National College of Ireland

An Enhanced Version of Data Classification based on Confidentiality for Cloud Security

MSc Research Project
Cloud Computing

Kirthikesh Parthasarathy
Student ID: 21195391

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Kirthikesh Parthasarathy
Student ID:	21195391
Programme:	Msc in Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	14/12/2023
Project Title:	An Enhanced Version of Data Classification based on Confidentiality for Cloud Security
Word Count:	6349
Page Count:	28 LastPage

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Kirthikesh Parthasarathy
Date:	29th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Enhanced Version of Data Classification based on Confidentiality for Cloud Security

Kirthikesh Parthasarathy
21195391

Abstract

The rapid proliferation of cloud technologies has motivated the organizations to store the data over cloud platforms since they offer high scalability and better performance in terms of software as a service. However, the increased deployment of cloud services has also increased the need for ensuring the protection of sensitive information stored in cloud servers. Privacy protection has become one of the critical aspects for various organizations that move their data to the cloud. Since the data can be of different types, the security requirements for data protection also vary. The crucial issue of securing data in cloud environments is addressed in this work by deploying an effective classification framework. This paper presents the design of a unique classification framework for securing the confidential data stored in the cloud. The classification model is developed in this work using the Random Forest (RF) classifier and the model is trained using the data features. The essential features are extracted using a hybrid CNN-LSTM model and a K-means SMOTE algorithm is used for addressing the class imbalance issues. Furthermore, the trained model is deployed into a Container as a Service (CaaS) environment and the deployed model is known as AUG-ConvoLSTM-RF. The model combines both data augmentation and Natural Language Processing (NLP) techniques for accurately classifying the data as confidential and non-confidential. The efficacy of the AUG-ConvoLSTM-RF model was experimentally evaluated and results show that the model exhibits an excellent classification accuracy of 84.36 % compared to other existing models.

***Keywords*— Cloud Security, Confidentiality, Data Classification, Data Augmentation, Natural Language Processing, CNN, LSTM, SPECK Algorithm**

Table of Contents

Contents

1 Introduction

In recent years, the increased deployment of cloud computing has revolutionized the storage and processing capabilities for organizations across diverse sectors. The inherent advantages of scalability, accessibility, and cost-efficiency have propelled the widespread adoption of cloud services. Different cloud services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) provide data security to the organizationsGeorge and Sagayarajan (2023). Although these services offer excellent data storage and processing abilities, the security and confidentiality of the data poses a significant threat to the adaptation of cloud services. The critical challenges associated with cloud security pertains to safeguarding sensitive information against unauthorized access and breachesAbdulsalam and Hedabou (2021). In addition, the nature of cloud infrastructures, characterized by shared resources and remote data storage, makes them more susceptible to data leakage. It requires robust protective measures to maintain the confidentiality of cloud data. Among these measures, the classification of data based on confidentiality emerges as a pivotal strategy to fortify the security posture within cloud ecosystems. One of the potential tools to secure the stored data from illegal authorities is data encryption. Encrypting the data stored in the cloud by understanding the security requirements of different data types requires more resources and this increases the computational burden. Considering this problem, there is a need to develop a confidentiality based classification model which is resource efficient. One such model is proposed in. The primary objective of this model was to classify different data security levels in order to prevent high computational burden on the cloud servers. Since the data in the cloud is remotely stored by the third party entities, it is vulnerable to security breaches and data leakage. It is highly challenging to identify and predict the security levels for all data types without thoroughly analyzing the security requirements. In this context, there is a great demand for a classification model which can categorize the security levelsKumar and Bhatia (2020). The two major categories for classifying data security levels are confidential and non-confidential data. Before designing a security approach for data security, the model must identify the security level; there are chances that the model might protect non-confidential data instead of confidential data. In order to mitigate such cases, this research proposes a data classification model based on the confidentiality level of cloud data.

Designing an accurate classification model in a heterogeneous cloud environment can be a complex task due to the constant evolution of the attacks on the data security. It is not practically feasible to depend on fundamental techniques to classify the data. In this case, machine learning (ML) techniques can be a potential tool for classifying the level of the data confidentiality since they can automatically classify the data using their self-learning mechanism. This research presents a novel approach which ensures secure data classification in cloud computing.

The contributions of this work are outlined as follows:

- A novel classification model known as AUG-ConvoLSTM-RF is designed in this paper for classifying confidential and non-confidential data stored in cloud servers.
- The AUG-ConvoLSTM-RF model is trained using the standardized data features that are extracted using the hybrid CNN-LSTM model and the model is deployed in a CaaS environment.

- The proposed model integrates data augmentation and NLP techniques for achieving a better data classification accuracy. In addition, the issue of data imbalance is addressed using a K-means SMOTE algorithm which ensures that all feature classes are equally balanced.
- The classification performance of the AUG-ConvoLSTM-RF model is evaluated and is compared with other ML models such as KNN, Naive Bayes (NB), Linear Regression, Gaussian NB, and Decision TreeChowdhury and Schoen (2020).

1.1 Research Question

How to enhance data security in the cloud by utilizing lightweight cryptographic using advanced techniques with respect to high volume of data?

This study flows accordingly in the following sections: Section 2 of the study reviews existing methodologies for data classification in the cloud environment. Section 3 briefs the design of the proposed classification model for cloud security along with the details of the implementation process. Section 4 discusses the experimental analysis and simulation outcomes. Section 5 is the conclusion of the research with prominent inferences and future scope.

1.2 Ethics Consideration

The ethical statement is included in Table 1, which contains relevant information. The study utilizes the IoT Healthcare Security Dataset supplied from the Kaggle website. The documentation pertaining to copyright concerns, Terms of Use, and Privacy Policy of the firm has been provided as follows:

- IoT Healthcare Security Dataset
- Copyright Dispute Policy
- Kaggle Terms of Use
- Kaggle Privacy Policy

Table 1: Table of Ethics Consideration Declaration

This project involves human participants	Yes / No
The project makes use of secondary dataset(s) created by the researcher	Yes / No
The project makes use of public secondary dataset(s)	Yes / No
The project makes use of non-public secondary dataset(s)	Yes / No
Approval letter from non-public secondary dataset(s) owner received	Yes / No

2 Literature Review

2.1 Data Augmentation (DA) techniques in Natural Language Processing

Data Augmentation techniques in Natural Language Processing (NPL). The authors highlight the main research gaps in the literature, including the lack of comparative studies and the need for a theoretical framework to support and explain DA methods in NLP. The authors also conducted an in-depth investigation of some of the most important NLP DA techniques, comparing their output and relative performance under various settings. The authors explored the relationship between DA, an implicit regularization technique, and an explicit regularization technique, namely the dropout procedure. The authors noted that NLP DA works rarely address any topic beyond model performance and that qualitative evaluations of the artificially generated data receive little to no attention. They suggested that enhancing the understanding of DA methods in NLP, both in terms of qualitative assessment and practical implementation, constitutes an important research agenda. The authors proposed several research directions, including the development of a policy optimization algorithm with lexical diversity and semantic fidelity, and the use of more formal theoretical tools or systematic procedures in DA NLP researchPellicer et al. (2023).

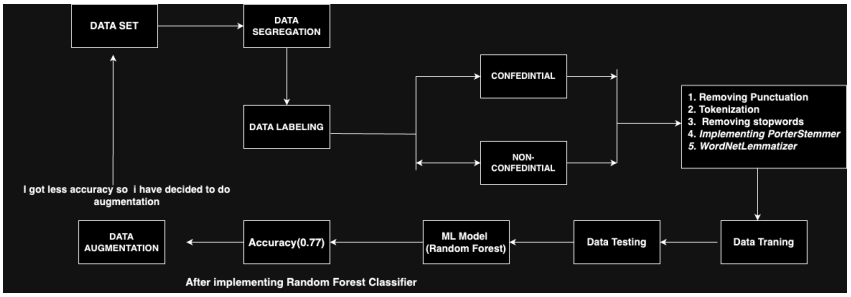


Figure 1: Architecture diagram for "Data Augumentation"

Data augmentation (DA) is approached from various angles to improve how well machine learning models work, especially in the context of (NLP). The authors explore a range of DA techniques, each with its own methodology for generating synthetic data that preserves the original labels and characteristics of the dataset. One common DA method mentioned is back-translation, which transfers words from one language to another and then back to the original language to create a paraphrased version of the text. This technique has been particularly successful in machine translation tasks and is one of the few NLP DA methods that have been widely adopted. Another approach discussed is the use of meta learning for data augmentation. This involves learning the most appropriate transformations from the training data itself, aiming to achieve better generalization power. Meta learning DA techniques are designed to overcome the limitations of handcrafted DA transformations.

2.2 An Inclusive Method Combining Lexical Analysis, Lemmatization, and Machine Learning Techniques

The rapid expansion of social media platforms like Twitter has led to an explosion of user-generated content, which presents both opportunities and challenges for sentiment analysis. Sentiment analysis aims to interpret the opinions, interests, and perspectives expressed in such content, which can be valuable across various academic and practical applications. Pre-processing is a key point in sentiment analysis, which is essential for cleaning and normalizing the data before feature extraction. Twitter language contains unique elements such as usernames, links, and hashtags that may not be relevant to the classification process and thus need to be removed or normalized. Lexical analysis is employed during pre-processing to reduce different word forms. Lemmatization is the process of converting words to their base or dictionary form. The WordNet Lemmatizer, which is part of the Natural Language Toolkit (NLTK), is a widely used tool for this purpose, despite not having a built-in lemmatization featureSaranya and Usha (2023). It relies on an English lexical database that organizes semantic relationships between words. The proposed solutions involve using machine learning techniques, such as Random Forest (RF) for sentiment classification, and Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction.

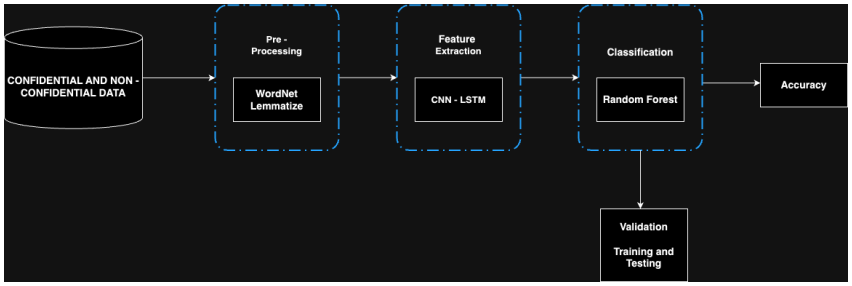


Figure 2: Workflow for Lemmatization

2.3 A Comparative Analysis of Stemming Algorithms

Stemming algorithms perform an important part in both natural language processing (NLP) and information retrieval (IR) systems. Their main purpose is to reduce words to their fundamental or basic form, commonly known as the stem. This procedure facilitates combining of several morphological variations of a word, hence improving the efficiency of text processing and information retrievalPolus and Abbas (2021). Although the Porter stemmer is commonly employed, it possesses many constraints. For example, the term "general" might be reduced to "gener" by removing the suffix "al." However, this process does not maintain the original meaning or grammatical structure of the word. In addition, the Porter algorithm lacks the ability to handle prefixes and is incapable of handling irregular word forms.

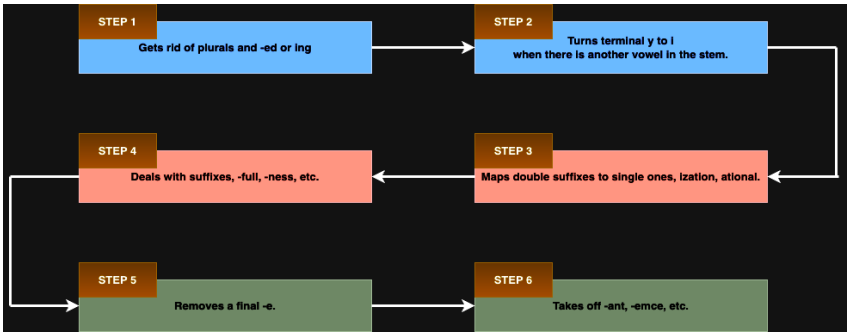


Figure 3: Workflow for Stemming Algorithm

2.4 An Deep Analysis of K-means Synthetic Minority Oversampling Technique (K-means SMOTE)

The literature review within the context of the paper under discussion focuses on the challenges and methodologies associated with Land Use/Land Cover (LULC) classification, especially when working with imbalanced datasets. Imbalanced Issues with datasets are prevalent. remote sensing and machine learning, where some classes (the minority classes) are underrepresented in comparison to others (the majority classes), leading to biased classifier performance towards the majority classes. To address the imbalanced learning problem, the paper discusses the use of oversampling techniques, which aim to balance the class distribution by artificially generating new samples for the minority classes. Among these techniques, the K-means Synthetic Minority Oversampling Technique (K-means SMOTE) is highlighted for its ability to improve the quality of newly generated digital information. K-means SMOTE not only addresses the classic oversamplers address the between-class imbalance, but they do not consider the within-class imbalance. It avoids the creation of noisy data and efficiently addresses data imbalance by creating synthetic samples that are more representative of the true underlying distribution of the minority classesFonseca et al. (2021).

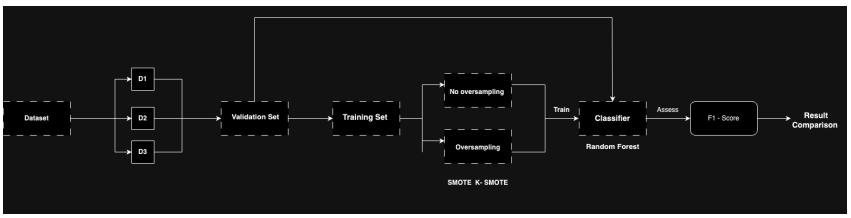


Figure 4: Workflow for SMOTE algorithm

2.5 Techniques for Dynamic Stopword Identification to Improve Natural Language Processing: Implementation in Text Classification and Information Retrieval Applications

Stopwords are a fundamental concept in natural language processing (NLP), often removed during the preprocessing phase to improve the performance of various applications

such as text classification (TC) and information retrieval (IR). The removal of stopwords can reduce the corpus size and improve metrics like precision, recall, and accuracy, as well as reduce space and time complexity for searching and indexing. Researchers have developed various techniques for identifying stopwords. These techniques range from simple frequency-based methods to more complex statistical models and knowledge-based approaches. For instance, employed a static approach using deterministic finite automata (DFA) for Arabic language stopword identification, while used an information theory model for Hindi. Studies have shown that stopword removal can significantly impact the performance of TC and IR systems. Demonstrated that removing stopwords from the Reuter-21578 corpus could substantially reduce the feature space and improve text classification accuracy. Similarly, Gunasekaran et al. applied statistical stopword removal techniques to Sinhala news classification and observed improvements in the average F-measure and accuracy. The literature suggests that while stopword removal is beneficial for NLP tasks, there is a need for more research on dynamic stopword identification techniques, especially for domain-specific applications and less-studied languages. The static approach may not handle new or out-of-vocabulary stopwords effectively.

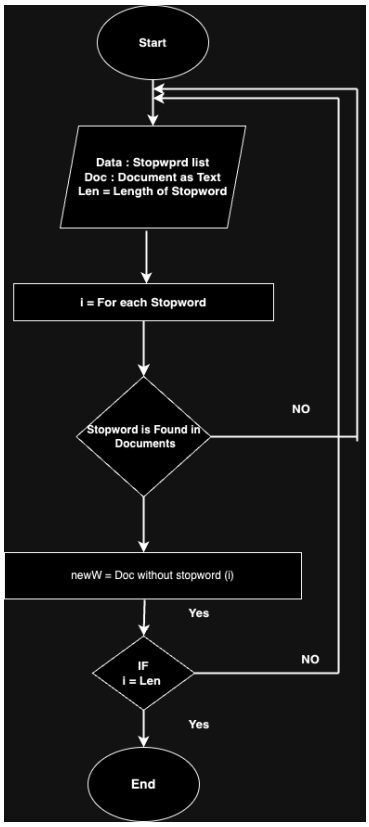


Figure 5: Flowchart for Stopwords

2.6 Tokenization Techniques in Natural Language Processing: A Detailed Review and Comparative Analysis

The literature review of the paper under discussion focuses on the application of supervised machine learning techniques for the classification of research papers into distinct

fields such as science, business, and social science. The importance of this research resides in the practicality of text categorization methods, which have gained popularity due to their wide range of applications, including sentiment analysis, product evaluation, and more specialized areas like poem classification and opinion mining. The methodology adopted in this research involves collecting abstracts from the three fields and pre-processing the textual data using natural language processing techniques. The pre-processing steps include tokenization, cleaning the text, removing stopwords, and stemming. These steps are essential for preparing the data for machine learning algorithms. The algorithms' performance is evaluated by comparing accuracy, recall, and F1 score standards. The study also investigates two techniques for representing text: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words.

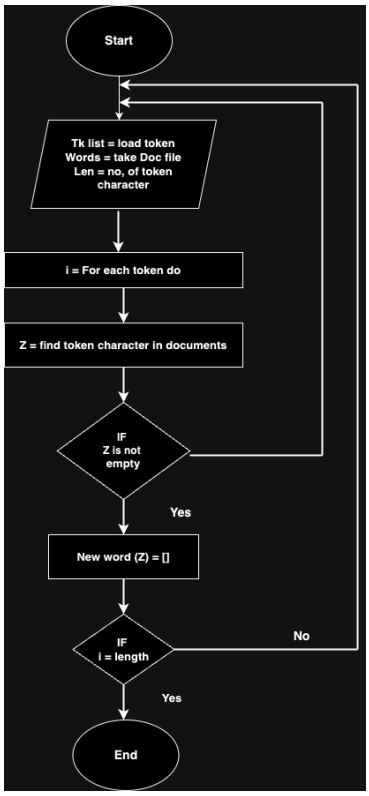


Figure 6: Flowchat for Tokenization

Tokenization is a crucial initial step in the pre-processing phase of the field of natural language processing. It involves breaking down a paragraph into words or other meaningful elements called tokens. This process is fundamental as it converts the input text into a list of arrays of words, which is a necessary step before any machine learning can be applied to the textual data. The Natural Language Toolkit (NLTK), a Python library, is utilized for tokenization in this work. The tokenized data is then cleaned by removing punctuation marks and converting all alphabets to lower case to ensure that identical words with different cases are not treated as separate tokens.

2.7 Improving Analysis of Sequential Data: A Thorough Study of CNN-LSTM Fusion Architectures in Natural Language Processing

The literature analysis of the research in question mostly centers around the advancement and assessment of network intrusion detection systems (IDS) through the utilization of machine learning and deep learning methodologies. The review is expected to discuss the importance of network security in relation to the fast expansion of technologies such as the internet of things, big data, and cloud computing, which have become crucial for networked computing. Conventional security measures such as firewalls and encryption methods are being tested by advanced assaults, underscoring the necessity for more effective network intrusion detection systems. Nevertheless, as networks continue to grow, there is an increasing demand for systems capable of accurately and efficiently identifying both familiar and unfamiliar assaults, while minimizing false positive alertsHalbouni et al. (2022).

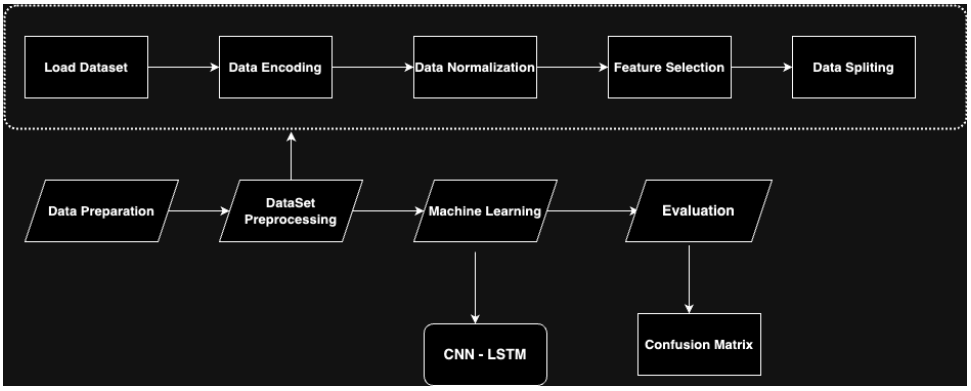


Figure 7: Architecture diagram for CNN-LSTM

The literature review may also include a discussion on the experimental setup used to evaluate the proposed models, including the hardware and software configurations, as well as the datasets used for binary and multiclass classification. The performance of different learning algorithms, such as CNN-alone, LSTM-alone, LSTM-CNN, and CNN-LSTM, is compared to determine the most effective model. The paper provides an overview of the current state of network intrusion detection systems, this paper explores the difficulties encountered by networked systems and the capacity of machine learning and deep learning techniques to enhance their capacity to protect against various types of cyber-attacks.

2.8 Word2Vec Integration: Exploring the Potential of Distributed Word Representations in Natural Language Processing

Word2Vec has emerged as a pivotal technique in the domain of Natural Language Processing (NLP), particularly in tasks such as sentiment analysis. The strength of Word2Vec lies in its ability to capture the contextual semantics of words, which significantly enhances the quality of feature representation for machine learning models . In the context of sentiment analysis of social media data, such as Twitter, Word2Vec has been

employed to understand and classify large collections of documents into positive and negative sentiments. The combination of Word2Vec with machine learning algorithms like Random Forest has shown to improve the accuracy of sentiment classification. For instance, a study on the sentiment analysis of 2019 election Twitter data demonstrated that Word2Vec with Random Forest outperformed traditional methods such as Bag-of-Words (BoW) and TF-IDF, achieving an accuracy of 86.87% . The application of Word2Vec is not limited to political sentiment analysis. It has also been used in analyzing sentiments towards products, such as electronic devices, where different machine learning algorithms were tested for classification accuracy. In these cases, Word2Vec's ability to understand the context of tweets has proven beneficial .

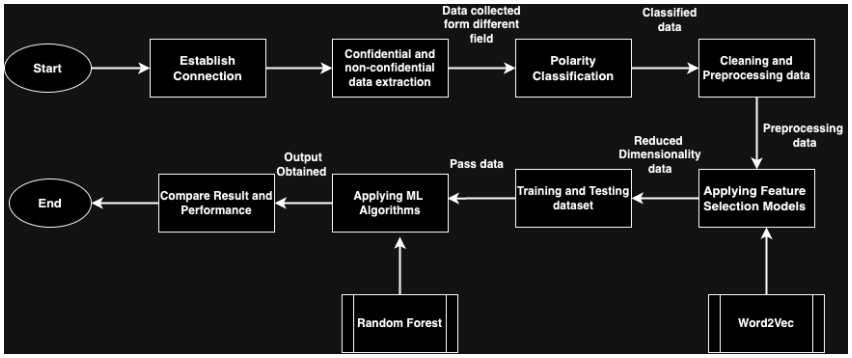


Figure 8: Word2Vec Convenstion diagram using ML Algorithm

The importance of training a Word2Vec model on domain-specific data cannot be overstated. While pre-trained word vectors are available, they may not capture the nuances of a specific corpus, such as tweets related to a particular event or topic. Training Word2Vec on the relevant data ensures that the vectors accurately reflect the unique language and style of the corpus in question. Word2Vec has proven to be a valuable tool in sentiment analysis, offering improved accuracy over traditional vector-space models by understanding the context of words. Its combination with various machine learning algorithms has been effective in analysing sentiments across different domains, from politics to consumer products. Future work could explore the integration of Word2Vec with additional machine learning techniques to further enhance sentiment analysis accuracyHitesh et al. (2019).

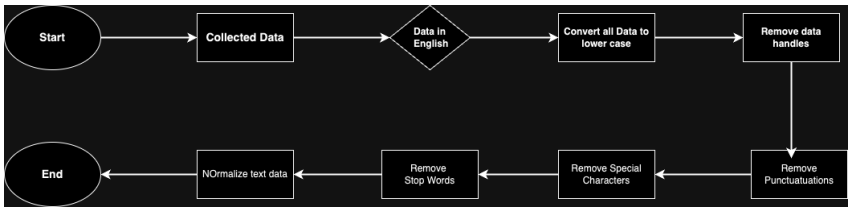


Figure 9: Workflow of Word2Vec

2.9 An In-Depth Investigation of RNN-LSTM Architectures in Sequential Data Processing

Features in the dataset have different units or scales, as it brings them to a common scale without distorting differences in the ranges of values. The authors likely used Standard Scaler to preprocess the input data for the RNN-LSTM model. By standardising the Bitcoin price data, Standard Scaler is a pre-processing technique used in machine learning to standardise the features of a dataset around the mean with a unit standard deviation. This scaling process is important because many machine learning algorithms, especially those that involve gradient descent optimisation or are distance-based (like k-nearest neighbours), assume that all features are centred around zero and have variance in the same order. Here's a brief explanation of how the Standard Scaler works. Calculation of Mean and Standard Deviation, For each feature in the dataset, the mean (average) and standard deviation are calculatedShivani et al. (2022).

Each feature value x is then transformed using the following formula:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

where :

z : standardized value,

x : original feature value,

μ : mean of the feature,

σ : standard deviation of the feature.

After applying the Standard Scaler, the distribution of each feature will have a mean value of 0 and a standard deviation of 1. Using Standard Scaler is particularly useful when the model can more easily learn the underlying patterns without being biased by the scale of the data, which can lead to improved performance in predicting cryptocurrency prices.

Figure 10: Standard Scaler Input/Output and Feacture Extraction of CNN_LSTM

3 Methodology

This paper proposes a hybrid approach that leverages advanced technologies such as ML, NLP, and data augmentation techniques to automate the data classification process and maintain the confidentiality of the data stored in the cloud. By analyzing the data attributes and patterns, this framework aims to dynamically classify the data, enabling adaptive security measures tailored to the confidentiality of each data type. A unique classification model known as AUG-ConvoLSTM-RF is designed in this research for identifying confidential data and ensuring secure data storage and transfer within the CaaS platform. The stages involved in the implementation process are as discussed in the below subsections.

3.1 Architecture diagram and Explanation

Following the data labeling, preprocessing was carried out to clean the data. Preprocessing in this research is performed using the steps of NLP which are discussed as follows:

- **Text Augmentation:** The text data is augmented using a NLP based BERT (Bidirectional Encoder Representations from Transformers) augmenterAtliha and Šešok (2020). In this process, the BERT model generates augmented text data. By augmenting the text data, the dataset is expanded and diversified which plays a critical role in improving the performance of the classification model. A "nlpaug" library is used to perform text augmentation using the BERT model. During augmentation it was ensured that the quality of the augmented data aligned to the original context of the dataset.

- **Remove Punctuation:** In this step, punctuation marks from the text are removed to improve the quality and consistency of the text. It is essential to make the text data more readable. In this work, a built-in string manipulation function was used to remove the punctuation marks. A "string.punctuation" constant is used to remove punctuation from the text using the translate() method.

- **Tokenization:** In this process, the text is broken into multiple smaller units, typically words or subwords, known as tokensMielke et al. (2021). A word tokenization approach is considered in this study wherein the text is split into words based on whitespace characters such as spaces, tabs, and newlines. Tokenization is performed using the Natural Language Toolkit (NLTK) library which provides different tokenizers for word tokenization.

- **Removal of Stop Words:** In this step, common and less informative words (stop words) are removed from the text. The NLTK library provides the list of stop words which are least or not significant in the text. Removal of stop words improves the accuracy of the NLP process.

- **Stemming:** It is a text normalization technique in NLP which reduces the words to their root or base form, known as the stem. The output of the stemming process might not always be the accurate representation of the actual word but it defines the core meaning shared by related wordsPramana et al. (2022). In this work, stemming is performed to reduce the variations of words and improve the text analysis by considering similar words as the same.

- **Lemmatization:** This process is similar to the stemming process but lemmatization ensures resulting words belong to the language. Although the lemmatization process is slower compared to stemming, it generates valid processes by using lexical knowledge to derive valid lemmasRazali et al. (2020).

- **Word2Vec:** In this step, the words are represented as vectors in order to enhance the text analysis processJang et al. (2020). The model is defined using TensorFlow's Keras API. In this research, Word2Vec is used to capture the semantic meaning of words by considering the context in which words appear in a large amount of text data.

After preprocessing the data, the essential features are extracted from the labeled data using a hybrid CNN-LSTM model.

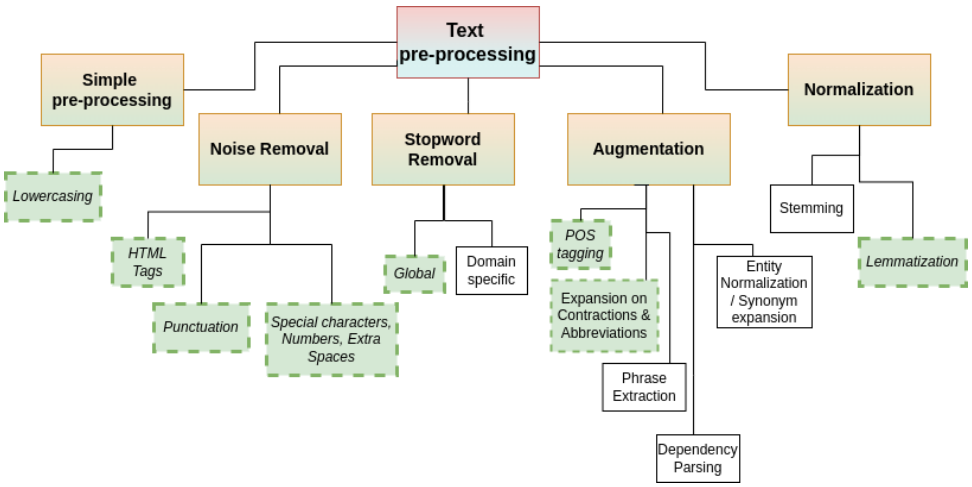


Figure 11: Architecture diagram for For Text Data Processing

4 Design Specification

Based on the below mentioned data labels, the confidentiality of the data is identified. The data accessed is preprocessed to make it suitable for the classification process.

4.1 Data Collection

The textual data was collected from the Kaggle datasets which can be accessed using the following link:

`https://www.kaggle.com/datasets/jensenbaxter/10dataset-text-document-classificatio`
`select=business`.

During this phase, the data was categorized into two main types:
confidential and non-confidential.

To achieve this, key attributes were utilized for distinguishing confidential data:

- Personal Identifiable Information (PII): This category included elements such as names, social security numbers, birthdates, addresses, phone numbers, and email

addresses. The identification and protection of PII are of paramount importance in data handling.

- Financial Data: Financial data is often highly sensitive, encompassing credit card numbers, bank account numbers, financial transaction records, profits, losses, sales forecasts, and more. Proper handling and classification of such data are crucial for regulatory compliance and security.
- Healthcare Information (PHI): Medical records, health insurance information, and prescription details are integral to healthcare systems. They must be identified and treated with the utmost care to safeguard individuals' privacy.
- Intellectual Property: Patents, trade secrets, copyrighted materials, and research and development data are vital assets for organizations. Protecting intellectual property is a core concern in many industries.
- Customer Data: Customer contact information, purchase history, and customer preferences are key components of business operations. Safeguarding this data ensures customer trust and regulatory compliance.
- Employee Data: Employee payroll information, human resources records, and employment contracts fall into this category. Proper handling of employee data is not only ethically important but also legally mandated.
- Legal Documents: Contracts, legal correspondence, and court documents require special attention in terms of classification due to their legal implications.
- Sensitive Business Data: This includes business plans, strategic documents, and financial reports, which can have a significant impact on a company's competitiveness and strategy.
- Logins and Passwords: Usernames, passwords, and access credentials are crucial for system security. Their classification helps ensure that security measures are in place.
- Confidential Communications: Emails, instant messages, and internal memos can contain sensitive information and need to be handled with care.
- Location Data: GPS coordinates and location history can reveal personal or organizational movements and must be protected.
- Biometric Data: Fingerprints, retina scans, and facial recognition data are unique identifiers and highly sensitive.
- Data Access and Authorization Information: This category includes access logs, user permissions, and encryption keys, which are vital for data security.
- Third-party Data: Data shared with third-party vendors and partner data require classification to maintain control over data sharing and compliance.
- Social Media Data: Social media account information, social media posts, and government-issued identifiers like Aadhar Number, ID card Number, and PAN Number are crucial for identity verification and privacy.

- Government-issued Identifiers: Passport numbers and driver's license numbers are highly confidential and essential for identity verification.
- Video and Audio Recordings: Security camera footage and voice recordings may contain sensitive information and need to be categorized accordingly.
- Research Data: Scientific research findings and experimental data have value and should be protected.
- Mailing Lists: Subscriber lists and newsletter recipients are valuable for marketing and need to be handled appropriately.
- Customer Feedback and Surveys: Feedback forms and survey responses provide insights into customer preferences and satisfaction.
- Backup and Recovery Data: Backup copies of sensitive data are critical for disaster recovery and should be properly categorized.
- Operational Data: Configuration data and network diagrams are essential for system operations and should be managed securely.
- Vendor and Supplier Data: Supplier contracts and vendor contact information are vital for business relationships and should be classified.
- Metadata: Timestamps and file histories provide context to data and can be valuable for analysis.
- User Profiles: User profiles contain information about individuals and should be protected.
- Government Data: Government-related information such as Cabinet Office orders, Freedom of Information Act data, Information Commissioner records, and legislative information like New rules on e-mail retention and Government Decisions are subject to specific regulations and need to be classified appropriately

5 Implementation

The implementation of Feature Extraction using CNN-LSTM model is used for extracting data related features to categorize confidential data. The hybrid model combines the attributes of CNN and LSTM to capture spatial features through convolutional layers and temporal or sequential information through recurrent layers Naeem and Bin-Salem (2021). The CNN-LSTM model identifies the data patterns and makes use of data patterns to identify the features and extract them. The hybrid model is constructed by stacking CNN and LSTM layers.

5.1 Model Architecture for Feature Extraction

A 1D convolutional layer (Conv1D) is designed with 256 filters, a kernel size of 5, and a rectified linear unit (ReLU) activation function. This layer is designed for causal padding, since it considers only the past values in the input sequence. The 1D convolutional layer is followed by two LSTM layers wherein one layer consists of 128 units and another with

64 units. Both use the hyperbolic tangent (tanh) activation function and the first LSTM layer returns sequences. Three dense layers with 32, 16, and 8 units are returned and the output is scaled by the lambda layer by a scaling factor of 100. The scaling balances the output of the LSTM's tanh activation function between -1 and 1.

The CNN in this work is modeled with a single input layer, multiple hidden layers and one output layer for extracting features from input data. The convolutional layer embedded in the hidden layer extracts the features and the temporal dependency between the features is learnt using the LSTM layer. The fully connected (FC) layer performs convolution operation and activation function for transforming the nonlinear features.

The convolution operation of the CNN can mathematically defined as follows:

$$C = f(W \cdot X + b) \tag{2}$$

Where, X is the input data, W is the filter weight, b is the bias output, and C is the output of the convolution layer. The convolutional operation followed by pooling layers, extracts relevant features.

On the other hand, the LSTM layers can take the output from the CNN layers or process the input directly, depending on the data. LSTM is known for its ability to capture sequential data and remember the data over long sequences. In the proposed hybrid model, the LSTM layers capture the dependencies on the data obtained from the CNN layers for feature extraction. The LSTM layer controls the flow of information through the gates and stores the information in its memory gate. Mathematically, it is defined using the below equations.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

Where σ is the sigmoid function, W_i is the input text sequence, h_{t-1} is the previous state of the forget gate, and bf represents the bottleneck features. The input gate is responsible for controlling the new information flow which is stored in the cell state Ct.

The operation of input gate is given as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

The current state of the cell and the memory unit status of the cell is defined in equations 5 and 6 respectively.

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \tag{6}$$

The output gate of the LSTM model controls the output of the sigmoid function defined in equation 7 and the output of the hidden layer is defined in equation 8.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$ht = o_t * tanh(Ct) \tag{8}$$

The terms b and W represent the offset terms and weight coefficient matrix respectively, tanh is defined as the hyperbolic tangent activation function.

Overall 8 features are extracted using the hybrid CNN-LSTM model. Further, these 8 features are then fed into the K-Means SMOTE algorithm. This algorithm addresses the issue of class imbalance by oversampling the minority class.

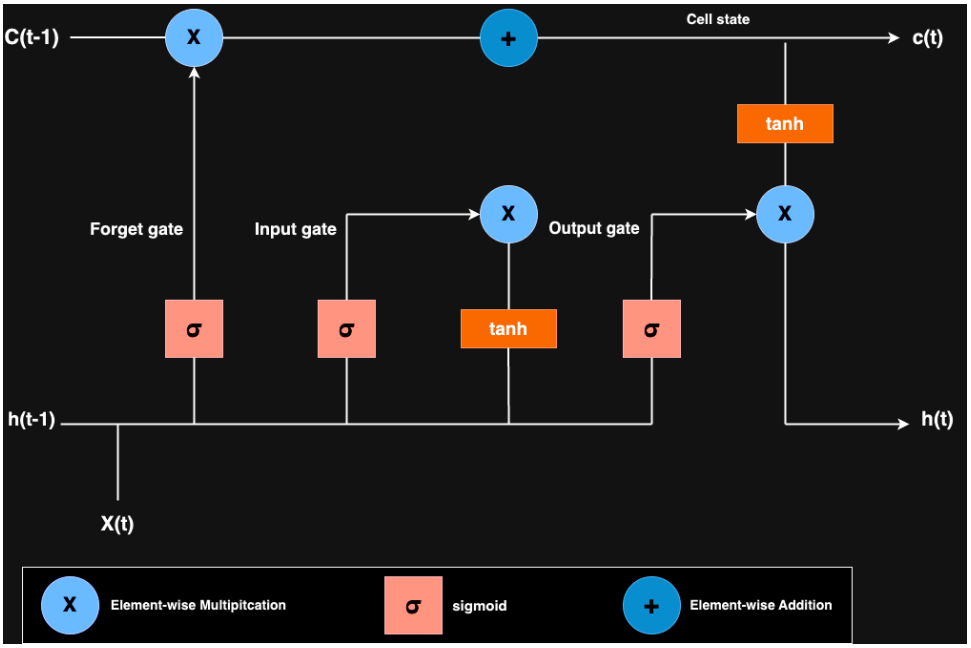


Figure 12: Architecture diagram for Feature Extraction (CNN-LSTM)

5.2 K-Means SMOTE Algorithm

The algorithm combines K-Means clustering (KCM) along with SMOTE for balancing the data classes. Initially, the KCM algorithm identifies the clusters in the minority class samples and representative clusters within the minority class are recognized. For each cluster obtained from K-Means, the SMOTE technique is applied independently within each cluster. Further, synthetic samples are generated for the minority class by interpolating between the existing minority samples in that cluster. The original minority class samples are combined with the synthetic samples generated by the SMOTE algorithm within each cluster. In this way, a balanced dataset is created and this dataset is used to train the classification model. Further, a standard scaling is applied to the data to maintain the features to be within a specific range (specifically between 0 and 1). This ensures that all features contribute equally to the data classification process and prevents the dominance of any particular feature due to differences in scale. The balanced features are used for classifying the data.

5.3 Analysing the Speck Cipher in Lightweight Cryptography for Data Security

It is the system is optimised for both software and hardware implementations, especially for applications with limited resources. SPECK is part of the ARX (add-rotate-xor) family of ciphers, which means its operations consist of modulo addition, rotation by a fixed number of bits, and XOR (exclusive-or) operationsThabit et al. (2021). These simple operations are chosen for their ease of implementation and efficiency on a wide range of platforms. The confidentiality provided by the SPECK Cipher Algorithm is attributed to its use of varying key lengths and block sizes, which can be tailored to the security requirements of the application. The algorithm's structure, which includes a series of rounds where each round involves specific shifting and mixing of the input plaintext and key material, makes it resistant to several types of cryptographic attacks when properly implementedAlkamil and Perera (2020).

The encryption process of SPECK involves a round function defined by the equation:

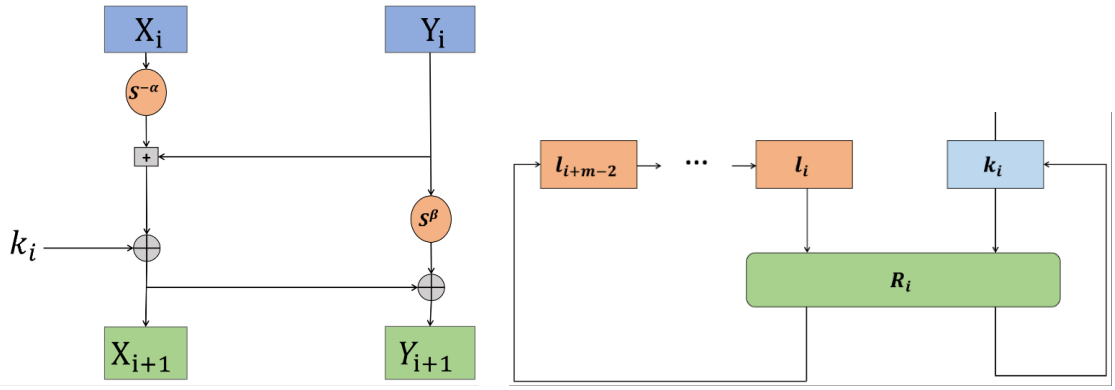
$$R_k(x, y) = ((S - \alpha x + y) \oplus k, S\beta y \oplus (S - \alpha x + y) \oplus k) \tag{9}$$

$$R^{-1}(x, y) = (S_\alpha((x \oplus k) - S_{-\beta}(x \oplus y)), S_{-\beta}(x \oplus y)) \tag{10}$$

The SPECK Cipher Algorithm involves the following steps:

$S - \alpha$**:** Bit shift to the right.

$S\beta$**:** Bit shift to the left.



(a) SPECK round function (b) SPECK key generation function

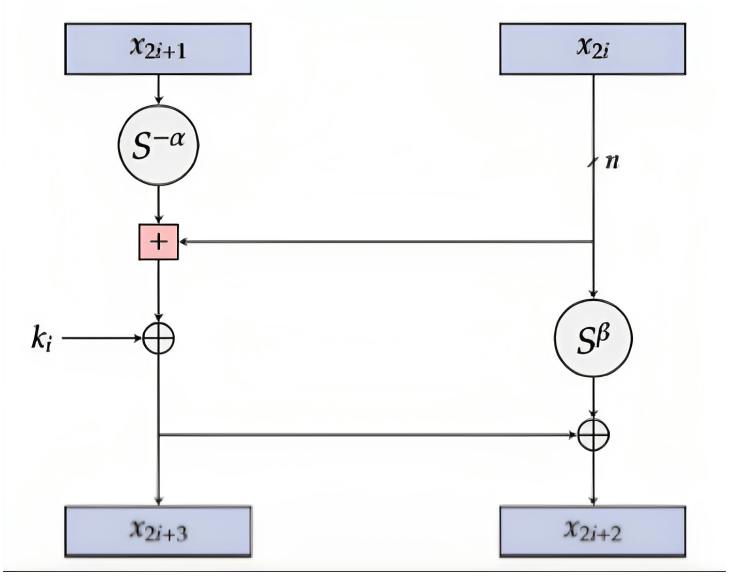
Figure 13: Structure and Functionality of SPECK

This process is repeated for a predefined number of rounds, ensuring that the plaintext is thoroughly mixed with the key material to produce the ciphertext. The SPECK Cipher Algorithm ensures secrecy by employing these measures, which render it computationally impractical for an adversary to retrieve the original plaintext from the ciphertext without the proper key.

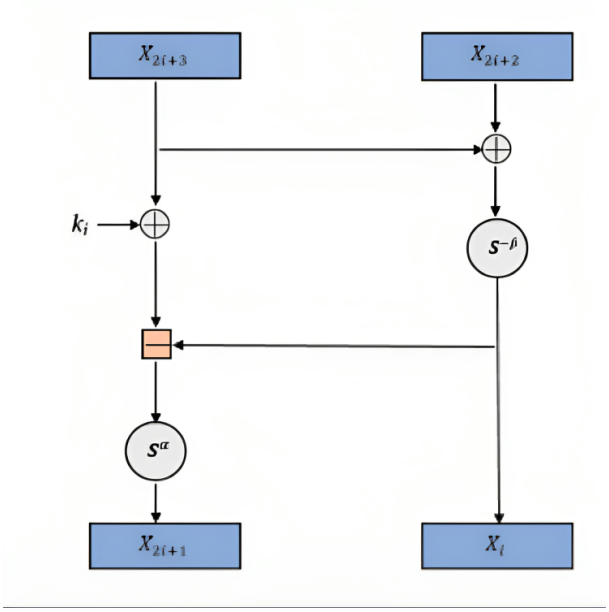
Figure (b) represents the architecture of the SPECK key generation function, which produces the round keys based on the original key K. The key creation function utilises the previously described round function to generate the round key (k_i) for each round. Given the values $\mathrm{K} = K = (l_{m-2}, \ldots, l_o, k_0)$ and $m = \{2, 3, 4\}$, the key generation function may be expressed using equation (3), where k_i represents the i-th round key, with $0 < i < T$:

$$l_{i+m-1} = ((k_i + S_{-\alpha} l_i) \oplus i) \tag{11}$$

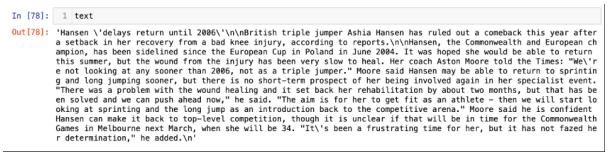
$$k_{i+1} = S_\beta k_i \oplus l_{i+m-1} \tag{12}$$



(a) SPECK encryption function



(b) SPECK decryption function



```
In [78]: 1 text
Out[78]: 'Hansen \'delays return until 2006\'\n\nBritish triple jumper Ashia Hansen has ruled out a comeback this year after
a setback in her recovery from a bad knee injury, according to reports.\n\nHansen, the Commonwealth and European ch
ampion, has been sidelined since the European Cup in Poland in June 2004. It was hoped she would be able to return
this summer, but the wound from the injury has been very slow to heal. Her coach Aston Moore told the Times: "We\'r
e not looking at any sooner than 2006, not as a triple jumper." Moore said Hansen may be able to return to sprintin
g and long jumping sooner, but there is no short-term prospect of her being involved again in her specialist event.
"There was a problem with the wound healing and it set back her rehabilitation by about two months, but that has be
en solved and we can push ahead now," he said. "The aim is for her to get fit as an athlete - then we will start lo
oking at sprinting and the long jump as an introduction back to the competitive arena." Moore said he is confident
Hansen can make it back to top-level competition, though it is unclear if that will be in time for the Commonwealth
Games in Melbourne next March, when she will be 34. "It\'s been a frustrating time for her, but it has not fazed he
r determination," he added.\n'
```

(a) Plain text

```
800684078707731535606946049527498842473649531012484426467957239700097575118716954671299759907542960798995619928376343487524306248658426464125125684845315654717417720638192645413536277093325500: Hanse
n 'delay
s return
until 2006
1
```

Figure 15: Plain text to Cypher text

5.4 Classification Model Development

The AUG_ConvoLSTM-RF model is developed using the RF model, which is an ensemble learning method that constructs multiple decision trees (DTs) during training and generates the classification output based on the collective outputs of the individual trees.

The process of RF for generating classification output are defined in the following steps:

Initially, a single DT is defined, which consists of nodes that make decisions based on feature values. Let T(x) represent the output of a DT for input. For a dataset with N samples and M features, a DT can be represented as a set of rules R defined as R1, R2,... RN, which also defines the nodes of the RF model. Each node Ri contains a feature set (fi) for classification, and the output of an individual DT is defined as follows:

$$T(x) = \sum_{i=1}^{n} \text{prediction}(R_i) \times \text{indicator}(x \in R_i) \tag{13}$$

Where, indicator($x \in R_i$) indicator is an indicator function that returns 1 if the value of x falls within the region Ri. Otherwise, it returns 0. The RF model makes the final classification by aggregating the output of all individual DTs.

The RF model is selected for designing the classification model since it avoids overfitting caused by the randomness introduced in sampling and feature selection. For designing the AUG_ConvoLSTM-RF model, the standardised data is fed into the RF classifier for training the model, and the trained model weights are stored for later deployment into a CaaS environment. The model distinguishes the confidential data from the non-confidential data. The proposed AUG_ConvoLSTM-RF model employs a combination of NLP, data augmentation, and feature extraction techniques to enhance its performance. The augmented data from the BERT model is processed by a hybrid CNN-LSTM model, which extracts the features for the classification process. Based on the features, the RF model classifies the data. After developing the RF model, a secure data transfer technique is implemented wherein the data is encrypted using a Speck lightweight encryption algorithm.

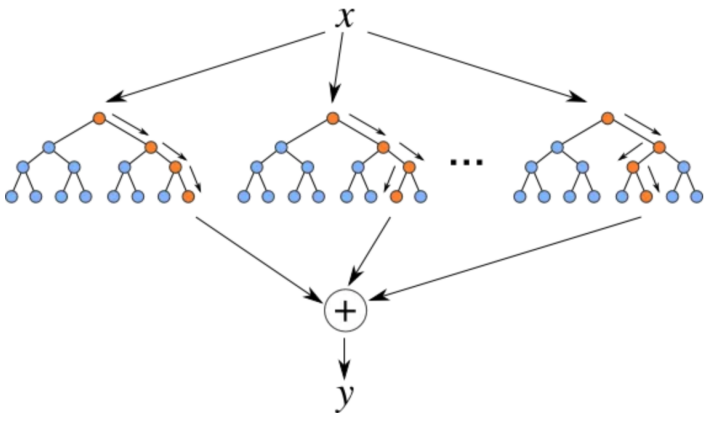
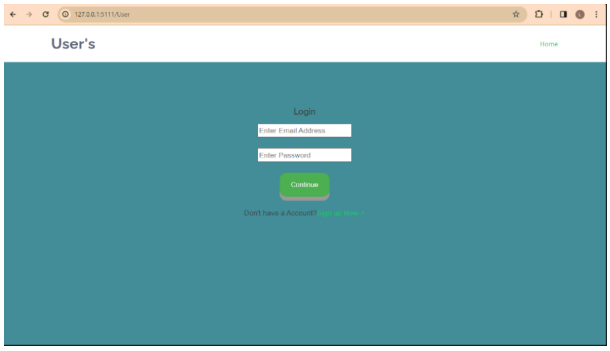


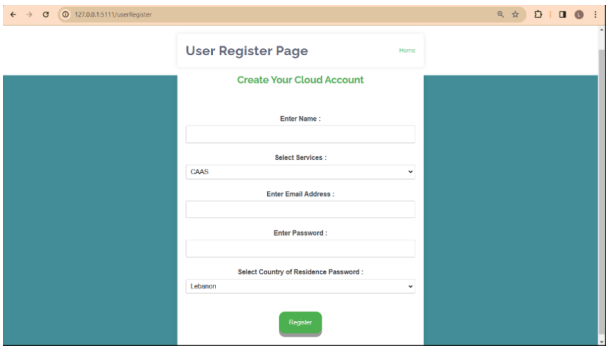
Figure 16: Architecture diagram for Classification Model (RF)

Framework Design and User Interface

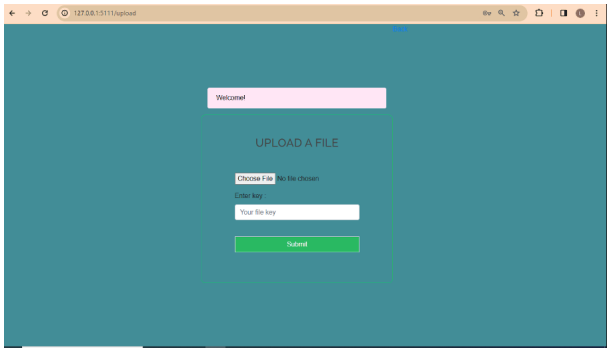
The entire framework is then designed into a user interface (UI), which allows the users to register themselves on the CaaS platform. Users are allowed to store and download their own data securely. The AUG_ConvoLSTM-RF trained model weights are used to categorise data into confidential and non-confidential segments. It must be noted that encryption is selectively applied only to confidential data to ensure secure storage and transfer. After uploading the file, you can see Encrypted and Decrypted file the encryption will happen only for the confidential data in the text document.



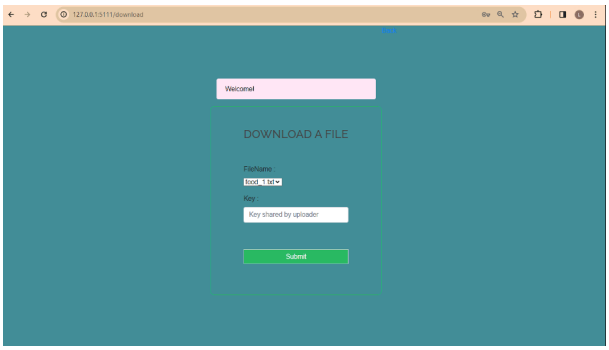
(a) User Login Page



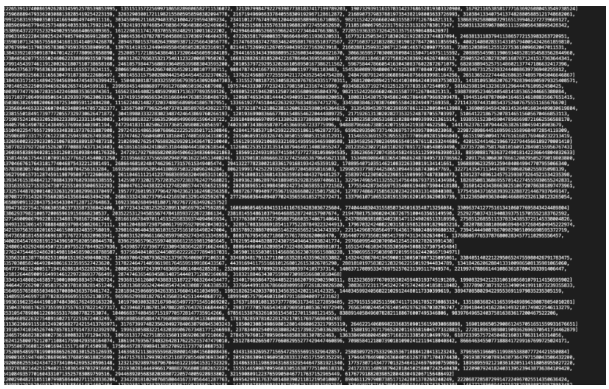
(b) User Register Page



(a) Upload confidential File and give security key



(b) download confidential file by entering security key



(a) Confidential Data

(b) Non-Confidential Data

6 Evaluation

The performance of the AUG_ConvoLSTM-RF classifier is determined using various different evaluations which are expressed using the below equations. These metrics are the elements of the confusion matrix which provides a clear output for multiple classes.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{F1 score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

Where TP, TN, FP, and FN represent True positives, True negatives, False positives, and False negatives respectively. The performance is evaluated with respect to classification accuracy, time complexity, processing time, and memory utilization.

RandomForest_accuracy : 0.8412726945892504
RandomForest_precision : 0.8675067841961733
RandomForest_recall : 0.8412726945892504
RandomForest_Balanced_F1 : 0.8383133725631672

The results of the simulation analysis are discussed as follows:

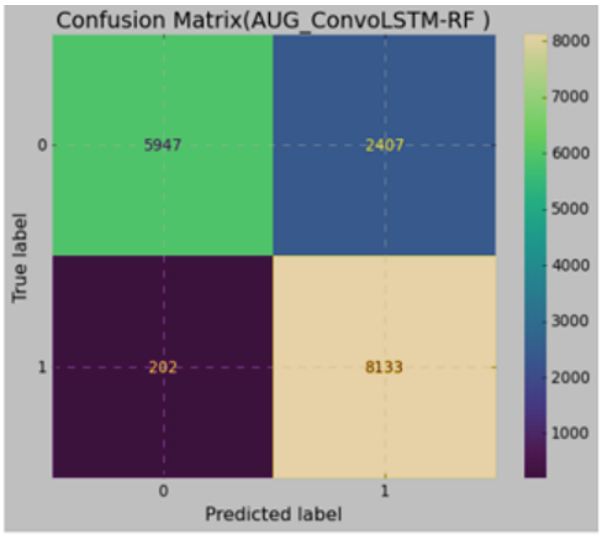


Figure 20: Confusion matrix of the AUG_ConvoLSTM-RF classifier

As observed from the confusion matrix, the AUG_ConvoLSTM-RF classifier achieves an accuracy of 84.36 %. The classification performance of the classifier is compared with other existing methods and the results are tabulated in table 1.

Decision Tree Metrics

As observed from the confusion matrix, the Decision Tree Metrics achieves an accuracy of 83.46 %. The classification performance of the classifier is compared with other existing methods and the results are tabulated in table 1.

DecisionTree_accuracy : 0.8301875486847624
DecisionTree_precision : 0.869150243235691
DecisionTree_recall : 0.8301875486847624
DecisionTree_Balanced_F1 : 0.8254867721333098

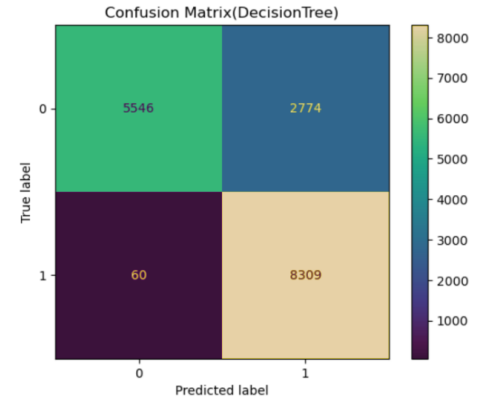


Figure 21: Confusion matrix of the Decision Tree Metrics

Gaussian Naive Bayes Metrics

As observed from the confusion matrix, the Gaussian Naive Bayes Metrics achieves an accuracy of 83.43 %. The classification performance of the classifier is compared with other existing methods and the results are tabulated in table 1.

GaussianNB_accuracy : 0.8312661034214153
GaussianNB_precision : 0.8610944929886988
GaussianNB_recall : 0.8312661034214153
GaussianNB_Balanced_F1 : 0.8276208499846731

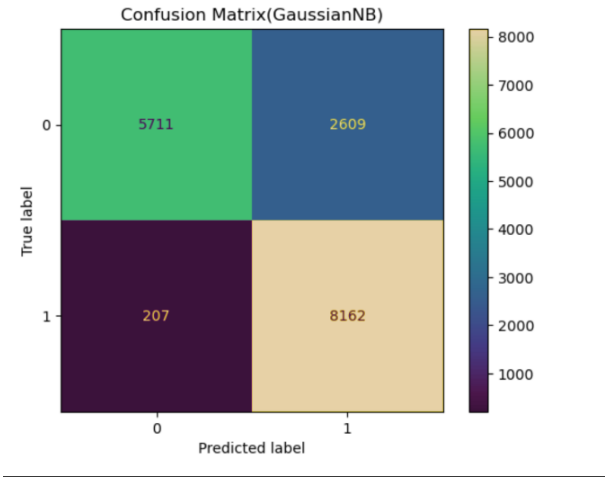


Figure 22: Confusion matrix of Gaussian Naive Bayes Metrics

Linear Regression Metrics

As observed from the confusion matrix, the Linear Regression Metrics achieves an accuracy of 82.94 %. The classification performance of the classifier is compared with other existing methods and the results are tabulated in table 1.

LinearRegression_accuracy : 0.8296482713164359
LinearRegression_precision : 0.8645408787143636
LinearRegression_recall : 0.8296482713164359
LinearRegression_Balanced_F1 : 0.8253773904298876

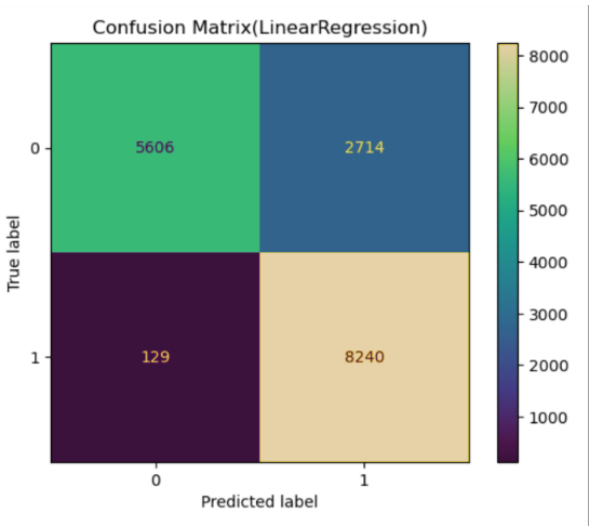


Figure 23: Confusion matrix of Linear Regression Metrics

Table 1: Performance Metrics of Existing Methods

Method	Accuracy	Precision	Recall	F1-Score
KNN	48%	34%	47%	40%
Naive Bayes	58%	42%	17%	27%
Semantic KNN	71%	43%	47%	44%
Linear Regression	82.94%	86.32%	82.94%	82.53%
Gaussian NB	83.43%	86.28%	83.43%	83.11%
Decision Tree	83.46%	86.92%	83.46%	83.06%
AUG_ConvoLSTM-RF	84.36%	86.95%	84.36%	84.09%

Results prove that the AUG_ConvoLSTM-RF classifier outperforms in terms of excellent classification performance. Both Gaussian NB and DT exhibit second best performance by achieving an accuracy of 83.43% and 83.46% respectively. The performance is also tested in terms of time complexity, processing time and memory utilization.

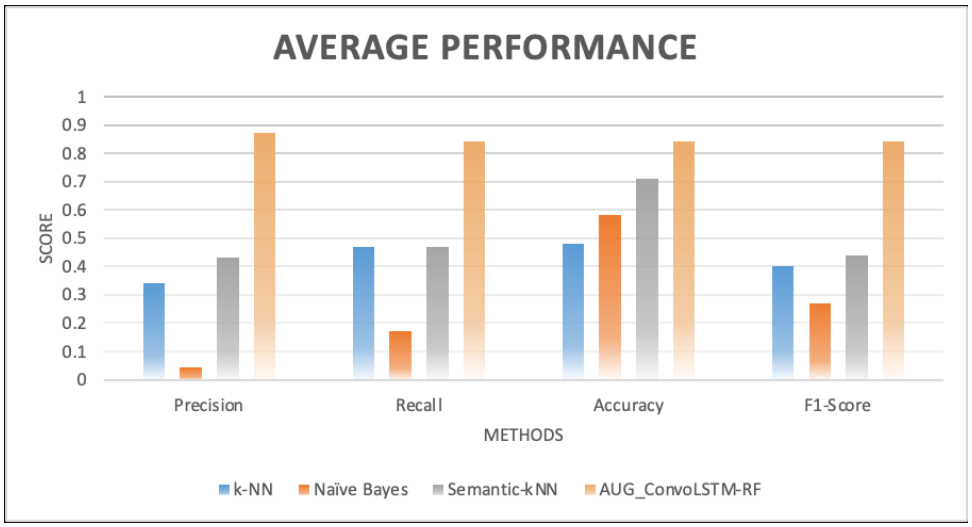


Figure 24: Average Performance

Table 2 represents the results for the time complexity.

Table 2: Analysis of Time Complexity

Methods	Time Complexity (MS)
Traditional Secure Method	14237148
C2aaS	3547661
CaaS based on AUG_ConvoLSTM-RF	53463

The time taken by the AUG_ConvoLSTM-RF classifier is significantly lower compared to other traditional methods and C2aaS. This validates the fact that the proposed classifier has a lesser time complexity. In addition, the classification performance with respect to confidential data is evaluated and the results are tabulated in table 3.

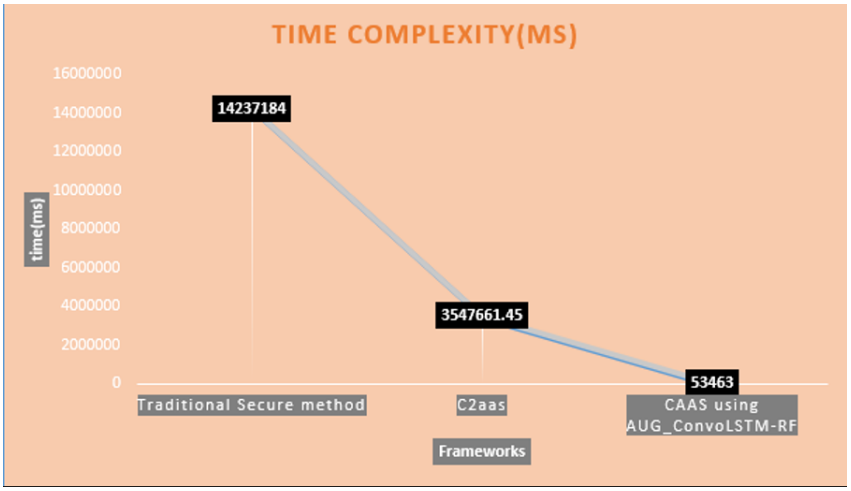


Figure 25: Time Complexity Analysis

Table 3: Dataset and Data Class Information

Data Class	Dataset Size (bytes)	Confidential Data	Non-Confidential Data
Business	198484	22356	176128
Entertainment	170051	34883	135168
Historical	532480	491520	40960
Politics	262144	32768	229376
Technology	274432	45056	229376
Sport	188416	61440	126976

The size of the confidential data is defined in table 3 and the time taken for processing the data is defined in table 4.

Table 4: Total Processing Time and Memory Utilization

Datasets	Total Processing Time (ms)	Total Memory Utilization (%)
Business	214	2.14
Entertainment	291	2.91
Historical	431	4.31
Politics	249	2.49
Technology	231	2.31
Sport	643	6.43

Results of the experimental analysis show that the proposed classifier is highly effective in terms of classifying and securing confidential data with better memory utilization and lesser time complexity.

Question: Explain the accuracy of the models and why they are almost the same.

Explanation: [In this project initial my dataset was small and I didn't get good accuracy on my trained model so that I have planned to do data augmentation for my dataset and once I did that I got good accuracy for all model and even my accuracy was (Decision Tree Metrics, Gaussian Naive Bayes Metrics, Linear Regression Metrics)

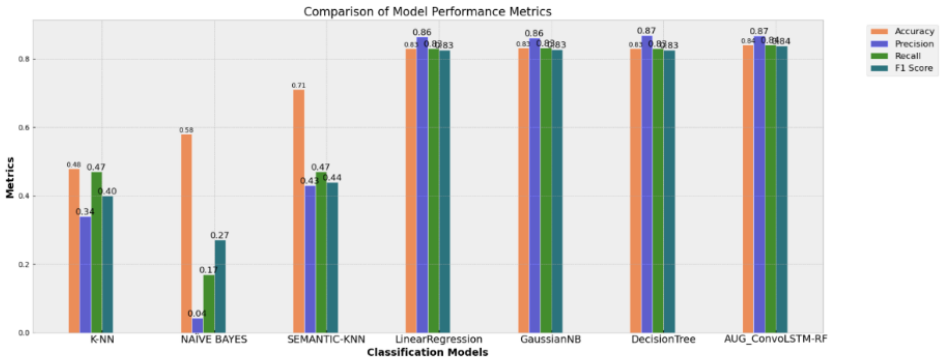


Figure 26: Graphical representation of the comparative results

different for all these model. But I got less accuracy for that the selected model (AUG-ConvoLSTM-RF). So, I have researched some paper and I got an idea of doing method name called oversampling and after doing that I got good accuracy for the selected model and same accuracy for the existing models like (Decision Tree Metrics, Gaussian Naive Bayes Metrics, Linear Regression Metrics). When models are trained on similar datasets, they might learn similar patterns and relationships. Consequently, with identical or highly correlated training data, the accuracy of the models might be similar as well.]

7 Conclusion and Future Work

This work aimed to address the critical challenge of enhancing data confidentiality in cloud environments by implementing a RF based AUG_ConvoLSTM-RF classification framework which was deployed in a CaaS environment. The increasing threats related to the data security within cloud infrastructures necessitate potential measures, and this research has contributed a comprehensive approach towards ensuring confidentiality through effective categorization of data security levels. The AUG_ConvoLSTM-RF classification model was trained to automate the categorization process by using the features extracted from the CNN-LSTM model. By harnessing its capabilities of the K-means SMOTE algorithm in handling imbalanced datasets and accommodating multiple attributes, this framework demonstrated considerable accuracy in classifying data according to varying levels of confidentiality. The integration of NLP techniques into the classification process facilitated adaptive and scalable security measures, ensuring that the confidential information is secured against unauthorized access and data breaches. Results reveal that the AUG_ConvoLSTM-RF model exhibits better performance compared to other models by achieving an accuracy of 84.36 % with a lesser time complexity and memory utilization. For future research, this work can be extended to address scalability concerns by optimizing the classification framework which can handle large-scale datasets efficiently. Techniques such as parallel processing, distributed computing strategies, or optimization techniques can be employed strengthening the security in cloud environments.

References

Abdulsalam, Y. S. and Hedabou, M. (2021). Security and privacy in cloud computing: technical review, *Future Internet* **14**(1): 11.

Alkamil, A. and Perera, D. G. (2020). Towards dynamic and partial reconfigurable hardware architectures for cryptographic algorithms on embedded devices, *IEEE Access* **8**: 221720–221742.

Atliha, V. and Šešok, D. (2020). Text augmentation using bert for image captioning, *Applied Sciences* **10**(17): 5978.

Chowdhury, S. and Schoen, M. P. (2020). Research paper classification using supervised machine learning techniques, *2020 Intermountain Engineering, Technology and Computing (IETC)*, IEEE, pp. 1–6.

Fonseca, J., Douzas, G. and Bacao, F. (2021). Improving imbalanced land cover classification with k-means smote: detecting and oversampling distinctive minority spectral signatures, *Information* **12**(7): 266.

George, A. S. and Sagayarajan, S. (2023). Securing cloud application infrastructure: Understanding the penetration testing challenges of iaas, paas, and saas environments, *Partners Universal International Research Journal* **2**(1): 24–34.

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M. and Ahmad, R. (2022). Cnn-lstm: hybrid deep neural network for network intrusion detection system, *IEEE Access* **10**: 99837–99849.

Hitesh, M., Vaibhav, V., Kalki, Y. A., Kamtam, S. H. and Kumari, S. (2019). Real-time sentiment analysis of 2019 election tweets using word2vec and random forest model, *2019 2nd international conference on intelligent communication and computational techniques (ICCT)*, IEEE, pp. 146–151.

Jang, B., Kim, M., Harerimana, G., Kang, S.-u. and Kim, J. W. (2020). Bi-lstm model to increase accuracy in text classification: Combining word2vec cnn and attention mechanism, *Applied Sciences* **10**(17): 5841.

Kumar, R. and Bhatia, M. (2020). A systematic review of the security in cloud computing: data integrity, confidentiality and availability, *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, IEEE, pp. 334–337.

Mielke, S. J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., Raja, A., Si, C., Lee, W. Y., Sagot, B. et al. (2021). Between words and characters: a brief history of open-vocabulary modeling and tokenization in nlp, *arXiv preprint arXiv:2112.10508* .

Naeem, H. and Bin-Salem, A. A. (2021). A cnn-lstm network with multi-level feature extraction-based approach for automated detection of coronavirus from ct scan and x-ray images, *Applied Soft Computing* **113**: 107918.

Pellicer, L. F. A. O., Ferreira, T. M. and Costa, A. H. R. (2023). Data augmentation techniques in natural language processing, *Applied Soft Computing* **132**: 109803.

Polus, M. E. and Abbas, T. (2021). Development for performance of porter stemmer algorithm, *Eastern-European Journal of Enterprise Technologies* **1**(2): 109.

Pramana, R., Subroto, J. J., Gunawan, A. A. S. et al. (2022). Systematic literature review of stemming and lemmatization performance for sentence similarity, *2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA)*, IEEE, pp. 1–6.

Razali, A., Daud, S. M., Zin, N. A. M. and Shahidi, F. (2020). Stemming text-based web page classification using machine learning algorithms: A comparison, *International Journal of Advanced Computer Science and Applications* **11**(1).

Saranya, S. and Usha, G. (2023). A machine learning-based technique with intelligent-wordnet lemmatize for twitter sentiment analysis., *Intelligent Automation & Soft Computing* **36**(1).

Shivani, C., Anusha, B., Druvitha, B. and Swamy, K. K. (2022). Rnn-lstm model based forecasting of cryptocurrency prices using standard scaler transform, *J. Crit. Rev* **10**: 144–158.

Thabit, F., Alhomdy, S., Al-Ahdal, A. H. and Jagtap, S. (2021). A new lightweight cryptographic algorithm for enhancing data security in cloud computing, *Global Transitions Proceedings* **2**(1): 91–99.