Securing Cloud Data: A Machine Learning based Data Categorization Approach for Cloud Computing

Fahad Burhan Ahmad (asif.nawaz@uaar.edu.pk)
Pir Mehr Ali Shah Arid Agriculture University: University of Arid Agriculture
Asif Nawaz
Pir Mehr Ali Shah Arid Agriculture University: University of Arid Agriculture
Tariq Ali
Pir Mehr Ali Shah Arid Agriculture University: University of Arid Agriculture
Azaz Ahmed Kiani
Pir Mehr Ali Shah Arid Agriculture University: University of Arid Agriculture
Ghulam Mustafa
Pir Mehr Ali Shah Arid Agriculture University: University of Arid Agriculture

Research Article

Keywords: Cloud Computing, Data classification, KNN, Naïve Bayes, Random Forest, SVM

Posted Date: February 7th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1315357/v1

Abstract

Cloud computing (CC) a new systematic model that allows users to store data on remote servers that are accessible via the internet. Due to this approach, the personal and essential data are being stored with easy to access and move. Because of this, it demands is increasing day by day. On this, one can store different data such as financial transactions, paperwork based files and multimedia content. Not only this, but CC also reduce the services dependency on local storage by reducing operational and maintenance costs. Existing systems such as encrypt all data with the same key size, regardless of the data's level of confidentiality due to which the processing cost and time become increased. Moreover, all such techniques classify the data with a low accuracy rate and don't provide better confidentiality. In this research, a cloud computing approach based on automated data classification has been presented for data sensitivity. The proposed model is based on three level of sensitivity i.e. basic, confidential and highly confidential using Random Forest (RF), Naïve Bayes (NB), k-nearest neighbor (KNN) and support vector machine (SVM) classifiers for training the proposed model with automated feature extraction. The proposed model achieved 92% accuracy that has been showed in simulation results. From this, we conclude that RF, NB, KNN performs better than SVM. The proposed research also provides useful guidelines for cloud service providers (e.g., drop box and Google drive) and researchers.

1 Introduction

Cloud computing (CC) is the most demanding platform that comprises of different advanced level technological applications where a person can store, retrieve and securely store their personal documents. Users can access cloud services and applications using cellular devices such as mobile, laptop and android phones. As far as the security of the documents is concern, cloud computing can be consider as most reliable and secure platform. Whereas, other storage devices like mobile phones and laptops may unable to product such secure platform for data storage due to lack of battery storage, performance and storage capacity [1, 3]. Cloud storage services are frequently used to store and backup arbitrary data in a cost-effective, user friendly and in fast accessible manner [1]. They also provide the facility of data sharing and device synchronization in easier way.

There is no standard set of attributes for cloud storage architecture, and several cloud storage architecture schemes exist across different cloud storage systems. However, to provide cloud storage services to clients, cloud storage frequently consists of thousands of storage devices clustered together and connected via a distributed file system, a network, and other storage middleware. Cloud storage includes a storage resource pool, service level agreements (SLAs), a distributed file system, and services interfaces. The primary goal of cloud storage designs is to provide on-demand storage in a multi-tanned, highly scalable manner. A common cloud storage design includes a front end with an API for storage access. In traditional storage systems, this API is the SCSI protocol; however, in the cloud, these APIs are becoming more prevalent, and we may find file service front ends, web service front ends, and even classical front ends like (iSCSI and internet SCSI). Storage logic is a layer of middleware that sits behind the front end. Several functionalities, such as data minimization and replication, are implemented using

this layer. The back end of the cloud storage architecture, which could be an internet protocol that implements specific features or a traditional back end for actual drives, is responsible for the physical storage of data. On-demand self-service, broad network access, resource pooling, quick elasticity, and measured service are the five main aspects of the cloud model. Service models are divided into three categories by the National Institute of Standards and Technology (NIST). Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), as well as private, public, hybrid, and community deployment options, are all available through cloud services [1, 2, 3].

Figure 1 gives a high-level picture of NIST's cloud computing standard architecture [3]. Five major actors Cloud customer, cloud supplier, cloud carrier, cloud auditor, and cloud broker defined by the architecture. A individual or a company that engages in a transaction or process and/or fulfils responsibilities is referred to as an actor in CC. The actors identified in the NIST cloud computing standard architecture are summarized in Table 1 [3]. Security is a component of the Cloud Provider in this case.

Table 1
Actors in NIST Cloud Computing reference architecture adopted from [26]

Actor	Definition
Cloud Consumer	A person or organization that maintains a business relationship with, and uses service from, Cloud Provider.
Cloud Provider	A Person, organization, or entity responsible for making a service available to interested parties.
Cloud Auditor	A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.
Cloud Broker	An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.
Cloud Carrier	An intermediary that provide connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

The data on the cloud is stored randomly. As the amount of data increases, the user has more difficulty for searching that data. Conversely, if the data is stored in organize form, the user will be able to easily access the required data. Therefore, there is a need to develop a model that allows data to be easily stored on cloud organized form. The benefits of classification are mentioned below:

• Ensures accurate classification with fewer false positives by employing compound word search.

• Has an index, allowing you to search for sensitive phrases without having to re-crawl your data storage.

• Comes with a flexible taxonomy manager that allows you to tailor your classification parameters.

• Offers procedures for automating tasks like transferring sensitive data from public sharing.

• Supports on-premises and cloud content sources, as well as structured and unstructured data.

Users are hesitant to upload personal and confidential files to the internet storage because they are concerned that the service provider may misuse them. They are also concerned that their data may be hacked and compromised as a result of the widespread adoption of effective cloud storage attacks [2]. Existing cloud based architectures use the same key length to encrypt all data, which may be infeasible, and do not consider the data's level of secrecy. Treating low and high secret data the same creates extra overhead and slows processing.

As a result of the previous facts, this study concentrates on three key components of cloud computing: data sensitivity, automatic classification, and high-level accuracy. Before the transmission and storage activities, we provide an effective system. Which ensures automatic data classification by using machine learning algorithms to maintain confidentiality and integrity in cloud storage. However, on the fact that data confidentiality is particularly essential in the cloud environment. Moreover, this framework will reduce manual efforts for classification and will achieve high-level accuracy rate.

The proposed model consists of three classes with respect to sensitivity levels that are shown in Fig: 2 which are named as Basic Class, Confidential Class and Highly Confidential Class. The proposed work has three phases, the first phase is the preprocessing of text datasets, the second phase is feature extraction using python sk learn library, the third phase is to train the model on the basis of features and the last phase produced three classes of text data according to data sensitivity. We are using different classifiers for the text classification to obtain the accuracy.

The information classification assists in deciding which security standards and policies are appropriate for protecting that data. Personal and non-exclusive (non-distinct) comprehension are the two types of data. The features of the data are used to classify the material. The sensitive data are classified as "confidential" and "highly confidential," while the remaining data are labeled as "basic." For better performance, the proposed cloud data classification framework using the RF, NB, KNN, and SVM algorithms to obtain high-level accuracy and automatic classification.

As a result of the previous facts, this study concentrates on three key components of cloud computing: data sensitivity, automatic classification, and high-level accuracy. In both transmission and storage activities, we provide an effective system that ensures data confidentiality and integrity in cloud storage, based on the fact that data confidentiality is particularly crucial in the cloud environment.

Moreover, this framework will reduce manual efforts for classification and a high-level accuracy rate. Following is the organization of remaining of this paper. Related work is mentioned in the second portion. The proposed work is presented in the third part. Experiments are presented in the fourth part. Results and discussions are discussed in fifth and future direction are mentioned in the part sixth, the evidence has been achieved with coming analysis directions.

2 Related Works

In [4] digital signatures were optimized to enhance security, while the RSA method was used to secure the confidentiality component of security. The encryption process is performed in five steps. The first step consists of key generation. A digital signature is implemented in step 2. After that encryption and decryption are performed in steps 3 and 4. In step 5 signature verification is performed. [5] Proposed an architecture that includes digital signatures and exchanges Diffie Hellman keys with the Advanced Encryption Standard (AES) encryption technique to secure the confidentiality of data stored in the cloud. The Diffie Hellman key exchange facility renders the key in transit ineffective even if it is compromised because it is useless without the user's private key, which is only available to authorized users. Hackers will have a hard time breaking into the security mechanism because of the architecture's three-way approach, which protects data stored in the cloud.

Sinha N et.al in [6] give a general overview of cloud computing, including its advantages, architecture, and implementation, as well as any potential issues. It provides an overview of the various sorts of security, data, and performance concerns that exist in the cloud. In [7] different cryptographic algorithms are explored and taken into consideration in this study to secure data confidentiality. Several criteria, such as block size, key length type, and characteristics, are used to compare different cryptographic algorithms. This study examined many cryptographic techniques that can be used to ensure cloud data security.

The KNN technique to classify data, to maintain data confidentiality. The primary goal of data classification is to ensure security. Using the KNN technique, they divide the data into sensitive and non-sensitive categories[22]. Encryption is used on sensitive data to offer security. The fundamental reason for classification is that it makes it easier to choose an appropriate security level for data based on the data's needs. As a result, security will be improved. [9] This paper presented another essential technique to improve the security of data in the cloud. Different parameters are taken for data classification, and then encryption is performed on the classified data. Access control, content, and storage are some of the classification aspects that are considered. To improve security and efficiency, data classification is done on these properties, and then encryption is conducted.

The approach of K-NN and Improved Naïve Bayes has been used to identify data and to provide data confidentiality. The basic purpose is to assure the protection of confidential data. Using the K-NN, Improved Nave Bayes algorithm, they were able to classify the data into sensitive and non-sensitive labels. Encryption is used to protect confidential information. According to the need for classification of data, it was simple to pick a suitable protection method [21]. They produce 72% accuracy by using improved Naïve Bayes. So, in this way, protection can be annealed.

Using the creational set of tests defined at each node or branch decision tree, the training dataset is recursively partitioned into smaller sub-divisions [20]. This feature has one value on each branch descending from the node, and each node of the tree represents a test of a feature from the training dataset. The dataset is categorized by starting with the root node and then testing each characteristic. Then, according to the value of the feature in the given dataset, going down the tree branch, and this method is repeated recursively [17].

From the above discussion it has been concluded that most of the exiting work may have the data's level of confidentiality, encrypt all data with the same key size, increasing the processing cost and time. Furthermore, all of these methods manually classify data with a low accuracy rate and do not offer better security. The proposed approach tries to reduce the manual classification for cloud computing. To high certainty and to provide the familiarity level required for data, the second specification has been ratified to compute the results on different classifiers and compare with our proposed approach, in three different levels with machine learning algorithms, classification is automatically being performed. The levels i.e. Basic, Confidential, and highly confidential level are based on the sensitivity of data. Four machine learning algorithms are being used to provide an automatic classification of data up to a great extent.

3 Proposed Work

This research article analyzed the impact of various machine learning data categorization techniques such as the NB, RF, SVM, and KNN algorithms. Data categorization is performed on sensitivity levels for the cloud. Our proposed model consisting of three classes: Basic Class, Confidential Class, and Highly Confidential Class, as shown in Figure 2.

3.1 Basic Class

Our proposed model's basic class comprises a common type of data, such as text documents, with a low level of confidentiality. Basic information such as advertising, announcements, and notices can be found in text documents. As a result, this level provides a basic level of data security. The basic class does not require encryption on the client side; nevertheless, when sent, it will be encoded on the server-side using the backup service's key.

3.2 Confidential Class

Personal files, such as private accounts, web accounts, and professional details, are covered in this class. Our confidential class is intended for data with a medium level of confidentiality. Security is necessary to protect our data because this class keeps track of secret and private information. At the confidential level, encryption methods such as AES can be utilized for this purpose. In this class, encryption will be used on the client-side.

3.3 Highly Confidential Class

This class is responsible for financial transactions, organization-wide secret documents, and military data. Users may have reservations about the data's high level of confidentiality, so they avoid all newly provided services. This level will provide security by using two standard recommended algorithms due to the high degree of confidentiality and integrity. The US National Security Agency recommended AES 256 to prevent unwanted access to top-secret material (NSA). The SHA-2 algorithm, on the other hand, ensures data integrity. This algorithm will be used to compute the hash value of data before changing or

transferring it. Also, construct a hash value for data retrieval on user request; the value must be the same to ensure that the data is not tampered with.

3.4 Dataset

We have collected the Reuters-21578 text categorization collection dataset from the UCI ML repository. We also collect confidential and highly confidential data from the CIA public library for text composition to test the recommended system. The compatible material like commercials, announcements, news articles data, accounts information documents (of the organization) and military information are compiled from the different mentioned international repository.

The datasets generated during and/or analyzed during the current study are available in the [UCI, CIA] repositories (Access links are provided in Table 2).

Table 2
Dataset with corresponding repositories

S.No	Dataset Type	Number of documents	Dataset Link
1	Public and Confidential Data	4000	https://miguelmalvarez.com/2015/03/20/classifying-reuters-21578-collection-with-python-representing-the-data/
2	Highly Confidential Data	2010	https://www.archives.gov/research/intelligence/cia

3.5 Data Processing

Natural language processing is a method of analyzing, manipulating, and extracting meaning from human language in such a way that computers can understand it. Before the text input is sent to the algorithm, it is transformed using the NLTK library. The unstructured text data is subsequently transformed into a structured format. Many machine learning techniques depend on processing as a significant component. It has a noticeable impact on the classification process as well [15].

3.5.1 Tokenization

Tokenization is the process of breaking down a character arrangement into components, each represents a word or phrase. In natural language processing, there are two types of tokenization: word tokenization and sentence tokenization. The list of tokens, which can be a word or a phrase, is then utilized to process the data [15]. Fig. 3 shows the tokenization process.

3.5.2 Filtering

Filtering a text file is a common practice to remove some of the more inconsequential terms. A reciprocal filtering mechanism prevents the removal of words. Stop words are terms that regularly appear in text

that lacks substantive information (for example relational words, conjunctions, and so on.).

Thus, words that appear frequently in the content are said to have insufficient information to distinguish between reports, whereas terms that appear infrequently may similarly be of low significance and can be eliminated from the content document [17].

3.5.3 Lemmatization

It is the study that considers the feature extraction of the words. Taking the different varieties of a word, for example, they can be broken down into a single piece. In other words, lemmatization approaches aim to separate structures into several tenses and items into a single structure. To lemmatize the texts, we must first identify the part of speech of each word in the document, and stemming approaches are preferred over POS because POS is repetitive and prone to errors [16].

3.5.4 Stemming

This process converts words to their root forms, such as mapping a group of words to a common stem, regardless of whether the stem is a valid word in the Language. As a result, stemming a word or a sentence can produce non-words. Stems are formed by deleting all prefixes and suffixes from a word. There are language-dependent stemming algorithms [17].

3.6 Feature extraction

Before being fed into the classifier, the data from the text document is represented in indexes. Words could be used to describe features. The Bag of Words technique, which represents the document as a collection of words, is a commonly used structure. We must first define several terms and variables that will be regularly used in the following to allow for formal or formal descriptions of feature extraction. If there are distinct terms or words in a set of documents D=d1,d2,..., dD, then V=w1,w2,...,wv exists, then V is known as the vocabulary [18]. fd(w) represents the occurrence of the word w V in the document dD, and fD represents the number of documents containing the word w. (w). tD = (fd(w1), fd(w2),..., fd(wv) represents the feature vector for document t.

Algorithm for creating a BoW model

Step 1: Creating the Bag of Words model

Step 2: word2count = {}

Step 3: for data in dataset:

Step 4: words = nltk.word_tokenize(data)

Step 5: for word in words:

Step 6: if word not in word2count.keys():

Step 7: word2count[word] = 1

Step 8: else:

Step 9: word2count[word] += 1

Step 10: End

There are two general approaches for symbolizing a document using a list of features, namely the local dictionary technique and the global dictionary methodology [13, 18]. The international dictionary will be built using just relevant texts. As a result, if a term appears in the relevant document, it can be added to the lexicon as a feature. As far as the local dictionary technique is concerned, it can produce better results [19].

3.7 Feature Vector

Transforming documents into numeric vectors is the most universal approach to represent them. The “Vector Space Model” is another name for this demonstration. Its structure, on the other hand, is simple and was designed with information retrieval (IR) and indexing in consideration. The vector space model is widely used in various text mining techniques and IR classifications, and it allows for intelligent analysis of a huge number of documents [19]. Each word in VSM is identified by a numeric number that signifies the word's weight or 'importance' in the document. The first of two basic feature weight models is the Boolean model. If a feature is present in the document, it has a weight of 1; otherwise, it has a weight of 0 if it is not present in the document. The second way is term frequency and inverse document frequency (TF-IDF), which is the most general term weighting scheme. This phrase comes from IR, which uses both TF and IDF to determine the relevance of a characteristic (IDF) [19]. The number of times a feature appears in the document is represented by TF, and the frequency or rarity of the feature is shown by IDF over all papers. The weight assigned to each word w D is derived as follows using the TF weighting method as an example:

$$f(w) = fd(w) * \frac{log|D|}{fD(w)} (1)$$

Algorithm for Tfidf Vectorizer to calculate tf-idf score

```
tokenizer = RegexpTokenizer(r'\w+')

stemmer = PorterStemmer()

path ="E:/Thesis/dataset/"

docs = []

for subdir, dirs, files in os.walk(path):
```

```
for file in files:
file_path = subdir + file
shakes = open(file_path, encoding="latin-1")
text = shakes.read()
docs.append(text)
'''print(docs) '''
def stem_tokens(tokens, stemmer):
stemmed = []
for item in tokens:
stemmed.append(stemmer.stem(item))
return stemmed
def tokenize(text):
tokens = tokenizer.tokenize(text)
stems = stem_tokens(tokens, stemmer)
return stems
'''Initializing Vector'''
vectorizer = TfidfVectorizer(tokenizer=tokenize, stop_words='english')
DocumentVectorizerArray = vectorizer.fit_transform(docs).toarray()
with open('E:/Thesis/fahad.csv') as csv_file:
csv_reader = csv.reader(csv_file, delimiter=',')
line_count = 0
rowcount = 0
vocabularyIndex = 0
Idfscore = 0
```

```
for row in csv_reader:

if line_count == 0:

for docIndex,doc in enumerate(docs):

newrow = [0] * len(row)

for index,column in enumerate(row):

if column in doc:

vocabularyIndex = vectorizer.get_feature_names().index(column)

Idfscore = DocumentVectorizerArray[docIndex][vocabularyIndex]

newrow[index] = Idfscore

else:

newrow[index] = 0

with open('E:/Thesis/fahad.csv', 'a', newline='') as f:

writer = csv.writer(f)

writer.writerow(newrow)

line_count = line_count+1

f.close()

else:

csv_file.close()

break
```

The documents in the collection is represented by |D|. The word frequency is divided by the IDF, in the TF-IDF formula. This normalization lowers the value of terms that appear more frequently in the documents collection, guaranteeing that particular features that appear less frequently in the collection excite the same documents more. In figure 5 below, the estimated feature vector is shown.

3.8 Sensitivity Base Classification

Classification is a method of supervised learning. Which classifier should be used to predict the class label on new data while also learning from the training data? It's used in a variety of disciplines, including

medical diagnosis, picture processing, document management, and text classification. It is also taken into consideration in a variety of communities, including machine learning, database, IR, and data mining [23].

The fundamental goal of classification is to assign specified classifications to text documents [17]. The following is a clear definition of the classification problem. We need a D = {d1,d2,.. .,dn} training set of documents, so that each document di is associated with a label ℓi from the collection L = {ℓ1,ℓ2,. .,ℓk}

Some of well-defined machine learning algorithms that were used to classify this document. These techniques include artificial neural networks, decision trees, the KNN technique, NB, rule-based classifiers, and SVM. The classification of the document supplied below is more appropriate out of the classifier [17].

A random forest is simply a series of decision trees with their outcomes aggregated into a single ultimate result. They're so effective because they can reduce overfitting while also reducing bias-related inaccuracy. A decision tree is essentially a categorized tree of the training dataset in which the data is hierarchically segregated using a feature value condition [24, 25].

3.9 Training Module

In the training phase preprocessing is performed using NLP and obtained features automatically. So extracted features are used for prediction by applying different classification algorithms. The predicted output is matched with the input instance to train it. The suggested methodology takes as input text documents including basic, sensitive, and highly confidential data, and at the end of the training phase, a final predictive model is chosen to forecast class labels. We provide 6010 text documents to train our model. KNN, NB, RF, and SVM are the four classifiers used to analyze the data.

The class labels are a set of outputs in this module that are used to train a prediction model by combining them with features (also known as variables). To allow a machine-learning system to predict class labels, use a training model. To train the classifier and test the efficiency of the trained model, cross-validation is performed. After being trained, the model will be able to predict the class label for any new text documents based on the features they include. The suggested method uses the training dataset, which was manually labelled with class labels, to create a classifier. Text documents are mapped to one of the predefined categories, the class labels, in this study. A training model for the automatically retrieved features with sparse matrices would be created using the suggested architecture. Because we developed 2048 automatic features, each one has its unique frequency, which is calculated using TFIDF. To predict class labels, classification algorithms were used for all of them, and three models with higher accuracy were chosen. We chose Random Forest, Nave Bayes, and k-nearest neighbor classifiers over other machine learning algorithms because they performed better. Multiple classifiers are available, and it is critical to choose the appropriate one for our problem to predict correct class labels. Furthermore, the SVM technique is inefficient for large data sets. SVM does not perform well when the data set contains more noise, such as overlapping target classes.

3.10 Testing Module

The final trained model is evaluated by using a new set of testing data to see how successfully our model was trained. A new dataset of text documents was used as input in the testing phase, and the new text documents were pre-processed. For pre-processing, the learning objects were tokenized into words based on the properties they contained. The newest cases were loaded into the trained model at the end. Which predicted the text's class label accurately to evaluate the classifiers, researchers employed about 2030 testing datasets from various publications. This is quite helpful to the testing module.

3.11 Development of Prototyping

The proposed methodology's goal is to construct the application's modules such that they can be validated. The system's back end was built in Python 3.7, which is a powerful language to work with for data analysis when combined with the correct tools and modules.

Document categorization further group into two phases, first consist of training and the second is testing. The training phase is broken down into several parts, including NLP pre-processing, feature extraction, and feature vector generation, followed by the application of various classification algorithms in the prediction module.

A final train model is chosen from the training phase and used to predict document classifications in the testing phase. Before being preprocessed, the data is sent into the pdf to text software (pdf to txt.exe), which converts pdf documents to text files. Before being fed into the preprocessing machine.

The system's back end was built in Python 3.7, which is a powerful language to work with for data analysis when combined with the correct tools and modules. Python is a free and open-source programming language designed to be simple to learn and powerful. We used a variety of Python libraries, including nltk, sklearn, numpy, os, csv, and scipy. These libraries are used to preprocess data, extract features automatically, construct feature vectors, produce dataset csv files, and then give the csv files to the classifier to train and test the models.

4. Experiment

This section contains a variety of experiments that are carried out to assess the proposed work on automatic data classification. RF, NB, KNN, and SVM are four types of classifiers used in experiments to obtain accuracy on a given dataset. For experiments, The Reuters-21578 text categorization data was obtained from the UCI ML repository. To evaluate the recommended system, we additionally take sensitive and highly confidential data from the CIA public library for text composition.

We must first define several terms and variables that will be regularly used in the following to allow for formal or formal descriptions of feature extraction. Given a set of documents D = {d1,d2,...,dD}, and the set of various terms or words in the set V= {w1,w2,...,wv},V is referred to as the vocabulary. fd(w) represents the occurrence of the word wV in the document dD, and fD represents the number of documents containing the word w. (w). tD = (fd(w1), fd(w2),..., fd(wv) represents the feature vector for

document t. The extracted features subset is then used to create a feature vector, in which the features are given weights. If a feature is present in the document, the TF-IDF is calculated for it, and a value is assigned to it; if the feature is not present, it is indicated as zero. The value of a characteristic in those texts is determined by its weightage. Feature vectors are constructed from all extracted features, according to the proposed design depicted in Figure 4. A weight is allocated to each feature. Sample feature vectors in the term matrix for automatically retrieved features are shown in Figure 4. To reflect the natural structure of the documents collection, a documents term matrix is used. Every row in the matrix represents a document, and each denotes an unique phrase or feature.

4.1 Model Evaluation Metrics

The objective is used to assess the classification model's performance. For this purpose, we reserved a random number of the categorized documents test set. Classify the test set and connect the expected labels with the true labels, as well as assess the quantitative performance, after the classifier has been trained with a labelled dataset. Accuracy is defined as the proportion of accurately classified documents to the total number of records. Precision, recall, and F-Measure are three standard objective evaluation or qualitative measurements for classification. The recall is an assessment of the system's intelligence in classifying important documents, while the precision is an assessment of the system's intelligence in identifying irrelevant documents. F-measure will also be used during evaluation to overcome the bias issue in precision and recall.

4.2 Precision

Precision is the calculation of the number of documents that are accurately classified by an entire number of documents.

$$precision(p) = \frac{tp}{tp + fp} = \frac{numberofdocuments}{numberoflabeldocuments}$$

4.3 Recall

The recall is the calculation of the number of retrieved documents that are significant or accurately categorized by several applicable documents.

$$Recall(r) = \frac{tp}{tp + fn} = \frac{numberoflabeldocuments}{numberofdocuments}$$

4.4 F1-Measure

In F1 Measure the calculated accuracy and calling off is used to dig out the symphonic purpose among them. The matchless records are 1 of F1 measure when the accurateness and precision are perfect in sequence and lowest when the F1 measure is 0.

$$F1 - Measure(f) = \frac{2*Precision*Recall}{Precision + Recall}$$

4.5 Accuracy

It is a mostly accustomed framework to assess the accomplishment of the graded algorithms.

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn}$$

Accuracy is computed with the help of the above equation.

5. Results And Discussion

The proposal's goal is to validate the application by developing modules for it. The use of Python 3.7 at the system's back end is a great example of data analysis with the correct tools and frameworks. The first phase of document classification is training, followed by the second phase of testing. Multiple parts of the training phase include NLP pre-processing, feature extraction, and feature vector, while the prediction module includes applying different classification algorithms for comparison. Figures 5 and 6 show the RF, NB, KNN, and SVM algorithms using various methodologies. In the graph above, the proposed approach is compared with the existing method in terms of performance. In the performance graphs, it is obvious that the proposed technique outperforms the old approach. Each classifier's average precision, recall, and F-measure are presented, and Figure 5 shows a graphical depiction of the precision, recall, and F-measure of three class labels, whereas Figure 8 shows the accuracy comparison of data classification algorithms RF, NB, KNN, and SVM Algorithms. SVM has a classification accuracy of 43%, KNN algorithm has an accuracy of 83%, NB has a classification accuracy of 72% and RF has a classification accuracy of 92%, indicating that the proposed algorithm has classified data more accurately. In terms of precision, recall, and F1-measure, Figure 6 compares the RF, NB, KNN, and SVM methods. Cloud data is categorized according to its security requirements using machine learning algorithms to reduce encryption time. The proposed methodology, according to the findings, performs better in terms of high accuracy, precision, recall, and F1-measure.

6 Conclusion And Future Work

A method for data confidentiality and automatic text document classification in a cloud context is proposed in this study. The research's main goal was to define the data and achieve high-level accuracy. Information security requirements that use machine learning algorithms to categorize data into basic, confidential, and highly confidential categories. This security model's primary contribution is data confidentiality and categorization utilizing a machine learning classification approach. The proposed methodology's goal is to construct the application's modules such that they can be validated. The system's back end was built in Python 3.7, which is an excellent language to work with for data analysis when combined with the correct tools and modules. The results reveal that the proposed method is more compatible than just storing data without first identifying its security needs. Furthermore, in terms of

accuracy, precision, recall, and F1measure, the random forest technique outperforms the NB, SVM, and KNN classification techniques. We will pass this classified data across TLS, AES, and SHA in the future. We intend to expand our system by filling another research gap. Other cryptographic techniques will be used, which can provide a higher level of reliability and security.

Declarations

We affirm that the Submitted Research Paper is my original work, with no part of it previously published. I accept full responsibility for the fact that if the document is found to be in violation of basic rules in the future, the final decision will be made by the relevant authorities. Plagiarism, in any form, will result in the paper being disqualified.

References

1. Sun, X., Wang, Z., Wu, Y. et al. A price-aware congestion control protocol for cloud services. J Cloud Comp 10, 55 (2021). https://doi.org/10.1186/s13677-021-00271-5
2. Al-Said Ahmad, A., Andras, P. Scalability resilience framework using application-level fault injection for cloud-based software services. J Cloud Comp 11, 1 (2022). https://doi.org/10.1186/s13677-021-00277-z.
3. Song, D., E. Shi, I. Fischer and U. Shankar, “Cloud data protection for the masses”, IEEE Computer. Soc., Vol. 45, Issue 1, pp.39-45, 2012.
4. Somani U, Lakhani K, Mundra M. Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. In Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on 2010 Oct 28 (pp. 211-216). IEEE.
5. Rewagad P, Pawar Y. Use of digital signature with Diffie Hellman key exchange and AES encryption algorithm to enhance data security in cloud computing. In Communication Systems and Network Technologies (CSNT), 2013 International Conference on 2013 Apr 6 (pp. 437-439). IEEE.
6. Sinha N, Khreisat L. Cloud computing security, data, and performance issues. In2014 23rd Wireless and Optical Communication Conference (WOCC) 2014 May 9 (pp. 1-6). IEEE.
7. Diwan V, Malhotra S, Jain R. Cloud security solutions: Comparison among various cryptographic algorithms. IJARCSSE, April. 2014 Apr.
8. Dubey AK, Dubey AK, Namdev M, Shrivastava SS. Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment. In Software Engineering (CONSEG), 2012 CSI Sixth International Conference on 2012 Sep 5 (pp. 1-8). IEEE.
9. Yellamma P, Narasimham C, Sreenivas V. Data security in cloud using RSA. In Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on 2013 Jul 4 (pp. 1-6). IEEE.

10. Lo'ai Tawalbeh NS, Raad S. Al-Qassas and Fahd Al Dosari,"A Secure Cloud Computing Model based on Data Classification". In First International Workshop On Mobile Cloud Computing Systems, Management and Security (MCSMS-2015) 2015 (Vol. 52, pp. 1153-1158).
11. Jacovi A, Shalom OS, Goldberg Y. Understanding convolutional neural networks for text classification. arXiv. 2018.
12. Pennington J, Socher R. Manning CD. Glove: Global Vectors for Word Representation; 2014. p. 1532–43.
13. Thiyagarajan, D., Shanthi, N.: A modified multi objective heuristic for effective feature selection in text classification. Cluster Comput. 22(5), 10625–10635 (2019).
14. Shaikh, Rizwana, and M. Sasikumar. "Data Classification for achieving Security in cloud computing." Procedia Computer Science 45 (Elsevier-2015): 493-498.
15. Guo, J. (1997). Critical tokenization and its properties. Computational Linguistics, 23(4), 569–596.
16. Saif, H., Fernández, M., He, Y., & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of twitter.
17. Mitchell, T. M., & others. (1997). Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 45(37), 870–877.
18. Deng, B., Li, Q., Liu, X., Cao, Y., Li, B., Qian, Y., Xu, R., Mao, R., Zhou, E., Zhang, W., Huang, J., Rao, Y. (2019). Chemoconnectomics: Mapping Chemical Transmission in Drosophila. Neuron 101(5): 876--893.e4.
19. Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. Expert Systems with Applications, 66, 245–260.
20. Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment, 61(3), 399–409.
21. Kumar, Rajeev. "Secure Cloud Model using Classification and Cryptography." International Journal of Computers and Applications 159.6 (2017).
22. Zardari MA, Jung LT, Zakaria N. " K-NN classifier for data confidentiality in cloud computing. In Computer and Information Sciences (ICCOINS), 2014 International Conference on 2014 Jun 3 (pp. 1-6). IEEE.
23. Chen, Hanting, et al. "Pre-trained image processing transformer." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
24. Molla, MM Imran, et al. "Cardiotocogram Data Classification Using Random Forest Based Machine Learning Algorithm." Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019. Springer, Singapore, 2021.
25. Latchoumi, T.P., Parthiban, L. Quasi Oppositional Dragonfly Algorithm for Load Balancing in Cloud Computing Environment. Wireless Pers Commun 122, 2639–2656 (2022). https://doi.org/10.1007/s11277-021-09022-w.

26. NIST cloud computing reference architecture, Special Publication 500-292, http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505.

Figures

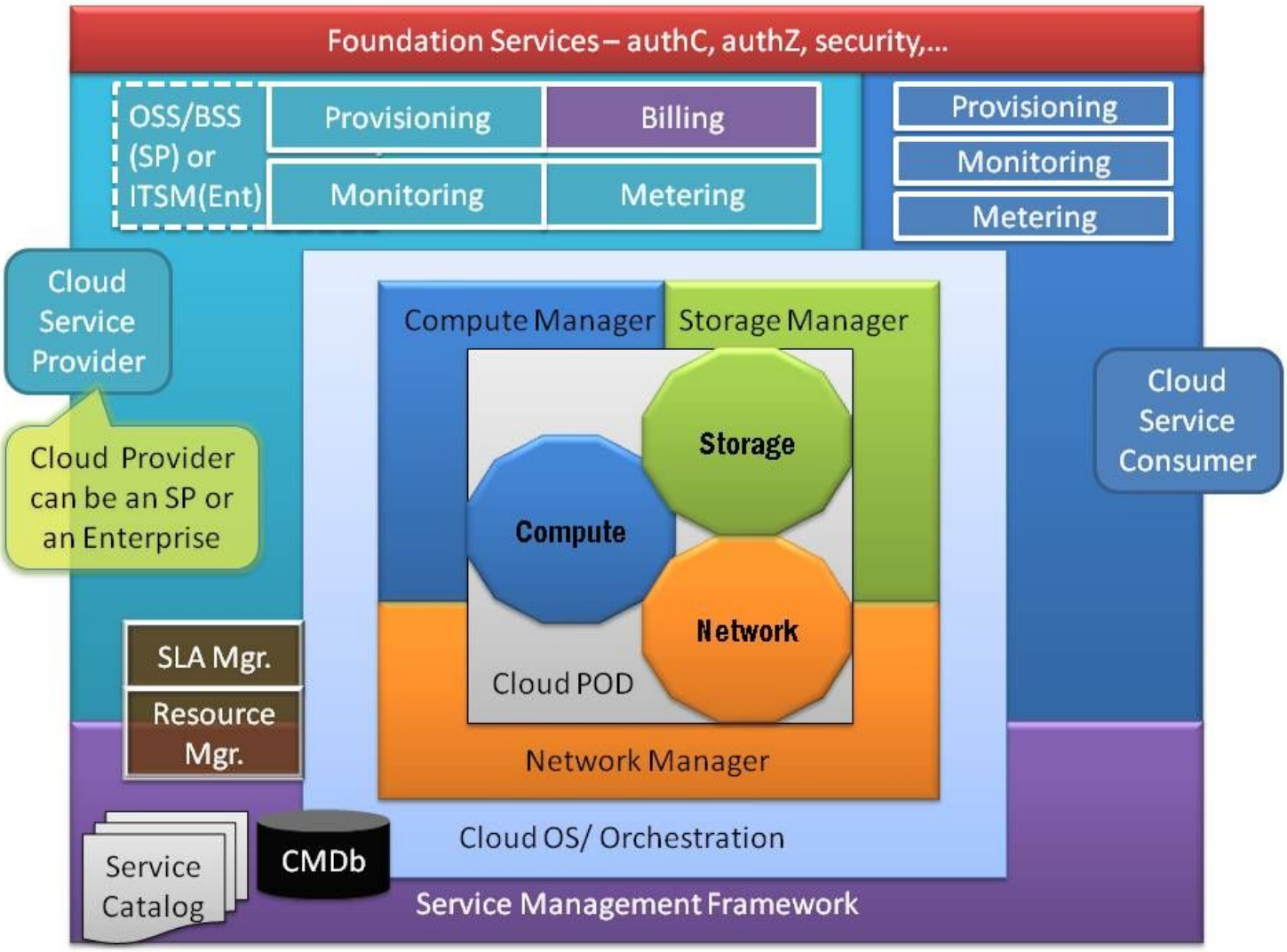

Figure 1

Structure of Cloud based storage adopted from [26]

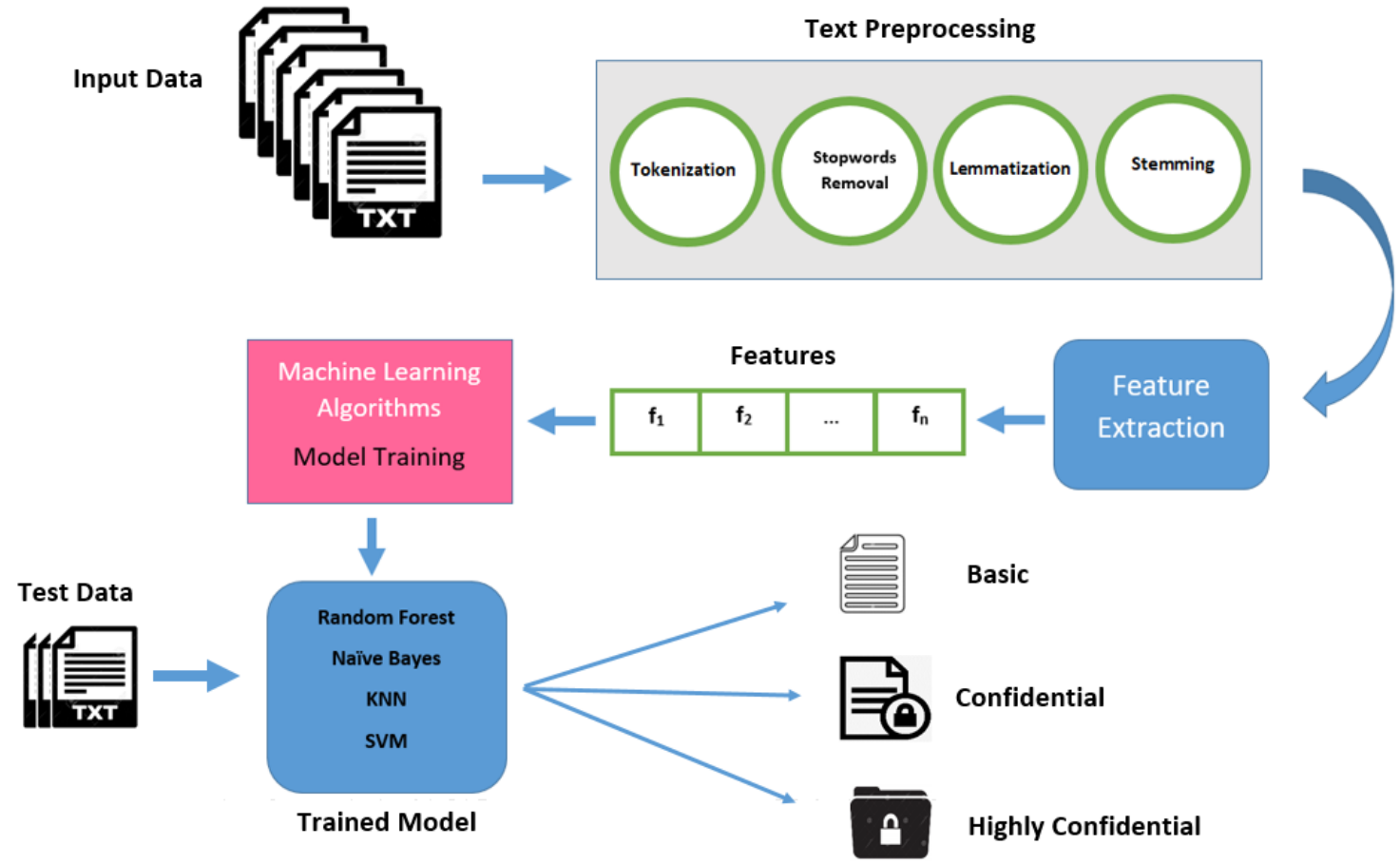

Figure 2

Proposed Framework for Automatic Data Classification for Cloud

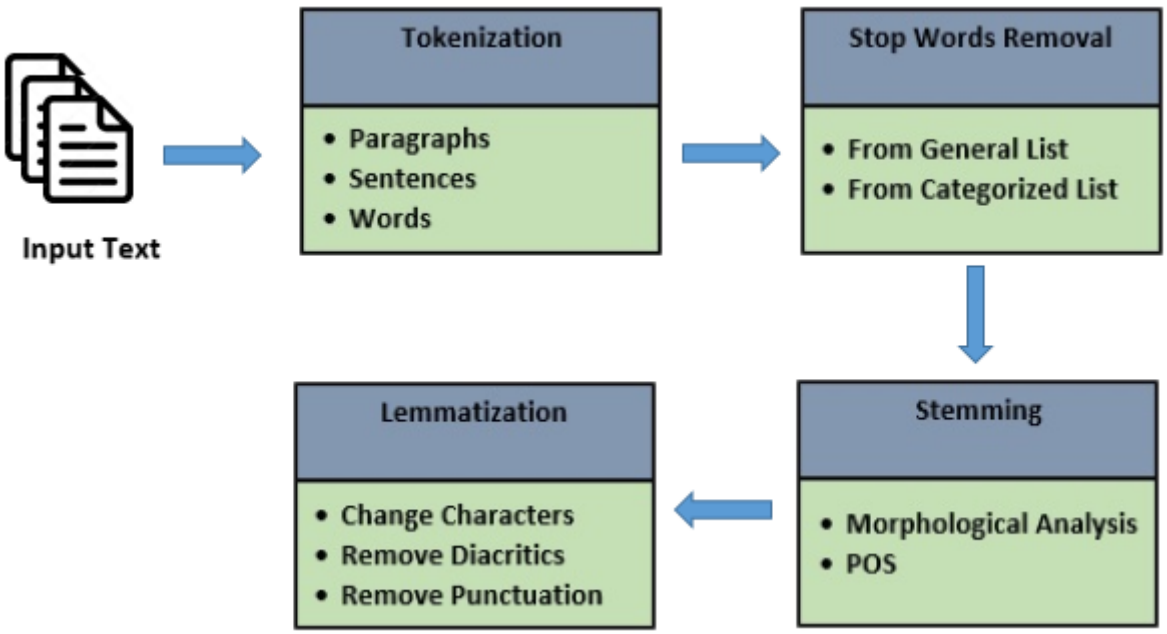

Figure 3

Data Preprocessing

abort	accept	access	account	action	adapt	address	label
0	0	0.001472	0.005158	0.077632	0.000716	0.004422	basic class
0	0.006418	0.026382	0.01284	0.00561	0	0.009908	basic class
0.106395	0	0	0	0.017273	0	0	basic class
0	0	0	0	0	0	0	basic class
0.151239	0	0	0	0.038195	0	0	basic class
0	0	0.104648	0	0	0	0	basic class
0	0	0.007775	0	0	0	0.013139	basic class
0	0	0	0	0	0	0.047921	basic class
0	0	0.093773	0	0	0	0	confidential
0.00276	0	0.004214	0.011484	0.001792	0	0.00633	basic class
0	0	0.040269	0	0	0	0	confidential
0	0	0.007428	0	0	0	0	confidential
0	0	0.014323	0	0	0	0	confidential
0	0	0.006674	0	0	0	0	basic class

Figure 4

Feature Vector

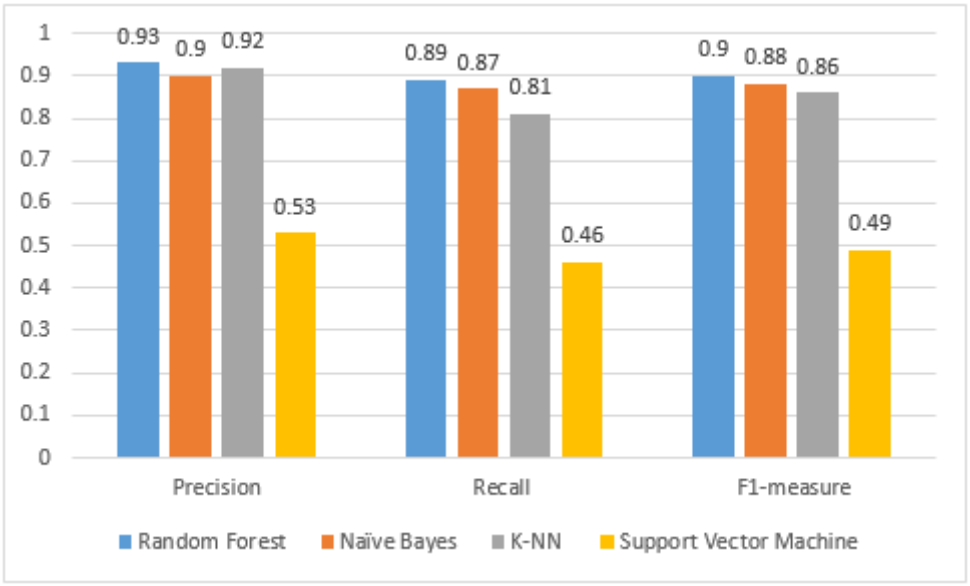

Figure 5

Performance Evaluation

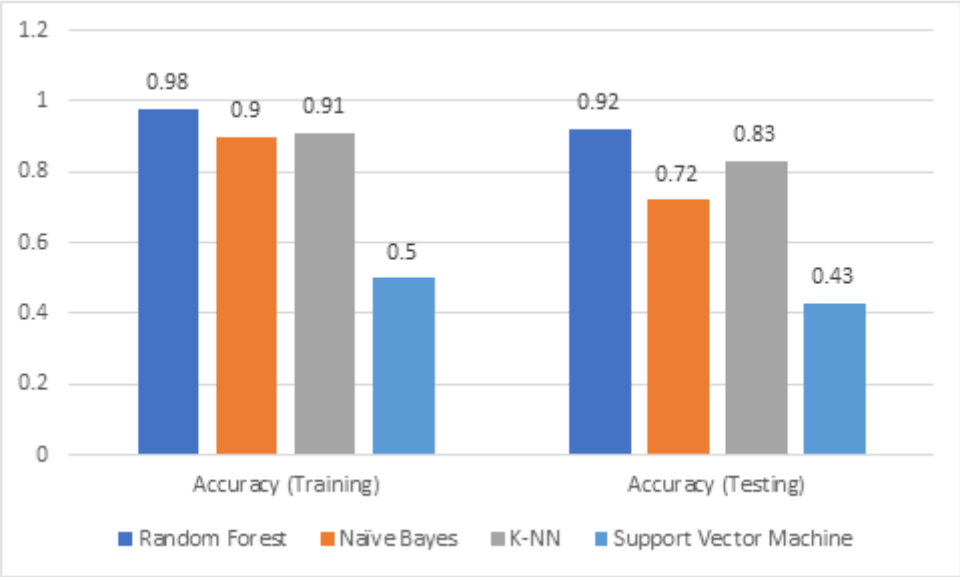

Figure 6

Accuracy of the proposed Classifiers