveness of the developed text clustering algorithm in the retrieval of academic research data from scholarly document repositories and report very promising results.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the materials and methods. Section 4 presents the results while Section 5 discusses the results. Finally, Section 6 presents the conclusions and highlights future work.

2. RELATED WORK

2.1 Information retrieval

The subject of information retrieval, information retrieval systems and information retrieval models has been well covered in literature. Information retrieval is the process of extracting relevant information resources in a predefined and automated manner from an available lot of information resources [8]. In most occasions, this information is usually stored in an information retrieval system. These information retrieval systems make use of information retrieval technology to provide or even suggest documents or information that the application user would find relevant based on their information needs. The information needs of the user are often expressed or represented to the information retrieval systems by the use of a search string or search query. Those documents that the user finds suitable are called relevant documents [9].

An information retrieval system should be in a position to support the following functionalities [9]:

i. The storage and representation of the content or information in the information retrieval system
ii. The representation of a user's information need. In many applications this is usually done via a search query.
iii. The comparison of the two representations (stored information and user's search query). This is usually performed by information retrieval technology or algorithms that search the information retrieval system in order to find information that the user would find relevant.

Information retrieval systems make use of information retrieval models to find suitable information for a particular user. Some of the information retrieval models discussed in literature include the exact match models [9], the vector space model [9] and the latent semantic indexing model [8].

2.2 Text Clustering in Information Retrieval

Text clustering is the process of partitioning an unstructured set of objects into groups of similar objects [10]. The goal is usually to have documents in one cluster being as similar as possible, while still being as different from documents in other clusters as possible [10].

Text clustering algorithms are divided into two main groups: hierarchical algorithms (which produce a hierarchy of clusters) and partitioning algorithms (which give a flat partition of the data set) [11]. In addition to these two categories of text clustering algorithms, a distinction is also made between hard and fuzzy clustering. Hard clustering means that a document can only be assigned to one cluster, while on the other hand, fuzzy clustering means that a document can be assigned to more than one clusters [11].

In hierarchical clustering algorithms, clusters are constructed in two main ways: the bottom-up approach and the top-down approach [11]. The bottom up approach is used in agglomerative algorithms. Agglomerative algorithms are deterministic in nature, meaning that they will generate the same cluster hierarchy every time the algorithm is run. On the other hand, the top-down approach is used in divisive algorithms, where any partitioning algorithm are used to split clusters. An example of divisive algorithms is the Bisecting K-Means algorithm [12]. The stopping criteria for both the agglomerative and the divisive algorithms could be the achievement of the required number of clusters, some limit on a criterion function, or any internal evaluation measure.

The K-Means algorithm is probably the most known and most common text clustering algorithm in the category of partitioning text clustering algorithms [13] This algorithm has a time complexity of $O(knI)$, where k is the number of clusters, n is the number of objects to be clustered while I is the number of iterations that the algorithm runs [14]. Some of the advantages of the K-Means algorithm are that first, it produces tighter clusters than the hierarchical clustering algorithms, and more so if the clusters are globular and second, if variables are huge, the K-Means algorithm most of the times is computationally faster that hierarchical clustering, if K is kept small. One limitation of the K-Means algorithm is that without known information about the data to be clustered, it is difficult to predict a K value (number of clusters to form) that will lead to optimal clustering [11]. Another limitation is that different initial partitions may result in different final clusters.

2.3 Clustering of SQL data Records

The integration of data mining algorithms with relational database management systems is both important and challenging at the same time [15]. This has led to the introduction of the concept of clustering among database programmers to aid the process of data mining and analytics. Several research works have explored the use of text clustering algorithms to cluster SQL data records from relational databases. For example, Ordonez [15] explores how the K-Means algorithm can be integrated with a relational database programming application using SQL. In addition, another SQL data clustering algorithm that merges Markov Chain Monte Carlo methods with the EM algorithm is presented by Matusevich and Ordonez [16]. The K-Means algorithm generally performs well, is independent of the operating system used by the programmer and is linear to the

size of the used dataset [16]. Another algorithm for clustering SQL data is proposed by Sun et al., [17], and is based on depth neural networks and is used to cluster data in distributed databases.

2.4 Applications of Text Clustering in Information Retrieval

The role of text clustering in information retrieval cannot be underestimated, as it has a number of applications in this field. These applications differ in the set of texts they cluster (whether it is the search results, the entire collection of text, or a subset of that set of collections) and the aspects of the information retrieval system they try to improves(such as the effectiveness or efficiency of the search system, user interface or user experience) [18]. Some of the applications of text clustering in information retrieval include search result clustering, scatter-gather, collection clustering and cluster-based retrieval [18].

2.5 Information Retrieval of Academic Research Data

Information retrieval of academic research data refers to the process of using information retrieval technology to obtain research data from a consolidated store such as a central database or repository. Easy access and retrieval of academic research data has many benefits to researchers. The benefits include easier identification of scientific communities [2], improved efficiency of research and acceleration of innovation.

The increased volume of scholarly data being produced by academic researchers has led to the emergence of the term "Scholarly Big Data" ([2], [19]). Due to this continued growth in the number of publications being produced all over the world as well as other research related data, researchers all over the world get overwhelmed and spend a lot of time when trying to access and retrieve this research data ([2], [20]). In order to help researchers obtain relevant information in this time of information overload, information science specialists need to "develop reliable and effective automated systems that support an easy and effective access to the relevant information" [2]. This demand has necessitated the application of data mining and analysis techniques in the field of scholarly big data, leading to what is now termed as scholarly data mining. In the recent past, various scholars have investigated the application of data mining to solve the problem of access to scholarly data. Some of the main problems attracting attention include general scholarly information access and retrieval ([2], [20], [21]), author disambiguation ([2], [19]), academic recommendation (such as collaborator recommendation as well as literature recommendation) ([2], [19], [22]), expert searching [2], scientific impact evaluation (which could include article or paper impact evaluation, journal impact evaluation, as well as author impact evaluation) ([2], [19], [23]), scholarly data visualization [3] and identification of trending and emerging research topics that are receiving much attention from scholars [22]. Sumba et al., [21] also proposed an architecture using ontologies and data mining to detect similar research areas among researchers as well as potential collaboration networks.

Despite the benefits of having research data being readily available and accessible, availability and accessibility of this data remains a challenge to many scholars in Kenyan universities. ([4], [5], [24]). Needless to say, there is need for an information retrieval model that can cluster data, match the resultant clusters with the user's information needs to identify the relevant clusters, and finally rank the information based on criteria set by the application users (researchers). This will ensure effective information retrieval from scholarly document repositories and enable researchers to realize the aforementioned benefits.

2.6 Research Gaps

Existing work did not reveal an available approach for application when the data to be clustered exists both in the form of text documents as well as SQL data records in a relational database. In addition, existing algorithms do not include the matching and ranking operations, hence cannot be used directly when one needs to apply them for information retrieval when issuing search queries. Furthermore, it is clear that accessibility and retrieval of research data is a challenge to the process of carrying out research among Kenyan scholars.

This paper presents a multi-level text clustering algorithm capable of matching and ranking so as to retrieve the most relevant research information and meta-data.

3. MATERIALS AND METHODS

3.1. Research Methods

3.1.1 The Research Process

The research process involved comprehensive literature review, development of the multi-level text clustering algorithm, construction of information retrieval model, and evaluation of the effectiveness of the developed model and associated algorithm as depicted in Figure 1.

The multilevel text clustering algorithm clusters the research metadata (researchers, research projects, funding opportunities among others) in the form of SQL records at the first level, then retrieves and clusters research outputs (research papers, theses, journal articles among others) in the form text documents (PDFs).

The research data information retrieval model is composed of four elements: the developed multilevel text clustering algorithm, an academic research data repository that contains research outputs and metadata, a web interface used by researchers and other system users to access and retrieve information from the repository, and lastly an Application Programming Interface (API) that allows communication between the web interface and the developed algorithm.

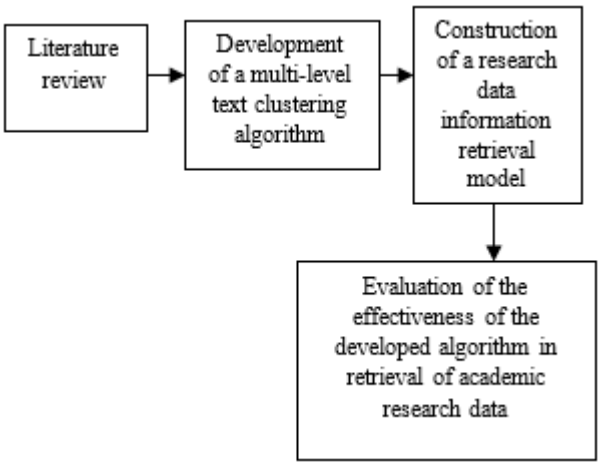

Figure 1. The research process.

The last step in our research process involves evaluating the suitability of the developed multilevel text clustering approach in the retrieval of academic research data from scholarly document repositories.

3.1.2 Steps Involved in the Development of the Multilevel Text Clustering Algorithm

i. Understanding text clustering

We first set out to study the field of text clustering so as to know the general working and steps of text clustering techniques.

ii. Identification of an existing text clustering technique to use in developing the multilevel approach

After an extensive literature review on text clustering algorithms, we chose to use the K-Means algorithm since it has been well explained in literature and generally performs well in clustering text documents. Since this (K-Means) algorithm was developed to cluster text documents, we tailored it so that it can cluster both SQL records (research metadata) and text documents (research outputs).

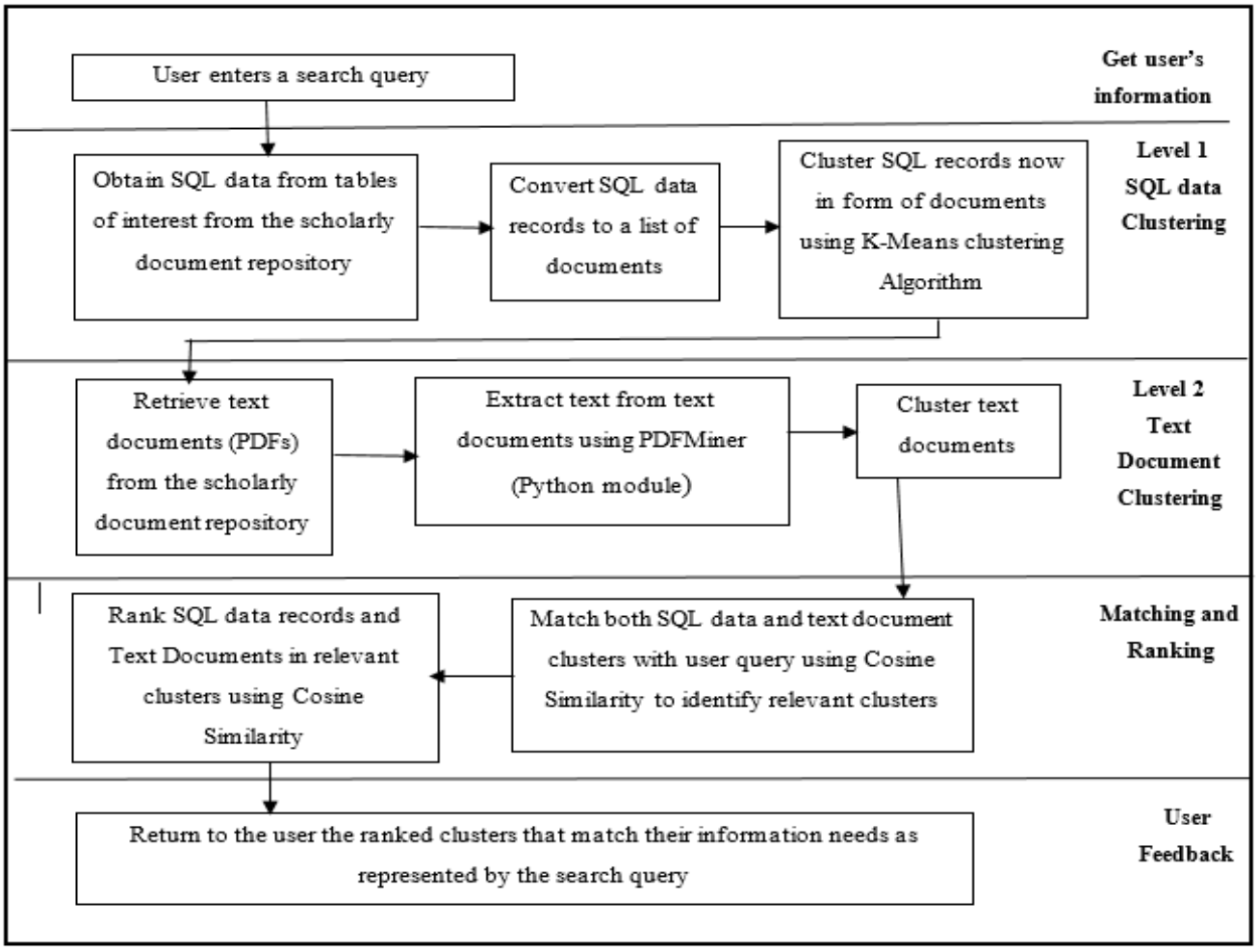

Figure 2. A Graphical Depiction of the Developed Approach to Multi-Level Text Clustering Algorithm

iii. Choosing a similarity measure

The algorithm has matching and ranking phases that require the use of a similarity measure. The matching phase identifies relevant clusters that are related to a search query from all the clusters in a given category of data. The ranking phase produces relevant information in a ranked order starting with the most relevant when compared to the search query. When various techniques used for similarity measure were reviewed, the Cosine Similarity was chosen to be used in developing the multilevel text clustering approach.

iv. Designing the algorithm

In designing the algorithm we first came up with the logical steps of the algorithm execution. We also came up with the four phases of the algorithm, as depicted by Figure 2.

v. Implementation of the multilevel text clustering approach

Once we had made all the decisions regarding the algorithm, we embarked on its implementation. It was developed in such a way that it could cluster both SQL data as well as text documents, as well as perform matching and ranking of information so that the application user would receive ranked information.

3.1.3 Evaluation of the Developed Multilevel Text Clustering Algorithm

Once the multilevel text clustering approach was developed, we evaluated its effectiveness in retrieval of research outputs and metadata from scholarly document repositories. The following are the steps that were followed in evaluating the algorithm.

i. Identification of a scholarly document repository

Because the developed text clustering approach is meant to cluster both SQL data records as well as text documents, we wanted to work with a repository that has this type of data. We settled on Kenya Research Information System [7] since it has the kind of data we needed.

ii. Choosing the metrics to use in evaluating the algorithm

After carrying out a study on metrics used in evaluating text clustering algorithms, we opted for the Silhouette Coefficient ([25], [12]) and the Davies-Bouldin Index ([25], [12]) metrics. This is because these two metrics are used to measure the internal quality of clusters when the ground truths about the data used for clustering are unknown, which was the case for our data. However, we also evaluated the algorithm on other metrics that are suitable for its application in our application domain, such as its ability to cluster both SQL data and text documents, as well its matching and ranking phases.

iii. Performing Experiments

We performed experiments in a bid to evaluate the effectiveness of the developed text clustering algorithm in the retrieval of research outputs and metadata from scholarly document repositories. We conducted lab experiments in which we used the application to retrieve academic research data like a normal application user would. We did this by supplying a user query as a representation of our information needs. The results of this evaluation are given in section 4 of this paper.

3.2 System Design

This section describes the architectural design of the research data information retrieval model, the components of the research data information retrieval model, and the design of the multilevel text clustering algorithm.

3.2.1 Architectural Design

The client server architecture was chosen for the research data information retrieval model. This architecture has three components:

i. The client

In the context of the developed research data information retrieval model, the client refers to a web interface that is used by application users such as researchers when accessing information from the scholarly document repository. The web interface is also used to display search results after information has been retrieved form the repository.

ii. The Server

The server stores the scholarly documents and research metadata in the form of SQL data records in a MySQL database. Research outputs in the form of text documents are stored in the KRIS application file system. The server receives incoming requests from the client, for example request for data in a given research area, then does processing and returns relevant data to the client via a communication network.

iii. Communication Network

The communication network acts as a link between the client and the server, allowing them to communicate. It allows requests for information from the client to reach the server, as well as retrieved information to be relayed from the server to the client. Figure 3 depicts the architecture.

3.2.2 Key Components

The information retrieval model developed in this research has four key components as explained below:

i. The multilevel text clustering algorithm

This algorithm aids in information retrieval by carrying out clustering of the available research data (research outputs like research papers as well as research metadata in the form of SQL data records). The algorithm clusters both the text documents and the SQL data into clusters of related data. The algorithm also performs matching and ranking so as to return to the user only relevant information based on their information needs as represented by their search query.

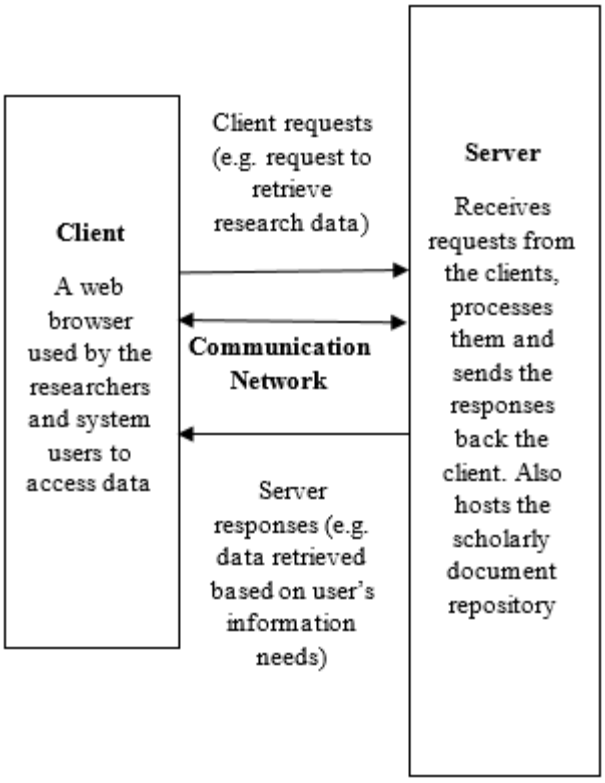

Figure 3. The client server architecture of the developed information retrieval model

ii. A Web interface

This web interface was developed by Muriithi et al., [7] and is used by researchers and other system users to

access the data in the scholarly document repository. It has a search text box where the application user provides a search string to represent the kind of information they are interested in retrieving. Search results after information retrieval are also displayed to the user on this web interface.

iii. The scholarly document repository

Research data that is created and used by academic researchers and scholars is consolidated and stored in the repository. The data includes research outputs such as theses, research papers, and journal articles, as well as research metadata such as researchers, research projects, and research project funding opportunities, among others [7].

iv. A Python Application Programming Interface

The multilevel clustering approach is developed using Django, which is a Python framework. On the other hand, both the web interface and the scholarly document repository are developed using Laravel, a PHP framework. Due to having different applications in different languages and frameworks, it became necessary to create an API to allow the two applications to communicate.

Figure 4 shows how these four components are layered in the academic data information retrieval model. On the other hand, Figure 5 shows the relationship between the Laravel framework application, the Django framework application and the API.

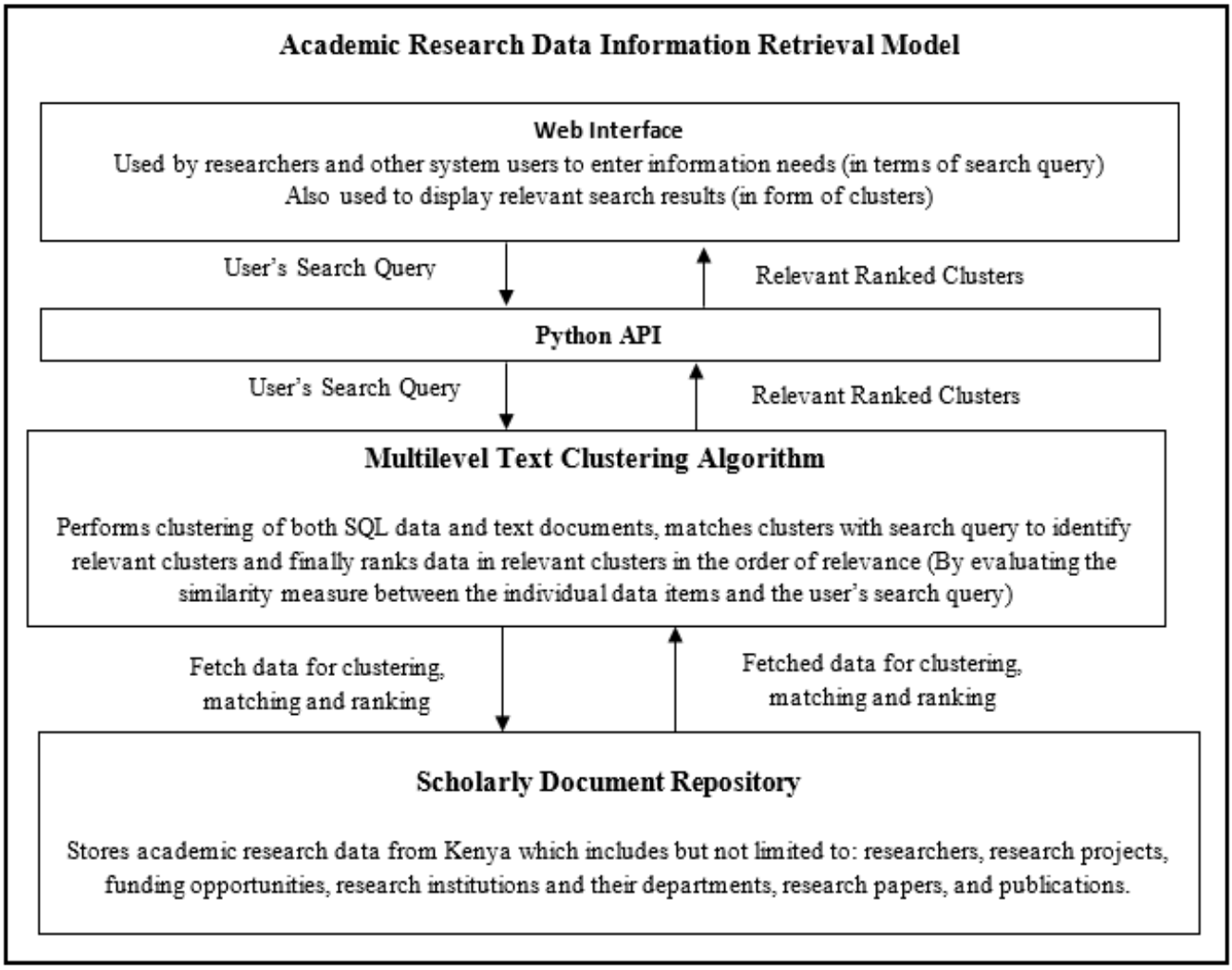

Figure 4. Components of the Constructed Academic Research Data Information Retrieval Model.

3.2.3 The design of the multilevel text clustering algorithm

The algorithm presented in this paper clusters both SQL data records in a database as well as text documents in the form of PDFs. The algorithm also performs matching and ranking of clusters in relation to their degree of similarity with the user query. In designing the algorithm, we came up with four logical steps in the execution of the algorithm. Figure 2 depicts these four logical steps described below:

i. Retrieving data from the scholarly document repository

This step gets data from the repository so that it can be clustered. This data includes research metadata such as researchers and research projects, in the form of SQL records in a MySQL database, as well as research outputs such as research papers in the form of PDFs.

ii. SQL Data clustering

This steps leads to the grouping of similar data in same clusters. The algorithm clusters research metadata in the form of SQL data records in the first level, and then clusters research outputs (text documents like research papers) in the second level.

iii. Matching

During this step, all the clusters in a given category of data are compared with the search query supplied by the application user in order to identify the relevant clusters.

iv. Ranking

This is the last phase of the algorithm. It takes as input the search query and the relevant cluster elements identified in the matching phase. It then calculates the similarity measure of each element in the relevant clusters when compared to the search query using Cosine Similarity. The output of this step is a ranked list of relevant data in each category of data that is the given back to the application user as search results.

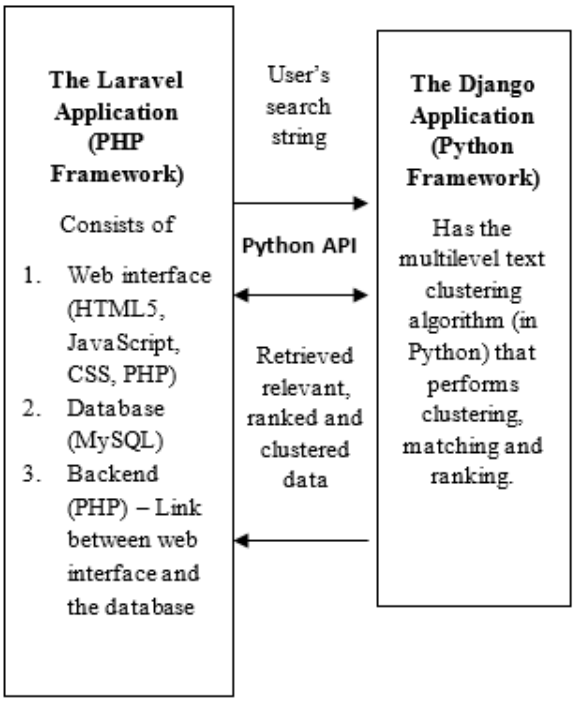

Figure 5. The relationship between the Python API, the Laravel application and the Django application.

3.3 Implementation of the Multi-level Text Clustering Algorithm

To implement the multi-level text clustering approach, we used Python, the Django framework, and the Pycharm Integrated Development Environment (IDE).

3.3.1 Clustering of Text Documents (Research Outputs)

i. Extraction of Text From PDF Documents

As noted earlier, the scholarly document repository contains, in PDF format, research outputs like postgraduate theses, research papers, etc. To cluster the documents, first, we used PDFMiner, a Python library, to extract the text from the documents. PDFMiner is able to extract the text such that all the text from one document forms one long string of characters. These strings of characters are then stored in a Python list, which then becomes the corpus that the algorithm clusters.

ii. Clustering

As earlier mentioned, K-Means algorithm was used in text clustering. We installed scikit-learn, an open source Python machine learning library that has a number of data mining and data analysis tools, in order to implement K-Means.

iii. Feature Extraction Using TF-IDF

Since the K-Means algorithm only works with numbers, we had to do feature extraction so as to obtain numbers from the document corpus. In our multilevel text clustering approach, feature extraction was performed using a scikit-learn tool called TfidfVectorizer. TfidfVectorizer uses an in-built Python dictionary to map the words in a document to feature indices and thereafter compute a word frequency matrix. The resultant word frequencies are then reweighted using the Inverse Document Frequency (IDF) vector collected feature-wise over the document corpus.

iv. Pseudo Code for the Clustering Step

```
Procedure: Text Clustering Algorithm
Input: A list of text documents
Output:  Text document clusters
Begin
        Convert text documents list to a vector space model
        Cluster text documents using K-means
        Return text document clusters
End
```

Figure 6. Algorithm for clustering text documents (research papers)

The algorithm shown in Figure 6 shows the steps involved in the clustering of text documents.

3.3.2 Clustering of SQL Data Records (Research Metadata)

The research metadata in the scholarly document repository is stored in a MySQL database in the form of SQL data records. To be able to cluster this data, first we fetch it from the database and convert records to a list of strings. Each record in the form of string of characters becomes a document in the list. This document list is then clustered using K-Means algorithm, and the resultant clusters assigned to a global variable so that they can be available for both the matching and ranking phases of the algorithm. Figure 7 shows the algorithm for the important steps of clustering SQL data. In this example we are clustering data in the 'researchers' table.

```
Procedure: Cluster researchers
Input: None
Output: Clusters of researchers
Begin
        Retrieve SQL data
        Convert SQL data records to document list
        Cluster SQL data records using K-means
        Return clusters of researchers
End
```

Figure 7. Algorithm for clustering of SQL data

3.3.3 The Matching Phase of the Algorithm

The importance of the matching phase of the developed multilevel text clustering algorithm is to identify relevant clusters from all the cluster formed. All the clusters formed are compared to the search query using Cosine Similarity, and those found to be similar are identified as candidates whose elements will be used in the ranking phase.

Before matching, the user's search string has its stop words removed. We wrote a Python function for that purpose. The function receives the search string as entered by the application user as an argument, and returns it without stop words. To achieve this, we calculated the set difference between the Scikit-learn's frozen set of English stop words and the set of words in the user's search query.

The algorithm in Figure 8 shows the step by step process of the matching phase, clearly indicating the inputs and outputs of that process. In this example we were matching researchers' clusters with the user's search query in order to get the clusters containing relevant researchers.

```
Procedure: Match researchers clusters
Input:User's search query; Researchers clusters
Output: Relevant researchers clusters
Begin
        Retrieve the search string minus stop words
        Match researchers clusters with user's search query
          (without stop words) using Cosine Similarity
        Return relevant researchers clusters (those with
          similarity measure greater than zero (0))
End
```

Figure 8. Algorithm for matching clusters to user query

3.3.4 The Ranking Phase of the Algorithm

In this phase, the documents or SQL data records in the relevant clusters identified in the matching phase are arranged in the order of their similarity when compared to the user's information needs as expressed by their search query, starting with the most similar. Just like in the matching phase, we also use the Cosine Similarity to measure the similarity between the search query (without its stop words) and the elements in the relevant clusters. In addition to creating a ranked list of relevant documents and data records, the ranking phase helps to remove documents or data records that mistakenly end up in the identified relevant clusters (outliers). This is because documents with a similarity measure of zero (0) (meaning that they are completely dissimilar to the search query) when compared to the search query are not included in the list of relevant and ranked documents and data records to be returned to the application user. The pseudo code for this ranking phase is shown in Figure 9.

```
Procedure: Rank relevant researchers
Input: Relevant researchers clusters; User's search string
Output: Ranked relevant researchers
Begin
        Retrieve the search string minus stop words
        Rank relevant researchers according to their
          degree of relevance based on the user's search string
            using Cosine Similarity
        Return ranked relevant researchers
End
```

Figure 9. Algorithm for ranking cluster results

4. RESULTS

The aim of this section is to evaluate the effectiveness of the developed multilevel text clustering algorithm in the retrieval of academic research data, both in the form of SQL data records and also text documents in PDF format. The data used for evaluation was consolidated research data from a central research data repository [7]. This data is composed of research papers (text documents) from academic researchers as well as research metadata stored in a MySQL database. The tables used for evaluation from this database keep data about researchers, research projects, and funding opportunities that researchers can apply for to facilitate their research. More details about this scholarly document repository can be found in [7].

4.1 Results of Evaluating the Algorithm's Ability to Cluster Both SQL Data Records and Text Documents

The aim of developing the algorithm was so that it can be able to cluster both SQL data and text documents from scholarly document repositories. We evaluated the algorithm against this criteria and it was indeed able to cluster both SQL data and research papers (text documents).

With regard to SQL data clustering, we clustered data in three tables: the researchers' table, the research projects table and the funding opportunities table. We chose to cluster data in both the researchers' and research projects tables into seven (7) clusters while data in the funding opportunities table was clustered into five clusters. The choice of number of resultant clusters was based on prior knowledge of the data that was being clustered.

In the graphs shown in Figure 10, Figure 11, Figure 12 and Figure 13, the X and Y coordinates represent the principal components for the two artificial dimensions created by the Principal Component Analysis (PCA) algorithm after dimensionality reduction.

The researchers' table stores bio data about researchers such as their research institution, department, titles (Prof., Dr., etc) and research interests. Figure 10 shows the 7 clusters resulting from clustering the data in this table. The cluster centroids for these clusters are indicated using digits (1-7). These digits also serve as the cluster labels.

The common characteristic of all the researchers in one cluster was the research area of interest. For example, cluster 7 was composed of researchers who had interests in banking, all of them belonging to either Jomo Kenyatta University of Agriculture and Technology or the University of Nairobi. In addition, cluster 4 had researchers belonging to various departments and research institutions, but having text clustering as their research interest. The cluster labeled "3" was made up of researchers specializing in HIV and AIDS, but from various research institutions and departments. In addition, the cluster labelled "6" had researchers with interests in accounting and finance, from different universities and departments. Cluster 2 was composed of clusters of researchers who were cancer specialists. The cluster labeled

"1" was made up of food security specialists from The University of Nairobi. Last but not least, the fifth cluster had researchers with research interests in cloud computing. All the researchers in that cluster (5) came from either Kenyatta University, the University of Nairobi or Jomo Kenyatta University of Agriculture and Technology.

The research projects table has data relating to research projects carried out by scholars such as the projects' title and project abstract. Figure 11 represents the seven clusters resulting from clustering data in this SQL table. Of these 7 clusters, five of them can be clearly seen. These five clusters represent research projects in the research areas of cancer (cluster 1), food security (cluster 2), HIV and AIDS (cluster 3), cloud computing (cluster 5) and finally text clustering (cluster 6). However, there are two clusters (clusters 4 and 7) that appear very close together. These clusters belong to projects in the banking (cluster 4) and accounting and finance (cluster 7) sectors, and they have very similar terms. That is why the cluster centroids for the data in these two clusters appear very close. The seven clusters from the research projects table shown in Figure 11 represents projects in seven different research areas of data used in this research.

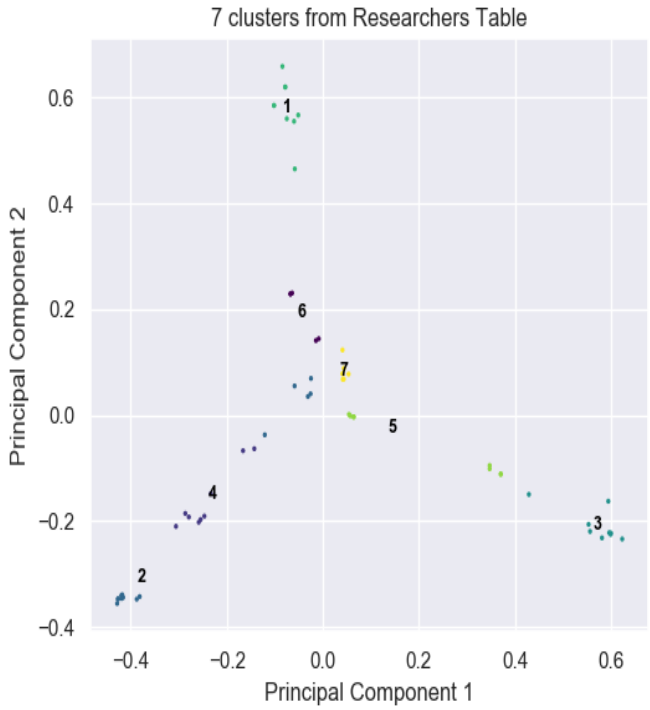

Figure 10. Results of clustering SQL data in the researchers table.

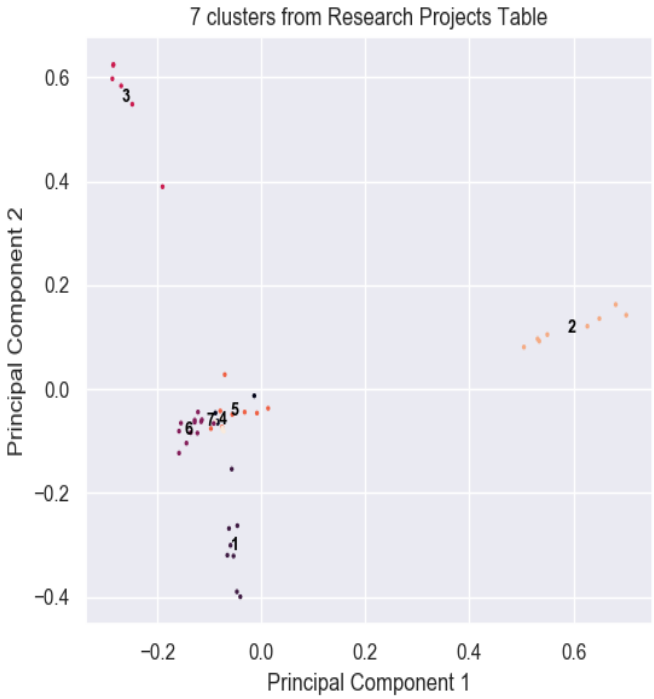

Figure 11. Clusters resulting from clustering data in the research projects table

In the funding opportunities table, the data stored include the funding body, the research areas being funded, and the deadline for applying for the funding opportunity. The five (5) clusters resulting from clustering these SQL data records are represented in Figure 12 with the cluster centroids shown by digits (1-5). Each of these five clusters contains all the research opportunities for projects in various research areas provided by one funding body. So the unique attribute in these five clusters is the name of the funding organization or body.

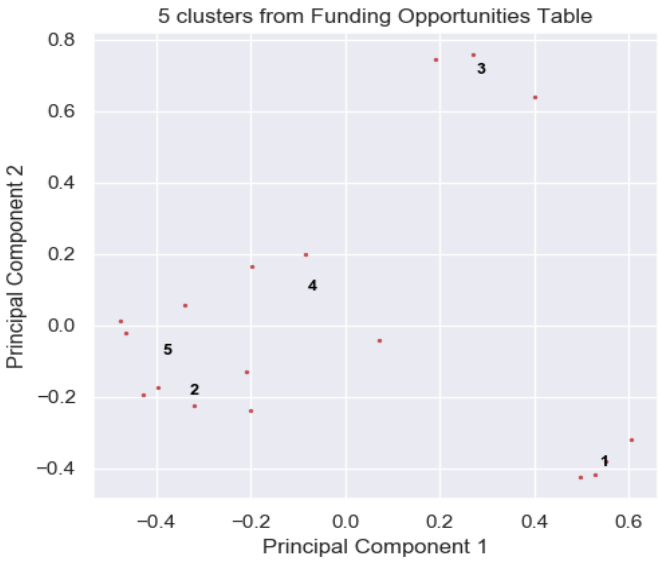

Figure 12. The 5 clusters formed after clustering data in the Funding opportunities table.

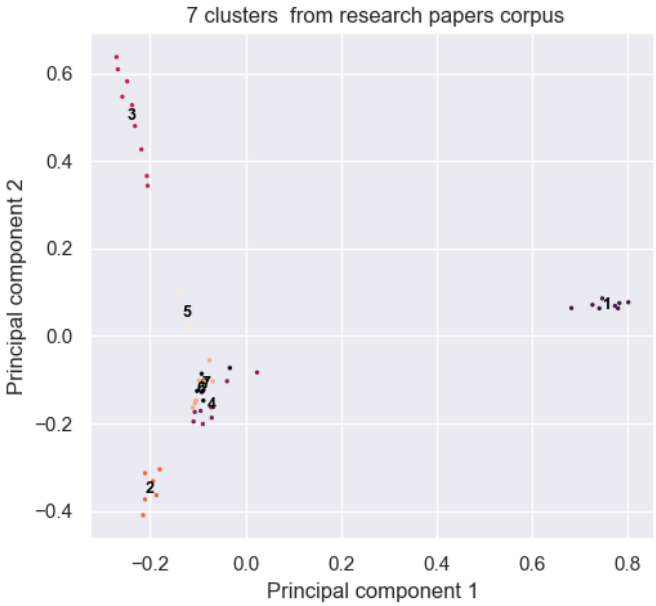

Figure 13. Research papers clustering results

To evaluate the ability of the developed multilevel text clustering approach to cluster text documents, we used research papers with content from seven research areas. These research areas include: text clustering, cancer, HIV and AIDS, banking, accounting and finance, food security and cloud computing. Because of this, we chose seven (7) to be the number of clusters to be generated by the algorithm. The algorithm generates seven clusters, grouping all the research papers in one research area in one cluster. Clusters 6 and 7 appear very close together because they represent research papers with content from two similar fields, hence making them to have similar terms. These two fields are banking and accounting and finance respectively. The 7 resulting clusters are depicted in Figure 13.

4.2 Results of Evaluating the Quality of Clustering Using the Silhouette Coefficient Metric

The Silhouette Coefficient is an internal quality measure of clustering results, and it is used when the ground truth labels of data being clustered are not known ([25], [12]) (which is the case for the data used to evaluate the developed algorithm). The Silhouette score indicates how similar a point is to the cluster it has been assigned. The Silhouette values are bounded between -1 and 1. The result is 1 for perfectly formed clusters and -1 for poorly formed clusters.

Table 1 shows the Silhouette Coefficient scores for the researchers' clusters shown in Figure 10, the research projects clusters represented in Figure 11, the funding opportunities clusters shown in Figure 12, and lastly the research papers clusters illustrated in Figure 13. According to these scores, which is something that can also be noticed from the graphical representation of the clusters, data in the researchers table forms the best quality of clusters.

Table 1. Silhouette Coefficient scores for the four categories of clusters formed

S.NO	Categories of Clusters	Silhouette Coefficient scores
1	Researchers	0.2137977853179148
2	Research projects	0.10267977158553505
3	Funding opportunities	0.1976613904708869
4	Text documents (research papers)	0.20310426032008072

4.3 Results of Evaluating the Quality of Clustering using the Davies-Bouldin Index

Table 2. The Davies-Bouldin Index scores for the researchers, research projects, funding opportunities and research papers clusters

S.NO	Categories of Clusters	Davies-Bouldin Index scores
1	Researchers	1.3715487874474703
2	Research projects	1.9340321303767844
3	Funding opportunities	1.445894695752895
4	Text documents (research papers)	1.4357250322273192

Just like the Silhouette Coefficient, the Davies-Bouldin Index is an internal quality measure, i.e., it uses the clusters themselves and not any other known external information such as labels ([25], [12]). The evaluation metric returns the ratio between the intra cluster distances and inter cluster distances. The lowest value is zero, and the lower the scores, the better the clustering. The Davies-Bouldin Index scores for the formed clusters are shown by Table 2. According to these results, clusters resulting from the researchers table resulted in the best formed clusters, since they have the least Davies-Bouldin Index score. This is in tandem with the results obtained from using the Silhouette Coefficient metric.

4.4 Results of Evaluating the Matching Phase of the Algorithm

Table 3. Relevant funding opportunities clusters identified at the matching phase of the algorithm

Cluster Number	Funder Name	Research Area being Funded
1	Africa ai Japan	Accounting, Finance
	Africa ai Japan	Text Clustering
	East Africa Research Fund	Text Clustering
	Africa ai Japan	Food Security
2	Kenya Research Fund	Accounting, Finance
	Kenya Research Fund	Text Clustering
	Kenya Research Fund	Cloud Computing
	Kenya Research Fund	Cancer
3	National Commission for Science, Technology and Innovation	HIV and AIDS
	National Commission for Science, Technology and Innovation	Accounting, Finance
	National Commission for Science, Technology and Innovation	Text Clustering

The importance of this evaluation is to find out if the algorithm is able to identify all the clusters that contain elements similar to the search string provided by the application user in KRIS. The elements in these relevant clusters are candidates for ranking in the ranking phase of the algorithm.

In this particular run of the algorithm, the search string (user query) provided was "text clustering". In the matching phase of the algorithm, from the five (5) funding opportunities clusters shown in Figure 12, the matching phase identified three (3) relevant clusters that contain "text clustering". These identified funding opportunities clusters are shown by Table 3.

4.5 Results of Evaluating the Ranking Phase of the Algorithm

The ranking phase of the algorithm uses Cosine Similarity measure to compare all the elements in the relevant clusters identified in the matching phase with the user's search string. These elements are then arranged in a list starting with the most similar. This ranked list of relevant data is then returned to the user via the KRIS web interface.

Table 4 shows the Cosine Similarity measures for all the relevant funding opportunities present in the relevant clusters shown in Table 3. On the other hand, Table 5 shows the ranked relevant funding opportunities based on the user's query ("text clustering").

Table 4. Cosine Similarity measures of funding opportunities in identified relevant clusters

Cluster Number	Funder Name	Research Area being Funded	Cosine Similarity Measure
1	Africa ai Japan	Accounting, Finance	0.0
	Africa ai Japan	Text Clustering	0.566345204707
	East Africa Research Fund	Text Clustering	0.437223120979
	Africa ai Japan	Food Security	0.0
2	Kenya Research Fund	Accounting, Finance	0.0
	Kenya Research Fund	Text Clustering	0.61920901471
	Kenya Research Fund	Cloud Computing	0.0
	Kenya Research Fund	Cancer	0.0
3	National Commission for Science, Technology and Innovation	HIV and AIDS	0.0
	National Commission for Science, Technology and Innovation	Accounting, Finance	0.0
	National Commission for Science, Technology and Innovation	Text Clustering	0.457674795248

Table 5. Relevant and ranked funding opportunities after searching for "text clustering"

Cluster Number	Funder Name	Research Area being Funded	Cosine Similarity Measure
2	Kenya Research Fund'	Text Clustering	0.61920901471
1	Africa ai Japan	Text Clustering	0.566345204707
3	National Commission for Science, Technology and Innovation	Text Clustering	0.457674795248
1	East Africa Research Fund	Text Clustering	0.437223120979

For data to be added to the list of what is to be returned to the user, the similarity measure has to be greater than zero. So from Table 4, only the four funding opportunities shown in Table 5 are returned to the user as relevant information. The data is ranked, starting with the most relevant.

4.6 Results of Using the Algorithm to Retrieve Data from KRIS via a web interface

Since the algorithm is invoked from KRIS through a Python API, this evaluation is to find out whether the algorithm indeed is able to return ranked and relevant data to the application user. Figures 14-16 show the results returned by the multilevel text clustering algorithm after searching for the word "cancer". Specifically, Figure 15 shows the relevant research projects, and Figure 16 contains a list of all relevant research papers. For the research papers, only the title of the paper is displayed. All these data have been ranked so that the most relevant data is on the top of the list. Figure 14 is basically the KRIS search results interface. It indicates when the search is complete, the number of relevant items retrieved, the user's search string, and the amount of time it took to retrieve the information from the database and file system (for research papers) and display it in the search results interface. The time is given in milliseconds.

In the event that what the KRIS application user has provided as a search query does not match any of the SQL data stored in the database or any of the research outputs stored in the applications file system, the application notifies the user that no relevant information was found for retrieval and requests them to redefine the search query and try again.

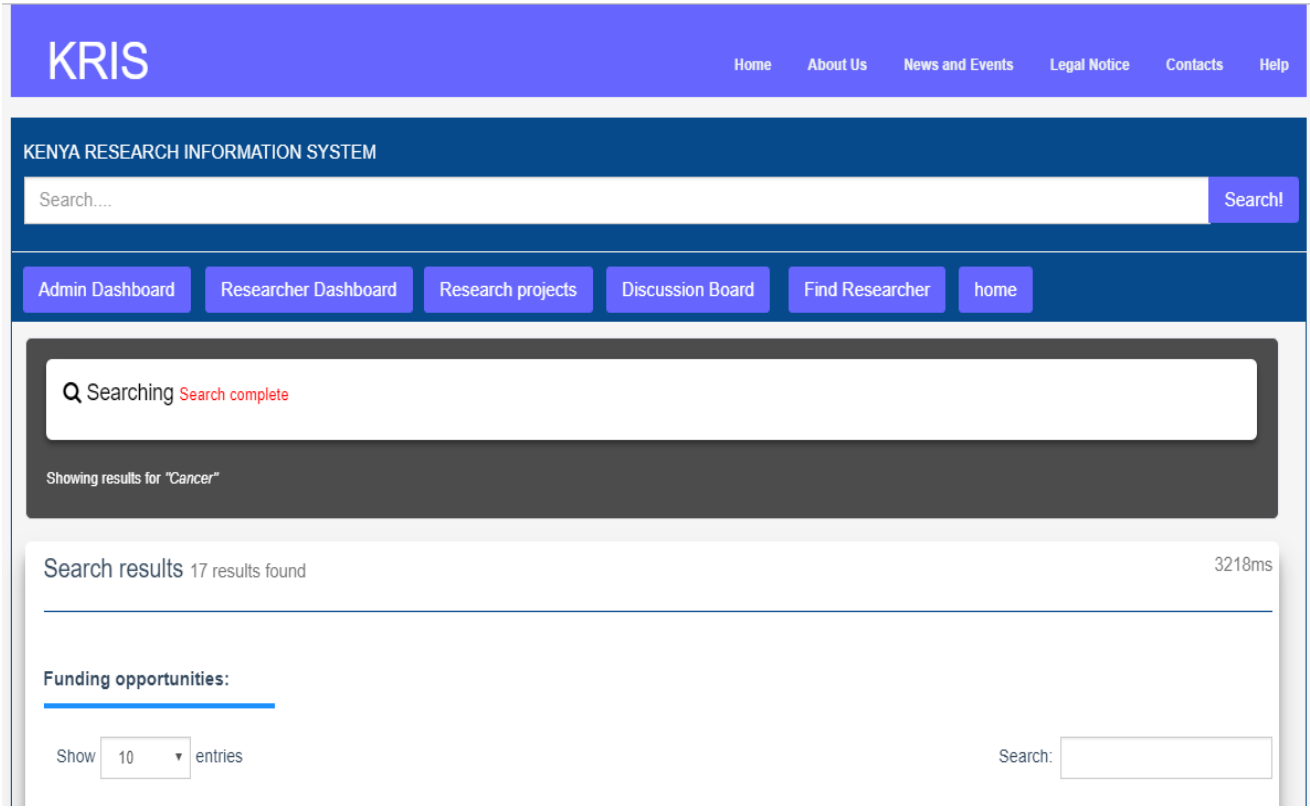

Figure 14. KRIS application search results interface

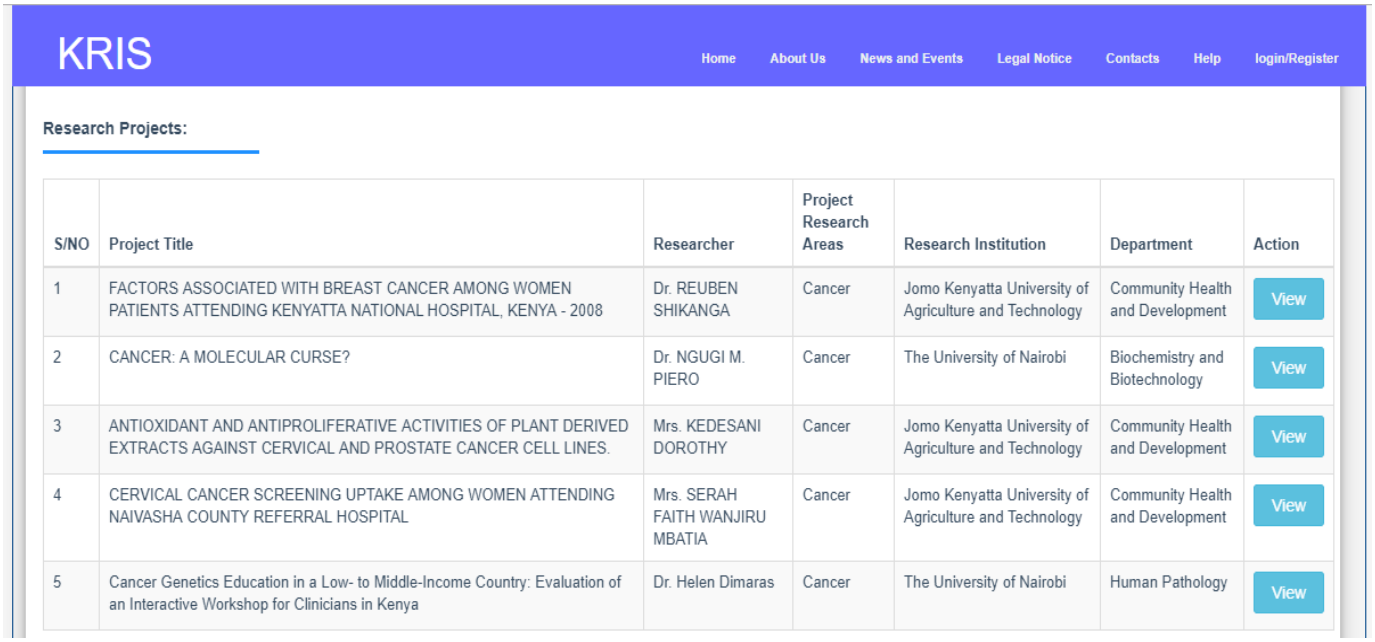

Figure 15. Relevant and ranked research projects information retrieved from the database after searching for “cancer”

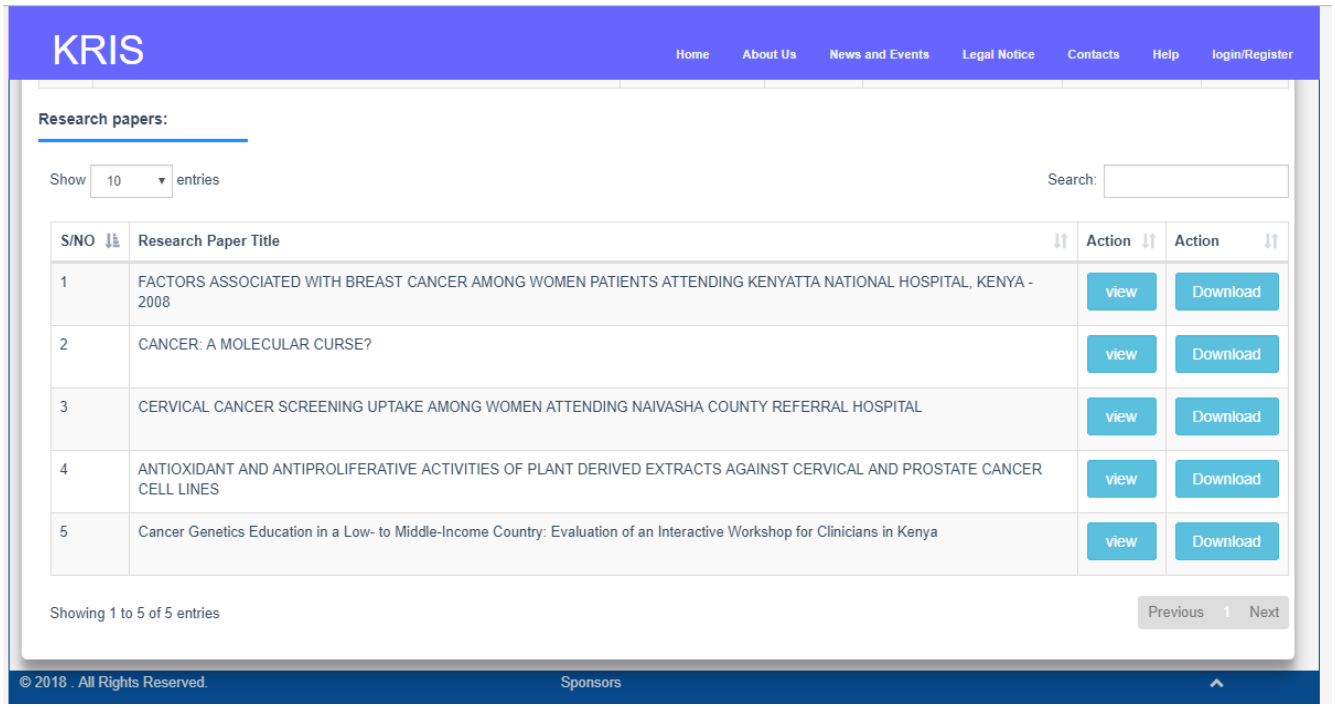

Figure 16. Relevant and ranked research papers retrieved from the file system of KRIS application after searching for “cancer”

5. DISCUSSION

From the evaluation results, it is clear that the developed multilevel text clustering algorithm meets its objective: to be able to cluster both SQL data and text documents, making it applicable in the retrieval of research outputs and metadata from scholarly document repositories.

Results from evaluating the algorithm on its ability to cluster both SQL data and text documents show that the algorithm is indeed able to fetch data from an SQL database and cluster it, as well as extract the content of text documents (research papers) in PDF format and cluster them. The resultant clusters have related data being grouped together, achieving the goal of clustering.

The Silhouette Coefficient scores are bounded between -1 and 1. A score of one means perfect clusters while a score of -1, which is the lowest possible, means that the clusters are very poorly formed. A score of zero (0) means that the cluster elements are at the border of other clusters. Generally, negative values indicate bad clustering while positive values indicate good clustering. On the positive side, the larger the value the better the clustering. According to Table 1, all the scores for the four categories of clusters are positive values,

indicating relatively good clustering. However, these values are still far from 1 (which is the perfect score). This can be attributed to the values of *K* (number of clusters to form per group of data) chosen when initializing the clustering algorithm. The values of *K* chosen to initialize the algorithm were just based on the researchers known information of the data to be clustered. For example, for the research papers, a value of seven (7) was used as the number of clusters to form (*K*) since in total there were research papers from seven research areas.

According to Table 1, the research projects clusters, with the least score (0.10267977158553505) were the worst formed compared to the other three categories of clusters, while the researchers clusters were the best formed with a Silhouette Coefficient of 0.2137977853179148. Possibly, choosing an optimal number of clusters would lead to better clustering and higher values for the Silhouette Coefficient metric.

The Davies-Bouldin Index scores shown in Table 2 do validate the evaluation results given by the Silhouette Coefficient metric in Table 1. In Table 2, the research projects clustering are the worst formed since they have the largest score, which is what the Silhouette Coefficients in Table 1 also imply. The researchers clusters are still the best formed according to the Davies-Bouldin Index scores, with a score of 1.3715487874474703. The choice of the number of clusters to form at the end of the clustering process as well as the choice of the maximum iterations when initializing the algorithm are the reason why the Davies-Bouldin Index scores are not zero (to imply perfect clustering).

Results from evaluating the algorithm on its ability to match resulting clusters with the user's search query in order to identify matching clusters show correct working of the matching phase of the algorithm. Table 3 shows the three clusters (out of the total five funding opportunities clusters) that were found to have information matching the user's search string (text clustering). In those clusters, the first cluster was found to be relevant because it contained two funding opportunities, one by the East Africa Research Fund and the other one by Africa ai Japan for text clustering research projects. Likewise, the second cluster has information about the Kenya Research Fund providing funding for text clustering projects. Lastly, the third cluster has the last record indicating that the National Commission for Science, Technology and Innovation is also providing a funding opportunity for research projects dealing with text clustering. In all the five funding opportunities clusters produced by the algorithm, only the clusters in Table 3 matched the user's information needs (text clustering).

Table 4 has all the funding opportunities in the three funding opportunities clusters in Table 3 identified in the matching phase to be relevant. These funding opportunities (in Table 4) also have a similarity measure score, indicating the extent to which they are similar to the user's search string (text clustering). In Cosine Similarity, the scores are bounded between zero (0) and one (1). A score of 0 means no similarity at all while a score of 1 means a perfect match. This implies that the larger the score, the more similar two strings are. Based on the similarity scores in Table 4, only four records have a degree of similarity with the search string, since the rest have a Cosine Similarity measure of 0.0, meaning no similarity at all. Table 5 then shows those four matching funding opportunities ranked in a single list, starting with the one that is most similar (and therefore most relevant when it comes to information retrieval). The first one provided by the Kenya Research Fund has a score of 0.61920901471, while the last one by East Africa Research Fund has the least score of 0.437223120979. Obviously, from Table 4 and Table 5, and the explanation given so far about the results of the ranking phase of the algorithm, the ranking phase of the developed algorithm is working as it should, ensuring that the most relevant information is kept on top of the information returned by the algorithm to the calling program (KRIS).

The evaluation of the algorithm on its ability to enable the user to retrieve data from a scholarly document repository tested all the steps of the processing of the algorithm illustrated in Figure 2. It tests whether the user's information needs in the form of a user query or search string can be send to the algorithm via the API. It also tests the ability of the algorithm to retrieve SQL data from the KRIS database and as well as research papers from the application's file system, including clustering, matching and ranking the relevant data according to its degree of similarity with the user's information needs. In addition, this evaluation also tests the ability of the algorithm to send back the relevant and ranked information retrieved from the database and the relevant text documents (research papers) to the KRIS search results interface. Figure 15 and Figure 16 do show that the algorithm does allow a researcher or any other application user to retrieve academic research data from KRIS via a web interface. For the data from the database (funding opportunities, researchers and research projects), the application user can click on the "view" button in the data tables to have access to more information about a given record that is not displayed in the data tables. In as far as the returned research papers are concerned, the researcher can opt to just view them to read the text or download the research papers to read them later by using the respective button.

6. CONCLUSION AND FUTURE WORK

This paper addresses the problem of access and retrieval of research information among researchers in Kenyan institutions of higher learning as well as research institutes. It develops a multilevel text clustering algorithm to be applied in the clustering and retrieval of academic research data from scholarly document repositories. The algorithm is able to fetch and cluster SQL data (research metadata such as researchers' biodata) at the first level, and extract text from text documents (research outputs such as research papers in the form of PDFs) and cluster them in the second level. In addition to text clustering, the developed approach to text clustering also performs matching and ranking, both of which are important operations in information retrieval. In addition, this paper also develops a research data information retrieval model by integrating the algorithm with a research data repository to facilitate retrieval of academic research data from the repository through a web interface. This information retrieval model is applicable not only for the retrieval of scholarly data, but also in any other situation where information to be retrieved consists of text documents and SQL data in a relational database.

Evaluation results show that the approach produces promising results. For instance when we evaluated the internal cluster quality of the produced clusters, we obtained results of the Silhouette Coefficient of value that are above zero (0), indicating relatively good clustering. In addition, evaltuating the ranking phase of the algorithm using the Cosine Similarity measure indicated that only information with a similarity measure greater than zero (0) when compared to the user query were included in the final list of ranked information to be returned to the user. Generaly, our evalution indicated that

the developed multilevel text clustering approach was effective in the retrieval of academic research data.

The future work of this research lies in the choice of the number of clusters that the developed multilevel approach will produce. In our experiments, the number of clusters was chosen based on prior knowledge of the consolidated data in the repository used. As the amount and diversity of the data in the repository continues to increase, choosing the number of cluster this way may prove inefficient because it becomes difficult to manually go through all the data so as to determine the number of clusters to produce. Research can therefore explore a technique to automatically choose the optimal number of clusters based on the data to be clustered and embed it onto the algorithm for efficient clustering.

7. ACKNOWLEDGEMENTS

I want to express my gratitude to my employer Jomo Kenyatta University of Agriculture and Technology for sponsoring my studies and supporting this research work.

8. REFERENCES

[1] Muriithi, M. P. 2013. Computer Mediated Collaboration among the Academic Research Community: A Case Study of Kenya: Doctoral consortium paper. *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, 28-31st May 2013, Paris**.** DOI: https://doi.org/10.1109/RCIS.2013.6577731.

[2] Xia, F., Wang, W., Bekele, T. M., and Liu, H. 2017. Big Scholarly Data: A Survey. *IEEE Transactions on Big Data*, 3(1), 18-35. **DOI:** https://doi.org/10.1109/TBDATA.2016.2641460.

[3] Liu, J., Tang, T., Wang, W., Xu, B., and Kong, X. 2018. A Survey of Scholarly Data Visualization. IEEE Access, 6, 19205 – 19221. DOI: https://doi.org/10.1109/ACCESS.2018.2815030.

[4] Muriithi P.M 2015. *Academic Research Collaborations in Kenya: Structure, Processes and Information Technologies* (Unpublished doctoral dissertation). University of Brighton, United Kingdom.

[5] Mugambi, W.C., and Ongus, W.R. 2016. Analysis of the Implementation of an Institutional Repository: A Case Study of Dedan Kimathi University of Technology, Kenya. In *International Journal of Information and Communication Studies*, 2(1), 22-30.

[6] Erima, J., Masai, W., and Wosyanju, M, G. 2016. Preservation of Digital Research Content in Academic Institutions: A Case Study of Moi University, Kenya. In *2016 IST-Africa Week Conference*, 11-13th May 2016, Durban, South Africa. DOI: https://doi.org/10.1109/ISTAFRICA.2016.7530620.

[7] Muriithi, P., Okeyo, G., and Waema, D. 2017. Towards improved availability and access to research data: A web based solution for Kenya. *12th JKUAT Scientific Conference*, Nairobi, 16–17th November 2017.

[8] Manju, K., Amita, J., Sonakshi, V., and Manoj, K. 2016. Analysis of Various Information Retrieval Models. In *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 16-18th March 2016, New Delhi, India.

[9] Goker, A., and Davies, J. 2009. *Information retrieval: Searching in the 21st Century*. Hoboken: John Wiley and Sons, Ltd.

[10] Desai, S. S., & Laxminarayana, J. A. 2016. A Review of Semantic Based Techniques for Document Clustering. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, 22(2), 214 – 217.

[11] Namratha, M., and Prajwala, T. R. 2012. A Comprehensive Overview of Clustering Algorithms in Pattern Recognition. *IOSR Journal of Computer Engineering (IOSRJCE), 4(6), 23-30.* DOI: 10.9790/0661-0462330.

[12] Banerjee, S., Choudhary, A., and Pal, S. 2015. Empirical Evaluation of K-Means, Bisecting KMeans, Fuzzy C-Means and Genetic K-Means Clustering Algorithms. In *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 19-20 December 2015, Dhaka, Bangladesh. 168-172. DOI https://doi.org/10.1109/WIECON-ECE.2015.7443889.

[13] Jain, A., Bajpai, A., and Kumar R. M. 2012. Efficient Clustering Technique for Information Retrieval in Data Mining. *International Journal of Emerging Technology and Advanced Engineering*, 2(6), 12-20

[14] Hand, D. J., H. Mannila, H., and Smyth, P. 2001. *Principles of data mining*. MITPress, Cambridge, MA, USA. ISBN 0-262-08290-X.

[15] Ordonez, C. 2006. Integrating K-means Clustering with a Relational DBMS using SQL. *IEEE Transactions on Knowledge and Data Engineering*, 18(2), 188 – 201. **DOI:** https://doi.org/10.1109/TKDE.2006.31**.**

[16] Matusevich, S.D., and Ordonez, C. 2014. A Clustering Algorithm Merging MCMC and EM Methods Using SQL Queries. *JMLR: Workshop and Conference Proceedings*, 36, 61-76.

[17] Sun, Q., Fu, L., Deng, B., Pei, X., and Sun, J. 2017. An Efficient Distributed Database Clustering Algorithm for Big Data Processing. In *2017 3rd International Conference on Computational Systems and Communications* CSP, 25 -26th March 2017, Beijing, China. DOI: 10.23977/iccsc.2017.1012.

[18] Manning, D.C., Raghavan, P., and Schütze, H. 2009. *An introduction to information retrieval*. Cambridge, UK: Cambridge University Press.

[19] Khan, S., Liu, X., Shakil, K.A., and Alam, M. 2017. A survey on scholarly data: From big data perspective. *Information Processing and Management*, 53(4), 923–944. DOI: http://dx.doi.org/10.1016/j.ipm.2017.03.006.

[20] Saggion, H., and Ronzano, F. 2017. Scholarly Data Mining: Making Sense of Scientific Literature. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 19-23 June 2017, Toronto, ON, Canada. DOI**:** https://doi.org/10.1109/JCDL.2017.7991622.

[21] Sumba, X., Sumba, F., Tello, A., Baculima, F., Espinoza, M., and Saquicela, V. 2016. Detecting Similar Areas of Knowledge Using Semantic and Data Mining Technologies. *Electronic Notes in Theoretical Computer Science*, 329, 149-167. DOI: http://dx.doi.org/10.1016/j.entcs.2016.12.009.

[22] Katsurai, M. 2017. Bursty research topic detection from scholarly data using dynamic Co-word networks: A preliminary investigation. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 10-12th March 2017, Beijing, China. **DOI:** https://doi.org/10.1109/ICBDA.2017.8078788**.**

[23] Cormode, G., Muthukrishnan, S., and Yan, J. 2014. People like us: Mining scholarly data for comparable researchers. In *Proceedings of the 23rd International Conference on World Wide Web*, 07 – 11th, April 2014, Seoul, Korea. DOI: https://doi.org/10.1145/2567948.2579038.

[24] Muia, M.A., and Oringo, J. 2016. Constraints on Research Productivity in Kenyan Universities: Case Study of University Of Nairobi, Kenya. *International*

Journal of Recent Advances in Multidisciplinary Research, 03 (8), 1785-1794.

[25] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. 2010. Understanding of Internal Clustering Validation Measures. In *IEEE International Conference on Data Mining*, 13-17 Dec. 2010, Sydney, NSW, Australia. DOI**:** https://doi.org/10.1109/ICDM.2010.35.