IJCSI International Journal of Computer Science Issues, Volume 14, Issue 1, January 2017
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201701.7986

Biterm for spam filtering in short message service text

Richard Omolo Midigo[1], Prof. Waweru Mwangi[2], Dr. George Onyango Okeyo[3]

[1]Computing Department, Jomo Kenyatta Univerity of Agriculture and Technology
P.O. Box 62000
Nairobi, 00200, Kenya

[2]Computing Department, Jomo Kenyatta Univerity of Agriculture and Technology
P.O. Box 62000
Nairobi,00200, Kenya

[3]Computing Department, Jomo Kenyatta Univerity of Agriculture and Technology
P.O. Box 62000
Nairobi, 00200, Kenya

Abstract

Due to rapid growth in mobile phones usage and reducing cost of sending text messages across mobile networks, short message service has become the most popular communication mode. This move has attracted spammers to mobile networks. Although several machine learning methods have been developed to filter out SMS spam from mobile phone users' inboxes, Short Messaging Service has issues that posse challenges to the use conventional document models that rely on proportion of word distribution. For instance, SMSs suffer from severe sparse context information, which hampers classification of content based on proportion of word distribution. This paper proposes an algorithm that uses biterm topic model (BTM) to model SMS text message. Biterm topic model directly models the generation of word co-occurrence patterns (i.e. biterms)in the whole document. Finally, support vector machine (SVM) was used classification. The algorithm has proved that it can effectively model SMSs for classification using SVM.

Keywords: Support Vector Machine, Biterm Topic Model, Short Message Service, Spam Filtering

1. Introduction

Spam has been recognized as a universal problem for both email and mobile phone users [1]. Generally, spam mail is an unsolicited bulk e-mail or junk mail or internet mail that is sent to a group of recipients who have not requested for it [2]. In Asia for instance, Short Messaging Service (SMS) spam constitutes 20 − 30% of all text messages transmitted via mobile phone networks. Spam has many disadvantages. To email users, spam messages are annoying, as they waste users time and clutter their mailboxes, cost money to users with dial-up connections, waste network bandwidth, as well exposing minors to unsuitable content (e.g. when advertising pornographic sites) [3]. Spam messages may fill user's mailbox engulfing important personal mails, wasting network bandwidth, consuming user's time and energy to sort through it [4]. Similarly, SMS spam messages are also nuisance to mobile subscribers as many mobile subscribers have suffered financial losses resulting from responding to SMS spam senders on premium rate numbers and signing up to expensive subscription services [5], [6].

Many factors have contributed to increase of spamming on phone networks. These factors include: 1). Availability of very cheap bulk pre-pay SMS packages. 2). Enhancement of SMS systems to concatenate the short message, 3). Point-to-broadcast capabilities. 4). Availability of Bulk SMS software for sending quick and short message from computer PC via GSM device or internet connection network to numerous people [7]–[9].

To reduce the effect of Spam on email and mobile subscribers, filtering techniques have been and continue to be developed to help to rule out these unsolicited e-mails and SMSs automatically from a user's mail stream. Machine learning-based text filtering techniques have been used since they are capable of eliminating human effort required to analyze data. Machine learning also improve accuracy in analyzing text. Machine learning algorithms such as K-Nearest Neighbor (k- NN), Decision Tree and Support Vector Machines (SVM), Neural Networks, Naïve Bayes classifiers, among others have been used in the of spam filtering systems [7]

Performance of any machine learning based filtering algorithm in making accurate predictions depends on the

quality and size of data available to the learner [10]. The problem therefore remains how to identify a method that would extract the required quality and quantity of data that adequately represent the documents to be categorized.

According to [11], SMS messages lack enough word counts that can convey how words are related when using conventional topic models and due to the severe sparse context information, revealing topics (proportion of word distribution) from short texts, it is challenging to model SMS messages using traditional frameworks that were initially designed to handle normal text documents.

In this paper proposes a hybrid SMS spam filtering model based on Biterm topic model, SVM and frequent itemset techniques. the input data for the model comprises SMS spam and ham datasets taken from the University of California Irvine machine learning repository Biterm topic model is used to generate biterms (unordered word pairs) from which frequent words are selected based predefined frequency threshold. In this research, feature selection is carried out first to select the relevant features that adequately represents the documents to be classified. After feature extraction, TF-IDF weighting is conducted for every feature before the weights with feature identifiers are passed over to the SVM classifier. Four confusion matrix performance evaluation metrics have been employed to gauge the performance of the algorithm in classifying the messages

This paper is organized as follows: section 1: introduction. Section 2: literature review. Section 3: methodology. 4: Discusion of experimental results of performance evaluation. Finally, section 5: Conclusion and future research.

2. Literature review

2.1 Machine-learning based text categorization

Text categorization (TC) structures a repository based on a scheme given as input. Meaning, in text categorization a set of classes to be used are predefined and known [12].Text categorization has been used in automatic indexing for Boolean information retrieval systems, document organization, text filtering, word sense disambiguation, hierarchical categorization of Web pages among others. Application of machine learning in text categorization dates back to Maron's seminal work on probabilistic text classification of 1961.

Machine learning employs classification of text to determine the class label of a given example, out of a finite set of classes, based on a description of the example [13] to fulfill the aim of text categorization which is approximating a category assignment function for given set of documents. Machine learning techniques help in reducing the manual effort required in analyzing data and also to improve accuracy in performance of the systems [7]. K-Nearest Neighbor (k- NN), Decision Tree, Support Vector Machines (SVM), Neural Networks, Naïve Bayes classifiers are some of machine learning techniques that have been used in categorizing text documents [7]

2.2 Short message service (SMS)

Short Message Service (SMS) commonly referred to as text messaging is a store and forward mechanism for delivering short text messages over the mobile networks [14] enabling subscriber to receive all message sent when their phone were turned off. SMS uses wireless network for transportation and delivery of the text messages [9] hence does not require that subscribers develop any infrastructure . SMS only allows for exchanging short text messages with a maximum of 160 simple characters [9]. Several enhancements have been introduced in SMS to make the service more effective [8], [9]These enhancements include: 1). Ability to concatenate the short message. 2). Point-to-broadcast capabilities. 3). Bulk SMS software for sending quick and short message from computer PC via GSM device or internet connection network to many people. SMS text messages fall into two categories namely spam and non spam messages. Spam messages are sent without legitimate or explicit opting by the receiver. Non spam messages (also known as ham) are the legitimate messages sent with explicit opting by the receiver.

2.3 Support Vector Machine

Support Vector Machines (SVMs) are a classification technique used in data mining and machine learning. SVM is particularly well suited for application with sparse data sets [15]. SVM is a new machine-learning approach based on statistical learning theory [16] which uses a blend of linear modeling and instance-based learning[17]. Support vector machine (SVM) is capable of taking care of the interdisciplinary nature of today's advanced analytics by drawing equally from three major areas namely computer science, statistics, and mathematical optimization theory [18]. Its classification involves looking for the optimal separating hyperplane between two classes by maximizing the margin between the classes' closest points. SVMs revolve around the notion of a "margin" on either side of a hyperplane that separates two data classes [19] and aims at maximizing the margin thereby creating the largest possible distance between the separating hyperplane and the

instances on either side of it. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples based on the experience gained during the learning process[20]. The method has been proven to reduce an upper bound on the expected generalization error. In 1995, Cortes and Vapnik provided first formal introductions to the concept of SVM while investigating algorithms for optical character recognition at the AT&T Bell Labs[19].

[21] Tested a number of email spam filtering algorithms using two SMS datasets. They wanted to a find out if the filtering algorithms previously used for email spam filtering could also be work SMS spam filtering systems. They found out that the techniques could be effectively transferred to SMS spam filtering and even went further to identify SVMs as one the most suitable methods for SMS spam filtering. [6] cited works which analyzed performance of SVM in SMS spam filtering and concluded According to [19] if the training data is linearly separable, then a pair (w , b) exists such that:

$$W^T xi + b \geq 1;\ for\ all\ x_i \in P \quad (1)$$

$$W^T x_i + b \geq -1, for\ all\ x_i \in N \quad (2)$$

Whre W is the weighted vector, T is transformation, b is the bias and P donoted positvive examples while N is negative examples

SVM's decision rule is given by

$$f_{w,b}(x) = sgn(w^T x + b) \quad (3)$$

where w is termed the weighted vector and b the bias (or – b is termed the threshold). If the data remains linearly inseparable in the higher dimensional space, the SVM tries to maximize the margin while minimizing the number of misclassified data points. The cost of misclassification, typically denoted as C, can be specified by the user and, together with γ, effectively drives the level of fitting for the SVM.

For example, given a set of training data = $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where $x_i \in R_n$ and $y \in \{+1, -1\}$ (here +1 denotes spam and −1 stands for legitimate mail). Training a support vector machine equals to finding the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} W^T . W + C \sum_{i=1}^{N} \xi_i \quad (4)$$

that SVM is an appropriate method for SMS spam filtering. Xiang et al. (2015) also noted that Support Vector Machines (SVMs) would be appropriate for the SMS spam filtering even though he did not give supportive experimental evidence.

[22] surveyed a number of text categorization techniques in order to compare their performances against that of SVM and discovered that S.V.M out performs most of text categorization techniques. Furthermore, the fact that SVM be used for both linearly separable problems as well as non-linearly separable problems makes it more suitable for text categorization because textual data may several dimensions. Another peculiar property of S.V.M. method is its ability to learn independently of the dimensionality of feature space. In text mining, it works well with documents having high dimensionality of feature space and so, and therefore it can performs well with text documents.

Subjected to $y_i(W^T \varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (5)$

In this example, training vectors $\mathbf{x_i}$**'s** are mapped into a higher (may be infinite) dimensional space by the function **φ**. W is a weight vector that should be minimized in finding an optimal linear separating hyperplane in this higher dimensional space. $\boldsymbol{\xi_i}$**'s** are slack variables and are used together with constant $C \geq 0$ to find solution of (2) in non-separable cases.

The most important advantage of SVM technique is that good generalization performance is achieved for pattern classification problems without incorporating knowledge from the problem domain [23].

2.4 Biterm topic model

In biterm topic model (BTM), topics are learned by directly modeling the generation of word co-occurrence patterns (i.e. biterms) in the whole corpus [11]. The authors outline major advantages of BTM as follows:

1. BTM explicitly models the word co-occurrence patterns to enhance the topic learning
2. BTM uses the aggregated patterns in the whole corpus for learning topics to solve the problem of sparse word co-occurrence patterns at document-level.

According to [24], the notation of "biterm", denotes an unordered word pair co-occurring in a short context (i.e., an instance of word co-occurrence pattern) where short context refers to a small, fixed-size window over a term sequence. In short texts with limited document length, such as tweets and text messages, each document is simply taken

as an individual context unit in which any two distinct words in the document construct a biterm. For example, a document with three distinct words will generate three biterms:

$$(w_1, w_2, w_3) \Rightarrow \{(w_1, w_2), (w_2, w_3), (w_1, w_3)\}$$

where $(\cdot, \cdot)$ is unordered. After extracting biterms in each document, the whole corpus now turns into a biterm set. The biterm extraction process can be completed via a single scan over the documents following this process.

1. For each topic z
   (a) Draw a topic-specific word distribution ϕ_z $\backsim$Dir(β)
2. Draw a topic distribution $\theta \backsim$Dir(α) for the whole collection
3. For each biterm b in the biterm set B
   (a) draw a topic assignment $z \backsim$Multi(θ)
   (b) draw two words: $w_i, w_j \backsim$Mulit(ϕ_z)

A major difference between BTM and conventional topic models is that BTM does not model the document generation process. Biterm topic model extracts any two distinct words in a short text document as a biterm. To infer the topics in a document, it is assumed that the topic proportions of a document equals to the expectation of the topic proportions of biterms generated from the document.

Probability of biterms in a document can be calculated as follows:

$$P(z|d) = \sum_b P(z|b)P(z|d) \quad (6)$$

2.5 Association rule mining

Association rules are rules presenting associations or correlation between itemsets [25]. An association rule is in form $A \Rightarrow$B. where A and B are two disjoint itemsets referred to respectively as lhs (left-hand side) and rhs (right hand side) of the rule.

Association rule mining has three most commonly used measures:

i. **Support** – refers to the percentage of cases that contains both A and B.

$$\text{Support } (A \Rightarrow B) = P(A \cup B) \quad (7)$$

ii. **Confidence** – The percentage of cases containing A that also contain B

$$\text{Confidence } (A \Rightarrow B) = P(B|A),$$

$$= \frac{P(A \cup B)}{P(A)} \quad (7)$$

Lift – the ratio of confidence to the percentage of cases containing B.

$$\text{lift}(A \Rightarrow B) = \frac{\text{confidence}(A \Rightarrow B)}{P(B)}$$

$$= \frac{P(A \cup B)}{P(A)P(B)} \quad (8)$$

Where in all the measures P(A) is the percentage (probability) of cases containing.

2.6 Term Frequency Inverse Document Frequency

Supervised machine learning involves learning and generalizing an input-output mapping. In case of text categorization the input comprises a set of documents, and the output is their respective categories .[26]. Again, supervised learning methods prescribe input and output format and uses attribute-based representation of the documents. Therefore the attributes must be transformed into vector space whereby each word belongs to only one vector space and identical words belong to the same vector space.

TF-IDF is the most commonly used method of conversion into vector space. **TF-IDF** stands for term frequency-inverse document frequency, and the **TF-IDF** uses weighting system. TF-IDF is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

TF-IDF is formulated as follows:

$$\text{TFIDF}(i.j) = \text{TF}(i.j).\text{IDF}$$

$$\text{IDF}(i) = \log \frac{N}{\text{DF}(i)} \quad (9)$$

Where (i.j) refers to the number of times i^{th} word occurs in j^{th} document. N is the numbers of documents and DF (i) counts the documents containing i^{th} word at least once. The documents so far transformed together for term-document matrix. Document normalization achieved by using TF-IDF ensures that all documents have the same length in the vector space irrespective varied original lengths. Formula for document–normalization in TF-IDF is:

$$TFIDF'(i.j) = \frac{TFIDF(i.j)}{\sqrt[\beta]{\sum_i TFIDF(i.j)^\beta}} \quad (10)$$

3. Methodology

3.1 Data collection

Data has been collected from University of California Irvine machine Learning Data Repository SMS spam collection. The SMS Spam Collection dataset consisted of 4823 non spam (ha) messages and 747 spam messages. Messages are labeled already in the repository as in table 1.

Table 1: Dataset representation

Label	*Messages*
ham	4823
spam	747
Total	5570

3.2 Data sampling

4-fold cross validation was used to split the dataset into training and test sets. In four-fold cross validation, the dataset D is randomly split into four mutually exclusive subsets (folds) D_1, D_2----D_4 of approximately equal size. Each time, the model will be trained on $\frac{3}{4}$ of the dataset. This means that k (k=4) iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k – 1 (4-1) folds are used for learning. In K-fold cross validation, the process is repeated *k* times with each of the *k* subsamples used exactly once as the validation (test) data. . Table 2 represents number of messages held out for test in every iteration.

Table 2: Data splitting

label	*Training set*	*Test set*	*Total*
ham	3620	1203	4823
spam	557	190	747

3.3 Algorithm implementation

The filtering algorithm was implemented using Scikit learn which is a free software machine learning library for the Python programming language.

3.3 Data cleaning

The data was preprocessed to eliminate special characters and stop-words using Sklearn non-words characters remover and Natural Language Toolkit (NLTK) English corpus respectively. Automatic word spacing was also implemented in the same function. Table shows a sample of messages before they are cleaned and then after they are cleaned.

- Eliminate special characters
- Automatic words spacing
- Eliminate non-using words

3.4 Data modeling

Cleaned data was modeled using biterm topic model which generates unordered word pairs. This topical model was developed to help deal with sparcity problem in short text document modeling. Table shows biterms generated from documents sampled from the research dataset.

3.5 Feature selection

Feature selection component generates networks of words using biterm topic model (BTM). In machine learning, feature (or variable) selection consists of choosing a subset of available features that capture the relevant properties of the data. Feature selection helps to enhance accuracy in many machine learning problems and also improves the efficiency of training (Le Thi, Le, and Pham Dinh 2014).

In this algorithm biterm generation pseudo code is as follows:

1. For each document *d*
   (a) Draw two words: $w_i, w_j \backsim \text{Mulit}(\phi_z)$
   (b) Repeat the process each and every word with successive words until all the words have been used

2. Count words in biterms generated in the whole document

Features to be used for classifying the SMS messages as spam or non-spam are composed of words selected based on association rule support measure defined in the algorithm.

For this research, rules have been specified based on the number of biterms generated from documents with different lengths. The rules here are as follows:

1) For documents with less than ten (10) biterms, five top most frequent words are selected
2) For documents with ten (10) and less than fifty (50) biterms, upto fifteen most frequent words are selected
3) For documents with fifty (50) and above biterms, top twenty (20) most frequent words are selected

3.6 Feature extraction

The sklearn.feature_extraction module was used to extract features in a format supported by Support Vector machines from the training and test datasets.

In order to re-weight the count features into floating point values suitable for usage by a classifier TF–IDF transform was used. TF-IDF also normalizes the documents. SVMs need numeric inputs and TF-IDF helps to convert text-based features into numbers before they are passed to the classifier.

3.7 SMS classification

Support vector machine learning algorithm was used to reclassify the already labeled SMS messages into spam and non-spam (ham) after being training on a subset of the dataset (test set).

3.8 The proposed system

The proposed SMS spam filtering system has three major components.

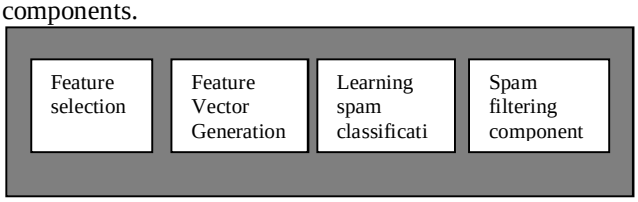

Fig. 1: Overall structure of the SMS spam filtering system

The first component generates biterms from which words that have the highest frequencies are selected to become features. In the second step selected features are passed over to the feature vector extraction component which generates feature for training using TF-IDF. In the third step, the support Vector machine learner is trained on the generated feature vectors. Lastly, spam filtering component categorizes messages using the completed classifier.

4. Performance evaluation results

The research has used subsets of SMS spam collection dataset from UCI machine Learning Data Repository. 4-fold cross validation was used to split the dataset into training and test datasets. In every instance, one-fold was left out to form the test set and the rest three folds formed the training set. The SMS Spam Collection dataset used consisted of 4823 non spam (ham) messages and 747 spam messages. Table 3 represents data split implemented using sklearn_cross validation.

Table 3: Instance of sampling using dynamic 4-fold cross validation

label	*Training set*	*Test set*	*total*
ham	**3620**	**1203**	**4823**
spam	**557**	**190**	**747**
Total	**4177**	**1393**	**5570**

4.1 Accuracy

After running several instances with 4-fold cross validation it was realized that accuracy of the filtering model remained unchanged at 98% as generated from sklearn metrics accuracy score confusion_matrix report. Accuracy refers to the proportion of the total number of predictions that were correct. Accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{total}}$$

Where true positives (TP) refer cases in which the model predictions for spam matched the original labels and true negatives (TN) refers to cases where the model predictions for ham matched original labels.

Other confusion matrix performance metrics used in the study included precision, recall and F1-score as contained in Table 4. The values were obtained by running sklearn metrics classification report.

Table 4: Confusion matrix

	Precision	*Recall*	*F1-score*	*Support*
ham	0.99	1.00	0.99	1203
spam	0.97	0.92	0.94	190
Avg/total	0.98	0.98	0.98	1393

4.2 Precision

Precision refers to fraction of cases labeled declared ham out of all instances where model declared ham and vice versa. For the model precision average was 0.98. Precision is computed as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{Predicted Yes(False Positive+True Positive)}}$$

4.3 Recall

Recall also known as true positive rate (TP) refers to the proportion of positive cases that were correctly identified. Recall is calculated using the equation:

$$TP = \frac{\textbf{True Positive}}{\textbf{False Negative + True Positive}}$$

In the study, recall average was 0.98

4.4 F1-Score

F1-score also called the F score or the F measure conveys the balance between the precision and the recall. The F1-Score is calculated as:

$$F1 - Score = 2 * \frac{(precision * recall)}{(\text{precision + recall})}$$

5. Conclusions

From the study, the researcher can conclude that biterm topic model (BTM) can be used to model SMS messages for classification. SMS messages can be categorized in spam and non-spam. SMS spam which messages are sent to recipients without explicit request while non-spam (ham) is a legitimate SMS message. In this research, the research tried to understand the different conventional data modeling techniques that have been used to model SMS messages for categorization by machine learning algorithms and also technologies used for sending SMS spam. For example, spamming software provides various facilities like multi targets, spoofing identity, using relays etc. Due to these tracing senders is difficult and therefore using content based filtering would be more appropriate

Acknowledgment

R. O. Midigo (author) thanks my supervisors, Prof. Waweru Mwangi and Dr. George Onyango Okeyo for their exemplary effort in shaping my MSc. thesis research from which this paper has been extracted. I also thank Jomo Kenyatta University of Agriculture and Technology for financial support for the MSc. course in Computer Systems whose taught content included artificial machine learning and document modeling. These techniques formed the basis for my thesis research and by extension this paper.

References

[1] T. a Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259–262.

[2] L. Zhang, J. Zhu, and T. Yao, "An Evaluation of Statistical Spam Filtering Techniques Spam Filtering as Text Categorization," *ACM Trans. Asian Lang. Inf. Process.*, vol. 3, no. 4, pp. 243–269, 2004.

[3] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach," no. September 2000, pp. 1–12, 2000.

[4] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, "Comparative study on email spam classifier using data mining techniques," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2012, vol. 1, pp. 14–16.

[5] B. Coskun and P. Giura, "Mitigating SMS spam by online detection of repetitive near-duplicate messages," in *IEEE International Conference on Communications*, 2012, pp. 999–1004.

[6] S. J. Delany, M. Buckley, and D. Greene, "SMS spam filtering: Methods and data," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9899–9908, 2012.

[7] H. Alshalabi, S. Tiun, N. Omar, and M. Albared, "Experiments on the Use of Feature Selection and Machine Learning Methods in Automatic Malay Text Categorization," *Procedia Technol.*, vol. 11, pp. 748–754, 2013.

[8] H. Kale, G. Rane, S. Shende, and S. Shinde, "Short Message Service Offline Notification System through Bulk SMS for Android Application," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 12, pp. 101–103, 2014.

[9] M. Unmehopa, K. Vemuri, and A. Bennett, *Parlay / OSA: From Standards to Reality*. John Wiley & Sons, 2006.

[10] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

[11] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *WWW '13 Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 1445–1456.

[12] F. Sebastiani, "Text Categorization," *Stud. Health Technol. Inform.*, vol. 129, no. Pt 2, pp. 968–72, 2005.

[13] S. Bengio, *Multimodal Signal Processing*. Elsevier, 2010.

[14] P. Gupta, "Short Message Service: What, How and Where?," *Wireless Developer Network*. 2000.

[15] S. Ryan, S. Kandanaarachchi, and K. Smith-Miles, "Support Vector Machines for Characterising Whipple Shield Performance," *Procedia Eng.*, vol. 103, pp. 522–529, 2015.

[16] G. Shi, *Data Mining and Knowledge Discovery for Geoscientists*. Elsevier, 2014.

[17] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.
[18] V. Kotu and B. Deshpande, *Predictive Analytics and Data Mining*. Elsevier, 2015.
[19] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, 2006.
[20] OpenCV, "Introduction to Support Vector Machines — OpenCV 2.4.13.1 documentation," 2014. [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html#introductiontosvms.
[21] G. Gomez and R. Sanchez, *End-to-End Quality of Service Over Cellular Networks: Data Services Performance Optimization in 2G/3G*. John Wiley & Sons, 2005.
[22] R. Jindal and S. Taneja, "Text Categorization – A Review," pp. 444–449.
[23] I. Joe and H. Shim, "An SMS Spam Filtering System Using Support," *LNCS Springer-Verlag*, pp. 577–584, 2010.
[24] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM : Topic Modeling over Short Texts," vol. 26, no. 12, pp. 1–14, 2014.
[25] Y. Zhao, "9 Association Rules Basics of Association Rules," in *R and data mining: Examples and cases*, 2013, pp. 89–103.
[26] I. Pilászy, "Text Categorization and Support Vector Machines," in *The Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, 2005, vol. 1.
[27] H. A. Thi, H. Le, and T. P. Dinh, "Feature selection in machine learning: an exact penalty approach using a Difference of Convex function Algorithm," *Mach. Learn.*, 2014.

Biography

R. O. Midigo was born in 1972 in Nyanja Province of Kenya and is currently pursuing MSc. Degree in Computer Systems at Jomo Kenyatta University of Agriculture and Technology (JKUAT). He received Bachelor of Science degree in Information Science (IT option) from Moi University, Eldoret (Kenya) 2008. In 1996, he joined JKUAT Library Department as an employee. He has risen within the ranks in the library and serves as Systems Librarian.

Prof. Waweru Mwangi is an associate professor and a Senior Lecturer in the School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology. He holds PhD in Information Systems Engineering from Hokkaido University (Japan) 2004, MSc Operations Research and Cybernetics, Shanghai University (China)1995. Bed Mathematics Kenyatta University (Kenya) 1989.

Dr. G. O. Okeyo is a holder of PhD in Activity Recognition in Smart Environments from the University of Ulster (UK) 2013, Mater of Science in Information Technology from University of Nairobi (Kenya) 2007. He is a lecturer in the School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology.