Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19669-9

Security to text (S2T): multi-layered based security approaches for secret text content

Shamal Kashid[1] · Lalit K. Awasthi[1] · Krishan Berwal[2]

Received: 5 June 2023 / Revised: 3 April 2024 / Accepted: 10 June 2024

Abstract
In the digital world, text data is produced in an unstructured manner across various communication channels. Extracting valuable information from such data with security is crucial and requires the development of techniques in text mining, information retrieval, and natural language processing (NLP). To solve this issue, we introduce two novel approaches: keyword extraction (KE) and a multi-layered secret sharing scheme (MLSS) to provide security to extracted keywords rather than overall text documents. The KE approach encompasses a sequence of text pre-processing procedures, including tokenization, stopword removal, stemming, and bag of words representation, followed by indexing. This methodology aims to efficiently extract keywords from text datasets. For this research work, we have proposed three datasets, including text messages, whatsapp messages, and electronic mail. MLSS enhances the security of extracted textual data by leveraging text pre-processing steps. This scheme ensures better confidentiality and the non-revealment of sensitive information. Additionally, we evaluate our KE model on our dataset as well as on standard datasets. Experimental results demonstrate the effectiveness of our proposed security to text (S2T) model, which outperform existing state-of-the-art approaches. The model obtains a 100% correlation between the reconstructed text and the original text.

Keywords Keyword extraction · Text security · Secret sharing scheme · Natural language processing

Lalit K. Awasthi and Krishan Berwal contributed equally to this work.

✉ Shamal Kashid
kashid.shamalphd2021@nituk.ac.in

Lalit K. Awasthi
lalitdec@gmail.com

Krishan Berwal
k2b@ieee.org

[1] Department of Computer Science and Engineering, National Institute of Technology Uttarakhand IN, Srinagar, India

[2] Faculty of Communication Engineering, Military College of Telecommunication Engineering, Mhow, IN, India

1 Introduction

Technology development and easy Internet accessibility generate enormous amounts of textual content daily. This text content takes a lot of space and time to be processed and securely communicated, which decreases the system's performance [1]. The fair utilization of data with security and the fast delivery of content to various users or systems with guaranteed quality of service are essential yet challenging areas [5]. With the ease of editing and creation in the digital world, protecting ownership and preventing unauthorized tampering with text data have become significant issues. Most people use the Internet daily for different social activities on platforms like facebook, whatsapp, instagram, and twitter. Semantic differences in text content pose a massive challenge to data extraction [8].

Identifying meaningful or relevant information from enormous textual data and ensuring the safety of such extracted sensitive information has become critical in many areas of this digital world [10]. The notable financial sector, commercial applications, military communications networks, and national security are the various sectors of government that require text data security. Customer information such as user names, telephone numbers, debit and credit card numbers, addresses, and credentials are the most informative data in banking applications, referred to as Keywords in Banking, Emails, Business, Feedback, Surveys, and Social media posts [9, 11].

Ensuring the safety and security of textual data is crucial across various sectors, including finance, commercial applications, military communications networks, and national security. It helps mitigate risks, protect against threats, and maintain confidentiality. Robust security measures can enhance trust, preserve intellectual property, and safeguard national interests [12, 13]. These approaches extend beyond specific sectors to areas like healthcare, legal, and government agencies, where confidentiality, privacy, and data protection are essential. It can assist businesses in enhancing productivity, providing better customer service, and staying in contact with the consumer base. Once the keywords are identified, data can be combined with additional text analysis to categorize the texts based on sentiment, topic, language, and other factors [1]. Table 1 shows a text file processed by KE along with ground truth.

Text pre-processing plays an important part in any NLP system. The most commonly used terms unlikely to assist keyword extraction include pronouns, articles, and prepositions; these terms can be eliminated. Text pre-processing procedure effectively removes unnecessary data from input dataset [2]:

- To minimize the size of the indexing (or data) files for text documents to stopwords occupying up to 20-30% of the overall number of words on a written page, indexing size can be reduced by 50% by using stemming.
- To increase the efficiency and effectiveness of the information retrieval system to stopwords do not help search or text mining, and they may cause the retrieval system to get confused, stemming checks similar terms in a text document [2].

Several automatic data extraction (ADT) tools (like YAKE [4], RAKE [5, 8], BERT (Bidirectional Encoder Representations from Transformers) [6, 7] and Textrank [4]) are already present in NLP. Improving text data pre-processing efficiency in ADT is challenging, and its modeling requires large-scale training datasets. Developing a large text dataset is unreasonable due to the need for more data sources and unnecessary cost and time [19]. YAKE [3, 4] is a simple unsupervised automatic keyword extraction approach that uses statistical text features obtained from single documents to find the most relevant terms in a text. RAKE [5] is a procedure for extracting keywords from separate documents that are

Table 1 Extracted Keyword from a text file

Dear Customer,
We refer to your e-mail regarding your Credit Card 1234XXXXXXXX7009.
We wish to clarify that as per terms and conditions if the customer makes a
payment of only the minimum amount due or any amount less than the total
amount due, then appropriate charges and interest are levied on the account.
We also wish to clarify that the refund received in case of the canceled
transaction is not considered a payment. If there is a refund received from
the merchant equivalent to the total amount due, the payment of at least the
minimum outstanding due needs to be paid. So that no financial charges can
be generated on your credit card statement. Please be informed that no merchant
refund/cashback/credit due to transactions converted to EMI /canceled
transactions reversals/promotional cashback will be considered as a payment
towards the outstanding card. In case a card member makes an excess payment
compared to the outstanding of the card, there will be a credit balance in
the card account. This will be adjusted against the subsequent transactions
on the card. However, no interest can be claimed on this excess credit amount.
We request you to click on the below-mentioned link for the details on
scenarios where the financial charges are levied.

Keywords/Ground Truth:
Dear, Customer, e-mail, regarding, Credit, Card,
1234XXXXXXXX7009, clarify, terms, conditions, customer, only, any, less,
appropriate, wish, received, transaction, not, refund, equivalent, minimum,
needs, paid, generated, statement, Please, no, refund/cashback/credit, converted,
canceled, reversals, promotional, cashback, member, compared, balance, request,
click, mentioned, link, details, scenarios.

unsupervised approach, domain, and language independent. The main challenge for extracted keywords/information is to be confidential, authenticated, and untampered before sending it to its intended recipient. Cryptography is one aspect of the solution to these issues [16], however, cryptography and watermarking algorithms are not suitable for situations where a secret or master key has to be distributed and controlled among the group of participants [24]. This work applies a branch of cryptography known as a secret-sharing scheme (SSS), where the secret is shared and allocated to group members.

The KE model and MLSS to addresses the critical issue of text data security and extraction. The KE model aims to extract essential keywords from textual data efficiently, enabling accurate analysis and interpretation. These keywords play a vital role in various applications, such as information retrieval, document categorization, and sentiment analysis. On the other hand, the MLSS scheme focuses on enhancing the security of extracted text data. By employing encryption, secret sharing, and XOR operations, the MLSS scheme ensures the confidentiality and integrity of sensitive information extracted from textual data. Together, the KE model and MLSS scheme provide a comprehensive solution for both extracting valuable insights from text data and safeguarding it against unauthorized access and revelation. The following are the main contributions of the proposed work:

- The proposed KE model effectively extracts sensitive data from text datasets. Additionally, a bag of word and indexing techniques effectively implement our proposed KE model.
- Three types (whatsapp text messages, electronic mail, and textual messages) of large data volume datasets have been proposed for this research work.
- The proposed MLSS model provides security to the extracted non-revealing text data by achieving a 100% correlation between the reconstructed and original text.
- KE proposed model outperforms baseline models on the three datasets (Inspec, SemEval201, and Krapivin) as well as on our three proposed datasets based on F1-Score.
- The proposed technique takes less computational time to create secret share generation and secret share reconstruction than the existing SSS techniques.

2 Background study

Many applications, such as marketing, product management, academics, and governance, need to evaluate and extract information from textual data. It can be utilized for indexing purposes in information retrieval programs like Google, Microsoft Internet Explorer, and Firefox [2]. The primary aim of the preprocessing text is to extract essential features or keywords from raw textual documents to increase the significance of words to their document files, which takes up to 80% of the time on the total text documents classification procedure.

2.1 Keyword extraction

Nanta Janpithak et al. [22] demonstrated the utility of the GATE (General Architecture of Text Engineering) framework in extracting regulatory requirements from documents. A Nearly New Information Extraction System can be used to extract only the most important content and in its implementation; unstructured phases like subject, object, target, and action may all be retrieved and converted into structured data such as a table. The sentiment of web film reviews investigated [23] and several preprocessing methods are used to minimize the noise in the text, as well as the chi-squared method to remove unnecessary attributes which are not affected by the text's alignment. Experiment assessment shows realistic text pre-processing procedures, like data transformation and filtering, significantly enhancing the classifier's capability. The accuracy gained on the pair of data sets is analogous to subject categorization, a much simpler work. Results indicated significant enhancement in classifier performance due to realistic text preprocessing techniques like data transformation and filtering.

Key Information Extraction (KIE) [24] was used to improve capabilities by automatically adding textual and visual aspects within documents. The methodology incorporates improved graph learning modules to rectify the graph architecture on complex documents along graphically rich contexts. Using a BiRNN- bidirectional recurrent neural network, Yang et al. (2012) [25] learned the conditional distribution features of every keyword in texts where the model reaches nearly 100% recalls and precisions. To estimate the ability of the hidden informative data within the developed system to utilize the differences in feature distribution, resulting in state-of-the-art outcomes. YAKE, introduced by Campos et al. [3], emerged as a feature-based model for multilingual KE from single documents, surpassing existing methods without requiring training on specific collections or dictionaries. The extension of YAKE by Campos et al. [4] to select the most relevant keywords based on statistical text features outperformed numerous supervised and unsupervised methods across various text lengths, languages, and domains.

Allahyari et al. [34] elaborated various extraction methodologies for multi-document and single-document summarization. The author discussed the most often used techniques, including subject representation methods, frequency-driven approaches, graph-based methods, and machine learning context technology. Gambhir et al. [35] studied some crucial information about the record of text summarization, the recent SOTA and future potential. The survey conducted in this work would be a useful beginning point for new researchers to understand the fundamental challenges of text summarization. The majority of studies took an extractive strategy. It is necessary to place a greater emphasis on abstractive and hybrid techniques. El-Kassas et al. [36] provide a survey with a complete assessment of the various components of Automatic Text Summarization (ATS): methodologies, techniques, datasets, approaches, evaluation methods and research goals.

2.2 Text security

Generated models [12, 29] which were providing a personal collection of shares (secret) when using the n-counting-based SSS method. In this study, unmanaged allocated shares created by the system utilizing an authenticated receiver key creation technique were served using personalized remembrance tools. The shares employed in the original SSS method were difficult to remember, in contrast to standard password assignments, which provide full personal selection. The proposed learning context or keeping shares secreted within personally a section of lines or sentences with the help of improved Arabic textual steganography. Table 2 shows the analysis of text data security using the SSS approaches.

Parul et al. [32] proposed a BEMSS- Blockwise Encryption based Multi Secret Sharing scheme for Securing Visual Content based on (n,n) multiple secret visual data encoding technique with a SSS, which is proposed in this study by utilizing blockwise encryption. The experimental studies demonstrate the performance of the BEMSS model which delivers a computationally effective method for data security. The secret-sharing procedure is considered a classification issue by the generative adversarial networks (GANs) network, which assumes the role of a dealer. The key concept is to treat any secret texts as images. The secret text image is then broken into sub-images using image segmentation, and the sub-images are then encoded and decoded using DNA (Deoxyribonucleic acid) coding. Next, the proposed model is trained to identify the SS results. The results illustrate the scheme's high communication efficiency, flexibility, and security. This technique is a considerable extension of the GANs and a unique approach for the key SSS methodology [13].

Table 2 Analysis of technique based SSS for textual data

Scheme	Type	Secret Information	SSS Approach
Gutub et al. [29]	(k,n)	Text	Counting based SSS with Stegnography
Adnam et al. [12]	(k,n)	Text to Image	Counting based SSS with Stegnography
Zhen et al. [13]	(k,n)	Text	Generative Adversarial Networks-Basedd SSS
Esraa et al. [14]	(k,n)	Text Watermarking	Watermarking with Counting based SSS
Guttikonda et al. [17]	(k,n)	Text File	Polynomial Based SSS
Shamal et al. [9]	(n,n)	Email,Text and Whatsapp Message	Multilayered SSS based on X-OR

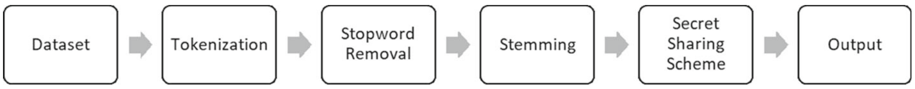


Fig. 1 Proposed Keyword Extraction Model with Seurity Steps

The two proposed approaches use Arabic language characteristics steganography based on Kashida extension characters to hide confidential secrets inside the texts. The comparisons were evaluated at several techniques on the same stage with the 40 standard benchmark textual assertions, and the findings were remarkable and showed promising results for further study. The Kashida steganography technique used in the Arabic text watermarking system is proposed [14]. MSSS (Multilayered Secret Sharing Scheme) for text uses X-OR to provide (n+1, n+1) multilayered text encryption with the help of a SSS. MSSS [9] includes two different levels for share generation and share reconstruction. While elaborating on the reconstruction algorithm, the authors used Blakley's technique and determined the scheme's access structure as well as evaluated its information rate. The authors motivated and applied Secret sharing scheme in this proposed work as discussed in Section 3.

3 Proposed model

In this work, two novel approaches have been proposed. First is to extract important words using the KE model with an aim to extract important words from text files effectively. Second is to provide text data security using a multi-layered secret-sharing scheme. Figure 1 shows the proposed S2T model steps.

3.1 Keyword extraction (KE) model

Our proposed KE model is based on different text preprocessing steps. This section briefly describes the KE model steps in detail as shown in Fig. 2.

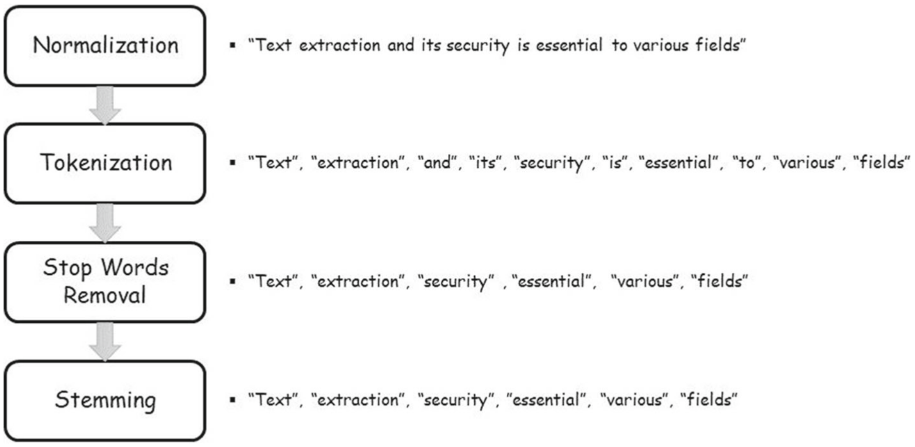

Normalization
"Text extraction and its security is essential to various fields"
Tokenization
"Text", "extraction", "and", "its", "security", "is", "essential", "to", "various", "fields"
Stop Words Removal
"Text", "extraction", "security" , "essential", "various", "fields"
Stemming
"Text", "extraction", "security", "essential", "various", "fields"

Fig. 2 Proposed Keyword Extraction Model Steps

3.1.1 Step 1 tokenization

The procedure of converting a set of words to significant or meaningful words is called tokenization. It follows some steps. First, split the input text into paragraphs and second, split the paragraph into sentences. After that, the third step splits the sentence into words. A tokenization filter, also known as tokens, separates these words and characters. The tokens collection included question marks, symbols, and characters outside common morphology. The nouns, verbs, pronouns, and other tokens are then organized using the Part of Speech (POS) grammar rules. We can obtain language-dependent properties by tokenization, which continues using the stopword [20]. The outcome can be a single sentence, a paragraph, a document, or a group of sentences.

Algorithm 1 Tokenization Algorithm.

```
Require: Document
Ensure: List of Word Tokens
1: Split the document into the paragraph
2: for each paragraph in the document do
3:     split a paragraph into sentences
4:     for each Sentence in the document do
5:         split sentence into tokens
6:     end for
7: end for
```

Algorithm 1 provides a foundational tokenization process essential for NLP tasks. It starts by segmenting a document into paragraphs, then iterates over each paragraph, breaking it down into sentences. Within each sentence, the algorithm further divides the text into individual word tokens, enabling the next analysis [20, 21].

3.1.2 Step 2 stopword removal

The stopwords contain a listing of frequently occurring words which are appeared in all textual documents [9, 20, 21]. Common features of text such as conjunctions (and, or, but) and pronouns (she, he, it) should be deleted for the reason that they don't have any impact, which contributes less or does not provide any value to the classification process (every feature was removed whenever the model matches the feature in the stopwords dictionary). If the feature is a special character or a number, it should be eliminated. For the same reason, if the feature is a special character or a digit, it was removed. To identify the stopwords,

Algorithm 2 Stopword Removal.

```
Require: List of Word Tokens
Ensure: Preprocessed text
1: for each document do
2:     search the stopword list
3:     if (word=stopword) then
4:         remove the word from the sentence
5:     else if then
6:         do nothing
7:     end if
8: end for
```

we can sort our collection of phrases by frequency and choose the ones with the highest frequency based on their lack of semantic value [21]. So, Algorithm 2 shows the procedure of stopword removal. This algorithm takes a list of word tokens as input and aims to produce preprocessed text as output by removing stopwords. It iterates over each document in the dataset and searches for each word in a predefined stopword list. If a word is identified as a stopword, it is removed from the sentence. Otherwise, if the word is not a stopword, no action is taken. By systematically removing stopwords from the text data, the algorithm helps reduce noise and improve the quality of the next text analysis tasks, such as sentiment analysis or keyword extraction processes.

3.1.3 Step 3 stemming

Stemming removes unnecessary features from a word's form while maintaining proper spelling. At the same time, all stemmed words are converted to their basic form. Stemming is a normalization technique for natural language processing employed in this step to minimize the number of computations created on the test dataset.

This stage consists of breaking down words or phrases into root words. Stemming is the procedure of extracting affixes, prefixes, and suffixes from characteristics of decreasing modulated terms to their original stem. The stemmed word is not needed to connect the word's initial root, but it is usually connected the reason that words match a similar stem. In order to reduce the number of features in the feature space and enhance the performance of the classifier, many different types of features are combined into a single feature. Consider the following scenario: (Interchang, Interchange, Interchanger, Interchangeable, Interchanging). To produce the single-feature relation, the total features are conflated into a single feature by deleting the different suffixes -e, -er, -able, and -ing to accomplish only one (single) feature connection, as shown in Fig. 3.

Bag of words A BOW [18] represents text data that describes the occurrence of particular words within a text data file, as shown in Fig. 4. We keep records of the number of words while ignoring grammar issues and word order. Since any data about the sequence or structure of words contained within the document is deleted, it is referred to as a "bag" of words. The

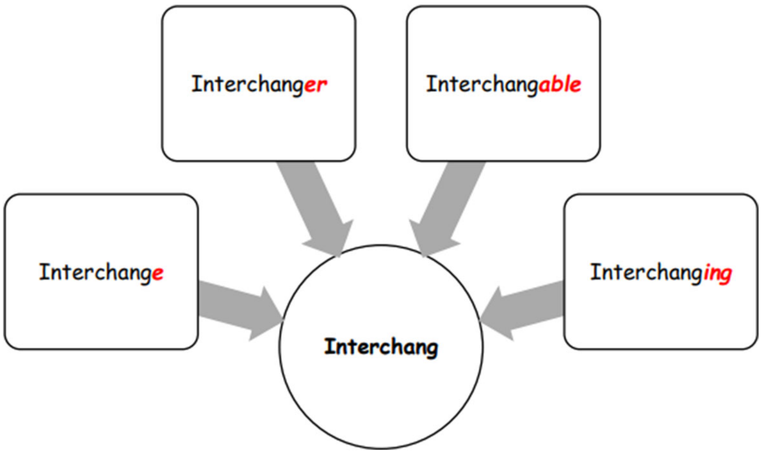
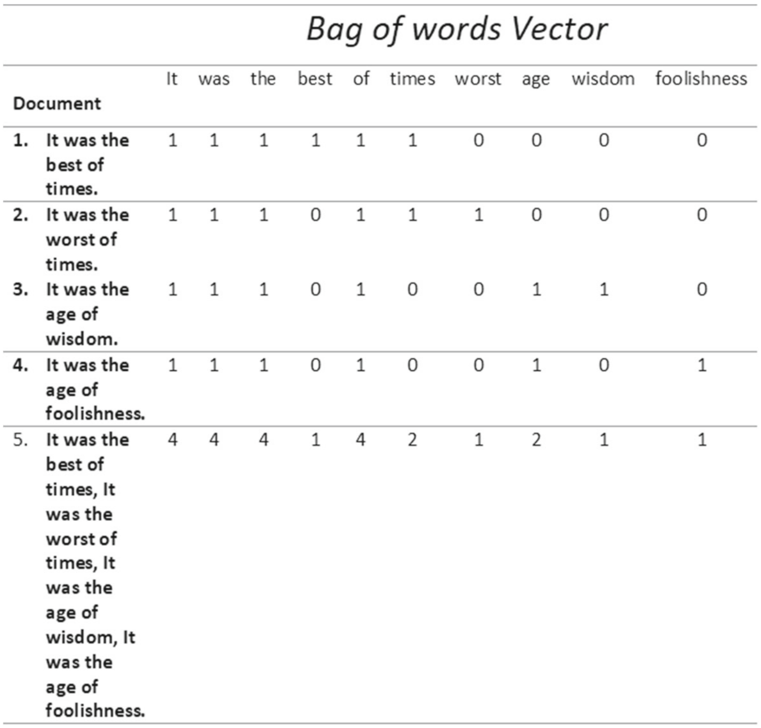


Fig. 3 Example of stemming



	Bag of words Vector									
Document	It	was	the	best	of	times	worst	age	wisdom	foolishness
1. It was the best of times.	1	1	1	1	1	1	0	0	0	0
2. It was the worst of times.	1	1	1	0	1	1	1	0	0	0
3. It was the age of wisdom.	1	1	1	0	1	0	0	1	1	0
4. It was the age of foolishness.	1	1	1	0	1	0	0	1	0	1
5. **It was the best of times, It was the worst of times, It was the age of wisdom, It was the age of foolishness.**	4	4	4	1	4	2	1	2	1	1

Fig. 4 Example of Bag of Words

model is only concerned with recognized terms appear in the document, not with where they appear. To identify the location of terms we use indexing.

Indexing The main goal of word indexing is to boost efficiency by extracting a set of terms from the final text document to index the text document. Word indexing entails choosing a suitable collection of keywords from a large number of documents and identifying those keywords for each text document, essentially customary documents, into such a vector of keyword weights.

The KE model employs preprocessing techniques to extract important keywords from textual data. Initially, the text undergoes tokenization, where it is segmented into individual words or tokens, establishing the fundamental units for analysis. Subsequently, stopwords are the common words removed to reduce noise and focus attention on more meaningful terms. Stemming techniques further refine the data by reducing words to their root or base form, capturing variations of the same word. Once preprocessed, the text is represented using the BoW model, which represents each document with a vector that indicates the frequency of each word in a predefined vocabulary. This representation allows for efficient analysis while disregarding word order. Finally, we organize and store the BoW representations using indexing techniques like inverted indices or term-document matrices, which enable quick access and retrieval of textual data. The effectiveness of the KE model relies on the quality of these preprocessing techniques and the suitability of the chosen algorithms. Overall, by employing these preprocessing techniques, the KE model enhances the quality of extracted keywords, enabling more accurate analysis and interpretation of textual data.

3.2 Multi-layered secret-sharing scheme

The MLSS ensures the security of extracted non-revealing text data through a multi-step process. Initially, the text undergoes preprocessing to the KE approach. Then, encryption techniques are applied to secure the extracted text data, ensuring confidentiality and integrity. Each word or token is encrypted by using a cryptographic algorithm. Next, the encrypted text data is divided into multiple shares using a SSS, distributing the shares among multiple parties to enhance security. XOR operations may be employed during encryption and reconstruction to securely combine data. This process ensures that the original text remains confidential and non-revealing, with reconstruction requiring the collaboration of multiple parties. In many cases, keeping a secret is necessary. A password, an encryption key, a secret message, and other items may need to be hidden. Encryption can help protect data, but it is also crucial to keep the secret key safe for encryption. Consider that you encrypt all of your critical files with a single secret key and if that key is lost, all of your important files are inaccessible. To overcome this issue, SSS allows you to divide your secret into multiple parts and distribute them to specific individuals [15].

A SSS is a methodology via which a vendor allocates shares to parties so that authorized subsets of those parties can only rebuild the secret or original data. These schemes are essential cryptographic methods that operate as the basis for various secure protocols, including the general protocol for multi- ple parties computation, threshold cryptography, Byzantine agreement, key control, and attribute-based enciphering [31]. The original secret text is recovered during decoding by combining the shares. Blakley and Shamir [33] extended this idea of SSS to a (k, n) system in which n represents the total numeral of shares and the minimum or required number of shares represented by k to retrieve the secret. Confidential details can be shared in a group through a SSS method without a single user having access to that secret. When required or more additional members of the grouping cooperate, the secret will be recovered. Any media, including video, text, images, audio, and numbers, can contain the secret.

There are several SSS approaches [9, 10, 15] available to manage the different types of multimedia data. For text data security, this work relies on XOR-based secret-sharing scheme methodology. MLSS scheme divides a secret input text into n+1 shares, where n represents the length of the text directed as shares. The proposed MLSS approach protects textual data. Figure 5 shows the general steps of the secret-sharing scheme. The size of the textual data is minimized once the pre-processing of the textual data is done. The keywords that were extracted are also the most informational or disclosing data. Consequently, ensuring safety is crucial for data or information security. Data security of text data is guaranteed using the secret sharing method at the primary level, and the secret share generated is communicated as a binary file at the secondary level.

Algorithm 3 describes the text encoding and share generation procedures within the Multi-layered Secret Sharing Scheme. The algorithm takes a set of secret keywords represented by variable W as input and generates (n+1) secret shares denoted by variable S using an MLSS scheme. Initially, each word W undergoes XOR encryption with a randomly generated

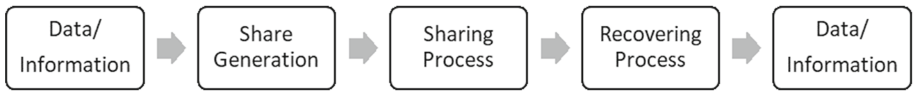


Fig. 5 General Secret Sharing Scheme Steps

Algorithm 3 Proposed Share Generation Scheme.

```
Input : n secret texts ($W_1$,$W_2$,$W_3$,$W_4$,....$W_n$)
Output : n+1 shared texts ($S_1$,$S_2$,$S_3$,$S_4$,....$S_(n + 1)$)

Share Creation
1. Binary random number generation
   R=random binary number(i)
2. Calculate ($X_1$,$X_2$,$X_3$,$X_4$,....$X_n$) using XOR operation
   For (i=1 to n):
   (a) $X_i$= $W_i \oplus$ R                    Where,(i=1,2,3,...n)
3. Creation of Shares ($S_1$,$S_2$,$S_3$,$S_4$,....$S_(n + 1)$)
   based on XOR operation
   (a) $S_1$= $X_1$
   (b) For (i=2 to n):
       $S_i$= $X_i \oplus X_(i - 1)$
   (c) For (i=n+1):
       $S_(n + 1)$= $W_1 \oplus X_n$
```

binary number R, ensuring unique encryption for each word. Afterwards, we generate shares by performing XOR operations between consecutive encoded words (X_i). The first share, S1, corresponds to the encoded representation of the first word W1, while the next shares (S2 to Sn) are computed as XOR combinations of consecutive encoded words. The final share, S(n+1), is generated by XORing the last encoded word with the original first word. This process generates (n+1) secret shares representing the original keywords, ensuring confidentiality and security in the sharing process.

Algorithm 4 proposes a secret reconstruction scheme that aims to recover the original secret texts from the (n+1) shared texts generated by the SSS. It follows a step-by-step process: first, intermediate texts are reconstructed from the shared texts through XOR operations. Then, the first secret text is reconstructed using the final shared text and the last intermediate text. Next, a binary random number is computed from the first intermediate text and the reconstructed first

Algorithm 4 Proposed Secret Reconstruction Scheme.

```
Input : n+1 shared texts ($S_1$,$S_2$,$S_3$,$S_4$,....$S_(n + 1)$)
Output : n secret texts ($W_1$,$W_2$,$W_3$,$W_4$,....$W_n$)

Share Reconstruction
1. Recover ($X_1$,$X_2$,$X_3$,$X_4$,....$X_n$) texts with the help of XOR operation
   (a) $S_1$= $X_1$
   (b) For (i=1 to n):
       $X_i$= $S_i \oplus X_(i - 1)$ where(i=1,2,3,4,....n)
   (c) For (i=n+1):
       $S_(n + 1)$= $W_1 \oplus X_n$
   (d) Reconstruct $W_1$ using
       $W_1 = S_(n + 1) \oplus X_n$
2. Find out binary random number using
   R= $X_1 \oplus W_1$                    Where,(i=1,2,3,...n)
3. Recovered shared texts ($W_1$,$W_2$,$W_3$,$W_4$,....$W_n$)
   using XOR operation
   $W_i$= $X_i \oplus$R                    Where,(i=2,3,4.....n)
```

secret text. Finally, the remaining secret texts are recovered by XORing each intermediate text with the binary random number. This process ensures the secure reconstruction of the secret texts while preserving confidentiality. The main advantages of the both proposed models are mentioned below:

- No necessity to deal with all words from proposed dataset.
- Without every one of the shares, the hacker is still unable to retrieve the secret.
- Shares generated of text comprise no information related to the original text.

All steps of MLSS are discussed in details below.

- Data / Information: Input for this MLSS model are text in which the input text is converted into binary numbers. The description of the input dataset is given in Table 3.
- Secret Generation: Share generation of input data is done on the basis of Algorithm 3. The proposed scheme is (n,n+1) secret share generation. For example, if the text has length 4 then generated share is 5 (n+1) where n=4. However, generated shares are represented by a binary number. An example of secret share generation using Algorithm 3 is explained in Table 7. The generated share does not reveal anything related to the original text.
- Sharing Process: Generated share is shared among the authorized group of persons. The secret can be reconstructed only when the authorized group agrees to reveal the secret.
- Recovery Process: Share reconstruction of text is done on the basis of Algorithm 4. The proposed scheme is (n+1,n) secret share reconstruction which means if the number of shares is 4 (n+1) then the original text length is 3. Also, the created shares are represented in binary numbers.
- Data / Information: In this step, generated binary shares are converted into original textual data, and Finally, original text data were successfully retrieved using the above steps.

The MLSS scheme offers unique advantages in providing text data security, including enhanced security through a multi-layered approach, power access management, robustness against single points of failure, scalability, flexibility, efficient key management, and significant contributions to the field of text data security, military, and defense systems. By safeguarding extracted sensitive information, these security approaches enable compliance with regulatory requirements, maintain individual privacy rights, and mitigate the risks associated with data breaches and unauthorized access.

4 Experimental result and discussion

This section evaluates the proposed model using the performance measures discussed in Section 4.1. Section 4.2 provides a detailed description of the datasets. We evaluated the proposed S2T model through the F1-score of the KE methodology and execution time for SSS scheme, as explained in Section 4.3.

Table 3 Overview of dataset

Dataset Type	Full Text Length (FTL) File (Char)	Full Text Length (FTL) File (Words)	Number
Whatsapp	03000	0740	300
Text Messages	04773	0836	400
E-mail	20929	3277	2000

4.1 Perfromance measure

The effectiveness of our proposed KE method alongside baseline models is evaluated using performance metrics: precision (P), recall (R), and F-score (F1). Prior research has widely used these metrics to assess the effectiveness of keyword extraction techniques [49–52]. We outline the equations for calculating precision, recall, and f1-score below.

$$P = \frac{\text{Correctly Identified Keywords}}{\text{All Extracted Keywords}} \tag{1}$$

$$R = \frac{\text{Correctly Identified Keywords}}{\text{All Text Keywords}} \tag{2}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{3}$$

4.2 Dataset used

The comparative analysis utilizes three standard datasets (INSPEC, SemEval2017 & Krapiv) with different sizes and domains to evaluate the proposed method against several benchmark models. Also we proposed datasets for this task.

- INSPEC: The first dataset, as utilized in [53], comprises 2000 English abstracts sourced from the Inspec database. These abstracts are segmented into three subsets: a training set, a validation set, and a test set, each containing 1000, 500, and 500 abstracts, respectively.
- SemEval2017: The second dataset, constructed by Kim et al. [54] designed for the keyphrase extraction task as part of the SemEval evaluation campaign. This dataset consists of 284 scientific articles written in English, sourced from the ACM Digital Libraries, encompassing conference and workshop papers. The 284 documents are segmented into three subsets: a trial set comprising 40 documents, a training set containing 144 documents, and a test set comprising 100 documents.
- Krapivin: The third dataset, constructed by Krapivin et al. [55] dataset comprises 2,304 full-text scientific articles from the computer science domain, published by the Association for Computing Machinery (ACM). Author-assigned and editor-corrected keyphrases for each article are given.

Additionally, dataset has been created manually for this research work. Three various types of text categorical data are included. The primary dataset encompasses whatsapp messages (text), the secondary dataset comprises text messages, and the third dataset type includes details of e-mails of bank transactions. We compared the KE model with our proposed dataset and three standard datasets INSPEC, SemEval2017, and Krapivin [46–48]. Table 3 indicates a detailed description of the proposed dataset in which the model has been executed using the three separate datasets.

4.3 Results and discussion

The proposed model has been designed by considering bank applications. So, we have kept the keywords (customer name, address, mobile no. etc.) manually, which are application-oriented. RAKE model extracts keyphrases or word phrases from text documents which is an inefficient way of extracting keywords for the banking domain. All the manually decided keywords are nearly available in the KE model. Due to this KE model is more efficient for

Table 4 Comparison of the proposed KE model with yake [4] and rake model [8]

Dataset Type	Full Length File (FLF)	Extracted Keyword using proposed KE	Extracted Keyword using YAKE [4]	Extracted Keyphrases using RAKE [8]
Whatsapp	0740	138	053	286
Text Messages	0836	154	107	456
E-mail	3277	208	126	1336

bank applications. The proposed KE model extracts more efficient words from a text file than the YAKE and RAKE models. Table 4 shows the keyword or essential word extraction result from the KE model compared with the YAKE model [17] and RAKE model [8].

Table 5 shows our proposed dataset performance based on the KE model with a detailed description. Table 6 represents the results of the F1-scores for KE and the baseline models on the three datasets. The outcomes show that the KE model performs better on almost all evaluation metrics across all three datasets, indicating the effectiveness of the proposed method.

Table 7 demonstrates the output of the proposed MLSS-based model employed to generate shares for text with the help of Algorithm 3 for text data having a length of three and shares equal to four. Each share is equal to the length of the word plus one.

Table 8 illustrates three different texts utilized to execute Algorithm 3, a method for creating secret shares. The outcome shows that similarity values between generated shares and actual text are not matched. The findings indicate that every share is distinct and not revealing anything related to the secret text. Additionally, the 100% correlation between the actual and reconstructed words shows that no data has been lost.

Table 9 shows the average execution time (in seconds) needed to construct (n+1) shares by utilizing Algorithm 3, and Table 10 shows the result of reconstructing the secret using (n+1) shares using Algorithm 4. Both tables contain seven different text files with varying text length (characters) [17]. The outcome of these two tables shows that our proposed algorithms perform better than the existing Polynomial-Based SSS for Text [17] approach in terms of execution time. Additionally, Tables 9 and 10 show that the execution time for share reconstruction is less than share generation.

In Table 11, we compare the results of our KE model with the YAKE and RAKE model on our proposed three types of a dataset, as shown in Table 3. The execution time (in seconds) for KE needed to produce (n+1) shares and uses that (n+1) shares to reconstruct the secret. From Table 11, it is observed that:

- KE requires less execution time to generate shares for extracted keywords than generation shares for the overall Full-Length-Text Files.

Table 5 Dataset performance based on proposed keyword-extraction model

DESCRIPTION	E-MAIL	MESSAGES	WHATSAPP
Full-Text FiLe (Char)	20929	4773	3000
Full-Text FiLe (Words)	2377	836	740
Vocab Size	1798	430	280
Tokens Words	2519	585	585
Filtered Tokens Words	1741	419	270
Posting List	689	351	346
Stopwords	179	179	179
Extracted Words	208	154	138

Table 6 Performance of the KE model as an F1-score on three benchmark datasets

MODEL	INSPEC	SemEval2017	Krapivin
TextRank [47]	27.62	30.50	17.32
SingleRank [48]	24.13	31.73	18.17
TopicRank [49]	19.33	24.87	15.85
MultipartiteRank [50]	20.52	26.87	17.37
YAKE [3]	13.65	20.55	13.87
EmbedRank(BERT) [51]	39.77	36.72	20.90
SIFRank(ELMo) [37]	39.82	37.25	22.05
MDERank(BERT) [52]	36.17	37.18	23.85
PromptRank(T5) [39]	38.17	41.57	28.04
Proposed	**41.0**	**42.0**	**30.0**

Best results are shown in bold

Table 7 Secret share generation using MLSS scheme

Sample Text	Share 1	Share 2	Share 3	share 4
NLP	01001111	00000010	00011100	00011111
Txt	01010101	00101100	00001100	00100001
SSS	01010010	00000000	00000000	00000001
Ykw	01011000	00110010	00011100	00101111

Table 8 Similarity score for text using SSS

Sample Text	Share1	Share2	Share3	Reconstructed text
Text 1	1.0000	1.0000	1.0000	1.0000
Text 2	1.0000	1.0000	1.0000	1.0000
Text 3	1.0000	1.0000	1.0000	1.0000

Table 9 Average execution time for text by using MLSS scheme (SHARE CREATION) (seconds)

Text Type	Text Length (char)	Share Generation [17]	Proposed Share Generation
Text 1	10	0.004854	**0.00099**
Text 2	374	0.006000	**0.00199**
Text 3	3391	0.015160	**0.01399**
Text 4	26,382	0.079933	**0.10846**
Text 5	70,352	0.381000	**0.27921**
Text 6	246,232	2.159500	**0.95673**
Text 7	378,142	3.435500	**1.29432**

Best results are shown in bold

Table 10 Average execution time for text by using MLSS scheme (SHARE CREATION) (seconds)

Text Type	Text Length (char)	Share Reconstructiom [17]	Proposed Share Reconstructiom
Text 1	10	0.00379	**0.00069**
Text 2	374	0.00496	**0.00099**
Text 3	3391	0.00533	**0.01099**
Text 4	26,382	0.04916	**0.06598**
Text 5	70,352	0.24123	**0.20432**
Text 6	246,232	0.57085	**0.04890**
Text 7	378,142	0.63653	**0.52093**

Best results are shown in bold

- KE requires less execution time to reconstruct shares for extracted keywords than reconstruct shares for the overall Full-Length-Text Files.
- Extracted Keywords and Full-Length Files (FLF) results show that reconstruction of the secret takes a shorter time than the construction of the shares.
- Results on the E-mail dataset shows that the KE model takes less computation time as compared to YAKE and RAKE model.
- Results on the Whatsapp dataset show that the KE model takes less computation time as compared to YAKE and RAKE models.
- Results on the Text messages dataset show that the KE model takes less computation time as compared to YAKE and RAKE models.
- KE model takes less execution time for secret share generation and reconstruction on three different datasets when compared to the YAKE and RAKE models.
- YAKE model takes less execution time for secret share generation and secret share reconstruction as compared to the RAKE model.

4.4 Discussion

Securing extracted keywords holds the main significance in ensuring the integrity and reliability of various applications such as information retrieval, NLP, and data mining. Unauthorized

Table 11 Average execution time for text using MLSS scheme (SHARE CREATION & share reconstruction) (seconds)

Data Type	Text Length (char)	Share Generation(s)	Share Reconstruction (s)
Email (FLF)	20929	0.08644	0.06373
Email (KE)	01770	0.00699	0.00599
Email (Yake)	01839	0.00799	0.00597
Email (Rake)	14875	0.06270	0.04761
Whatsapp (FLF)	04773	0.01896	0.01497
Whatsapp (KE)	01219	0.00499	0.00399
Whatsapp (Yake)	01302	0.00599	0.00415
Whatsapp (Rake)	04089	0.02198	0.01299
Messages (FLF)	03000	0.01198	0.00899
Messages (KE)	00379	0.00199	0.00185
Messages (Yake)	0.00285	0.00199	0.00192
Messages (RAKE)	03758	0.02498	0.01257

access to or manipulation of keywords could not only undermine the accuracy of search results but also pose significant risks to data privacy and system security. Additionally, the presence of sensitive information within keywords necessitates robust measures to prevent unauthorized disclosure or misuse. The proposed KE model consistently outperforms in accurately identifying and extracting essential keywords from the text corpus based on the F1-score. Secondly, in terms of text share generation and reconstruction, our proposed model exhibits enhanced performance compared to the baseline method. The proposed MLSS model achieves faster share generation times and reconstruction times without compromising the quality of the reconstructed text due to the use of XOR operations instead of complex algorithms for encryption. This improvement is particularly significant in real-time applications where rapid text share generation and reconstruction are important. The significance of the results in Tables 5-11 shows that our proposed model S2T is better than other methods of keyword extraction, share generation, and reconstruction time. Overall, the significance of our results lies in demonstrating the effectiveness and practical utility of our proposed model across multiple dimensions, including keyword extraction, summary generation, share generation time, and reconstruction time.

4.5 Limitations

The limitations of securing extracted keywords, particularly within the context of the proposed S2T model, can include the following:

- Loss of Contextual Information: Securing extracted keywords can potentially lose contextual information in the original text, as they are condensed representations of the underlying text, further abstracting away valuable context. The loss of context could potentially affect the interpretability and usefulness of secured keywords, especially in tasks requiring context, such as NLP or sentiment analysis.
- Semantic Ambiguity: Keywords in text can be ambiguous and misinterpreted, so securing them without considering their semantic context can lead to misunderstandings or misinterpretations.
- Limited Information Retention: Securing keywords involves reducing original text content to essential terms, enhancing security but discarding non-keyword information. This limitation may restrict tasks that require a comprehensive understanding of the text, such as document summarization or information retrieval.
- Dependency on Keyword Extraction Accuracy: The accuracy of the keyword extraction process is crucial for securing extracted keywords, as a flawed algorithm may not accurately represent the underlying text. Ensuring the robustness and accuracy of the keyword extraction process is crucial for maintaining the integrity and security of the secured keywords.
- Vulnerability to Adversarial Attacks: Securing keywords may introduce vulnerabilities to adversarial attacks aimed at manipulating or perturbing the extracted terms. Adversaries could exploit weaknesses in the keyword extraction algorithm or the security mechanisms used to protect the keywords, leading to the compromise of sensitive information or misleading terms.
- Scalability and Efficiency Concerns: Securing a large volume of extracted keywords from massive text datasets may pose scalability and efficiency challenges. The computational resources and processing overhead required to encrypt, transmit, and store a large number of keywords securely could be substantial, especially in real-time or high-throughput applications.

Addressing these limitations requires careful consideration of the trade-offs between security, usability, and performance in the design and implementation of keyword security mechanisms. It also highlights the importance of incorporating context awareness, robustness, and efficiency into keyword extraction and security processes to mitigate potential risks and maximize the utility of secured keywords. Future work for securing extracted keywords could focus on addressing the identified limitations and advancing the capabilities of keyword security mechanisms.

Overall, future work on securing extracted keywords should aim to advance the state-of-the-art in keyword security by addressing key challenges, enhancing semantic understanding, improving adversarial robustness, ensuring scalability and efficiency, integrating with privacy-enhancing technologies, empowering users with security controls, and establishing evaluation frameworks for comprehensive assessment and comparison.

5 Conclusion

We presented an unsupervised KE model with SSS to provide text security for sensitive data rather than overall data. The effectiveness of the S2T model enhances text data security through keyword extraction methodology and a secret sharing scheme. For this research, we have proposed three datasets, including text messages, whatsapp messages, and electronic mail. Initially, we preprocess text datasets using techniques such as stopword removal, tokenization, stemming, bag of words, and indexing to extract important data. Subsequently, we employ the KE model to extract keywords relevant to banking applications. The result analysis of the KE model showed its effectiveness on three publicly available datasets (inspec, semeval2017, and krapivins) as well as on our proposed datasets. The result analysis indicates a notable reduction in numeral words and includes more informative keywords through the KE model. Also, the KE model takes less execution time for secret share generation and secret share reconstruction than the existing methodology. Furthermore, we compare the execution times of the total text file with and without the extracted keywords based on MLSS. The findings highlight that our proposed MLSS model significantly improves the execution time of the KE model while ensuring a secure system environment. Experimental result analysis shows the efficiency and effectiveness of our proposed S2T scheme in enhancing text data security through optimized keyword extraction techniques.

For future research, the SSS-based approach will be able to reduce the amount of text data shares generated and expand the preprocessing of that text data using various deep learning techniques, which perform better when the text size is enhanced.

Author Contributions All the authors contributed equally.

Availability of data and materials The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interest None

References

1. Aggarwal CC, Zhai CX (2012) An introduction to text mining. Springer, Boston, MA, pp 1–10
2. Kannan S, Gurusamy V, Vijayarani S, Ilamathi J, Nithya M, Kannan S, Gurusamy V (2014) Preprocessing techniques for text mining. Int J Comput Sci Commun Netw 5(1):7–16
3. Campos R, Mangaravite V, Pasquali A, Jorge A, Nunes C, Jatowt A (2020) YAKE! Keyword extraction from single documents using multiple local features. Inf Sci 509:257–289
4. Campos R, Mangaravite V, Pasquali A, Jorge AM, Nunes C, Jatowt A (2018) Yake! collection-independent automatic keyword extractor. In: Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40, pp. 806-810. Springer International Publishing, rake
5. Rose S, Engel D, Cramer N, Cowley W (2010) Automatic keyword extraction from individual documents. Text mining: applications and theory pp 1-20
6. Borisov O, Aliannejadi M, Crestani F (2021) Keyword extraction for improved document retrieval in conversational search. Preprint at arXiv:2109.05979
7. Qian Y, Jia C, Liu Y (2021) BERT-based text keyword extraction. In: Journal of Physics: Conference Series, vol. 1992, no. 4, IOP Publishing, p 042077
8. Rinartha K, Kartika LGS (2021) Rapid automatic keyword extraction and word frequency in scientific article keywords extraction. In: 2021 3rd International conference on cybernetics and intelligent system (ICORIS), pp 1-4. IEEE
9. Kashid S, Kumar K, Saini P, Negi A, Saini A (2022) Approach of a multilevel secret sharing scheme for extracted text data. In: 2022 IEEE Students conference on engineering and systems (SCES), pp 1-5. IEEE
10. Kashid S, Awasthi LK, Kumar K, Saini P (2023) NS4: a novel security approach for extracted video keyframes using secret sharing scheme. In: 2023 International conference on computer, electronics & electrical engineering & their applications (IC2E3), pp 1-6. IEEE
11. Cirillo S, Desiato D, Scalera M, Solimando G (2023) A visual privacy tool to help users in preserving social network data. In: IS-EUD Workshops
12. Adnan G, Aziz A, Alaseri K (2021) Refining Arabic text stego-techniques for shares memorization of counting-based secret sharing. J King Saud Univ - Comput Inf Sci 33(9):1108–1120
13. Zheng W, Wang K, Wang F-Y (2021) GAN-Based key secret-sharing scheme in blockchain. IEEE Trans Cybern 51(1):393–404. https://doi.org/10.1109/TCYB.2019.2963138
14. Esraa A, Gutub A (2022) Novel arabic e-text watermarking supporting partial dishonesty based on counting-based secret sharing. Arab J Sci Eng 47(2):2585–2609
15. Saini P, Kumar K, Kashid S, Negi A (2022) MEVSS: Modulo encryption based visual secret sharing scheme for securing visual content. In: International conference on deep learning, artificial intelligence and robotics, pp 24-35. Cham: Springer International Publishing,
16. Luo Y, Yao C, Mo Y, Xie B, Yang G, Gui Huiyang (2021) A creative approach to understanding the hidden information within the business data using Deep Learning. Inf Process Manage 58(5):102615
17. Guttikonda P, Mundukur NB (2020) Polynomial-based secret sharing scheme for text, image and audio. J Inst Eng (India) B 101(5):609–621
18. HaCohen-Kerner Y, Miller D, Yigal Y (2020) The influence of preprocessing on text classification using a bag-ofthe -words representation. PloS One 15(5):e0232525
19. Feng D, Chen H (2021) A small samples training framework for, deep Learning-based automatic information extraction: case study of construction accident news reports analysis. Adv Eng Inform 47:101256
20. Abidin DZ, Nurmaini S, Malik RF, Rasywir E, Pratama Y (2019) A model of preprocessing for social media data extraction. In: 2019 International conference on informatics, multimedia, cyber and information system (ICIMCIS), pp 67-72. IEEE
21. Kadhim AI (2018) An evaluation of preprocessing techniques for text classification. IJCSIS 16(6):22–32
22. Janpitak N, Sathitwiriyawong C, Pipatthanaudomdee P (2019) Information security requirement extraction from regulatory documents using GATE/ANNIC. 2019 7th International electrical engineering congress (iEECON). IEEE
23. Haddi E, Liu X, Shi Y (2013) The role of text pre-processing in sentiment analysis. Procedia Comput Sci 17:26–32
24. Yu W, Lu N, Qi X, Gong P, Xiao R (2021) PICK: processing key information extraction from documents using improved graph learning-convolutional networks. In: 2020 25th International conference on pattern recognition (ICPR), pp 4363-4370. IEEE
25. Yang Z, Wang K, Li J, Huang Y, Zhang Y-J (2019) TS-RNN: Text steganalysis based on recurrent neural networks. IEEE Signal Process Lett 26(12):1743–1747
26. Papagiannopoulou E, Tsoumakas G (2020) A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10(2):e1339

27. Florescu C, Caragea C (2017) Positionrank: an unsupervised approach to keyphrase extraction from scholarly documents." In Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), pp 1105–1115
28. Yadollahi MM, Lashkari AH, Ghorbani AA (2021) Towards query-efficient black-box adversarial attack on text classification models. In: 2021 18th International conference on privacy, security and trust (PST), pp 1-7. IEEE
29. Gutub A, Alaseri K (2020) Hiding shares of counting-based secret sharing via Arabic text steganography for personal usage. Arab J Sci Eng 45(4):2433–2458
30. Phiri KK, Kim H (2019) Linear secret sharing scheme with reduced number of polynomials. Security and Communication Networks 2019
31. Khan AA, Shaikh AA, Cheikhrouhou O, Laghari AA, Rashid M, Shafiq M, Hamam H (2022) IMG-forensics: Multimedia-enabled information hiding investigation using convolutional neural network. IET Image Process 16(11):2854–2862
32. Saini P, Kumar K, Kashid S, Dhiman A, Negi A (2022) BEMSS- Blockwise Encryption based multi secret sharing scheme for securing visual content. 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp 1-6. https://doi.org/10.1109/UPCON56432.2022.9986417
33. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
34. Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) Text summarization techniques: a brief survey. Preprint at arXiv:1707.02268
35. Gambhir M, Gupta V (2017) Recent automatic text summarization techniques: a survey. Artif Intell Rev 47(1):1–66
36. El-Kassas WS, Salama CR, Rafea AA, Mohamed HK (2021) Automatic text summarization: a comprehensive survey. Expert Syst Appl 165:113679
37. Sun Y, Hangping Qiu Y, Zheng ZW, Zhang C (2020) SIFRank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model. IEEE Access 8:10896–10906
38. Hasan HMM, Sanyal F, Chaki D (2018) A novel approach to extract important keywords from documents applying latent semantic analysis. In: 2018 10th International conference on knowledge and smart technology (KST), pp 117-122. IEEE
39. Kong A, Zhao S, Chen H, Li Q, Qin Y, Sun R, Bai X (2023) PromptRank: Unsupervised keyphrase extraction using prompt. Preprint at arXiv:2305.04490
40. Kashid S, Kumar K, Saini P, Dhiman A, Negi A (2022) Bi-RNN and Bi-LSTM based text classification for amazon reviews. In: International conference on deep learning, artificial intelligence and robotics, pp 62-72. Cham, Springer International Publishing
41. Kumbhar A, Savargaonkar M, Nalwaya A, Bian C, Abouelenien M (2019) Keyword extraction performance analysis. In: 2019 IEEE Conference on multimedia information processing and retrieval (MIPR), pp 550-553. IEEE
42. Kim SN, Medelyan O, Kan M-Y, Baldwin T (2013) Automatic keyphrase extraction from scientific articles. Lang Resour Eval 47:723–742
43. Nomoto T (2022) Keyword extraction: a modern perspective. SN Comput Sci 4(1):92
44. Liao S, Yang Z, Liao Q, Zheng Z (2023) TopicLPRank: a keyphrase extraction method based on improved TopicRank. J Supercomput 1-20
45. Kılıç Ünlü H, Çetin A (2023) Keyword extraction as sequence labeling with classification algorithms. Neural Comput Appl 35(4):3413–3422
46. Delgado-Solano IP, Nunez-Varela AS, Perez-Gonzalez GH (2018) Keyword extraction from users' requirements using textrank and frequency analysis, and their classification into ISO/IEC 25000 quality categories. In: 2018 6th International conference in software engineering research and innovation (CONISOFT), pp 88-92. IEEE,
47. Mihalcea R, Tarau P (2004) Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing, pp 404-411
48. Wan X, Xiao J (2008) Single document keyphrase extraction using neighborhood knowledge. In: AAAI vol 8, pp 855–860
49. Bougouin A, Boudin F, Daille B (2013) Topicrank: graph-based topic ranking for keyphrase extraction. In: International joint conference on natural language processing (IJCNLP), pp 543-551
50. Boudin F (2018) Unsupervised keyphrase extraction with multipartite graphs. Preprint at arXiv:1803.08721
51. Bennani-Smires K, Musat C, Hossmann A, Baeriswyl M, Jaggi M (2018) Simple unsupervised keyphrase extraction using sentence embeddings. Preprint at arXiv:1801.04470
52. Zhang L, Chen Q, Wang W, Deng C, Zhang S, Li B, Wang W, Cao X (2021) MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction. Preprint at arXiv:2110.06651

53. Hulth A (2003) Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing, pp 216-223
54. Kim SN, Medelyan O, Kan M-Y, Baldwin T (2013) Automatic keyphrase extraction from scientific articles. Lang Resour Eval 47:723–742
55. Krapivin M, Autaeu A, Marchese M (2009) Large dataset for keyphrases extraction pp 1-4