Securing Cloud Data: An Approach for Cloud Computing Data Categorization Based on Machine Learning

Fahad Burhan Ahmad[1*], Azaz Ahmed Kiani[2], Yaser Hafeez[1], Hamza Imran[1], Muhammad Habib[1], Asif Nawaz[1], Muhammad Rizwan Rashid Rana[1], Muhammad Azhar[1]

[1]University Institute of Information Technology, PMAS Arid Agriculture University, Rawalpindi, Pakistan

[2]Department of Computer Science, National University of Modern Languages, Rawalpindi, Pakistan

***Corresponding author:** fahad.burhan@uaar.edu.pk

Citation | Ahmad. B. A, Kiani. A. A, Hafeez. Y, Imran. H, Habib. M, Nawaz. A, Rana. M. R. R, Azhar. M, "Securing Cloud Data: An Approach for Cloud Computing Data Categorization Based on Machine Learning", IJIST, Vol. 7 Issue. 1 pp 235-258, Feb 2025

Received | Jan 04, 2025 **Revised |** Feb 02, 2025 **Accepted |** Feb 03, 2025 **Published |** Feb 05, 2025.

Introduction/Importance of Study: A novel innovative technique known methodical approach is referring as cloud computing (CC), which allows users to store data on remote servers that are accessible through the internet. This method makes it simple to move and retrieve vital and personal data storage. As a result, the demand for it is rising daily. This can be used to store a variety of data, including multimedia content, paperwork-based files, and financial transactions. Furthermore, by lowering operating and maintenance expenses, CC lessens the reliance of the services on local storage.

Novelty statement: Current systems apply only one key size with which all data is encrypted without concerning the level of privacy of the data. This results in higher processing costs and longer processing times. Furthermore, none of these methods improves secrecy and only achieves a low accuracy rate in data classification.

Material and Method: This study presents a cloud computing strategy for data sensitivity that is based on automated data classification. The model suggested in this study utilizes Random Forest (RF), Naïve Bayes (NB), k-nearest neighbor (KNN), and support vector machine (SVM) classifiers to achieve automated feature extraction. This methodology is designed to operate effectively across three sensitivity levels: basic, confidential, and highly confidential.

Results and Discussion: The experiments were performed on the Reuters-21578 dataset, which consists of 21,578 documents. The simulation results demonstrated that the three proposed models achieved accuracy rates of 97%, 96%, and 95%, respectively. These findings indicate that SVM, RF, and KNN outperform NB in classification performance.

Concluding Remarks: Additionally, the suggested study offers helpful recommendations for researchers and cloud service providers (like Dropbox and Google Drive).

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

Introduction:

Cloud Computing (CC) is the need of this age that consists of several high-tech applications that allow users to safely store, retrieve, and store their documents [1][2]. Android phones, laptops, and other mobile devices can be used to access cloud services and applications. Regarding document security, cloud computing is considered the most trustworthy and safe platform. However, because they lack battery life, performance, and storage capacity, other storage devices like laptops and mobile phones might not be able to provide such a safe platform for data storage [3][4]. It is common practice to store and backup arbitrary data using cloud storage services since they are affordable, easy to use, and quickly accessible [3]. Additionally, they offer the convenience of data sharing and device syncing. Various cloud storage systems employ diverse architecture schemes, lacking a singular standard set of attributes. Yet, the utilization of cloud storage services commonly entails. Consolidating hundreds of storage devices into clusters. These are Interconnected through distributed file systems, middleware software storage, and computer networks. Services such as distributed file systems, storage resource pools, service interfaces, and service level agreements (SLAs) are all components that are incorporated in cloud storage operations.

Cloud storage solutions aim to deliver multi-tenant on-demand storage that is enormously expandable. It is a typical characteristic of cloud storage architectures to have a front end with API for storage access; the SCSI protocol in the conventional storage systems. However, these APIs are becoming more and more prevalent in cloud computing. These APIs comprise file service front ends, web service front ends, and conventional interfaces like internet SCSI (iSCSI). Data reduction and replication are two services that are made easier to deliver by the middleware's storage logic layer, which sits behind the front end. The handling of the physical storage of the data is the responsibility of the back end of the cloud storage structure. This back end might be an internet protocol that performs particular activities or it could be a standard back end for physical drives. The five primary components of the cloud concept are resource pooling, rapid elasticity, quantified service, on-demand self-service, and wide network connectivity. The National Institute of Standards and Technology (NIST) divides cloud services into three service models: Cloud deployment models are of four category known as communal, hybrid, private, and public Cloud. There are three primary forms of this; Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [3][5].

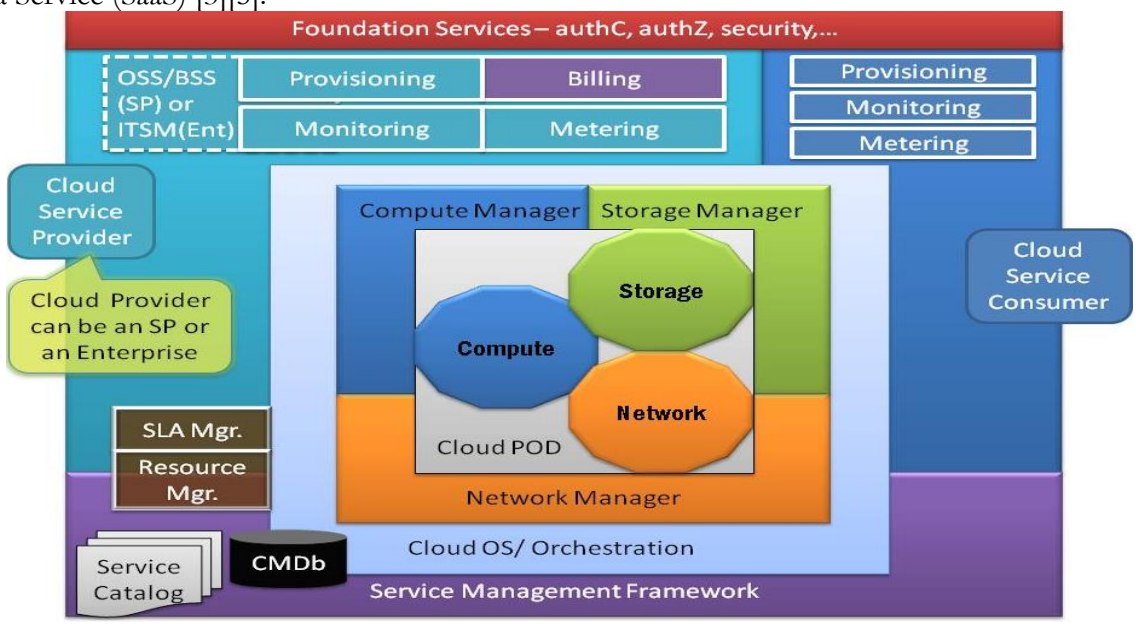

Figure 1. Cloud-based storage structure adapted from [6]

Figure 1 displays the cloud computing standard architecture developed by NIST in high detail. Five top performers Cloud customer, cloud provider, cloud carrier, cloud auditor, and cloud broker are among those that are specified by the architecture. In CC, an actor is an individual or an entity that engages in a procedure, a transaction, and/or carries out tasks. An overview of the players included in the NIST cloud computing standard architecture may be seen in Table 1 [4]. In this instance, security is a feature of the cloud provider.

Table 1. Actors in the reference architecture for NIST Cloud Computing obtained from [7]

Actor	Definition
Cloud Consumer	An individual or organization that enters into agreements with cloud providers and utilizes their services.
Cloud Provider	A person, group, or other entity responsible for providing a service to potential customers.
Cloud Auditor	A third party can independently assess cloud services, information system operations, security, and performance.
Cloud Broker	A business that mediates contracts between clients and cloud providers and oversees the use, functionality, and delivery of cloud services.
Cloud Carrier	A middleman that connects cloud service providers and consumers.

The cloud stores data in an arbitrary manner. The user finds it harder to search through data as it grows in volume. On the other hand, the user will find it easier to access the necessary data if it is organized and stored. As a result, a model that makes it simple to store data in an organized cloud format must be created. The following is a list of advantages of classification: Uses compound word search capabilities to ensure accurate classification with fewer false positives.

Indexed features enable sensitive phrase searches without the need for data storage re-crawling.

Incorporates a scalable taxonomy manager that allows classification parameters to be customized.

Offers automated processes for things like moving private information from open-access sharing sites.

Supports a wide range of content sources, including on-premises and cloud-based, and handles both structured and unstructured data.

They fail to upload their sensitive and confidential files on the online storage because they feel that the provider might exploit them. They are also concerned that because the best cloud storage attacks are intensifying now and then, their information can be hacked and get into the wrong hands [5]. Current cloud-based architectures do not take into account the degree of secrecy of the data and encrypt all data using the same key length, which may not be viable. Processing is slowed and additional overhead is created when low and high secret data are treated equally.

According to the facts provided above, this study focuses on three essential aspects of cloud computing: a high degree of accuracy, automatic categorization, and data sensitivity. Based on such an idea, we present a working system, which, before any transmission or storage operations happen, utilizes machine learning methods to classify the data and ensure the confidentiality and integrity of cloud storage [8]. However, in cloud contexts, data confidentiality is particularly crucial. While maintaining a high degree of accuracy, this framework will also lessen the need for manual classification operations.

The proposed model is sub-partitioned again into Basic, Confidential, and Highly Confidential classes depending on the sensitivity degree of text information. These classes are depicted in the Figure 2 below. The suggested work is broken down into three phases:

preprocessing text datasets, training the model based on features, and creating three classes of text data depending on data sensitivity in the last stage. Features are extracted using the Python Sk Learn Library. We use a variety of classifiers to obtain text categorization accuracy [9].

Classifying information makes it easier to find security guidelines and regulations that are suitable for safeguarding that information. The two kinds of data are personal and non-exclusive (non-distinct) comprehension kinds. The content is categorized using the qualities of the data. Information deemed sensitive is labeled as "confidential" or "highly confidential," and it is identified from other information by being referred to as "basic." SVM, NB, KNN, and RF algorithms are used in the suggested cloud data classification framework for improved performance, which results in high-level accuracy and automatic categorization. This is a summary of the study's contribution:

In this paper, Basic Class, Confidential Class, and Highly Confidential Class are presented as three new classifications for the text data sensitivity level based on the powerful classification model. This classification scheme facilitates the implementation of tailored security measures based on the sensitivity of the data.

Using SVM, NB, KNN, and RF classification algorithms to discover the most effective technique for accurate automatic classification. This comparative study improves the choice of the optimal algorithms for the intended objective.

Focusing on critical features of cloud computing, with a special emphasis on sensitivity level, automated classification, and high accuracy. During data transmission and storage in the cloud, this framework prioritizes data confidentiality and integrity.

Implementing an effective method to reduce manual work in the classification process, resulting in a significant boost in accuracy rates. This enhancement positively contributes to the improvement of protective measures for the data stored in cloud storage.

This paper's remainder is organized as follows: Section 2 describes the related work. Section 3 presents the proposed work. Section 4 discusses the experiment's specifics, while Section 5 presents the findings and next steps.

Literature Review:

Security was enhanced in [10] for digital signatures and the secretive aspect of security, RSA was used. There are five steps involved in the encryption process. The creation of keys is the initial step. In step 2, a digital signature is used. Steps 3 and 4 then involve the encryption and decryption processes. Verifying the signature is step five. [11][12] suggested an architecture to protect the privacy of data kept in the cloud by combining the Advanced Encryption Standard (AES) encryption method with digital signatures and the sharing of Diffie Hellman keys. Because the Diffie-Hellman key exchange protocol uses the user's private key, which is only accessible by authorized users, it eliminates the requirement for a compromised key even if it is compromised in transit [13][14]. The architecture's three-way method, of securing cloud-stored data, presents formidable challenges for hackers attempting to bypass the security mechanism.

In [15], Sinha N et al. provide a thorough analysis of cloud computing, including its advantages, design, use, and possible disadvantages. The discussion encompasses diverse data types, security concerns, and performance challenges within the cloud. Furthermore, [16][17][18] explore and evaluate a range of cryptographic techniques aimed at enhancing data security. Several cryptographic methods are compared using a variety of factors, including features, block size, and key length type. Numerous cryptographic methods that can be applied to guarantee cloud data security were investigated in this work [19].

Using the KNN approach, data classification and confidentiality are ensured. Security is the main objective of data classification. They employ the KNN technique to partition the data into sensitive and non-sensitive as indicated by [20][21]. Sensitive data is secured using encryption. The main justification for categorization is that, depending on the demands of the

material, it facilitates the selection of a suitable security level. Consequently, security will be enhanced. [22] Another crucial method for enhancing cloud data security was provided in this research. After several factors are selected for data classification, the categorized data is encrypted. Among the classification-related factors that are taken into account are storage, content, and access control. To boost efficiency and security, data is classified based on these attributes and then encrypted.

The proliferation of Internet applications and the resulting vast increase in online texts have paved the way for improved automated text mining classifiers, which play an essential role in one of the primary functions of natural language processing (NLP): text classification [23][24]. These classifiers can automatically organize and categorize documents, thereby enhancing the efficiency of various text-processing tasks. An automatic text classifier has been created using a variety of machine learning algorithms that were trained on a set of classified training texts [25][26][27]. Many additional languages, including Urdu, English, French, and Chinese, have models for text categorization developed [28][6][29]. Using classification techniques, the SVM is a supervised machine-learning model created to handle two-group classification issues [30].

K-NN and Enhanced Naïve Bayes techniques have been implemented for data identification and confidentiality [31][32]. The main goal is to offer adequate security for confidential information. The K-NN and Enhanced Naïve Bayes classifications have been used where data is being processed to discern the differences between sensitive and non-sensitive labeling. Confidential information is kept secure via encryption. Selecting an appropriate protection strategy was straightforward in light of the requirement for data classification [33]. By utilizing enhanced Naïve Bayes, they achieve 72% accuracy. Thus, protection can be annealed in this manner.

For the classification of financial data, the authors [34] proposed a hybrid strategy that combines Convolutional Neural Networks (CNN) with (SVM). A dataset of Chinese financial papers was used to evaluate this approach, and it obtained a high classification accuracy of 94.2%. Their research highlights the effectiveness of combining machine learning techniques with deep learning architectures to enhance document classification performance. A method for classifying financial data that combines SVM and Decision Tree algorithms was published in [35]. A collection of Indian financial records was used to evaluate their technology, and the accuracy rate was 91%. This study demonstrates how incorporating conventional machine learning algorithms can produce reliable categorization outcomes in the finance industry.

A hybrid strategy combining SVM and Long Short-Term Memory (LSTM) networks was developed for the categorization of financial documents [36][37]. When evaluated on a dataset of US financial documents, this method produced a 91% accuracy rate. Their work emphasizes the importance of incorporating sequential learning models like LSTM to capture temporal dependencies in financial texts, improving classification outcomes. SVM, Naive Bayes, and a rule-based classifier were used by the authors [38] to present a machine learning-based method for the automatic classification of sensitive financial documents. To train the models, they extracted appropriate features from the documents, and they analyzed the models' performance using execution time, accuracy, and F-Score. The results indicated that all methods performed satisfactorily, with Naive Bayes outperforming the others, achieving a 95% accuracy in classifying financial documents while also minimizing manual handling time for misclassification. Furthermore, a tolerance-leveled version of the Naive Bayes model demonstrated marginally reduced accuracy but increased precision. The study emphasized the potential of this automated approach to improve efficiency and reduce errors in the financial industry, where manual classification processes are traditionally time-consuming and prone to inaccuracies.

At every node or branch of a decision tree, a set of tests is defined to recursively divide the training dataset into smaller subsets [39]. Every node in the tree symbolizes a feature test from the training dataset, and every branch that descends from the node has a single value that corresponds with it. Testing each feature after moving on to the root node is how the dataset gets categorized. Next, moving down the tree branch by the feature's value in the provided dataset; this process is then repeated recursively [40].

Based on the discussion above, it can be determined that the majority of currently in-use work may have data confidentiality levels, encrypting all data with the same key size and adding time and expense to the processing process. Moreover, none of these manual classification techniques improve security and only classify data with a poor accuracy rate. The suggested method seeks to lessen cloud computing's need for manual classification. The second specification has been approved to obtain the outcomes on different classifiers and compare it with our suggested method which automatically performs classification using machine learning technologies in three different levels with a high degree of certainty and provides the necessary level of familiarity for the data. Which level of confidentiality Basic, Confidential, or Highly Confidential to choose depends on how sensitive the data is. A substantial degree of automatic data classification is achieved by the application of four machine learning techniques.

Novelty and Objectives:

This research introduces a novel approach to cloud data security through an intelligent classification system that automatically categorizes data based on sensitivity levels while maintaining optimal performance. The primary novelty lies in the integration of machine learning algorithms (SVM, NB, KNN, and RF) to create a three-tiered classification system (Basic, Confidential, and Highly Confidential) that determines appropriate security measures for different data types, thereby addressing the critical challenge of uniform encryption in current cloud architectures. The main objectives of this study are:

- To develop an automated classification framework that reduces manual intervention while maintaining high accuracy
- To implement sensitivity-based security measures that optimize resource utilization by applying appropriate encryption levels
- To create a scalable system that can handle both structured and unstructured data from various sources
- To enhance data retrieval efficiency through indexed searching and compound word capabilities, ultimately providing a more secure and efficient cloud storage solution for organizations dealing with varying levels of data sensitivity.

Proposed Work:

This study examined the effects of several machine learning methods for classifying data, including the NB, RF, SVM, and KNN algorithms. Sensitivity levels for the cloud are used to categorize data. The basic methodology is shown in Figure 2. Our suggested model, which is depicted in Figure 3, has three classes: This includes; Basic Class, Confidential Class, and Highly Confidential Class.

Basic Class:

The basic class of our proposed model consists of low-secrecy common data types, like text documents. Text documents contain basic information like notices, announcements, and advertisements. This level thus offers a minimal degree of data security. The basic class will be encrypted using the backup service's key on the server side before transmission, even if it doesn't need to be encrypted on the client side.

Confidential Class: In this session, personal files, web, private, and business accounts are covered. Our secret class is on data with a medium level of security. Since this class tracks

confidential and secret data, security is required to safeguard our data. Encryption techniques like AES [41] can be used for this at the confidential level. We will use client-side encryption in this class.

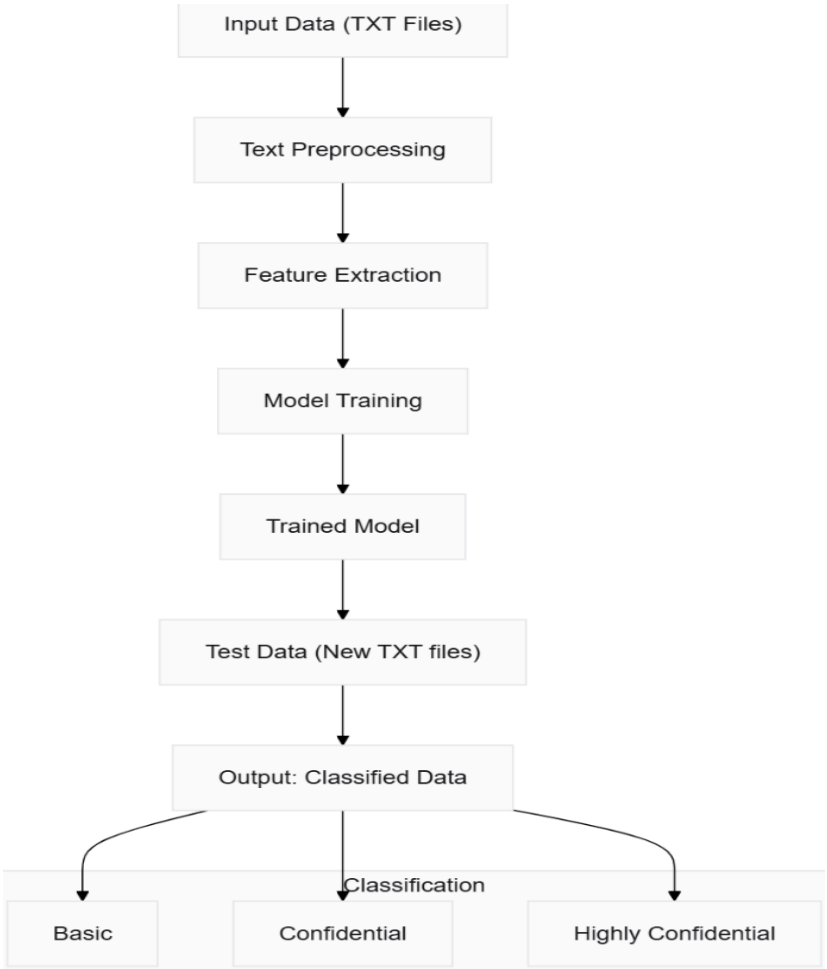

Figure 2: Basic Methodology

Highly Confidential Class:

This class involves financial transactions, any document that is restricted to circulation within the specific organization, and military information. Users may steer clear of all newly offered services due to concerns about the extreme confidentiality of the data. Because the level of confidentiality and integrity is so high, two standard recommended algorithms will be used to provide security. AES 256 was advised by the US National Security Agency [42][43] to guard against unauthorized access to top-secret information (NSA). Conversely, data integrity is guaranteed by the SHA-2 algorithm [44][45]. The hash value of the data will be determined using this algorithm before any modifications or transfers. Create a hash value as well for user-requested data retrieval; the value needs to match to guarantee that the data hasn't been altered.

Dataset:

As our source of data, we obtained the text categorization collection dataset from the Reuters-21578 [46] from the UCI ML repository. The Reuters-21578 dataset is classified into three categories: Basic, Confidential, and Highly Confidential, each containing 972 documents. After preprocessing, the dataset was split into 2332 training samples and 584 test samples. Data used and compared in the current study originate from the [UCI] repositories (the access links are provided in Table 2).

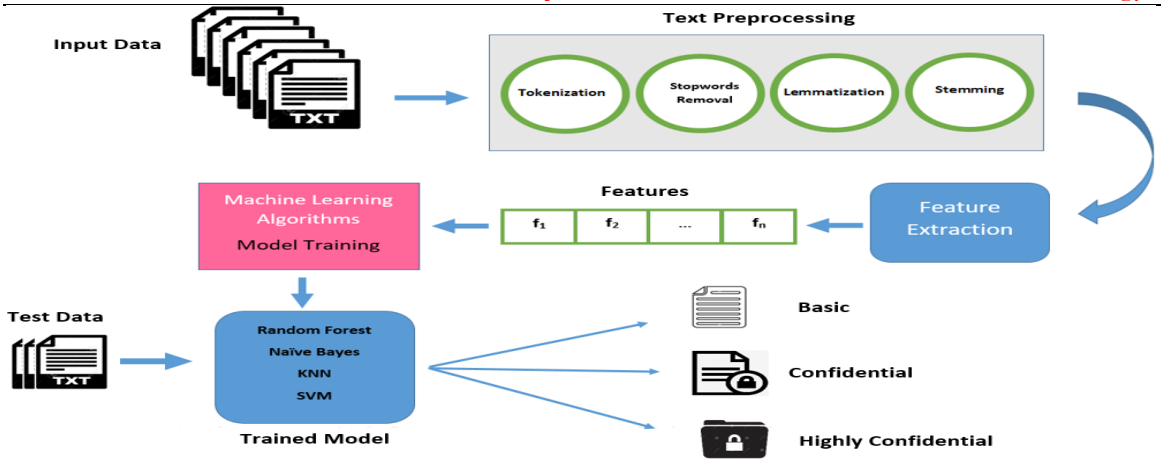

Figure 3. Proposed Automatic Data Classification Framework for Cloud Computing

Table 2. Dataset and matching repositories

S. No	Nature of Dataset	Size of Dataset	Type of Data	Web Link
1	Reuters-21578 Public, Confidential, and Highly Confidential Data	21578 Basic: 972 Confidential: 972 Highly Confidential: 972	Text Documents	https://miguelmalvarez.com/2015/03/20/classifying-reuters-21578-collection-with-python-representing-the-data/

The Reuters-21578 dataset has several limitations that can impact classification performance. One major issue is class imbalance, where certain categories have significantly more samples than others, leading to biased model predictions. This imbalance can cause classifiers to favor dominant classes while underperforming minority classes. Moreover, overlapping class labels can introduce ambiguity, making it difficult to distinguish between closely related categories.

Data Processing:

Natural language processor can be defined as the systematic study and analysis of natural language for making changes and supplying a meaning that computers can understand. The NLTK library is utilized to preprocess the text input before feeding it into the algorithm. There are various preprocessing steps which include this where information and specifically text data is converted from an unstructured format to a structured one. Processing serves as a pivotal component across numerous machine-learning techniques. It also has a discernible effect on the classification procedure [47]. The preparation algorithm for text documents is given below.

Algorithm 1: Text Preprocessing

Input:

W={w1,w2,w3,…,wr}, List of Words: where r is the word count
SW= {sw1, sw2,sw3,…, swm}, Stop Words List: which contains stop words.
U={u1,u2,u3,…, ur} Regular Expression List: to apply regular expressions on W.
S: New word list.
Procedure:
For each ni in W do
For each SJW in SW do
If ni is not in SJW, then
Append ni to S.
Else
continue to the next SJW.

End For
End For

Output:

S, New word list containing words from W excluding stop words.

Tokenization:

Tokenization pertains to the process of segmenting a sequence of characters where one segment represents a word or a phrase. In natural language processing, tokenization comes in two forms: of those, the most common processes include word tokenization and sentence tokenization [48][49]. After tokenization, the input undergoes processing using the resulting list of tokens, which can be either individual words or phrases [50][51][52][53][54]. Figure 4 illustrates how tokenization works.

In the above sentence tokens are as follows:

{“It” “was” “sent” “to” “the” “companies” “as” “a” “confidential” “document”}.

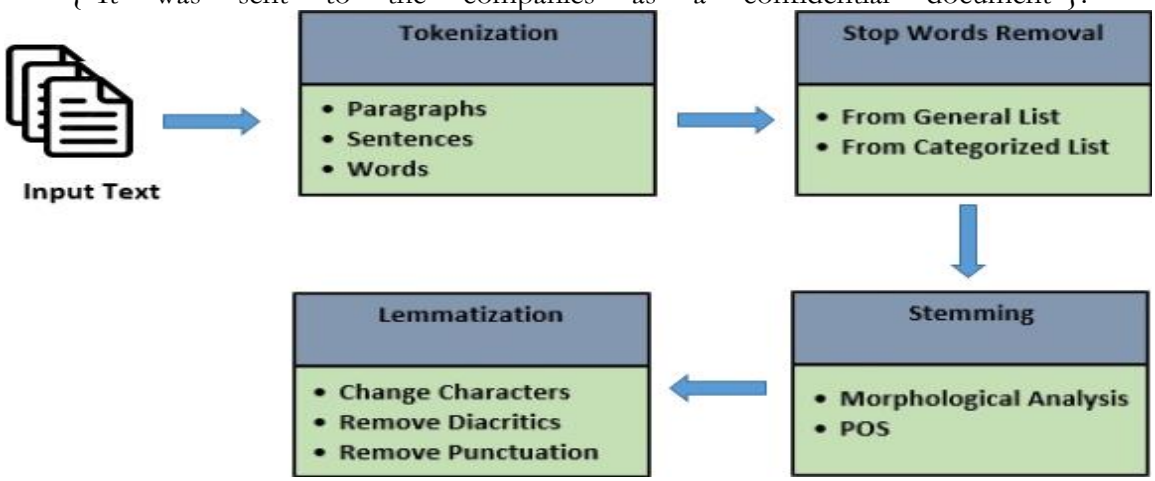

Figure 4. Data Preprocessing

Filtering:

It's usual practice to filter a text file to exclude some of the less important terms. The removal of words is prevented via a reciprocal filtering technique. Stop words are phrases that frequently occur in writing that is devoid of important details, such as {"It" "was" "to" "the" and "a"}) [55]. Consequently, the frequently used words in the content are considered irrelevant and can be excluded from the content document while the phrases that frequently appear in the content document might similarly provide insufficient information to distinguish between various reports.

Lemmatization:

The study takes into account the words' feature extraction. For example, it is possible to remove the various forms of a word and leave only one element remaining. In other words, lemmatization techniques are to deliver different tenses and components within a single, for example, complex structure. As for the lemmatizing approach, it begins with the step to determine the actual part of speech of a specific word in the given document. Since POS is repetitious and prone to inaccuracy, stemming techniques are preferable [56].

Stemming:

This method reduces words to their most basic forms, for example, by assigning a common stem to a set of words, even if the stem isn't a recognized term in the language [57]. Consequently, non-words may be derived when stemming a word or a sentence. Stem is the form which obtained after erasing all the prefixes and suffixes from the word. Words like "Continue," "Continued," and "Continuous" are changed to "Continue" [58][59]. Language-specific stemming algorithms exist.

Feature extraction:

Bag-of-words representation is the most straightforward and well-known approach [8]. Text is converted into fixed-length vectors by this algorithm. This can be accomplished by calculating the word's frequency of occurrence in a document. Find the words x and y for whose frequency (x and y in the same document) using equation 1 below. The product of x and y frequencies.

$$pmi(x;y) \equiv log\ \frac{p(x,\ y)}{p(x)p(y)} \quad \ldots \quad (1)$$

Before being processed by the classifier, the text document data is indexed. Features are typically described using words. The Bag of Words approach, a common method, represents the content as a word cloud. To facilitate formal descriptions of feature extraction, it's necessary to define several concepts and variables. The vocabulary, denoted as V = w1, w2, ..., wv, exists if a set of papers D = d1, d2, ..., dD contains unique words or phrases [60]. The total number of documents is denoted by D and 'D' represents of total number of documents containing this word 'w' while the total number of occurrences of the term 'w' in the document is denoted by 'fd(w)'. It is possible to determine the feature vector of the specific document t as (fd(w1), fd(w2),…,fd(wv)).

There are two general approaches to employing a feature list to represent a document: both the global dictionary approach and the local dictionary technique [60][61]. The construction of the worldwide dictionary will only employ pertinent materials. As a result, a term can be included in the feature list of a lexicon if it is used in the relevant text. In terms of outcomes, the local dictionary method can yield superior outcomes [62].

Feature Vector:

Documents are typically represented by converting them into numerical vectors. An alternative name for this demonstration is the "Vector Space Model." In contrast, its design is uncomplicated and specifically crafted for indexing and information retrieval (IR) purposes. The widely used vector space model, which allows for analyzing vast text data in detail, is actively integrated into various text mining techniques and IR classifications [62].

Words in VSM are assigned unique numbers that represent their relative weights, or "importance," within the text. The Boolean model is the first of the two fundamental feature weight models. Features that are included in the document are given a weight of 1, and those that are not get a weight of 0. The second approach is known as term frequency and inverse document frequency (TF-IDF) and is said to be the most encapsulating of all term weights methods [63]. This expression comes from the IR which measures the importance of an attribute (IDF) with the help of TF as well as IDF [62][63][64]. TF represents the total count of a feature's occurrences within the document, while IDF reflects its frequency or rarity across all documents. The TF weighting method is exemplified as a means to determine the weight assigned to each word w within the document.

$$tf_{idf(t,d)} = tf(t,d) * \mathrm{idf(t)} \quad \ldots \quad (2)$$

Where t stands the number of times a term appears in document d.

$$tf(t,d) = \frac{\text{number of time term (t) } appears\ in\ document\ (d)}{total\ number\ of\ terms\ in\ documents} \quad \ldots \quad (3)$$

To determine the t term for the inverse document frequency, one can follow this formula, with N representing the total number of documents in a collection:

$$idf(t) = log\ \frac{\mathrm{N}}{df(t)+1} \quad \ldots \quad (4)$$

In the TF-IDF calculation the documents in the collection are represented by |D|. Here the direct word frequency is divided by the IDF [65]. This normalization step seeks to minimize the influence of words that occur more frequently across the document collection.

By ensuring that less common traits in the collection have a more significant impact on the texts, this normalization helps maintain balance and accuracy in the computation.

Sensitivity Base Classification:

One technique of supervised learning is classification. To learn from the training data and predict the class label on new data, which classifier should be applied? It is utilized in several fields, including text classification, image processing, document management, and medical diagnostics. Various communities, such as machine learning, database, IR, and data mining, also take it into account [66].

Assigning specific classifications to text documents is the main objective of classification [39]. The challenge of classification is defined clearly in the following way. A set of training documents To assign each document di to the label ℓi drawn from a set of labels L= {ℓ1, ℓ2… ℓk } and set of documents D={d1,d2… dn}.

To classify this article, many precise machine-learning approaches were applied. These techniques include rule-based classifiers, support vector machines (SVM), k-nearest neighbors (KNN), k-nearest trees (KNN), Naïve Bayes (NB), and artificial neural networks. The classifier [39] believes that the document's classification below is more appropriate.

All that's needed for a random forest is a sequence of decision trees and the total sum of their outcomes to get a single final result. They may reduce both bias-related inaccuracy and overfitting, which is why they work so well. In essence, a decision tree is a training dataset's classified tree, where each feature value condition is used to segregate the data hierarchically [67][68].

Training Module:

In this research, the training module focuses on utilizing the Reuters-21578 dataset for a text classification task aimed at categorizing news articles into three sensitivity levels: basic, confidential, and highly confidential. First, the dataset is loaded using the Natural Language Toolkit (NLTK), where each document in the dataset is associated with one or more topics. Based on the predefined categories, these topics are mapped to sensitivity labels. For example, general news items like sports and agriculture fall under "basic," while finance and trade are labeled as "confidential," and military or security-related topics are categorized as "highly confidential." After assigning these labels, text preprocessing is conducted to prepare the data for machine learning models [9][69]. This will include, converting all documents to lowercase, stripping each document of all non-alphabetic characters, and omitting all stop words to minimize noise. To make the preprocessed text data in a form that can be processed by machine learning algorithms, the data is encoded using TF-IDF (Term Frequency-Inverse Document Frequency). The dataset is again divided into training and testing sets to provide each model with a sufficient amount of data to make it learn and test. Training is done with different algorithms such as K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machines (SVM), and Random Forests, all models fit on the TF-IDF vectorized training corpus [69]. During this training phase, the models learn patterns associated with each sensitivity category, adjusting their parameters to maximize classification accuracy. The training module also measures accuracy, precision, recall, F1-score ROC, and AUC to measure the models on how well they categorize between sensitivity levels.

Testing Module:

The testing module is crucial for assessing how well the trained models function on data that hasn't been seen yet and making sure they generalize well outside of the training set. After training, each model is applied to the test set, generating predictions for each document's sensitivity level. These predictions are then compared against the actual sensitivity labels to assess model accuracy. Accuracy for each model is calculated to compare the precision, recall, and F1 score so that the precision of each category can be known. Confusion matrices are also generated, providing insights into the models' error distribution across categories. For

example, if a model misclassifies "highly confidential" documents as "confidential," it indicates potential areas for refinement. The accuracies increase when plotting with ROC and AUC as axes for creating true positive /false positive trade-offs to assess each model's ability to differentiate between different sensitivity levels. This testing module allows us to compare model performances, providing a quantitative basis to select the best-performing classifier.

Development Prototype:

The development prototype integrates the training and testing modules [9], creating a streamlined pipeline for text categorization based on sensitivity. This prototype begins by loading and preprocessing the Reuters-21578 dataset, applying text-cleaning techniques to each document, and vectorizing the data for model compatibility. It includes multiple classifiers KNN, Naive Bayes, SVM, and Random Forest allowing for a robust comparison. The prototype is oriented to automate the model training and testing, storing the performance metrics of each model: accuracy, precision, recall, F1-score, and ROC-AUC. For visualization, the prototype includes functions to generate combined graphs, comparing metrics like accuracy across models, and displaying precision, recall, and F1-score in a single bar chart. Additionally, ROC and AUC curves are plotted together to allow side-by-side performance comparison across models. This user-friendly setup not only provides immediate performance feedback but also supports further experimentation, enabling model fine-tuning based on specific evaluation criteria. The prototype offers a comprehensive approach to classifying news articles by sensitivity, providing detailed insights through metrics and visualizations, which are crucial for understanding model behavior and ensuring reliable classification in real-world applications.

Experiment:

In this section, the proposed approach for automatic data classification is assessed based on a set of experiments. Four different classifier types SVM, NB, KNN, and RF are employed in these experiments to achieve accuracy on a specific dataset. The UCI ML repository provided the text classification data used in the testing, which was referred to as Reuters-21578 [45]. Let's create a few phrases and variables that will be used often to help with formal descriptions of feature extraction. The vocabulary V = {w1, w2, …,wv} ranges through a variety of words or a phrase passed in a set of documents D = {d1, d2, …, dD}. In dD, the terms wV, and w is contained as, fD, and fd(w) compared to, how many documents contain the word w (w). The general form for the document t feature vector is given by tD = [fd(w1), fd(w2), …, fd(wv)] Third, depending on the provided subset of features a feature vector is developed and some of the features are given weight.

Table 4. Feature Vector

abort	secret	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	0.56273	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	0.745226	0	0	0	**confidential**
0	0.624140	0.007428	0	0	0	0	**Highly Confidential**

0	0	0.014323	0	0	0	0	**confidential**
0	0	0.006674	0	0	0	0	**basic class**

Unlike TF and IDF individually, 'TF-IDF' is calculated and has a value when the feature is in the document, otherwise, it has 0 in case the feature is absent from the document. A characteristic's worth in these kinds of texts is determined by its weight. Using the recommended architecture displayed in Table 4, each retrieved feature is converted into a feature vector. Every characteristic has a weight assigned to it. A few feature vectors for the automatically acquired features are shown in the term matrix in Table 4. Documents collection organization is described using a documents term matrix. Each document is represented like a row and every element of the matrix corresponds to some unique phrase or feature.

Model Evaluation Metrics:

The evaluation of the classification model's performance is conducted using an objective measure. A random portion of the test batch containing classified documents is reserved for this purpose. Following the training of the classifier on a labeled dataset, the test set is organized, true and predicted labels are paired, and the quantitative performance of the classifier is assessed. The ratio of the number of documents accurately classified to the total number of records is called accuracy. Three widely used objective evaluation or qualitative measures for categorization are F-measure, accuracy, and recall. Precision evaluates the system's capacity to distinguish irrelevant material, whereas recall measures the system's capacity to recognize significant information. The F-measure will also be incorporated into the assessment process to reduce the memory and precision bias issue.

Precision:

Precision is given by the number of documents well classified by the total number of documents.

$$precision\ (p) = \frac{tp}{tp+fp} = \frac{number\ of\ documents}{number\ of\ label\ \ documents} \quad \ldots (5)$$

Recall:

The total number of documents that were successfully retrieved and correctly classified by several relevant documents.

$$Recall\ (r) = \frac{tp}{tp+fn} = \frac{number\ of\ label\ documents}{number\ of\ documents} \quad \ldots (6)$$

F1-Measure:

F1 Measure uses called-off and computed accuracy to identify the symphonic intent among them. When accuracy and precision are flawless in order, the matchless records have an F1 measure of 1, and when the F1 measure is zero, they have the lowest matchless records.

$$F1 - Measure\ (f) = \frac{2*Precision*Recall}{Precision+Recall} \quad \ldots (7)$$

Accuracy:

Interestingly, it is utilized as a standard to measure the graded algorithm's performance.

$$accuracy = \frac{tp+tn}{tp+fp+fn+tn} \quad \ldots. (8)$$

The equation above is used to calculate accuracy.

ROC:

The ROC equation is a way to compare TPR/FPR at various classification thresholds to measure the performance of the model.

$$ROC = \{(FPR(\theta), TPR(\theta)) | \ \theta \in [0,1]\} \quad \ldots. (9)$$

Result:

Thus, by creating modules for the classification the idea aims to prove the applicability of the approach. Python 3.7 powers the layout of the system's backend proven to be the appropriate tool and framework for data analysis. Fig 5 illustrates an ROC curve of the classification model, K-Nearest Neighbors (KNN) using a one vs rest approach.

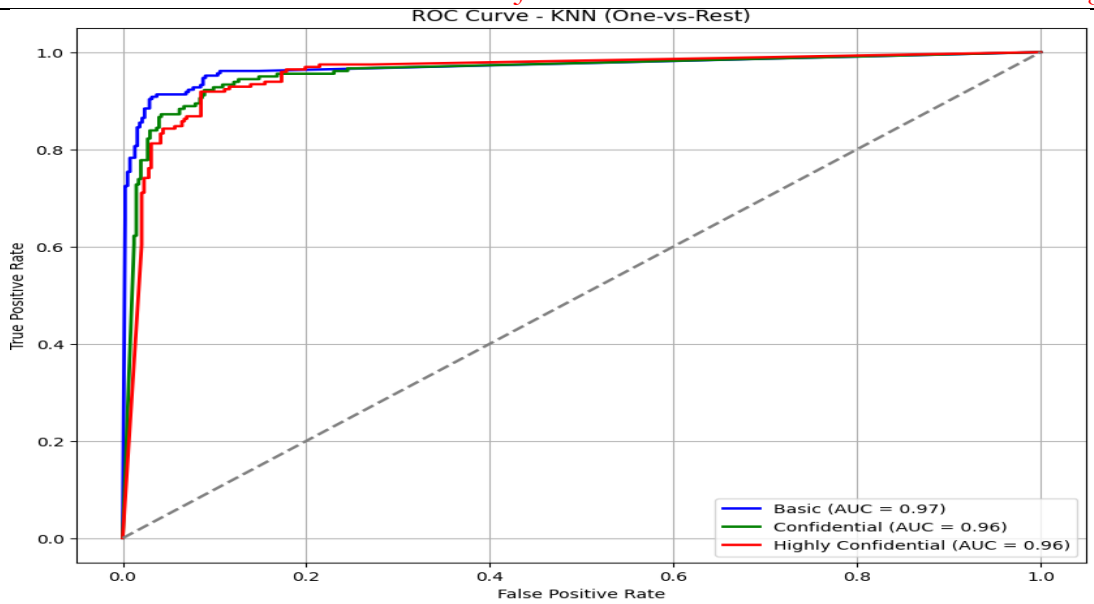

Figure 5. ROC Curve for KNN

It plots the True Positive Rate against the False Positive Rate on the y & x axes respectively and is compared with a diagonal dashed line to get a random classifier based on AUC = 0.5. Three curves are presented for different categories: The following security profiles include: Basic (blue, Mean AUC = 0.97), Confidential (green, Mean AUC = 0.96), and Highly Confidential (red, Mean AUC = 0.96). Every model's curve is significantly above the random baseline, hinting at efficient classification for all models. The Basic model presents the highest AUC at 0.97 which is somewhat better compared to the Advanced and Research models, which both have an AUC equal to 0.96. The fact that all the AUC values are fairly close to one another indicates that the classifier is accurate within all categories with little variation. In conclusion, the performances of the KNN classifier are quite consistent and the classifier is highly dependable. The results are shown in Figure 6.

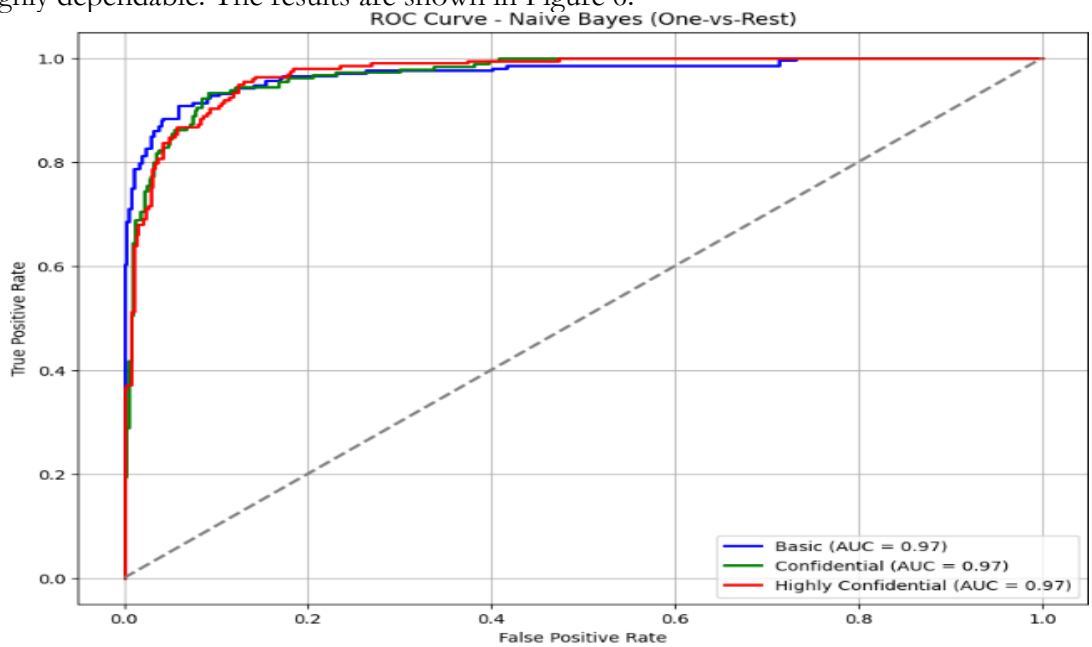

Figure 6. ROC Curve for Naïve Bayes

A dotted line of slash diagonal form shows the characteristics of completely random predictions where AUC equals 0.5 and y-axis defines the True Positive Rate and the x-axis denotes the False Positive rate. Three curves are displayed, representing the performance

across different categories: Tested schemes have "Basic" (blue), "Confidential" (green), and "Highly Confidential" (red) which yielded an Area Under the Curve (AUC) of 0.97. These results show that there is high and comparatively equal classification accuracy for the three categories. The curves stay far averted from the baseline line which further substantiates the measure of accuracy of the proposed model along with the appreciable class identification efficiency of the model.

The ROC curve of the SVM classification model used in this study is illustrated in Fig. 7 with a one-vs-rest strategy.

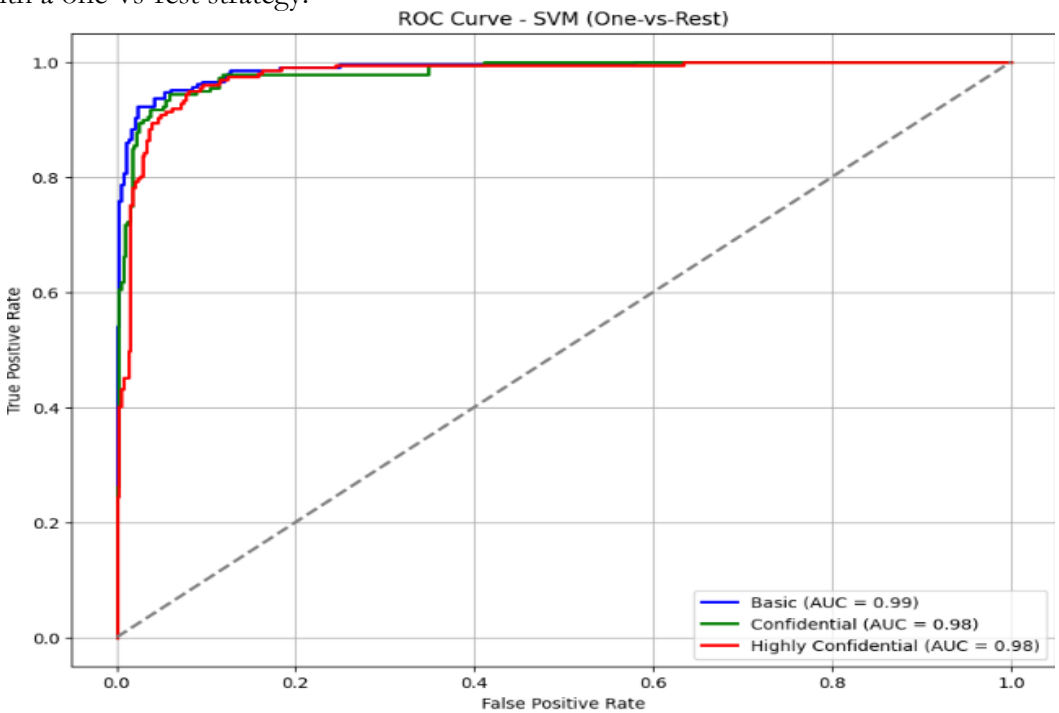

Figure 7. ROC Curve for SVM

A diagonal dashed line indicates the performance of a random model AUC = 0.5, while the graphic compares the True Positive Rate=(1-sensitivity) on the right y-axis with the False Positive Rate= (1-specificity) on the left x-axis. The three curves represent the model's performance across different categories: Depending on the level of confidentiality the decision can be of the following types: "Basic" (blue color including 1088 images with AUC = 0.99), "Confidential" (green color including 889 images with AUC = 0.98) and "Highly Confidential" (red color including 883 images with AUC = 0.98). From the above results, the Basic category has the highest AUC than that of other categories following a close second after it. All curves stay far above the baseline, which indicates accurate classification and a high level of effectiveness. This implies that the SVM model yields very good and fairly stable accuracy in all categories.

The ROC curve corresponding to the Random Forest classification model that uses a one-versus-rest approach is shown in Fig. 8. A broken diagonal line on the graph is the area of operation of a random classifier with an AUC value of 0.5 On the graph, TPR, which is on the y-axis, is compared with FPR on the x-axis. The model's performance is shown for three categories: Published" (blue, AUC = 0.98), "Internal"(green, AUC = 0.98), and "Sensitive"(red, AUC = 0.97). All of the curves are well above the baseline and show good classification performance for all categories.

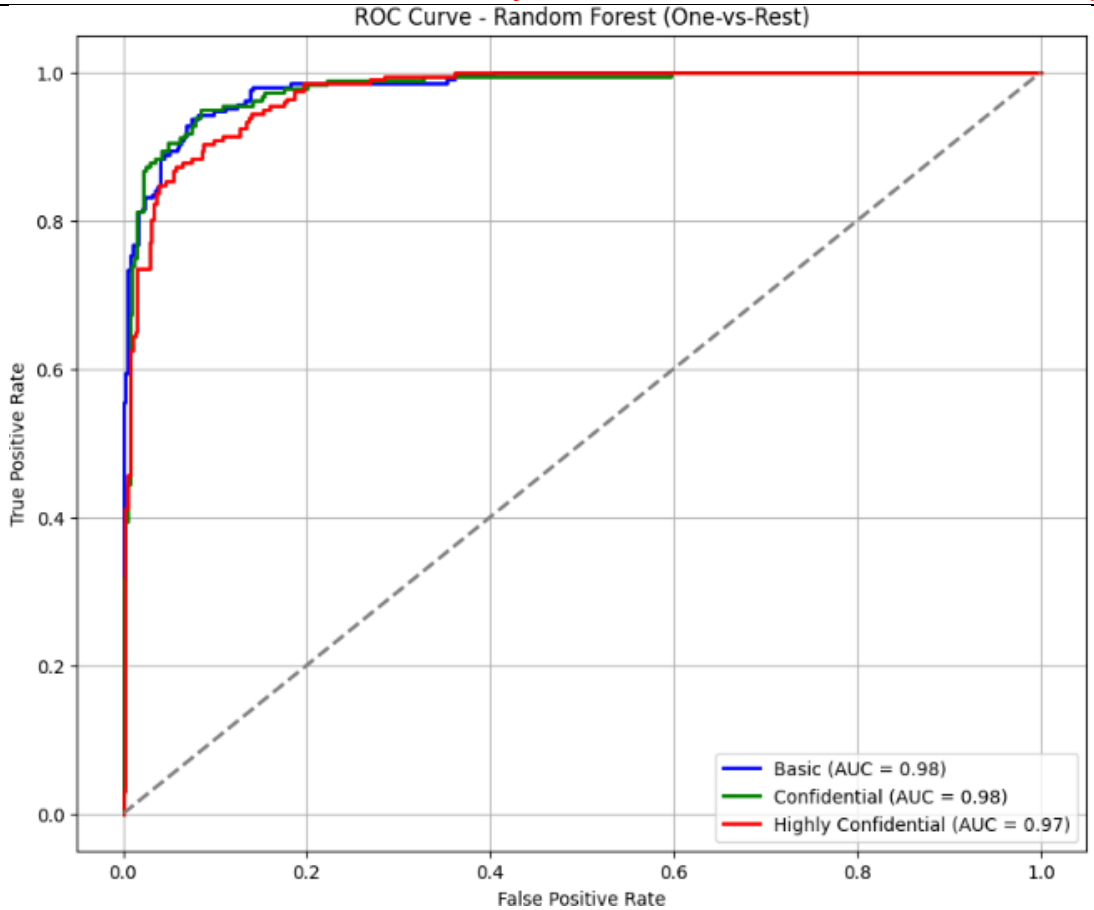

Figure 8. ROC Curve for Random Forest

Both Basic as well as Confidential categories reportedly have the best AUC of 0.98, with Highly Confidential not lagging far behind at a level of 0.97. This shows that the Random Forest model has similar and comparable results on all the groups and therefore is very reliable.

SVM, Random Forest, KNN, and Naive Bayes are the four models whose classification performance is compared in Fig. 9 combined ROC curve across three categories: There are three levels of classification namely: Basic, Confidential, and Highly Confidential.

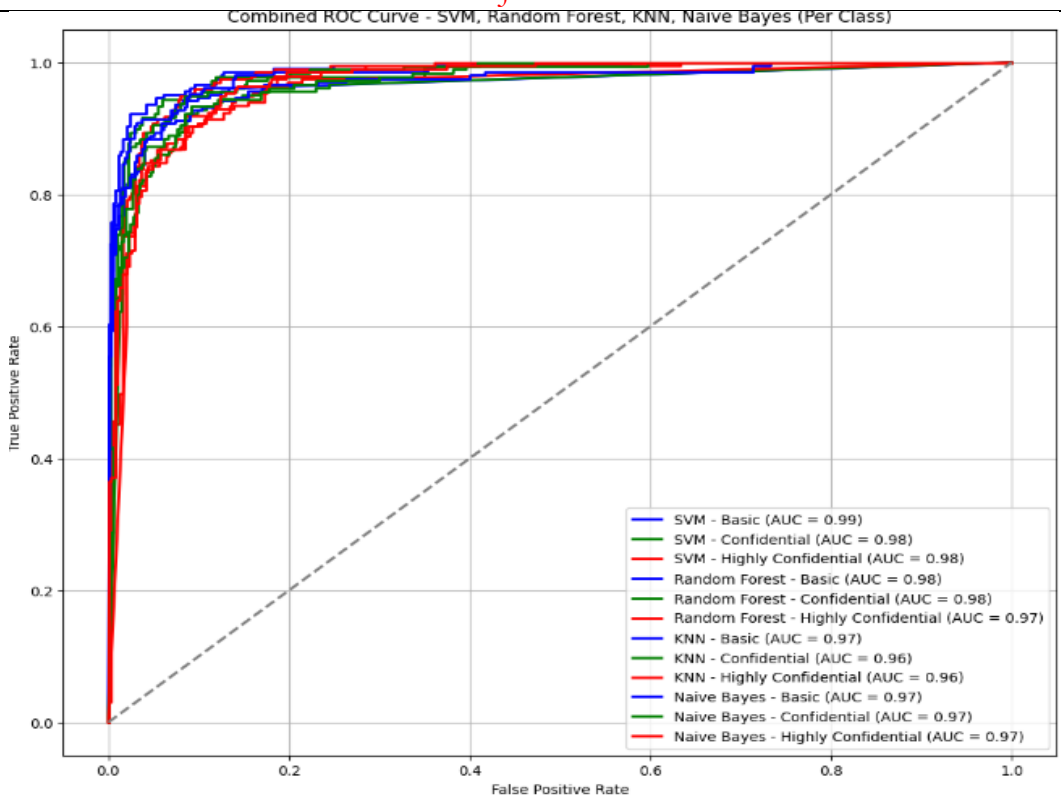

Figure 9. ROC Curve for All Models

With true positive and false positive rates, as the axes, for easy comparison with other models, a dashed diagonal line at 0.5 AUC is used in plotting the True Positive Rate against the False Positive Rate chart. The performance of the models is high with all the AUC values placed above 0.96 for all categories of the graphical output. random forests occupy the second place with AUC = 0.98 for all categories while the SVM model gives the highest AUC = 0.99 at the "Basic." The KNN yields AUCs of 0.97 for "Basic", and slightly lower of 0.96, for the rest of the categories. Once again, Naive Bayes is seen to be dominating all the other algorithms as can be seen by the crisp AUC value of 0.97 in all categories. Thus, as can be seen from the table, SVM slightly outperforms the others, but all the models are guaranteed a highly accurate classification.

Discussion:

There are two phases to document classification: training and testing. The training phase includes the NLP pre-processing phase the features pre-processing phase as well as the features vectoring phase The prediction class displays the variation of various categorization techniques. The SVM, NB, KNN, and RF algorithms which use different approaches are presented in Tables 5 and 6 The comparison of the proposed approach and current method are shown as follows in Table 5 below.

Table 5. Performance Evaluation

Classifier	Precision	Recall	F1-measure	ROC
Support Vector Machine	0.98	0.97	0.97	0.99
Random Forest	0.82	0.81	0.81	0.98
Naïve Bayes	0.88	0.87	0.87	0.97
K-NN N=3	0.95	0.95	0.95	96.5

Using the performance graphs we have presented strong evidence that indeed the proposed approach is better than the previous one. The Precision, Recall, and F1-measure of the three class labels are presented in Figure 10 While, figure 11 compares the accuracy of the

data classification algorithms SVM, NB, KNN, and RF [65]. Presented also in terms of recall, accuracy, and F-measure with each classifier's mean value.

Table 6. Accuracy of Classifiers for Training and Testing

Classifiers	Accuracy (Training)	Accuracy (Testing)
Support Vector Machine	0.99	0.97
Random Forest	0.98	0.96
Naïve Bayes	0.88	0.86
K-NN N=3	0.97	0.95

The accuracy comparison of several machine learning methods is presented in Table 5 above. The SVM, RF, and KNN methods have hence classified data more effectively than the NB algorithm, with the following testing accuracy: SVM=0.97, RF = 0.96, KNN = 0.95, and NB=0.86. Figure 10 shows the comparison of the precision, recall, and F1-measure of SVM, NB, KNN, and RF algorithms. Given its security and privacy level, cloud data is automatically classified using machine learning algorithms for encryption hence minimizing encryption time. The results show that the proposed approach achieves a significantly higher recall rate, accuracy, precision, and F1-measure than the recommended methodology. Recall, precision, and F1-score of four models are presented in the Fig 14 graph. The proposed K-Nearest Neighbors yields an F1-score of 0.95, precision of 0.95, and recall of 0.95. The following performances are achieved by Naive Bayes, precision is equal to 0.88, recall is 0.87 and F1-score is 0.88. While comparing the precision is 0.98, recall of 0.97, and F1-score of 0.98, SVM is still comparatively higher. Random Forest is least effective among all the models considered here with a poor accuracy of 0.82 for precision, recall, and F1-score. SVM, in any way, is better across the board.

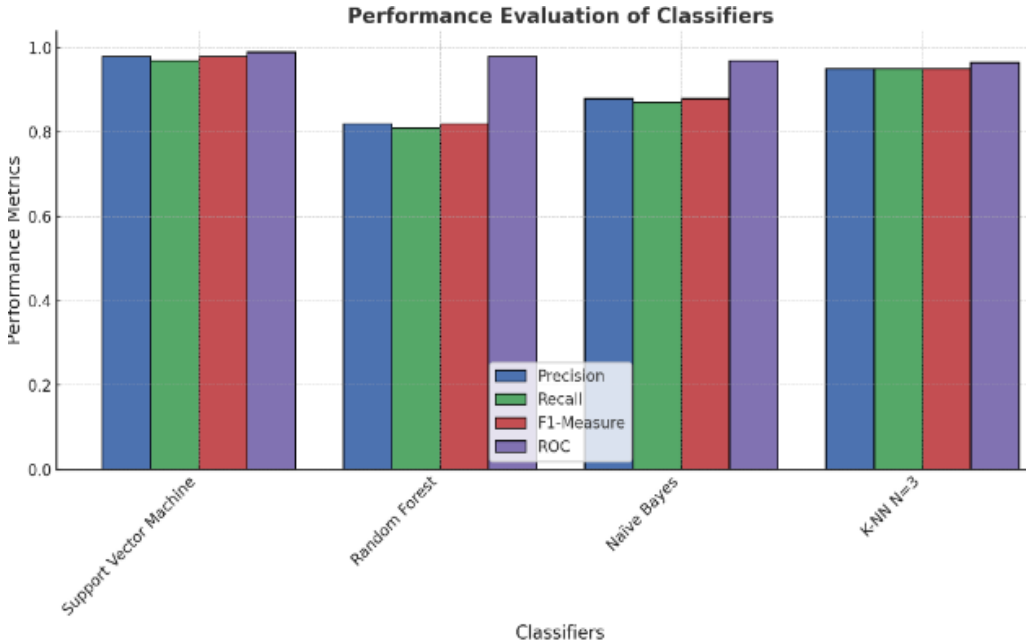

Figure 10. Performance Evaluation

Fig 11 below shows the training and testing accuracy of four models. The performance of K-Nearest Neighbors is 0.95 for testing and the performance of K-Nearest Neighbors is 0.97 for training. The testing accuracy of Naive Bayes is 0. 86 while its training accuracy is 0. 88 thus means lower values as compared to SVM. SVM has the highest level of testing accuracy, 0.97, as well as a training accuracy of 0.99. Random Forest also performs well with training accuracy and testing accuracy figures of 0.98 each and testing accuracy of 0.96. Even though SVM does slightly better than the others in general for both testing and training sessions in so far as the number of correctly classified patterns is concerned.

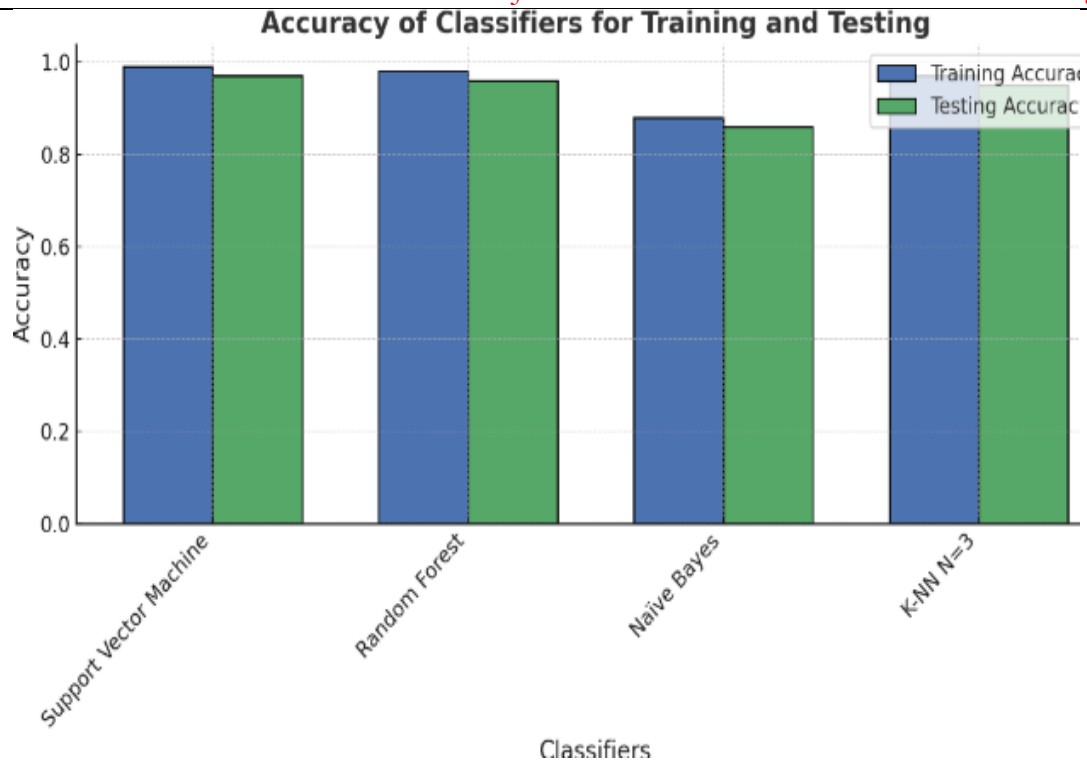

Figure 11. Accuracy of the proposed Classifiers

Support Vector Machine (SVM) outperformed the other classifiers due to its strong ability to handle high-dimensional data and effectively separate classes using optimal hyperplanes, making it particularly suitable for text classification tasks like those in the Reuters-21578 dataset. SVM's kernel trick further enhances its performance by mapping non-linearly separable data into higher-dimensional spaces, improving classification accuracy. On the other hand, Naïve Bayes (NB) had the lowest accuracy because it relies on the assumption of feature independence, which is often unrealistic in text classification, where words and phrases exhibit strong dependencies. This oversimplification leads to misclassifications, particularly for complex datasets with intricate relationships between features, thereby reducing NB's overall performance.

Conclusion and Future Work:

The current research proposes an approach for automated text document classification in cloud environments with a focus on data security. Its primary goals were to define data parameters and achieve high accuracy levels. Information security standards divide data into three categories: Using machine learning techniques Highly Confidential, Confidential, and basic. The use of machine learning classification methods and data security is the unique contribution of this security model. The idea behind the presented methodology is to build the application's modules with the identification of validation constructs in mind. Python 3.7 was used as the backend of the system because it is ideal for data analysis because of the tools and modules that accompany it. The results indicate that for the usage where the data's security has been pre-identified, the proposed method outperforms the approach of storing data without knowledge of its security requirements. The SVM, RF, and KNN, algorithms are outperformed by the NB classification approach in aspects of accuracy, precision, recall, and F1 measure as well. When disseminating this classified data in the future before uploading the documents to cloud storage we will use TLS, AES, and SHA. To extend our system, we plan to close another research gap as follows: We will also use other cryptographic techniques that, as we have come to notice, can be more reliable and secure.

Acknowledgment. The authors affirm that this study is original, has not been previously published, and is not currently under consideration elsewhere.

Author's Contribution. The corresponding author should explain the contribution of each co-author completely.

- **Fahad Burhan Ahmad and Azaz Ahmed Kiani:** The idea was proposed, and the study was conceptualized. Engaged in scientific discussions to refine the study. The manuscript was drafted and prepared accordingly.
- **Yaser Hafeez, Muhammad Habib, and Asif Nawaz:** Provided supervision for the study's execution. Ensured validation and accuracy of the outcomes.
- **Hamza Imran, Muhammad Rizwan Rashid Rana,[and Muhammad Azhar:** Conducted data analysis, performed experiments, and implemented algorithms.

Conflict of interest. The authors declare that there are no conflicts of interest associated with this study.

References:

[1] N. Antonopoulos and L. Gillam, Eds., "Cloud Computing," 2017, doi: 10.1007/978-3-319-54645-2.

[2] T. H. Noor, S. Zeadally, A. Alfazi, and Q. Z. Sheng, "Mobile cloud computing: Challenges and future research directions," *J. Netw. Comput. Appl.*, vol. 115, pp. 70–85, Aug. 2018, doi: 10.1016/J.JNCA.2018.04.018.

[3] H. J. xiaocui sun, Zhijun Wang, Yunxiang Wu, Hao Che, "A Price-Aware Congestion Control Protocol for Cloud Services," *J. Cloud Comput.*, 2021, doi: https://doi.org/10.21203/rs.3.rs-364078/v1.

[4] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses," *Computer (Long. Beach. Calif).*, vol. 45, no. 1, pp. 39–45, Jan. 2012, doi: 10.1109/MC.2012.1.

[5] P. A. Amro Al-Said Ahmad, "Scalability resilience framework using application-level fault injection for cloud-based software services," *J. Cloud Comput.*, vol. 11, no. 1, 2022, doi: Journal of Cloud Computing.

[6] N. Aljedani, R. Alotaibi, and M. Taileb, "HMATC: Hierarchical multi-label Arabic text classification model using machine learning," *Egypt. Informatics J.*, vol. 22, no. 3, pp. 225–237, 2021, doi: https://doi.org/10.1016/j.eij.2020.08.004.

[7] R. B. Fang Liu, Jin Tong, Jian Mao and L. B. and D. L. John Messina, "NIST Cloud Computing Reference Architecture," *Natl. Inst. Stand. Technol.*, pp. 500–292, 2011, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf

[8] Q. Z. Zhiying Jiang, Bo Gao, Yanlin He, Yongming Han, Paul Doyle, "Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports," *Math. Probl. Eng.*, 2021, doi: https://doi.org/10.1155/2021/6619088.

[9] "(PDF) Efficient Machine Learning Classifiers for Automatic Information Classification." Accessed: Feb. 03, 2025. [Online]. Available: https://www.researchgate.net/publication/339551576_Efficient_Machine_Learning_Classifiers_for_Automatic_Information_Classification

[10] U. S. K. L. M. Mundra, "Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing," *2010 First Int. Conf. Parallel, Distrib. Grid Comput.*, pp. 211–216, 2010, doi: 10.1109/PDGC.2010.5679895.

[11] M. P. Rewagad and M. Y. Pawar, "Use of digital signature with diffie hellman key exchange and aes encryption algorithm to enhance data security in cloud computing," *Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013*, pp. 437–439, 2013, doi:

10.1109/CSNT.2013.97.

[12] P. Kanagala and R. Jayaraman, "Effective encryption approach to improving the secure cloud framework through fuzzy-based encrypted cryptography," *Soft Comput.*, pp. 1–10, Apr. 2023, doi: 10.1007/S00500-023-08188-8/METRICS.

[13] P. Singh, B. Acharya, and R. K. Chaurasiya, "A comparative survey on lightweight block ciphers for resource constrained applications," *Int. J. High Perform. Syst. Archit.*, vol. 8, no. 4, pp. 250–270, 2019, doi: 10.1504/IJHPSA.2019.104953.

[14] S. Hussain, T. Shah, and A. Javeed, "Modified advanced encryption standard (MAES) based on non-associative inverse property loop," *Multimed. Tools Appl.*, vol. 82, no. 11, pp. 16237–16256, May 2023, doi: 10.1007/S11042-022-14064-8/METRICS.

[15] N. Sinha and L. Khreisat, "Cloud computing security, data, and performance issues," *2014 23rd Wirel. Opt. Commun. Conf. WOCC 2014*, 2014, doi: 10.1109/WOCC.2014.6839924.

[16] Jagriti Dhamija, "Cloud Security Solutions: Comparison among Various Cryptographic Algorithms," *Int. J. Nov. Res. Dev.*, vol. 3, no. 4, 2018, [Online]. Available: https://www.ijnrd.org/papers/IJNRD1804025.pdf

[17] G. Manik, S. Kalia, S. K. Sahoo, T. K. Sharma, and O. P. Verma, Eds., "Advances in Mechanical Engineering," 2021, doi: 10.1007/978-981-16-0942-8.

[18] S. Ahmad and S. Mehfuz, "Efficient time-oriented latency-based secure data encryption for cloud storage," *Cyber Secur. Appl.*, vol. 2, p. 100027, 2024, doi: https://doi.org/10.1016/j.csa.2023.100027.

[19] M. Y. S. M. I. K. M. S. S. A. M. Zhu, "Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security," *IEEE Access*, vol. 12, pp. 26334–26343, 2024, doi: 10.1109/ACCESS.2024.3351119.

[20] M. A. Zardari, L. T. Jung, and N. Zakaria, "K-NN classifier for data confidentiality in cloud computing," *2014 Int. Conf. Comput. Inf. Sci. ICCOINS 2014 - A Conf. World Eng. Sci. Technol. Congr. ESTCON 2014 - Proc.*, Jul. 2014, doi: 10.1109/ICCOINS.2014.6868432.

[21] K. A. Sayar Ul Hassan, Jameel Ahamed, "Analytics of machine learning-based algorithms for text classification," *Sustain. Oper. Comput.*, vol. 3, pp. 238–248, 2022, doi: https://doi.org/10.1016/j.susoc.2022.03.001.

[22] P. Yellamma, C. Narasimham, and V. Sreenivas, "Data security in cloud using RSA," *2013 4th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2013*, 2013, doi: 10.1109/ICCCNT.2013.6726471.

[23] X. Luo, "Efficient English text classification using selected Machine Learning Techniques," *Alexandria Eng. J.*, vol. 60, no. 3, pp. 3401–3409, 2021, doi: https://doi.org/10.1016/j.aej.2021.02.009.

[24] S. S. Sehra, "A REVIEW PAPER ON ALGORITHMS USED FOR TEXT CLASSIFICATION," 2013.

[25] V. T. Emmanouil K. Ikonomakis, Sotiris Kotsiantis, "Text Classification Using Machine Learning Techniques," *WSEAS Trans. Comput.*, vol. 4, no. 8, pp. 966–974, 2005, [Online]. Available: https://www.researchgate.net/publication/228084521_Text_Classification_Using_Machine_Learning_Techniques

[26] A. I. Anik, S. Yeaser, A. G. M. Imam Hossain, and A. Chakrabarty, "Player's performance prediction in ODI cricket using machine learning algorithms," *4th Int. Conf. Electr. Eng. Inf. Commun. Technol. iCEEiCT 2018*, pp. 500–505, Jul. 2018, doi: 10.1109/CEEICT.2018.8628118.

[27] N. Kamal, M. Andrew, and M. Tom, "Semi-Supervised Text Classification Using EM," *Semi-Supervised Learn.*, pp. 32–55, Oct. 2006, doi:

10.7551/MITPRESS/9780262033589.003.0003.

[28] I. Rasheed, V. Gupta, H. Banka, and C. Kumar, "Urdu text classification: A comparative study using machine learning techniques," *2018 13th Int. Conf. Digit. Inf. Manag. ICDIM 2018*, pp. 274–278, Sep. 2018, doi: 10.1109/ICDIM.2018.8847044.

[29] Y. Zhan, H. Chen, S. F. Zhang, and M. Zheng, "Chinese text categorization study based on feature weight learning," *Proc. 2009 Int. Conf. Mach. Learn. Cybern.*, vol. 3, pp. 1723–1726, 2009, doi: 10.1109/ICMLC.2009.5212257.

[30] B. P. Mayor Shweta, "Document Classification Using Support Vector Machine," *Int. J. Eng. Sci. Technol.*, vol. 4, no. 4, 2012, [Online]. Available: https://www.researchgate.net/publication/266593700_Document_Classification_Using_Support_Vector_Machine

[31] Y. Zheng, "An exploration on text classification with classical machine learning algorithm," *Proc. - 2019 Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2019*, pp. 81–85, Nov. 2019, doi: 10.1109/MLBDBI48998.2019.00023.

[32] "(PDF) Techniques for text classification: Literature review and current trends." Accessed: Feb. 03, 2025. [Online]. Available: https://www.researchgate.net/publication/301633216_Techniques_for_text_classification_Literature_review_and_current_trends

[33] R. K. Tamanna, "Secure Cloud Model using Classification and Cryptography," *Int. J. Comput. Appl.*, vol. 159, no. 6, 2017, doi: 10.5120/ijca2017912953.

[34] B. T. P. Quang Hung Nguyen, Hai-Bang Ly, Lanh Si Ho, Nadhir Al-Ansari, Hiep Van Le, Van Quan Tran, Indra Prakash, "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil," *Math. Probl. Eng.*, 2021, doi: https://doi.org/10.1155/2021/4832864.

[35] L. Morse, M. Teodorescu, Y. Awwad, and G. C. Kane, "A Framework for Fairer Machine Learning in Organizations," *SSRN Electron. J.*, Sep. 2020, doi: 10.2139/SSRN.3690570.

[36] S. C. Jaeyoung Kim, Sion Jang, Eunjeong Park, "Text classification using capsules," *Neurocomputing*, vol. 376, no. 1, pp. 214–221, 2020, doi: https://doi.org/10.1016/j.neucom.2019.10.033.

[37] J. Y. R. Cornejo and H. Pedrini, "Audio-visual emotion recognition using a hybrid deep convolutional neural network based on census transform," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2019-October, pp. 3396–3402, Oct. 2019, doi: 10.1109/SMC.2019.8914193.

[38] K. Akuthota, A. Ganesh, B. Reddy A, and S. K. Depuru, "Machine Learning Models for Classification of Sensitive Financial Documents," *5th IEEE Int. Conf. Cybern. Cogn. Mach. Learn. Appl. ICCCMLA 2023*, pp. 334–340, 2023, doi: 10.1109/ICCCMLA58983.2023.10346685.

[39] C. E. B. M.A. Friedl, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, 1997, doi: https://doi.org/10.1016/S0034-4257(97)00049-7.

[40] Ekta, "MACHINE LEARNING: A REVIEW OF LEARNING TYPES," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 9, 2022, [Online]. Available: https://www.irjmets.com/uploadedfiles/paper//issue_9_september_2022/29824/final/fin_irjmets1662994184.pdf

[41] X. Yan, L. Tan, H. Xu, and W. Qi, "Improved mixture differential attacks on 6-round AES-like ciphers towards time and data complexities," *J. Inf. Secur. Appl.*, vol. 80, p. 103661, Feb. 2024, doi: 10.1016/J.JISA.2023.103661.

[42] M. A. R. Pandu Adam, "IMPLEMENTASI SISTEM KEAMANAN DOKUMEN KEPEGAWAIAN MENGGUNAKAN METODE AES-256 DAN VIGENERE

CHIPER," *J. Komput. dan Teknol.*, vol. 3, no. 1, 2024, doi: https://doi.org/10.58290/jukomtek.v2i2.166.

[43] M. A. R. Fathur Setya Pratama, "PENGAMANAN DOKUMEN KEPEGAWAIAN PADA DINAS PENDIDIKAN TEMANGGUNG DENGAN ALGORITMA RC4 DAN AES-256," *J. Komput. dan Teknol.*, vol. 3, no. 1, 2024, doi: https://doi.org/10.58290/jukomtek.v2i2.167.

[44] and S. H. A. Sami, Teba Mohammed Ghazi, Subhi RM Zeebaree, "A Novel Multi-Level Hashing Algorithm to Enhance Internet of Things Devices' and Networks' Security," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, pp. 676–696, 2024, [Online]. Available: https://www.academia.edu/114844058/A_Novel_Multi_Level_Hashing_Algorithm_to_Enhance_Internet_of_Things_Devices_and_Networks_Security

[45] S. Sangheethaa, A. Korath, and C. R. Ranjana, "Improvisation in SHA Algorithm," *RASSE 2023 - IEEE Int. Conf. Recent Adv. Syst. Sci. Eng. Proc.*, 2023, doi: 10.1109/RASSE60029.2023.10363491.

[46] A. S. Babu, Ratnam Dodda, "Text Document Clustering Using Modified Particle Swarm Optimization with k-means Model," *Int. J. Artif. Intell. Tools*, vol. 33, no. 1, 2024, doi: https://doi.org/10.1142/S0218213023500616.

[47] C. J. I. H., Frank, E., Hall, M. A., & Pal, "Practical machine learning tools and techniques," *Witten*, 2016.

[48] D. G. Verma, Tanu, Renu Renu, "Tokenization and Filtering Process in RapidMiner," *Int. J. Appl. Inf. Syst.*, vol. 7, no. 2, 2014, [Online]. Available: https://research.ijais.org/volume7/number2/ijais14-451139.pdf

[49] C. C. Aggarwal, "Machine Learning for Text: An Introduction," *Mach. Learn. Text*, pp. 1–16, 2018, doi: 10.1007/978-3-319-73531-3_1.

[50] S. M. Gaurav Gupta, "Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example)," *Int. J. Comput. Appl*, 2015, [Online]. Available: https://www.researchgate.net/publication/339527155_Text_Document_Tokenization_for_Word_Frequency_Count_using_Rapid_Miner_Taking_Resume_as_an_Example

[51] H. Saif, M. Fernández, Y. He, and H. Alani, "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter," *Int. Conf. Lang. Resour. Eval.*, 2014.

[52] K. Spirovski, E. Stevanoska, A. Kulakov, Z. Popeska, and G. Velinov, "Comparison of different model⇔s performances in task of document classification," *ACM Int. Conf. Proceeding Ser.*, Jun. 2018, doi: 10.1145/3227609.3227668.

[53] J. Singh and V. Gupta, "Text Stemming," *ACM Comput. Surv.*, vol. 49, no. 3, Sep. 2016, doi: 10.1145/2975608.

[54] G. Sampson and P. M. Postal, "The 'language instinct' debate : revised edition," 2009, Accessed: Feb. 03, 2025. [Online]. Available: https://books.google.com/books/about/The_Language_Instinct_Debate.html?id=WkRDgytEWNYC

[55] Richard F. Xiang, "Use of n-grams and K-means clustering to classify data from free text bone marrow reports," *J. Pathol. Inform.*, vol. 15, p. 100358, 2024, doi: https://doi.org/10.1016/j.jpi.2023.100358.

[56] Y. R. Bowen Deng, Xinxing Liu, Wenxia Zhang, Juan Huang, "Chemoconnectomics: Mapping Chemical Transmission in Drosophila," *Neuron*, vol. 101, no. 5, pp. 876–893, 2019, [Online]. Available: https://www.cell.com/neuron/fulltext/S0896-6273(19)30072-8?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0896627319300728%3Fshowall%3Dtrue

[57] D. T. N. Shanthi, "A modified multi objective heuristic for effective feature selection in text classification," *Cluster Comput.*, vol. 22, pp. 10625–10635, 2019, doi: https://doi.org/10.1007/s10586-017-1150-7.

[58] K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification," *Expert Syst. Appl.*, vol. 66, pp. 1339–1351, Dec. 2016, doi: 10.1016/J.ESWA.2016.09.009.

[59] P. C. Miftahul Qorib, Timothy Oladunni , Max Denis, Esther Ososanya, "Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset," *Expert Syst. Appl.*, vol. 212, p. 118715, 2023, doi: https://doi.org/10.1016/j.eswa.2022.118715.

[60] S. Misra, Kousik Barik, "Analysis of customer reviews with an improved VADER lexicon classifier," *J. Big Data*, vol. 11, no. 10, 2024, doi: https://doi.org/10.1186/s40537-023-00861-x.

[61] H. Chen *et al.*, "Pre-trained image processing transformer," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 12294–12305, 2021, doi: 10.1109/CVPR46437.2021.01212.

[62] M. A. Aqsa Khalid, Ghulam Mustafa, Muhammad Rizwan Rashid Rana, Saeed M. Alshahrani, "RNN-BiLSTM-CRF based amalgamated deep learning model for electricity theft detection to secure smart grids," *PeerJ Comput. Sci.*, 2024, [Online]. Available: https://peerj.com/articles/cs-1872/

[63] T. P. Latchoumi and L. Parthiban, "Quasi Oppositional Dragonfly Algorithm for Load Balancing in Cloud Computing Environment," *Wirel. Pers. Commun. 2021 1223*, vol. 122, no. 3, pp. 2639–2656, Aug. 2021, doi: 10.1007/S11277-021-09022-W.

[64] A. A. Tahani Alsaedi, Muhammad Rizwan Rashid Rana, Asif Nawaz, Ammar Raza, "Sentiment Mining in E-Commerce: The Transformer-based Deep Learning Model," *Int. J. Electr. Comput. Eng. Syst.*, vol. 15, no. 8, 2024, doi: https://doi.org/10.32985/ijeces.15.8.2.

[65] A. I. Kadhim, "An evaluation of preprocessing techniques for text classification," *Int. J. Comput. Sci. Inf. Secur*, vol. 16, no. 16, pp. 22–32, 2018.

[66] N. I. M. and H. A. G. Mochamad Alfan Rosid, Arif Senja Fitrani, Ika Ratna Indra Astutik, "Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, 2019, doi: 10.1088/1757-899X/874/1/012017.

[67] "(PDF) Urdu Text Classification using Majority Voting." Accessed: Feb. 03, 2025. [Online]. Available: https://www.researchgate.net/publication/307539554_Urdu_Text_Classification_using_Majority_Voting

[68] M. Bilal, H. Israr, M. Shahid, and A. Khan, "Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, no. 3, pp. 330–344, 2016, doi: https://doi.org/10.1016/j.jksuci.2015.11.003.

[69] S. S. A. Balinsky, H. Balinsky, "Rapid Change Detection and Text Mining," *Proc. 2nd Conf. Math. Def. (IMA), Def. Acad. UK*, 2011, [Online]. Available: https://ima.org.uk/wp/wp-content/uploads/2011/10/Rapid-Change-Detection-and-Text-Mining.pdf