International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Novel Approach for Protecting Personal Sensitive Information in a Cloud Storage Environment

El Moudni Mohammed[1], El Houssine Ziyati[1]

[1] *Computer Science Department, High School of Technology, University Hassan 2, Casablanca, Morocco*

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: With the growing adoption of cloud computing and the increasing popularity of digital technologies, personal data storage and processing in cloud environments has become essential. However, as organizations and individuals embrace the benefits of cloud services, the security of personal sensitive information within this dynamic ecosystem has become a top priority. Ensuring the confidentiality, integrity, and availability of personal data in the cloud is critical to mitigate the risks associated with cyber threats. As cyber threats continue to evolve, it is essential to adopt innovative approaches to ensure personal data security measures. This article introduces a new approach, leveraging machine learning, and data masking techniques, using both serverless and secret vault services provided by most of cloud service providers (CSPs). Data masking techniques are employed to further protect sensitive information from unauthorized access. This paper explores and assesses the effectiveness of machine learning algorithms including LSTM, CNN, and MLP in classification tasks. The results highlight CNN's outstanding performance, achieving a remarkable 100% accuracy. This ensures perfect classification with double validation using the pattern matching technique. Furthermore, an analysis of download and upload time costs reveals that data processing using our model has no significant impact on the execution time metric.

Keywords: Machine Learning, Personally Identifiable Information, Data Privacy, Cloud Storage, Data Leakage, Cloud Computing, Data Security, Cyber Threats.

E-mail :mohammed.elmoudni2@gmail.com, ziyati@gmail.com

1. INTRODUCTION

Finding sufficient storage space to accommodate data poses a significant challenge for numerous IT professionals, researchers, and individuals [1]. However, cloud storage services enable individuals and organizations to embrace a digital paradigm characterized by flexibility, efficiency, and accessibility [2]. By taking advantage of the capabilities of cloud storage solutions, they can overcome the limitations of traditional data management, opening a wide range of possibilities [3].

Globally, the adoption of cloud storage systems is on the rise as organizations and individuals seek efficient solutions for storing and retrieving their data [4]. However, this escalating reliance on cloud storage introduces significant concerns related to data security. The susceptibility of cloud storage systems to diverse cyber threats poses a critical challenge, particularly in safeguarding the confidentiality, integrity, and availability of all stored data [5] [6] [7]. Achieving the right combination of effectiveness and practicality is one of the key challenges facing the design of a cloud storage security approach [8]. It is crucial to build a model that can accurately differentiate normal data from sensitive data [9], while maintaining enough simplicity and performance for realistic deployment in a cloud environment, rather than becoming either too complex or too consuming of resources [10] [11] [12].

Considering the above challenges, the primary objective of this research paper, is to look at the security issues associated with personal information in cloud systems, and to suggest a proactive approach to mitigate these risks. By adhering to machine learning (ML) classification algorithms and using data masking techniques.

ML approaches offer sophisticated ways of classifying sensitive data, relying on algorithms to systematically identify patterns and features that indicate sensitive information. A popular approach is supervised learning [13] [14], in which models are trained on labeled datasets to classify data into predefined categories, such as sensitive and non-sensitive [15]. Supervised learning algorithms such as random forests (RF) , support vector machines (SVM), or neural networks rely on historical data patterns and features to predict the sensitivity of new data [16] [17] [18]. In addition, unsupervised learning methods such as clustering can be used to group similar data items together, potentially revealing clusters of sensitive information [19]. Natural language processing (NLP) techniques can also play a key role, allowing analysis of textual data to identify sensitive content [20] [21] .

Data masking techniques may help also to protect sensitive data [22]. They are often used in data management and data leakage prevention systems (DLPs) to prevent data breaches and unauthorized access [23]. Common techniques include substitution, encryption, and tokenization [24]. These techniques preserve data confidentiality and integrity, making it possible to protect sensitive data while making it usable for legitimate purposes in a cloud environment .

The upcoming sections are structured as follows: Section 2 provides the context and background of the research. In Section 3, the related literature is reviewed. Moreover, Section 4 demonstrates the proposed framework by delineating the main elements of the system. Section 5 presents the datasets used, and both experimental design and results, while Section 6 contains the conclusion and outlines future work.

2. BACKGROUND

In this section, we seek to offer an overview and general background in relation to the topic explored in this paper. By giving a detailed examination of the context, we aim to establish a solid foundation for further discussion and analysis.

A. Cloud Computing

Cloud computing is a model for providing computing resources over the Internet on a pay-as-you-go basis. It allows customers to access servers, storage, databases, networks, and software applications without the need to possess or manage physical infrastructure. This flexible, scalable concept ensures cost-effective, efficient use of resources [25].

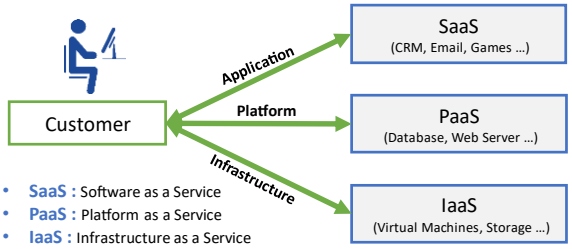

Figure 1. Cloud Service Models

Cloud service models, including Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS), provide various levels of abstraction and control. As shown in Fig. 1, IaaS provides basic computing resources on demand, while PaaS abstracts infrastructure to simplify development, and SaaS offers fully managed applications accessible over the Internet. Cloud environments are vulnerable to a wide range of security threats and attacks, originating from different categories of risks. These risks cover a spectrum of threats inherent to cloud infrastructures as shown in Table 1.

TABLE I CLOUD MAIN THREATS

Category	Threats
Authentication	- Weak or compromised credentials. -Inadequate authentication mechanisms.
Data Security	- Data breaches and leaks. - Insecure storage configurations.
Network Security	- Weak firewall rules. - Man-in-the-middleattacks.
Application Security	- Inadequate input validation. - Exposed APIs.
Human Factor	- Insider threats - Social engineering attacks.

B. Machine Learning

Machine learning is a field of artificial intelligence (AI) that allows learning and improvement from experience without the need to be explicitly programmed. It comprises algorithms that analyze data, identify patterns, and make predictions or decisions based on these patterns. The goal is to build models

that can be generalized from data to solve problems or to make specific predictions [13] [14].

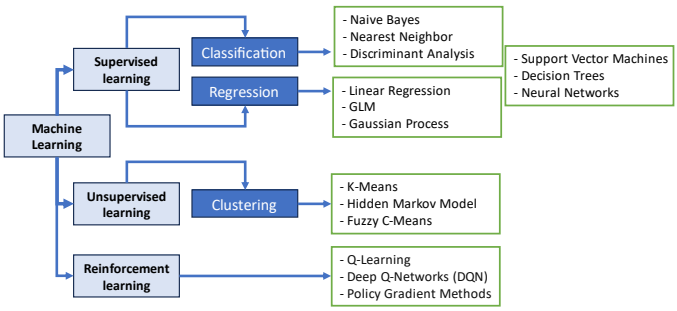

Figure 2. Machine Learning Models

As illustrated in Fig. 2, Each machine learning model operates separately, targeting specific problem areas. When they are combined, they achieve better performance than the other models. Hybrid models also minimize the limitations of individual basic models and take advantage of their different generalization mechanisms.

- *Supervised Learning:* These algorithms are trained using labeled data to create a mapping between input variables and output variables. This enables them to make predictions on new data. Examples of common algorithms in this category include linear regression, decision trees, and neural networks.
- *Unsupervised Learning:* In this approach, algorithms are trained on unlabeled data to discover hidden patterns or structures within the dataset. Clustering algorithms like K-means are commonly used for this purpose.
- *Reinforcement Learning:* This technique involves training agents to interact with an environment by taking actions and receiving feedback. The goal is to maximize cumulative rewards over time. Reinforcement learning has applications in fields such as robotics, games and autonomous systems.

C. *Data Masking*

Data Masking is a vital data security measure which aims to obscure sensitive information in order to protect it from being accessed by unauthorized parties [24]. This process usually involves substituting the original data with non-realistic, fictitious values, while maintaining the format and integrity of the data. Techniques such as tokenization, encryption and hashing can be employed to protect masked data [26] .

3. LITERATURE SURVEY

A. *Related Works*

Several studies have addressed the issue of data security in cloud computing. They fall into two categories: securing the container, which is the storage service in different contexts, namely the public, private and hybrid cloud. Securing the content, which refers to data in its three states: in transit, at rest and in use.

In [27], the author introduces a new secure cloud storage service designed to improve data security by implementing access control lists (ACLs), key rotation and metadata tagging. The author of [28] proposed a multi-layered defense mechanism to protect sensitive data stored in the cloud. Using techniques like Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), and Blockchain. The work in [29] introduced a user-side encrypted file system designed for cloud storage. It utilizes identity-based encryption scheme (IBE) and implements transparent encryption on a per-file basis using per-file keys. This approach enhances data security by encrypting files at the user's end before storing them in the cloud. Reference [30] propose a framework combining Ethereum blockchain technology with Ciphertext-policy attribute-based encryption (CP-ABE) to create a secure cloud storage solution.

The author of [31] proposed a new approach to guarantee data security using machine-learning classification algorithms. The Reuters-21578 dataset is used and trained using natural language processing (NLP) with four classifiers to evaluate the data. In [32], the author suggests a new model where cloud data is categorized based on its sensitivity, encrypted, randomized, and anonymized. Reference [33] proposed a differential Approach for Privacy-preserving Machine Learning Model (DA-PMLM) that ensures robust privacy protection for both data and classifiers. Experimented with a Naive Bayes classifier across multiple datasets, the model involves four entities: Data Owners (DOid), Classifier Owner (CO), Cloud Service Provider (CSP), and Request Users (RUid).

In [34], the authors present a novel three-dimensional CCDC sensitive information security storage algorithm, integrating advanced techniques such as feature combination for sensitive information filtration and encryption. Moreover, it implements a three-dimensional storage principle to ensure secure data storage. The system proposed in [35] utilizes JavaScript injection techniques and deep learning methods to sanitize sensitive on-premises data before uploading it to cloud storage. It consists of five components: Interceptor, Parser, Classifier, Sanitization, and Packer. The Interceptor intercepts HTTP/HTTPS traffic, while the Parser parses application protocols and extracts file content. The Classifier categorizes sensitive data, and the Sanitization module detects and sanitizes sensitive information. Finally, the Packer assembles redacted data into web requests, which are then sent to the cloud storage.

The work in [36] introduced the Scale-based Secure Sensitive Data (SSSD) cloud storage technique, aiming to provide personalized security levels for user data through a privacy score. The model utilizes Likert Scale assignment and Dichotomous Response Matrix generation to simplify data classification. Privacy scores identify common sensitive attributes across users, while association rule mining identifies user-specific sensitive attributes.

B. *Discussion*

Although models [27]-[30] present distinct advantages such as effective access control, cryptographic strength, and distributed architectures, they also encounter technical challenges such as management complexity and scalability limits. Nevertheless,

the research outlined above requires secure key management mechanisms to prevent exposure of cryptographic materials. In contrast, referenced approaches [31]-[33] offer promising methods to increase data security and privacy in cloud environments. However, these approaches lack comparative analyses, detailed evaluations, scalability considerations, and explicit discussions of threat models. Even though models proposed in [34]-[36] offer promising solutions to enhance data security in cloud storage, they encounter challenges related to performance, resource consumption, and practical implementation. Further research and comprehensive evaluations are needed to address these concerns and validate the effectiveness of these techniques in real-world scenarios.

From the current literature, it is evident that encryption techniques and the use of external tools to preprocess, classify, and manipulate data before storing it in the cloud are widely used. Despite this, existing research lacks studies that utilize existing cloud services to propose comprehensive data storage frameworks. Our approach takes advantage of hybrid techniques that combine data masking and classification algorithms with existing services provided by cloud service providers (CSPs). This strategy aims to reduce the complexity of the design while guaranteeing scalability. Additionally, the ability to be easily implemented in real-world scenarios is achieved by leveraging infrastructure as code (IaaC) tools.

4. PROPOSED MODEL

A. Dataset

In this research, as shown in Table 2 we only process personally identifiable information (PII) for dataset availability reasons, PII is defined as any data that can be used to identify a specific person. This includes information such as full name, social security number, date of birth, address, telephone number, e-mail address or bank account number. PII is considered sensitive because its exposure or unauthorized access can lead to a range of privacy and security risks, including identity theft, fraud, phishing, harassment, bullying and discrimination. Protecting these sensitive data is crucial to maintain the privacy and security of individuals in the cloud computing context .

TABLE II DATASET SOURCES

INFORMATION	SOURCE	ROWS
EMAIL	mailing list dataset / **pastebin.com**	2500
CREDIT CARD	credit card data / **kaggle.com**	800
PASSEPORT	passport synthetic data / **protecto.ai**	2498
IP ADDRESS	ip address blocks / **nirsoft.net**	2336
BIRTHDATE	indian women dataset / **kaggle.com**	2500

B. General Overview

As illustrated in Fig. 3, our proposed model is designed to guarantee the confidentiality and security of data sent to the cloud by individuals and companies. in fact, once the data is uploaded to the cloud, a data gateway is set up to capture the request and transfer it to the proposed framework for processing. once this is done, the output data is forwarded for storage.

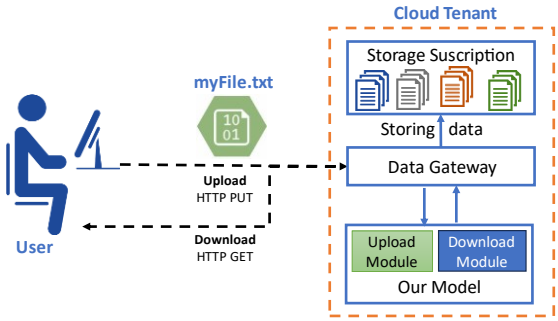

Figure 3. macro view of the proposed model

The cloud offers a wide range of data types, whether structured, semi-structured or unstructured. Table 3 shows the types of data storage services provided by the cloud service providers.

TABLE III CLOUD DATA STORAGE OFFERS

	Microsoft Azure	Amazon AWS	Oracle OCI	Google GCP
File	Azure File Storage	Amazon EFS	OCI File Storage	Google Cloud Filestore
Block	Azure Blob Storage	Amazon EBS	OCI Block Volume	Cloud Persistent Disk
Object	Azure Blob Storage	Amazon S3	OCI Object Storage	Google Cloud Storage

- *Object storage :* It arranges data into objects, which can be files, images, videos, or other unstructured data.
- *Block storage* : It splits data into blocks and stores them individually. It is employed for structured data .
- *File storage :* It offers a file system gateway through which data can be stored and accessed.

The proposed approach is divided into two modules, the first handles the upload flow and the second, the download flow, both of which operate in serverless mode, a service provided by the cloud service provider. Serverless computing, enables to run code without managing the underlying servers. Cloud service providers take care of infrastructure management, including provisioning, auto-scaling and maintenance, offering cost-effectiveness, scalability, flexibility and ease of use. We should mention that serverless computing in the cloud enables event-driven execution, in other words, in a serverless environment, code is initiated by specific events, such as HTTP requests (REST calls), file updates, file downloads and so on. Table 4 shows some serverless services offered by CSPs .

TABLE IV CLOUD SERVERLESS OFFERS

Microsoft Azure	Amazon AWS	Oracle OCI	Google GCP
Azure Functions	AWS Lambda	OCI Functions	Google Cloud Functions

C. Components

1) Upload Module

The proposed upload module serves as the access point for cloud-loaded data, featuring three essential elements, as illustrated in Fig. 4.

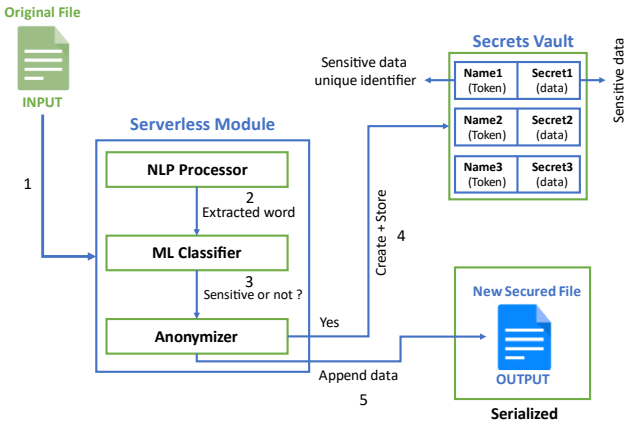

Figure 4. Upload Module

It seamlessly integrates with internal resources like Secrets Vault and Data Gateway, ensuring secure interactions. Strong authentication mechanisms enhance data security in transit, while its modular architecture efficiently handles a wide range of data formats and volumes. In summary, the upload module comprises:

a) NLP Processor

NLP utilizes computational methods to extract meaningful words from text by breaking it down into tokens, filtering out noise like punctuation and stopwords, and employing advanced techniques like part-of-speech tagging and named entity recognition. This process is fundamental for tasks such as sentiment analysis and information retrieval.

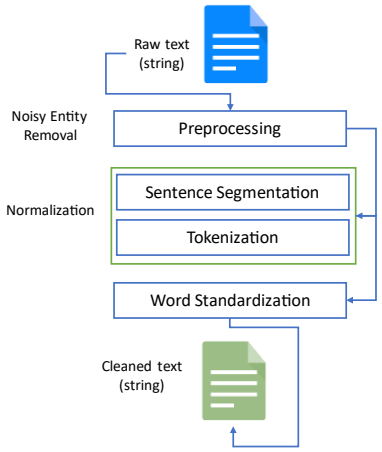

Figure 5. NLP Processor Pipeline

In our context, NLP Processor implements Spark NLP, an open-source library that provides simple, high-performance, and accurate NLP annotations for machine learning pipelines. It supports most NLP tasks and provides modules that can be used transparently. As shown in Fig. 5, Spark NLP processes data using pipelines, a structure that includes all the steps to be carried out on the given input data.

b) ML Classifier

In the classification phase, as indicated in Fig. 6, two techniques are employed. Pattern matching, which validates the sensitive data collected, using predefined patterns or regular expressions. Machine learning techniques, used to apply trained models to predict the category of the sensitive information, based on certain contextual features and characteristics related to the data itself.

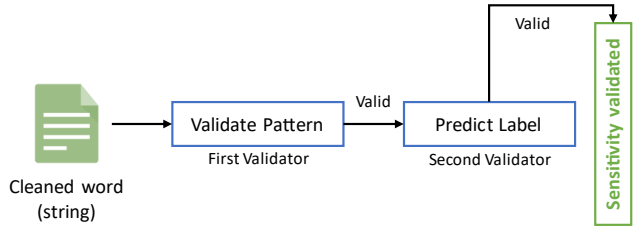

Figure 6. ML Classifier Phase

In the classification phase, as demonstrated in Fig. 6, two techniques are employed, and five sensitive information are evaluated as shown in Table 5.

TABLE V PROPOSED ITEMS AND PATTERNS

INFORMATION	PATTERN
EMAIL	^[\w\.-]+@[a-zA-Z\d-]+(\.[a-zA-Z\d-]+)+$
CREDIT CARD	\b(?:\d[-]*?){13,16}\b
PASSEPORT	^[A-Za-z0-9]{6,15}$
IP ADDRESS	**- (IPv4) :** \b(?:\d{1,3}\.){3}\d{1,3}\b **- (IPv6) :** \b(?:[0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}\b
BIRTHDATE	^(19|20)\d{2}-(0[1-9]|1[0-2])-(0[1-9]|[12]\d|3[01])$

In fact, pattern matching validates the sensitive data collected using predefined patterns and regular expressions, ensuring accuracy and consistency. On the other hand, machine learning techniques use trained models to predict the category of sensitive information . These approaches combine to accurately categorize sensitive information, improving the security and overall performance of the classification process. As indicated in the dataset section, we focus on sensitive personal information, listed in Table 5. We employ dataset to train our model using a set of algorithms. This methodology facilitates the evaluation of their performance, enabling us to determine the most appropriate algorithm for the given context.

c) Anonymizer

In this phase, we employ tokenization technique to replace sensitive information with randomly generated tokens. These tokens are stored in a separate mapping table, referred in Fig. 4 as Secrets Vault. Subsequently, the original information is substituted with a token, ensuring that the original value can be retrieved when necessary, by referencing the secret vault.

In cloud environments, a secret vault, also known as a secret manager, serves as a secure repository for storing sensitive information such as API keys, passwords, and cryptographic keys.

TABLE VI SECRET VAULT CLOUD OFFERS

Microsoft Azure	Amazon AWS	Oracle OCI	Google GCP
Azure Key Vault	AWS Secrets Manager	OCI Vault	Cloud Key Management Service

As demonstrated in Table 6, most cloud service providers offer a secret vault service under different names. This service provides centralized management and access control for secrets, ensuring their confidentiality and integrity. Additionally, it facilitates secure interaction among cloud services, applications, and users, while offering features such as versioning and auditing. During this phase too, symmetrical hashing with the MD5 algorithm is employed, as shown in (1), generating the token that serves as an identifier for sensitive information and replacing it in the stored data.

$$Token = MD5\,(sensitive\ information) \quad (1)$$

MD5 is a cryptographic hash function known for its pre-image resistance property, making it practically impossible to reverse the hashing process and recover the original input from the hash value. This property ensures robust security and data integrity.

2) Download Module

The proposed download module serves as an output point for secured data, ready to be served to the end-user, and comprises three essential elements, as shown in Fig. 7.

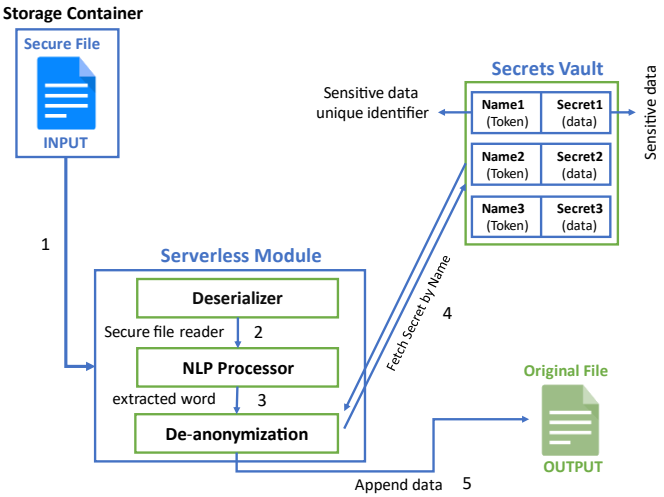

Figure 7. Download Module

The download module facilitates deserialization of stored data, performs sentence segmentation, and recovers sensitive information encrypted in the secret vault. It operates in three distinct phases to accomplish these tasks efficiently.

a) Deserializer

Data serialization is essential for security as it converts complex data into a format that can be efficiently stored, transmitted, and reconstructed, ensuring the integrity and confidentiality of data during transfer between systems. In our context, we opt for the "pickle" module from Python, commonly used for serialization. Serialization takes place in the last step of the upload module, as shown in Fig. 6, and serves as the entry point for the download process, as illustrated in Fig. 7.

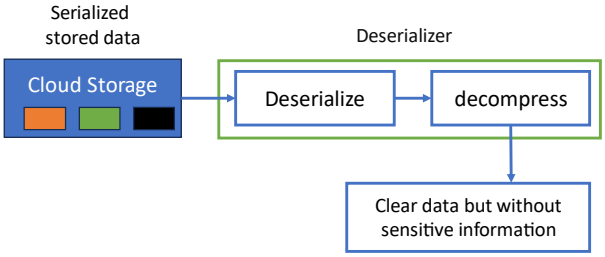

Figure 8. Deserialization phase

As we can see in Fig. 8, we have also thought about optimizing the serialization mechanism by compressing serialized data using algorithms such as GZIP or LZ4 to further reduce the size of the output and improve delivery performance.

b) NLP Processor

As explained in the upload module, this phase involves segmenting sentences to extract individual words, facilitating the retrieval of original information for reconstructing the original file. Additionally, a word size verification is implemented to minimize verification attempts on the secret vault. Specifically, the word size must match the size of the token previously configured for enhanced security.

c) De-anonymization

The secret vault operates in a key-value mode, where the value represents hidden information and corresponds to the token generated and stored during the upload phase.

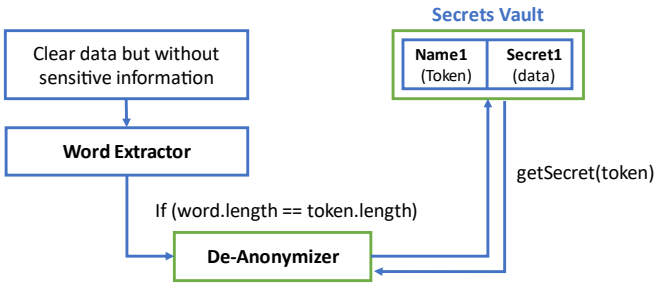

Figure 9. De-Anonymization phase

De-anonymization, as shown in Fig. 9, involves fetching tokens from the secured file and replacing them with the corresponding values found in the secrets vault to reconstruct the original clear file.

D. Implementation

In order to implement our system, we opted for Terraform. Terraform simplifies the provisioning and configuration of cloud resources by allowing users to define the infrastructure as code. With Terraform, infrastructure configurations are formulated in a declarative language, facilitating automation, consistency and scalability. This approach not only simplifies deployments, but also enhances collaboration and ensures infrastructure reproducibility across all environments. Terraform supports multi-cloud environments, providing greater flexibility and avoiding vendor lock-in, making it the ideal choice for modern cloud infrastructure management.

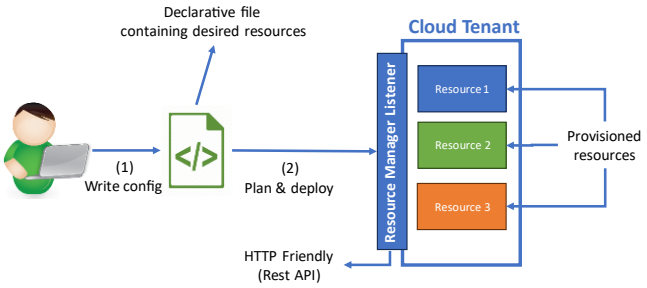

Figure 10. Implementation workflow

Terraform takes advantage of the http-based APIs offered by major cloud service providers, ensuring seamless compatibility with their platforms. The workflow shown in Fig. 10 enables Terraform to simplify the provisioning and configuration of cloud resources. As a result, Terraform functions as a versatile orchestrator, capable of provisioning resources on various cloud platforms through a unified set of commands. In addition, terraform's results provide a complete view of the provisioned infrastructure resources .

5. EXPERIMENTAL DESIGN AND RESULTS

A. Experimental Design

The metrics considered for evaluating the performance of the proposed system include sensitive data classification Accuracy, Recall, F1-Score data upload time, and data download time. The classifier performance is simulated using both Python environment in Microsoft Azure Functions and KNIME Analytics Platform, with a system configuration including 16GB RAM, Intel Core i7 processor, and Windows 11 operating system. To accomplish this, a random subset of categorized test data is reserved, and the predicted labels are compared with the true labels. Quantitative performance is evaluated after training the classifier with a labeled dataset. The mathematical representation of parameters employed to estimate the model's performance is shown from (2) to (4).

- *Accuracy :* It represents the ratio of correct predictions to the total number of input samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- *Precision :* It is the measure of the proportion of correctly predicted data records compared to the total predicted positive records. A higher precision indicates better performance of the classification model.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- *F1-Score :* It is also referred to as the harmonic mean, represents a balanced measure between recall and precision in the used classification model.

$$F1-Score = \frac{2TP}{2TP+FP+FN} \quad (4)$$

Table 7 illustrates the confusion matrix. TP, or True Positive, represents the number of correctly identified sensitive data. True Negative, or TN, indicates the number of correctly detected non-sensitive data. False Positive, or FP, denotes the number of incorrectly identified non-sensitive data as sensitive. False Negative, or FN, signifies the number of incorrectly identified sensitive data as non-sensitive.

TABLE VII CONFUSION MATRIX

		Actual Class	
		Positives	**Negatives**
Predicted Class	**Positives**	**TP** True Positive	**FP** False Positive
	Negatives	**FN** False Negative	**TN** True Negative

Three different ML models, namely LSTM, CNN and MLP, were trained and evaluated for a binary classification task. Parameters used for training include the number of epochs, set at 10, and a batch size of 1, indicating that each sample is treated individually during training. Models are compiled using the Adam optimizer and the binary cross-entropy loss function. After data pre-processing through tokenization and sequence filling, each model is trained on the dataset. The training time of each model is recorded in milliseconds.

TABLE VIII COMPARISON OF ALGORITHMS PERFORMANCE

	ML	*Train Time (ms)*	*Accuracy*	*Recall*	*F1 Score*
EMAIL	***LSTM***	*210271.44*	*99.92 %*	*99.86 %*	*99.93 %*
	CNN	*21741.12*	*100.0 %*	*100.0 %*	*100.0 %*
	MLP	*14396.72*	*99.08 %*	*98.6 %*	*99.22 %*
CREDIT CARD	***LSTM***	*44542.15*	*62.50 %*	*100 %*	*76.92 %*
	CNN	*7145.75*	*85.75 %*	*98.20 %*	*91.83 %*
	MLP	*14396.72*	*67.12 %*	*90.60 %*	*77.50 %*
BIRTHDATE	***LSTM***	*122794.02*	*100.0 %*	*100.0 %*	*100.0 %*
	CNN	*19040.19*	*100.0 %*	*100.0 %*	*100.0 %*
	MLP	*13872.15*	*99.96 %*	*99.93 %*	*99.96 %*
IP ADDRESS	***LSTM***	*196026.36*	*94.86 %*	*98.06 %*	*96.08 %*
	CNN	*15619.28*	*99.52 %*	*100.0 %*	*99.63 %*
	MLP	*9877.31*	*87.54 %*	*97.06 %*	*90.91 %*
PASSEPORT	***LSTM***	*78022.04*	*100.0 %*	*100.0 %*	*100.0 %*
	CNN	*15376.95*	*100.0 %*	*100.0 %*	*100.0 %*
	MLP	*10120.97*	*99.91 %*	*99.86 %*	*99.93 %*

The performance of each model was evaluated through a detailed analysis using a varied set of metrics, precisely described in Table 8. These metrics encompass critical aspects such as training time, accuracy, recall and F1 score, all of which

were measured carefully using the predictions generated by the trained models. These multi-faceted performance measures serve as essential benchmarks, providing a comprehensive overview of the models' classification feats.

As shown in Fig. 11, evaluation of data transfer prior to using our system illustrates varying upload and download times, measured in milliseconds (ms), for different data sizes. For example, for a data size of 10 KB, the upload time was 0.3726343 ms, while the download time was 0.6599721 ms. Similarly, for larger data, such as 10,000 KB, the load time increased to 5.7895292 ms, with a corresponding download time of 2.2125783 ms. However, as data size increased, download times increased significantly, download times showed minor variations, but remained stable overall, regardless of data size.

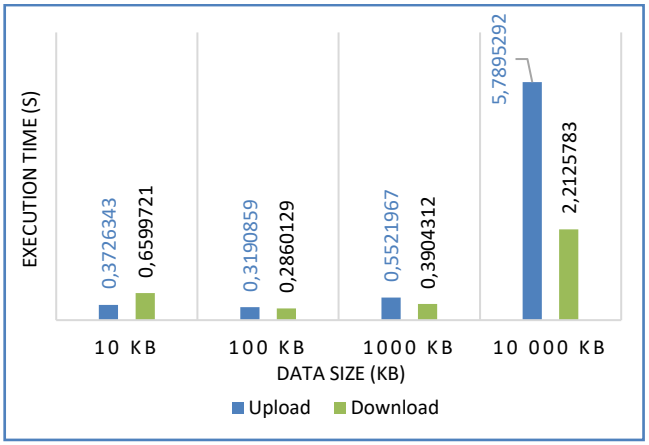

Figure 10. Data transfer (upload/download) pre- implementation

Following the implementation of our system, as illustrated in Fig. 11, the evaluation of data transfer reveals changes in upload and download times across various data sizes, measured in milliseconds (ms). For instance, for a data size of 10 KB, the upload time increased to 0.4607608 ms, while the download time decreased to 0.3886256 ms. Conversely, for larger data sizes such as 10,000 KB, the upload time was reduced to 6.8294536 ms, and the download time to 6.5997401 ms. These alterations signify the impact of our model on data transfer efficiency, showcasing both enhancements and adjustments in upload and download speeds for different data sizes.

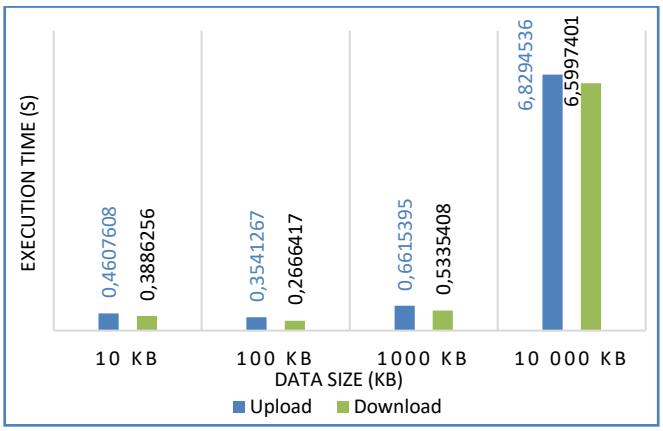

Figure 11. Data transfer (upload/download) post- implementation

The diagram depicted in Fig. 12 illustrates the comparison of upload times before and after the implementation of our system. Across various data sizes, discernible changes are observed. For instance, for data sets of 10 KB, the upload time decreased from 0.4607608 ms to 0.3726343 ms post-implementation. Conversely, for larger data sizes, such as 10,000 KB, a slight increase in upload time is noted, rising from 5.7895292 ms before implementation to 6.8294536 ms after.

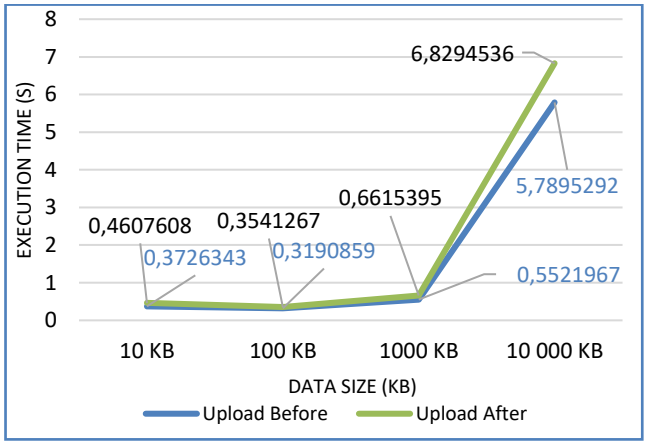

Figure 12. Upload time pre / post-implementation

As Fig. 13 shows, a comparison of download times before and after the implementation of our system reveals significant changes for different data sizes. For example, for a data size of 10 KB, the download time dropped from 0.6599721 ms to 0.3886256 ms after implementation. Similarly, for larger data, such as 10,000 KB, download time increased significantly, from 2.2125783 ms before implementation to 6.5997401 ms afterwards. These changes underline the impact of our system on download time.

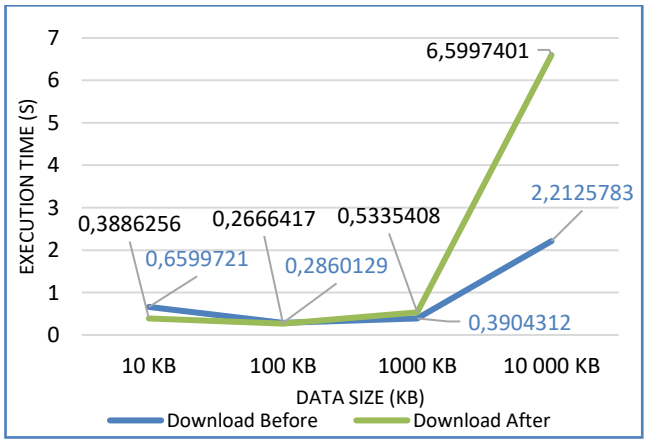

Figure 13. Download time pre / post-implementation

B. Finding

Our experiments evaluated the performance of machine learning models across various classification tasks. In email classification, the CNN model exhibited remarkable precision, achieving 100% accuracy, recall, and F1 score, outperforming both the LSTM and MLP models. For credit card classification, the CNN model demonstrated the highest precision (85.75%) and F1 score (91.83%), with significant recall improvements over the LSTM model. Similarly, in birthdate classification, all models achieved perfect accuracy, recall, and F1 score, with

variable training times: the LSTM model took 122794.02 ms, the CNN model 19040.19 ms, and the MLP model 13872.15 ms. In IPAddress classification, the CNN model attained superior accuracy (99.52%) and F1 score (99.63%), with perfect recall, while the MLP model showed competitive recall (97.06%) and the shortest training time. Moreover, in passport classification, all models performed exceptionally well, achieving 100% accuracy, recall, and F1 score. These results underscore the efficacy of machine learning models in accurately classifying diverse data types, with the CNN model emerging as the most suitable algorithm for our context. In the other hand, the evaluation of upload and download performance before and after the implementation of our model led to some important conclusions. Initially, download times ranged from 0.3726 ms for 10 Kb files to 5,789.53 ms for 10,000 Kb files, while upload times remained relatively stable. After implementation, download times improved significantly for all file sizes, decreasing to 0.4608 ms for 10 Kb files and 6,829.45 ms for 10,000 Kb files. Conversely, download times showed mixed results, with reductions observed for small files but marginal increases for large files. The optimized download and upload time is attributed to the use of serialization and compression techniques in both modules, minimizing transfer times for smaller data sets. However, despite these optimizations, the observed increase in upload and download times after implementing our model highlights the significant processing load on our serverless module, particularly for large data. Therefore, while serialization and compression techniques enhance performance for smaller datasets, they may not fully mitigate the impact of increased processing demands on transfer times for larger data volumes.

6. CONCLUSION AND FUTURE WORK

In a cloud environment, resources are shared among multiple tenants, making them susceptible to threats from both internal and external sources. Storing sensitive data in a shared storage space presents inherent risks, from accidental breaches to malicious attacks. Given these factors, it is essential to deploy appropriate security measures.

While several practical considerations exist for cloud storage security, the key objective of this work is to propose an effective framework that addresses such concepts and provides a viable solution for mitigating the privacy and security challenges of cloud storage.

This paper has focused on a limited aspect of using machine learning to improve the classification of sensitive data and has concentrated on the key features to consider when categorizing and masking personal information in a cloud context. It should be noted that the technical aspects discussed here are not perfect, and that it is possible to improve and refine the proposed approach in the future.

In future research, we plan to explore a hybrid approach integrating data profiling techniques with data mining algorithms . Additionally, we aim to investigate the potential benefits of integrating blockchain technology into cloud data storage security to enhance both transparency and accountability of all data transactions .

ACKNOWLEDGMENT

This research was made possible with the support of the C3S Research Laboratory. We would like to express our gratitude to the researchers whose ideas and expertise greatly contributed to this work. Additionally, we are extremely grateful to those who indirectly contributed to this research by sharing their valuable ideas and approaches through scientific papers and articles.

REFERENCES

[1] I. Gupta, A. K. Singh, C. -N. Lee and R. Buyya, "Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions," in *IEEE Access*, vol. 10, pp. 71247-71277, 2022.

[2] L. M'Rhaouarh, N. Chafiq and A. Namir, "Practices and usages of the cloud computing as a solution to rise to the challenge of the digitalization of Moroccan companies," *2018 4th International Conference on Optimization and Applications (ICOA)*, Mohammedia, Morocco, 2018, pp. 1-5.

[3] A. Kaur, V. P. Singh and S. Singh Gill, "The Future of Cloud Computing: Opportunities, Challenges and Research Trends," *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, Palladam, India, 2018, pp. 213-219.

[4] M. E. Moudni and E. Ziyati, "A Multi-Cloud and Zero-Trust based Approach for Secure and Redundant Data Storage," *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Istanbul, Turkiye, 2023, pp. 1-6.

[5] P. Anand, J. Ryoo and H. Kim, "Addressing Security Challenges in Cloud Computing — A Pattern-Based Approach," *2015 1st International Conference on Software Security and Assurance (ICSSA)*, Suwon, Korea (South), 2015, pp. 13-18.

[6] N. Tutubala and T. E. Mathonsi, "A Hybrid Framework to Improve Data Security in Cloud Computing," *2021 Big Data, Knowledge and Control Systems Engineering (BdKCSE)*, Sofia, Bulgaria, 2021, pp. 1-5.

[7] C. Choudhary, N. Vyas and U. Kumar Lilhore, "Cloud Security: Challenges and Strategies for Ensuring Data Protection," *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, Tashkent, Uzbekistan, 2023, pp. 669-673.

[8] Pottier and J. -M. Menaud, "TrustyDrive, a Multi-cloud Storage Service That Protects Your Privacy," *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, 2016, pp. 937-940.

[9] Zhe, W. Qinghong, S. Naizheng and Z. Yuhan, "Study on Data Security Policy Based on Cloud Storage," *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, Beijing, China, 2017, pp. 145-149.

[10] S. Nepal, C. Friedrich, L. Henry and S. Chen, "A Secure Storage Service in the Hybrid Cloud," *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, Melbourne, VIC, Australia, 2011, pp. 334-335.

[11] A. Batista De Carvalho, M. F. De Castro and R. M. De Castro Andrade, "Secure Cloud Storage Service for Detection of Security Violations," *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, Spain, 2017, pp. 715-718.

[12] J. . Hai, "Network Cloud Storage Service Architecture Analysis and Research," *2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Macau, China, 2016, pp. 413-416.

[13] S. Singhal, R. Srivastava, R. Shyam and D. Mangal, "Supervised Machine Learning for Cloud Security," *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, 2023, pp. 1-5.

[14] D. Bhamare, T. Salman, M. Samaka, A. Erbad and R. Jain, "Feasibility of Supervised Machine Learning for Cloud Security," *2016 International Conference on Information Science and Security (ICISS)*, Pattaya, Thailand, 2016, pp. 1-5.

[15] R. Kour, S. Koul and M. Kour, "A Classification Based Approach For Data Confidentiality in Cloud Environment," *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, Jammu, India, 2017, pp. 13-18.

[16] W. -T. Su and C. -Y. Dai, "QoS-aware distributed cloud storage service based on erasure code in multi-cloud environment," *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 2017, pp. 365-368.

[17] V. Bucur, C. Dehelean and L. Miclea, "Object storage in the cloud and multi-cloud: State of the art and the research challenges," *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania, 2018, pp. 1-6.

[18] M. A. Zardari, L. T. Jung and N. Zakaria, "K-NN classifier for data confidentiality in cloud computing," *2014 International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, 2014, pp. 1-6.

[19] A. N. Khan, M. Yu Fan, A. Malik and R. A. Memon, "Learning from Privacy Preserved Encrypted Data on Cloud Through Supervised and Unsupervised Machine Learning," *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2019, pp. 1-5.

[20] B. Jayaram, T. Sethukarasi, M. Sindhu and H. Jeyamohan, "A Summary on Privacy and Security in Cloud Data using Various Approaches," *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2023, pp. 1746-1752.

[21] F. Martinelli, F. Marulli, F. Mercaldo, S. Marrone and A. Santone, "Enhanced Privacy and Data Protection using Natural Language Processing and Artificial Intelligence," *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020, pp. 1-8.

[22] Y. Yuan, J. Zhang, W. Xu and Z. Li, "Enable data privacy, dynamics, and batch in public auditing scheme for cloud storage system," *2021 2nd International Conference on Computer Communication and Network Security (CCNS)*, Xining, China, 2021, pp. 157-163.

[23] Z. C. Nxumalo, P. Tarwireyi and M. O. Adigun, "Towards privacy with tokenization as a service," *2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST)*, Ota, Nigeria, 2014, pp. 1-6.

[24] Z. Aslanyan and M. S. Boesgaard, "Privacy Analysis of Format-Preserving Data-Masking Techniques," *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*, Copenhagen, Denmark, 2019, pp. 1-6.

[25] A. S. Al-Ahmad and H. Kahtan, "Cloud Computing Review: Features And Issues," *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, Shah Alam, Malaysia, 2018, pp. 1-5.

[26] M. F. Adak, Z. N. Kose and M. Akpinar, "Dynamic Data Masking by Two-Step Encryption," *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Sivas, Turkiye, 2023, pp. 1-5.

[27] C. A. Batista De Carvalho, M. F. De Castro and R. M. De Castro Andrade, "Secure Cloud Storage Service for Detection of Security Violations," 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, Spain, 2017, pp. 715-718.

[28] M. Y. Shakor, M. I. Khaleel, M. Safran, S. Alfarhood and M. Zhu, "Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security," in IEEE Access, vol. 12, pp. 26334-26343, 2024.

[29] O. A. Khashan, "Secure Outsourcing and Sharing of Cloud Data Using a User-Side Encrypted File System," in IEEE Access, vol. 8, pp. 210855-210867, 2020.

[30] S. Wang, X. Wang and Y. Zhang, "A Secure Cloud Storage Framework With Access Control Based on Blockchain," in IEEE Access, vol. 7, pp. 112713-112725, 2019.

[31] F. Ahmad, A. Nawaz, T. Ali, A. A. Kiani, and G. Mustafa, "Securing Cloud Data: A Machine Learning based Data Categorization Approach for Cloud Computing," in Proceedings of the IEEE, 2022.

[32] A. Singh, M. Bala, and S. Kaur, "Design and Implementation of Secure Multi-Authentication Data Storage in Cloud using Machine Learning Data Classification," International Journal of Computer Applications, vol. 161, pp. 48-51, 2017.

[33] R. Gupta and A. K. Singh, "A Differential Approach for Data and Classification Service-Based Privacy-Preserving Machine Learning Model in Cloud Environment," New Generation Computing, vol. 40, pp. 737-764, 2022.

[34] Z. Li and J. Wang, "Security Storage of Sensitive Information in Cloud Computing Data Center," International Journal of Performability Engineering, 2019.

[35] P. Han, C. Liu, J. Cao, S. Duan, H. Pan, Z. Cao, and B. Fang, "CloudDLP: Transparent and Scalable Data Sanitization for Browser-Based Cloud Storage," in IEEE Access, vol. 8, pp. 68449-68459, 2020.

[36] M. Sumathi and S. Sangeetha, "Scale-based secured sensitive data storage for banking services in cloud," Int. J. Electron. Bus., vol. 14, pp. 171-188, 2018.

El Moudni Mohammed received master's degree in Big Data Engineering from the Faculty of Science Rabat, graduating in 2019. Currently, he is a Ph.D. student at the C3S Laboratory affiliated with ENSEM CEDOC at the University of Hassan 2 Casablanca, under the supervision of Prof. ZIYATI El Houssaine. He is currently working on data and cloud security .

El Houssine Ziyati received PHD degree in Computer Science from Mohammed V University in 2010, presently, he is a Professor in Computer Engineering department in ESTC Institute of Technology, Casablanca, Morocco in Intelligence, Networking and Data warehousing.