International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1-14

http://dx.doi.org/10.12785/ijcds/1571016609

A Hybrid Approach to Enhancing Personal Sensitive Information Protection in the Context of Cloud Storage

Mohammed El Moudni[1] **and Elhoussaine Ziyati**[1]

[1]*C3S Research Laboratory, High School of Technology, Hassan II University, Casablanca, Morocco*

Received 5 April 2024, Revised 21 August 2024, Accepted 4 August 2024

Abstract: The growing use of cloud computing and increasing popularity of digital technologies have made it essential to store and process personal data in cloud environments. As organizations and individuals continue to adopt cloud services, the security of sensitive personal information in this dynamic environment has become a top priority. Ensuring the confidentiality, integrity, and availability of personal data in the cloud is critical for mitigating the risks associated with cyber threats. This study examines security issues related to personal information in cloud systems and proposes a new approach that leverages machine learning (ML) classification and data tokenization techniques using serverless and secret vault services provided by cloud service providers (CSPs). Supervised learning algorithms, including Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP), are used for data label prediction. Notably, we found that the CNN achieved a remarkable 100% accuracy on a large dataset, ensuring perfect classification with double validation using pattern matching. In addition, natural language processing (NLP) techniques are employed to clean and prepare data content, whereas data tokenization is used to ensure data confidentiality and integrity. Furthermore, an analysis of both model overhead and cloud performance revealed that our model is scalable, and data handling using our approach has no major significant impact on time costs. This study also provides an overview of cloud computing, its service models, and the main security threats inherent in the cloud infrastructure. The experimental design and results based on specific datasets validate the effectiveness of the proposed hybrid approach in enhancing the protection of sensitive personal information in cloud storage.

Keywords: Machine Learning, Personally Identifiable Information, Data Privacy, Cloud Storage, Data Security

1. INTRODUCTION

Finding sufficient storage space to accommodate data is a significant challenge for many IT professionals, researchers, and individuals [1]. Cloud storage services enable individuals and organizations to embrace a digital paradigm characterized by flexibility, efficiency, and accessibility [2]. By exploiting the capabilities of cloud storage solutions, they can overcome the limitations of traditional data management, thereby 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 has introduced significant concerns regarding 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 a key challenge in the design of cloud storage security approaches [8]. It is crucial to build a model that can accurately differentiate normal data from sensitive data [9] while maintaining sufficient simplicity and performance for realistic deployment in a cloud environment rather than becoming either too complex or consuming resources [10] [11]. To address these security challenges, our research introduces a novel approach that integrates machine-learning (ML) classification algorithms with data-masking techniques to enhance the protection of sensitive data in cloud environments.

Considering the above challenges, the primary objective of this study is to examine the security issues associated with personal information in cloud systems and suggest a proactive approach to mitigate these risks. By adhering to machine-learning (ML) classification algorithms and using data-masking techniques. Our approach aims to simplify design complexity while ensuring scalability and practical implementation, with potential applications across various industries, such as healthcare, finance, and government. This framework not only addresses current security concerns but also sets the stage for future improvements and

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

refinements.

ML approaches offer sophisticated ways of classifying sensitive data by relying on algorithms to systematically identify patterns and features that indicate sensitive information. A popular approach is supervised learning [12] [13], in which models are trained on labeled datasets to classify data into predefined categories, such as sensitive and insensitive [14]. 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 [15] [16] [17]. In addition, unsupervised learning methods such as clustering can be used to group similar data items together, potentially revealing clusters of sensitive information [18]. Natural language processing (NLP) techniques can also play a key role in allowing the analysis of textual data to identify sensitive content [19] [20].

Data masking techniques may also help protect sensitive data [21]. They are often used in data management and data leakage prevention systems (DLPs) to prevent data breaches and unauthorized access [22]. Common techniques include substitution, encryption, and tokenization [23]. These techniques preserve data confidentiality and integrity, making it possible to protect sensitive data while making them usable for legitimate purposes in a cloud environment.

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

2. BACKGROUND

In this section, we offer an overview and a general background on the topic explored in this paper. By examining the context in detail, we aimed to establish a solid foundation for further discussion and analysis.

A. Cloud Computing

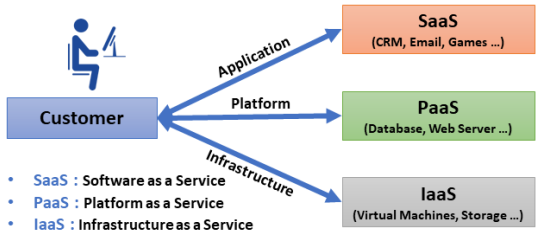

Figure 1. Types of Cloud Services

Cloud computing is a model that provides 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 and scalable concept ensures cost-effective and efficient use of resources [24].

Cloud service models, including Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS), offer varying levels of abstraction and control. As depicted in Fig. 1, IaaS provides basic computing resources on demand, PaaS abstracts infrastructure to simplify development, and SaaS offers fully managed applications accessible over the Internet. Cloud storage services, which exemplify the Infrastructure as a Service (IaaS) model, provide scalable storage solutions managed remotely by third-party providers, enabling users to store and access data over the internet.

The adoption of cloud storage has introduced significant security challenges. The confidentiality of sensitive information becomes compromised as it becomes vulnerable to unauthorized access and breaches. Ensuring data integrity is equally critical, with risks such as data tampering or corruption during transmission or storage. Furthermore, maintaining data availability is essential to mitigate the operational impacts and financial losses caused by disruptions or downtime. These challenges highlight the importance of robust security measures, including encryption, access control, authentication mechanisms, and continuous monitoring, to secure sensitive data against evolving cyber threats in dynamic cloud environments.

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-middle attacks.
Apps Security	- Inadequate input validation. - Exposed APIs.
Human Factor	- Insider threats. - Social engineering attacks.

Cloud environments are susceptible to a wide range of security threats and attacks originating from different risk categories. These risks cover a spectrum of threats inherent to cloud infrastructure, as shown in Table I.

B. Machine Learning

Machine learning is a field of artificial intelligence (AI) that allows learning and improvement from experience without requiring explicit programming. 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 make specific predictions [12] [13].

As illustrated in Fig. 2, each machine-learning model operates separately, targeting specific problem areas. When

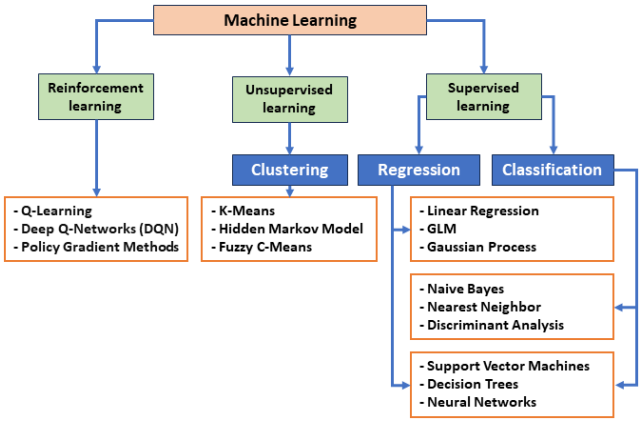

Figure 2. Machine Learning Models

combined, they achieved a better performance than the other models. Hybrid models also minimize the limitations of individual basic models and exploit their different generalization mechanisms.

- Supervised Learning: These algorithms are trained using labeled data to create a mapping between the input and output variables. This enables them to make predictions based on the new data. Examples of the common algorithms in this category include linear regression, decision trees, and neural networks.
- Unsupervised Learning: These algorithms are trained on unlabeled data to discover hidden patterns or structures within a dataset. Clustering algorithms such as K-means are commonly used for this purpose.
- Reinforcement Learning: This technique involves training agents to interact with an environment by taking action and receiving feedback. The goal is to maximize the 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 that 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.

3. LITERATURE SURVEY

A. *Related Work*

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

The authors of [25] developed a data protection strategy for cloud storage that integrates machine learning-driven encryption and anomaly detection. A comprehensive survey conducted by Patel et al. [26] explored machine-learning techniques, including neural networks and clustering, to enhance cloud data security. Chen et al. [27] reviewed encryption algorithms and access control mechanisms in cloud computing to improve data security. Sun et al. [28] proposed methods for preserving data privacy in cloud environments by discussing differential privacy and federated learning techniques. The implementation of machine-learning algorithms, such as decision trees and support vector machines [29], focuses on efficient and secure data storage in the cloud. Wu et al. [30] ensured secure and privacy-preserving cloud data storage by utilizing homomorphic encryption and blockchain technology. A machine-learning approach that incorporates deep learning and reinforcement learning [31] addresses real-time threat detection and mitigation in cloud storage. Li et al. [32] propose advanced encryption techniques and data masking methods using machine learning for protecting sensitive data in cloud storage. Patil et al. [33] explored anomaly detection and data obfuscation techniques to enhance data privacy and security in cloud environments. Kim et al. [34] effectively managed data security and privacy in cloud storage by employing cryptographic protocols and machine-learning-based intrusion detection systems.

In [35], the author introduced 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 [36] proposed a multilayered defense mechanism to protect the sensitive data stored in the cloud. Techniques include Elliptic Curve Cryptography (ECC), advanced encryption standards (AES), and blockchain. The study in [37] introduced a user-side encrypted file system designed for cloud storage. It utilizes an 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 end of the user before storing them in the cloud. Reference [38] proposed a framework that combines Ethereum blockchain technology with ciphertext-policy attribute-based encryption (CP-ABE) to create a secure cloud storage solution.

The authors of [39] proposed a new approach to guarantee data security using machine learning classification algorithms. The Reuters-21578 dataset was trained using natural language processing (NLP) with four classifiers to evaluate the data. In [40], the author suggested a new model in which cloud data are categorized based on their sensitivity, encrypted, randomized, and anonymized. Reference [41] proposed a differential approach for a 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

TABLE II. Comparison of techniques used in other models

Reference	Confidentiality	Integrity	Scalability	ML Based	Cloud Based
[25]	X	X	-	X	-
[26]	X	X	X	X	-
[27]	X	-	-	X	-
[28]	X	X	-	X	-
[29]	X	X	-	X	-
[30]	X	X	X	X	-
[31]	X	-	X	X	X
[32]	X	X	-	X	-
[33]	X	X	X	X	X
[34]	X	X	-	X	-

(CSP), and Request Users (RUid).

In [42], the authors presented a novel three-dimensional CCDC sensitive information security storage algorithm that integrates 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 [43] utilizes JavaScript injection techniques and deep learning methods to sanitize sensitive on-premise data before uploading them to cloud storage. It consists of five components: an interceptor, parser, classifier, sanitizer, and packer. The Interceptor intercepts the HTTP/HTTPS traffic, while the parser parses the application protocols and extracts the 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 [44] introduced a 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 the data classification. Privacy scores identify common sensitive attributes across users, whereas association rule mining identifies user-specific sensitive attributes.

B. Discussion

Researchers, presented in Table II including Yang et al. [25], encountered several challenges in dynamic cloud environments that affected scalability and operational complexity, which could limit the applicability of their solutions to large-scale deployments. Similarly, Patel et al. [26] pointed out practical complexities in implementing various machine learning techniques across diverse cloud infrastructures, hindering widespread adoption. Chen et al. [27] emphasized the importance of strong cryptographic algorithms, but they also acknowledged the scalability concerns associated with computational efforts in large-scale cloud deployments. Sun et al. [28] faced difficulties in integrating their privacy-preserving techniques with existing cloud architectures. Zhou et al. [29] achieved significant performance benefits but required adaptation to evolving cloud infrastructures. Wu et al. [30] emphasized the importance of security but required scalable management solutions. Khan et al. [31] developed deep learning models for threat detection, which needed further validation for broader scalability. Li et al. [32] proposed advanced encryption and data masking techniques that required integration with existing cloud frameworks. Patil et al. [33] enhanced anomaly detection capabilities but needed improvements in scalability. Kim et al. [34] employed cryptographic protocols that required adaptation to evolving threats in diverse contexts.

Although models [35]-[38] 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 management mechanisms to prevent exposure to cryptographic materials. In contrast, the referenced approaches [39]-[41] 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 the threat models. Although the models proposed in [42]-[44] offer promising solutions for enhancing data security in cloud storage, they encounter challenges related to performance, resource consumption, and practical implementation. Further research and comprehensive evaluations are required to address these concerns and validate the effectiveness of these techniques in real-world scenarios.

Current security measures in cloud storage, such as encryption, access controls, multifactor authentication, monitoring, and redundancy, are utilized to guarantee data integrity, availability, and confidentiality. However, ongoing challenges include evolving encryption standards, managing access policies, resource-intensive monitoring, and dependencies on third-party providers. Existing literature emphasizes the widespread adoption of encryption techniques and the use of external tools for preprocessing, classifying, and manipulating data before storage in the cloud. Despite this, research often lacks comprehensive studies utilizing cloud services to propose secure data-storage frameworks. Based on Table II, which compares various approaches previously proposed by the community that partially resemble our own, we can assert that our work makes a significant contribution,

as highlighted by the findings discussed in the experimental and results section. Our approach integrates hybrid techniques, combining data tokenization and classification algorithms with existing cloud services offered by CSPs. This strategy is designed to simplify design complexity while ensuring scalability and practical implementation in real-world scenarios, leveraging infrastructure-as-code (IaaC) tools to facilitate its deployment.

4. PROPOSED MODEL

A. Dataset

TABLE III. Dataset Sources

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

In this research, as shown in Table III, we only process personally identifiable information (PII) for reasons of dataset availability, which 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 are considered sensitive because their exposure or unauthorized access can lead to a range of privacy and security risks, including identity theft, fraud, phishing, harassment, bullying, and discrimination. Protecting sensitive data is crucial for maintaining the privacy and security of individuals in the cloud computing context.

B. General Overview

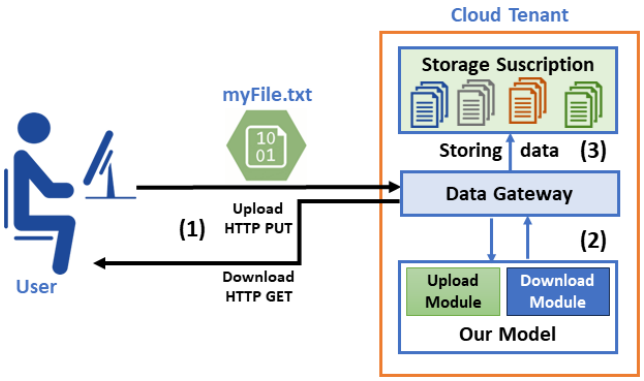

Figure 3. Macro View of the Proposed Model

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

TABLE IV. Cloud Data Storag 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

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

- Object storage: Data are arranged into objects that can be files, images, videos, or other unstructured data.
- Block storage: This splits data into blocks and stores them individually. It was employed for the structured data.
- File storage: This 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 handles the download flow, both of which operate in serverless mode, a service provided by the cloud service provider. Serverless computing enables code to run without managing the underlying servers. Cloud service providers manage infrastructure, including provisioning, auto-scaling, and maintenance, offering cost-effectiveness, scalability, flexibility, and ease of use. Table V shows some serverless services offered by the CSPs.

TABLE V. Cloud Serverless Offers

CSP	Serverless Offer
Microsoft Azure	Azure Functions
Amazon AWS	AWS Lambda
Oracle OCI	OCI Functions
Google GCP	Google Cloud Functions

Serverless computing in the cloud enables event-driven execution, that is, in a serverless environment, code is initiated by specific events, such as HTTP requests (REST calls), file updates, and file downloads.

C. Components

1) Upload Module

The proposed upload module serves as the access point for cloud-loaded data and features three essential

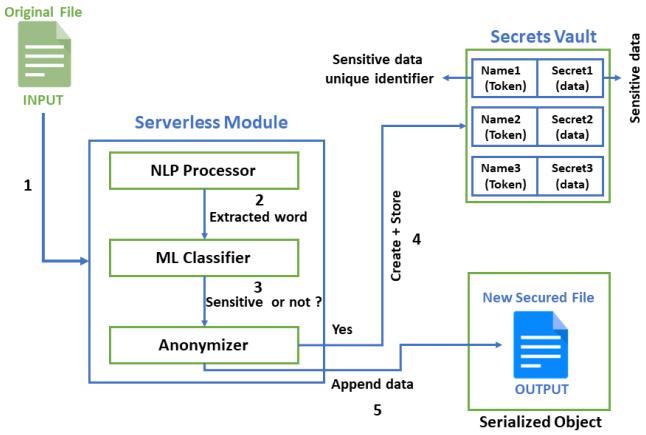

Figure 4. Upload Module

elements, as illustrated in Fig. 4. It seamlessly integrates with internal resources, such as the Secrets Vault and Data Gateway, ensuring secure interactions. Strong authentication mechanisms enhance data security in transit, whereas their modular architecture efficiently handles a wide range of data formats and volumes. In summary, the upload module comprises the following.

a) NLP Processor

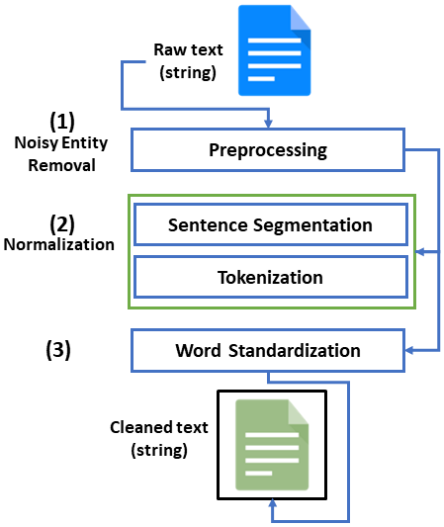

Figure 5. NLP Processor

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

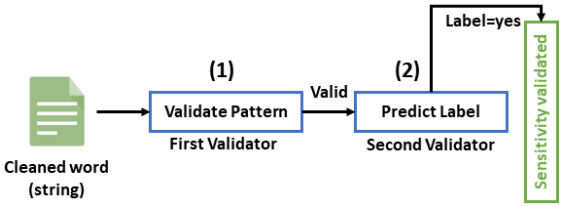

Figure 6. ML Classifier

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

TABLE VI. Proposed PII and Patterns

PII	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-`
BIRTHDATE	`^(19|20)\d{2}-(0[1-9]|1[0-2])-(0[1-9]|[12]\d|3[01])$`

In the classification phase, as shown in Fig. 6, two techniques were employed, and five types of sensitive information were evaluated. Pattern matching validates sensitive data collected using predefined patterns and regular expressions, ensuring accuracy and consistency. As depicted in experimental design sub-section, machine learning techniques use trained models to predict the categories of sensitive information. These approaches are combined to accurately categorize sensitive information and improve the security and overall performance of the classification process. As indicated in the dataset section, we focus on sensitive

personal information, as listed in Table III. We employed a dataset to train the model using a set of algorithms. This methodology facilitates the evaluation of performance, enabling us to determine the most appropriate algorithm for a given context.

c) Anonymizer

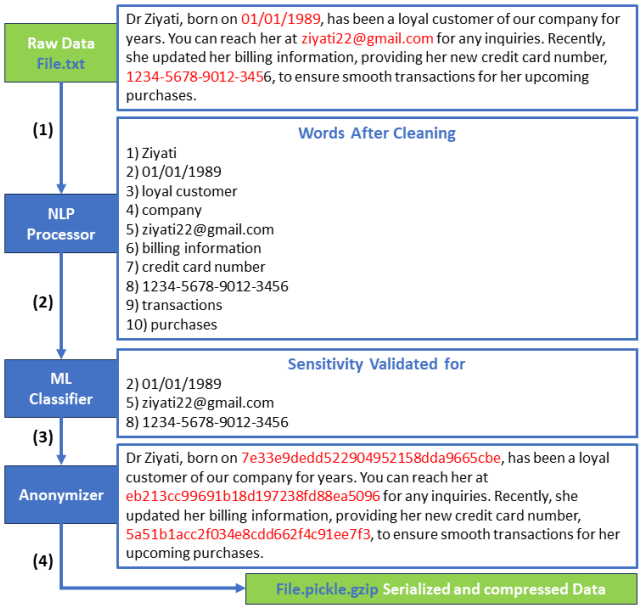

Figure 7. Data States during the Upload Process

In this phase, we employ a tokenization technique to replace the sensitive information with randomly generated tokens. These tokens are stored in a separate mapping table, referred to in Fig. 4 as the Secret Vault. As depicted in Fig. 7 the original information is substituted by a token, ensuring that the original value can be retrieved when necessary by referencing the secret vault.

TABLE VII. Cloud Secret Vault Offers

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

As shown in Table VII, most cloud service providers offer secret vault services under different names. This service provides centralized management and access control for secrets, thereby ensuring confidentiality and integrity. Additionally, it facilitates secure interaction among cloud services, applications, and users while offering features such as versioning and auditing.

$$\text{Token} = \texttt{MD5}(\textbf{sensitive information}) \quad (1)$$

During Tokenization step, symmetrical hashing with the MD5 algorithm is employed, as shown in (1), generating a token that serves as an identifier for sensitive information and replacing it in the stored data. MD5 is a cryptographic hash function known for its pre-image resistance property, which makes it practically impossible to reverse the hashing process and recover the original input from the hash value. This property ensures a robust security and data integrity using data tokenization. The process and the result of each step during the upload are illustrated in Figure 7 .

2) Download Module

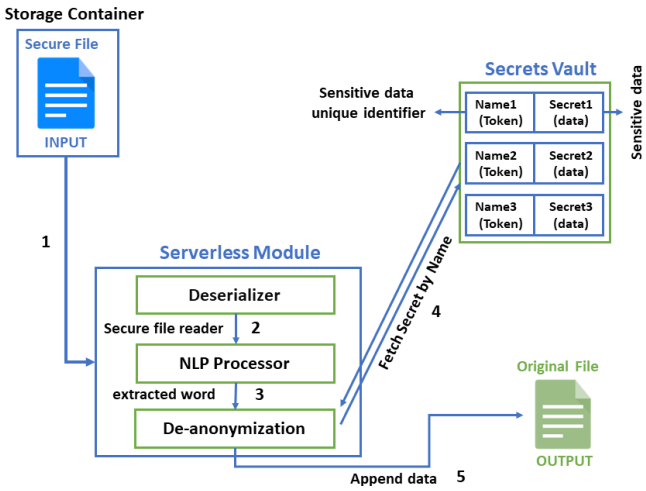

Figure 8. Download Module

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

a) Deserializer

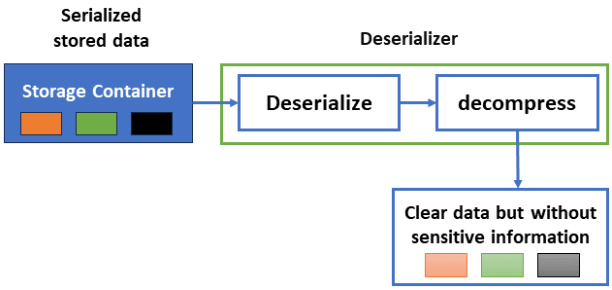

Figure 9. Deserialization Process

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 occurs in the last step of the upload module, as shown in Fig. 4, and serves as the entry point for the downloading process, as illustrated in Fig. 8.

As shown in Fig. 9, we also consider 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 the delivery performance.

b) NLP Processor

Natural Language Processing (NLP) employs computational techniques to extract meaningful words from text. This is achieved by breaking down the text into tokens, removing noise such as punctuation and stopwords, and applying advanced techniques such as part-of-speech tagging and named entity recognition. These methods are essential for tasks such as sentiment analysis and information retrieval.

c) De-anonymization

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

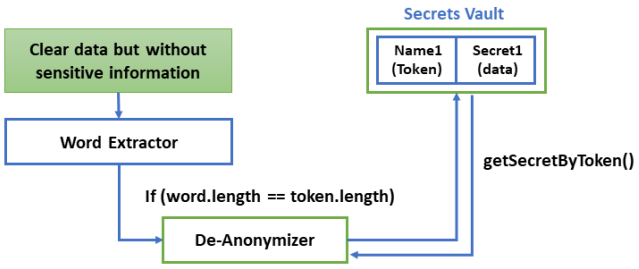

Figure 10. De-anonymization Step

As illustrated in Fig. 10, deanonymization involves fetching tokens from the secured file and replacing them with the corresponding values found in the secret vault to reconstruct the original clear file.

D. Implementation

We opted for terraform to implement our system. Terraform simplifies the provisioning and configuration of cloud resources by allowing users to define infrastructure as a code. With Terraform, infrastructure configurations are formulated in 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, provides greater flexibility, and avoids vendor lock-in, making it an ideal choice for modern cloud infrastructure management.

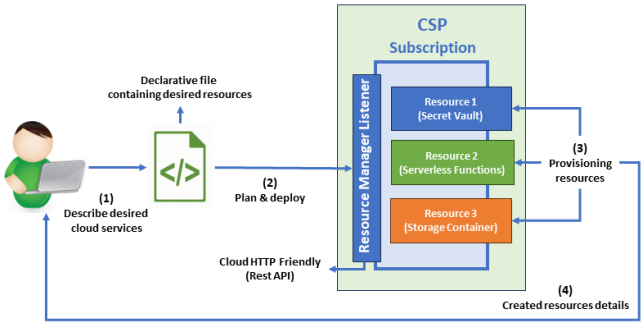

Figure 11. Automation of Model components Provisioning

Terraform takes advantage of the http-based APIs offered by major cloud service providers, ensuring seamless compatibility with their platforms. The workflow illustrated in Fig. 11 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, the terraform's results provide a complete view of provisioned infrastructure resources.

5. EXPERIMENTAL DESIGN AND RESULTS

A. Experimental Design

1) Machine Learning Classification

In this study, three distinct types of neural network algorithms—CNN, LSTM, and MLP—were employed within our model to accurately detect data sensitivity. Each algorithm was evaluated to identify the most performant and accurate option suitable for our specific context.

- Convolutional Neural Network (CNN): CNNs are designed for grid-like data. They learn spatial hierarchies of features through layers of convolutions, making them highly effective for recognizing and classifying patterns in data. In our context, CNNs help in identifying and classifying sensitive data patterns within structured datasets.
- Multi-Layer Perceptron (MLP): MLPs are feedforward neural networks with multiple layers of neurons, including input, hidden, and output layers. They are used for general-purpose classification and regression tasks due to their ability to model complex relationships. For our model, MLPs aid in the classification of data sensitivity levels, facilitating appropriate data masking techniques.
- Long Short-Term Memory (LSTM): LSTMs are a type of RNN that are well-suited for modeling temporal sequences and capturing long-term dependencies. They address the vanishing gradient problem and are ideal for tasks involving sequential data. In our model, LSTMs help in analyzing time-dependent data access patterns, enhancing the sensitivity validation process.

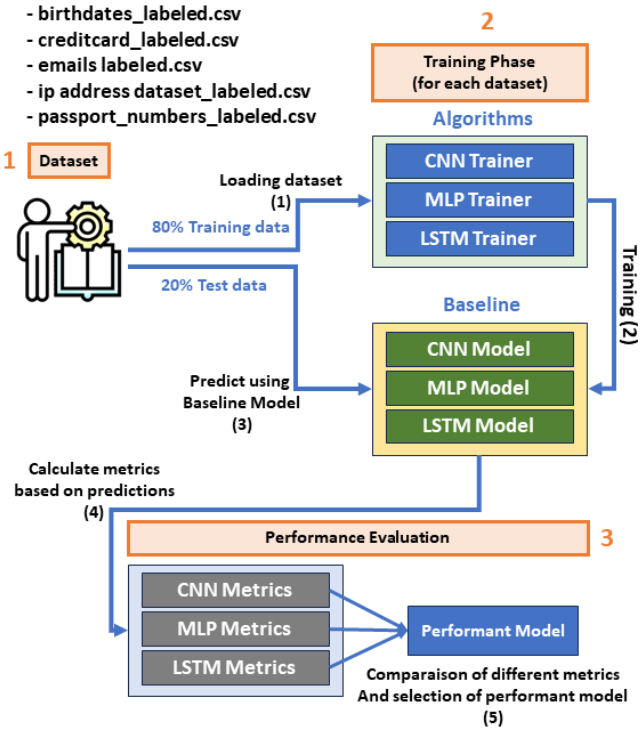

Figure 12. Machine Learning Workflow

To better understand the machine learning process, Fig. 12 illustrates the workflow applied to a set of labeled datasets representing personally identifiable information. The diagram outlines three phases: loading the labeled dataset, training the models using 80% of each provided dataset (training data), and evaluating the performance of each algorithm. This evaluation was concluded with the use of baseline models to predict labels on 20% of the dataset (test data). We opted for CNN, MLP, and LSTM algorithms to leverage their individual strengths and evaluate their performance on our dataset. CNNs are excellent for spatial feature extraction, MLPs are robust for general classification tasks, and LSTMs excel at handling sequential data. By evaluating the performance of each algorithm, we aim to identify the most suitable one for our specific context of securing sensitive data storage in the cloud. This evaluation ensures that we select the most effective algorithm for sensitivity validation and data masking, optimizing both security and accuracy. The binary classification performance was simulated using both the Python runtime environment in Microsoft Azure Functions and the 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 the categorized test data was reserved and the predicted labels were compared with the true labels.

The performance of the three compared models was evaluated based on their configuration using optimized parameters. Table VIII details the key parameters used for training and validation of each model, encompassing embedding size, dropout rate, optimizer, loss function, learning rate, epoch, batch size, and validation splits. Quantitative performance was evaluated after training the classifier with a labeled dataset. The mathematical representation of the parameters employed to estimate the performance of the model is shown in (2)-(4).

TABLE VIII. Used Parameters for Studied ML Models

Model	Key Parameters	Values
CNN	Embedding	input=2000, output=64
	Dense	units=128
	Dropout	rate=0.5
	Optimizer	Adam
	Loss	categorical_crossentropy
	Learning Rate	0.001
	Epochs	10
	Batch size	1
	Validation Split	0.2
LSTM	Embedding	input=2000, output=64
	units	units=100
	Dropout	rate=0.5
	Optimizer	Adam
	Loss	categorical_crossentropy
	Learning Rate	0.001
	Epochs	20
	Batch size	32
	Validation Split	0.2
MLP	Dense	units=128, 64
	Dropout	rate=0.5
	Optimizer	Adam
	Loss	categorical_crossentropy
	Learning Rate	0.001
	Epochs	10
	Batch size	32
	Validation Split	0.2

TABLE IX. Confusion Matrix

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

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

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

- Recall: This represents the ratio of correctly predicted positive observations to total actual positives. It measures the ability of a model to identify all the relevant instances in a dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- F1-Score: It is also referred to as the harmonic mean

and represents a balanced measure between recall and precision in the classification model.

$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Table IX presents a confusion matrix. TP (True Positive) denotes the count of correctly identified sensitive data points. TN (True Negative) represents the accurate detection of non-sensitive data. FP (False Positive) indicates the number of non-sensitive data points incorrectly identified as sensitive. FN (False Negative) signifies the count of sensitive data incorrectly identified as non-sensitive.

TABLE X. Comparison of Algorithms Performance

PII	ML	Training Time (ms)	Accuracy	Recall	F1-Score
Email	LSTM	210271.44	99.92%	99.86%	99.93%
	CNN	21741.12	100%	100%	100%
	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%	100%	100%
	CNN	19040.19	100%	100%	100%
	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%	99.63%
	MLP	9877.31	87.54%	97.06%	90.91%
Passport	LSTM	78022.04	100%	100%	100%
	CNN	15376.95	100%	100%	100%
	MLP	10120.97	99.91%	99.86%	99.93%

Table X presents the results of the classification using a varied set of metrics that 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 multifaceted performance measures served as essential benchmarks, providing a comprehensive overview of the model's classification achievements.

2) Model Overhead

To evaluate the impact of our model from user request to data persistence on the cloud environment, we analyzed time costs before and after the implementation of our model. In addition, we quantified the latency associated with the read and write operations. The cost of the Model is also evaluated. By assessing these metrics, we obtained valuable insights into the additional computational and resource costs incurred by our model, enabling us to optimize efficiency and reduce overhead.

The graph in Fig. 13 reveals the varying upload and download times in milliseconds (ms) for different data sizes,

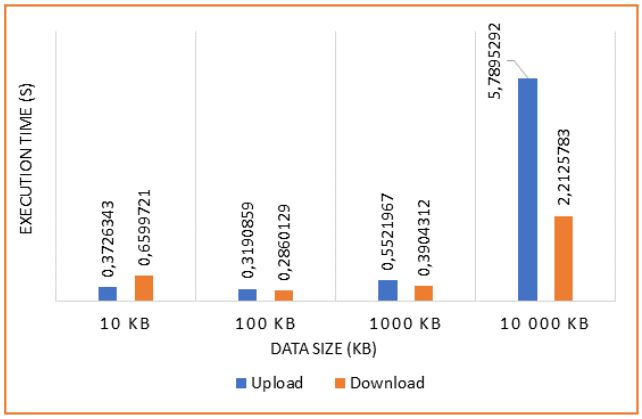

Figure 13. Data transfer (upload/download) pre- implementation

as the data size grows, the download times significantly increase. Although there are minor variations, download times remain stable overall, regardless of the data size.

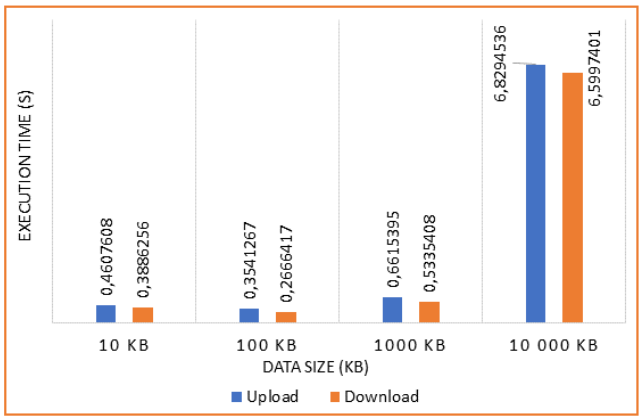

Figure 14. Data transfer (upload/download) post- implementation

Following the implementation of our system, as illustrated in Fig. 14, we evaluated its effect on data transfer performance by examining changes in upload and download times across a range of data sizes, measured in milliseconds (ms). For instance, with a data size of 10 KB, we observed that the upload time increased slightly to 0.4607608 ms, while the download time decreased to 0.3886256 ms. This indicates a modest adjustment in the system's handling of smaller data sizes, potentially due to overhead introduced by our model. In contrast, for larger data sizes, such as 10,000 KB, the system demonstrated notable improvements: the upload time was significantly reduced to 6.8294536 ms, and the download time decreased to 6.5997401 ms. These changes highlight the system's enhanced efficiency in managing larger data transfers, which is likely attributable to the optimizations embedded in our model. Overall, these results reflect a balance between upload and download performance improvements, suggesting that our model can effectively adjust to varying data sizes.

TABLE XI. Details of Model Cost

Module	Azure Service	Description	Configuration	Cost Details
Storage Container	Azure Blob Storage	Storing Secured Files	10,000 Write operations, 10,000 List and Create Container Operations 10,000 Read operations 10,000 Other operations. 1 GB Capacity	$0.03 Per 1GB $0.00036 Per 10,000 operations
Upload	Azure Functions	Executing functions in Python runtime	Consumption tier, Pay as you go, 128 MB memory, 100 milliseconds execution time, 100 executions/mo	The first 400,000 GB/s of execution and 1,000,000 executions are free.
Download	Azure Functions	Executing functions in Python runtime	Consumption tier, Pay as you go, 128 MB memory, 100 milliseconds execution time, 100 executions/mo	The first 400,000 GB/s of execution and 1,000,000 executions are free.
Data Gateway	API Management	Intercepting HTTP requests	Consumption tier 10,000 calls	$0.035 per 10,000 calls
Secrets Vault	Azure Secret Vault	Storing Sensitive Information	10,000 Operations	$0.030 Per 10,000 operations
			TOTAL PEER MONTH :	0.07 $

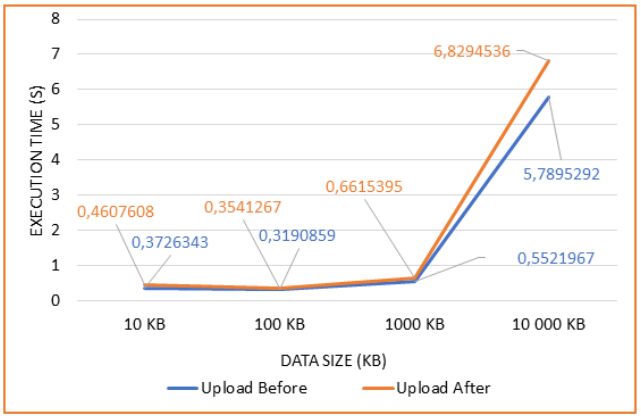

Figure 15. Upload time pre / post-implementation

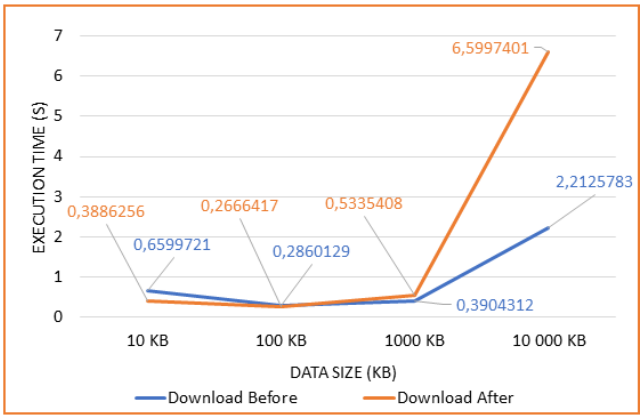

Figure 16. Download time pre / post-implementation

In Fig. 15, we compare the upload times before and after the implementation of our system. Discernible changes were observed across various data sizes.

In Fig 16, a comparison of the download times before and after the implementation of our system reveals significant changes for the different data sizes. For example, for a data size of 10 KB, the download time decreased from 0.6599721 ms to 0.3886256 ms after implementation. Similarly, for larger data, such as 10,000 KB, the download time increased significantly. These changes underline the impact of our system on the download time.

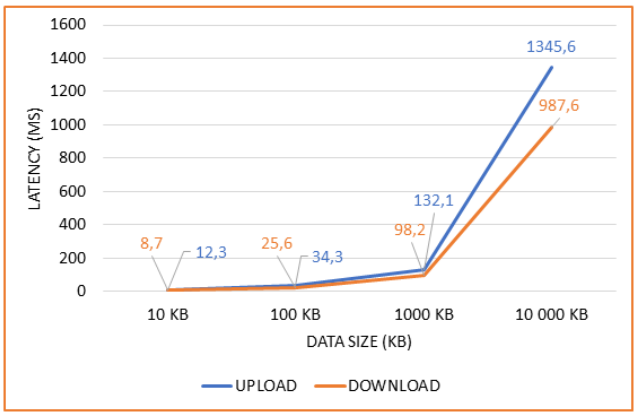

Figure 17. Data Storage Latency

In the context of our experiment, which uses Azure Storage, latency refers to the delay or response time for data read and write operations that are separate from the other modules of our model that run on Azure Functions. Latency is influenced by factors such as the distance between the user and the region in which the storage container is hosted. Fig. 17 provides the data latency measured in milliseconds (ms) across various data sizes. As the data size increases, for both upload and download requests, the latency generally exhibits an upward trend. For example, uploading 10 KB of data takes approximately 12.3 ms, whereas uploading 10 MB increases the latency to 1345.6 ms. Similarly, the downloading latency ranges from 8.7 ms for 10 KB to 987.6 ms for 10 MB.

We evaluated the cost of our model, which is estimated using the Azure Cost Calculator tool and can change based on several factors, including the chosen Azure region—France Central in our case—and the specific configuration of each service. The cost of Azure Blob Storage depends on the storage tier and the amount of data stored, whereas the pricing of Azure Functions varies based on the number of executions and resource consumption. Similarly, the cost of Azure Key Vault is influenced by the number of operations performed on stored secrets. Table XI shows the Azure services used to simulate the implementation of our approach and the monthly costs of each service.

3) Cloud Performance

Alongside the overhead analysis, we assessed the cloud performance metrics to evaluate the scalability of our model deployed in Azure cloud. Serverless computing, such as Azure Functions, utilized in our experimental design, offers inherent scalability by automatically adapting resources in response to demand. This dynamic scaling ensures that the functions can effectively manage the varying data volumes without requiring manual intervention. Cloud uptime is guaranteed through the utilization of availability zones for high availability.

Figure 18. Serverless Modules Scalability

The metrics depicted in Fig. 18 from Azure Monitor, illustrate this dynamic scalability, as the graphs exhibit real-time metrics, including the request rate, request duration, dependency call rate, dependency call duration, committed memory, and CPU usage. As demand fluctuates, the Azure Functions automatically adjust resources to maintain performance, as evidenced by the varying request rates and CPU usage. This demonstrates how serverless services efficiently scale up or down computational resources based on incoming requests,thereby ensuring optimal responsiveness and resource management. The dependency calls include crucial libraries and tools, such as Spark NLP, ML Python libraries for MLP, CNN, and LSTM, and the MD5 library, which are essential for the model's data processing and security functions.

B. Finding

Our model offers significant benefits, including high sensitivity classification accuracy, attributed to the superior performance of Convolutional Neural Networks (CNNs), which achieve perfect scores for various PII types with shorter training times. Our model ensures low latency for small data sizes and the efficient handling of large data transfers, making it suitable for diverse implementation scenarios. Cost-effectiveness was further validated with a low estimated monthly cost. In addition, its inherent dynamic scalability over the cloud maintains optimal performance by automatically adjusting computational resources based on real-time demand, ensuring consistent service quality, and efficient resource utilization. Our model effectively addresses crucial security and performance factors, as demonstrated by Table II and the examination of the results. It showcases strong capabilities in maintaining confidentiality, integrity, and scalability. Furthermore, the incorporation of machine learning enhances its overall effectiveness. The compatibility with cloud deployment provides additional versatility to the model.

6. CONCLUSION AND FUTURE WORK

In a cloud environment, resources are shared among multiple tenants, making them susceptible to threats from internal and external sources. Our research proposed a comprehensive framework that integrates machine learning and data-masking techniques to enhance the security of personally sensitive data storage in the cloud. Our approach involves integrating binary classification and data masking using cloud services, which may have broad applications in various industries, such as healthcare, finance, and government. This is particularly relevant in sectors in which strict data privacy regulations mandate robust security measures. For instance, in healthcare, our methodology can de-identify patient records while adhering to regulations such as HIPAA, facilitating secure data sharing for medical research, and preserving patient confidentiality. In the finance sector, our approach enhances fraud detection by identifying irregular transactions based on personally identifiable information patterns. In government agencies handling citizen data, our framework can ensure the security of personal sensitive data stored in the cloud, while enabling efficient data processing and sharing for public services. 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 the use of data profiling and data mining algorithms to optimize the process of data sensitivity validation. In addition, we aims to investigate the potential benefits of integrating blockchain technology into

cloud data storage security to enhance both the transparency and accountability of all data transactions.

Acknowledgements

This study was conducted 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. We are extremely grateful to those who contributed indirectly 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," *IEEE Access*, vol. 10, pp. 71 247–71 277, 2022.

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

[3] A. Kaur, V. P. Singh, and S. S. Gill, "The future of cloud computing: Opportunities, challenges and research trends," in *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 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," in *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," in *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," in *2021 Big Data, Knowledge and Control Systems Engineering (BdKCSE)*, Sofia, Bulgaria, 2021, pp. 1–5.

[7] C. Choudhary, N. Vyas, and U. K. Lilhore, "Cloud security: Challenges and strategies for ensuring data protection," in *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," in *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," in *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," in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, Melbourne, VIC, Australia, 2011, pp. 334–335.

[11] J. Hai, "Network cloud storage service architecture analysis and research," in *2016 Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Macau, China, 2016, pp. 413–416.

[12] S. Singhal, R. Srivastava, R. Shyam, and D. Mangal, "Supervised machine learning for cloud security," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, 2023, pp. 1–5.

[13] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, "Feasibility of supervised machine learning for cloud security," in *2016 International Conference on Information Science and Security (ICISS)*, Pattaya, Thailand, 2016, pp. 1–5.

[14] R. Kour, S. Koul, and M. Kour, "A classification based approach for data confidentiality in cloud environment," in *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, Jammu, India, 2017, pp. 13–18.

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

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

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

[18] A. N. Khan, M. Y. Fan, A. Malik, and R. A. Memon, "Learning from privacy preserved encrypted data on cloud through supervised and unsupervised machine learning," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2019, pp. 1–5.

[19] B. Jayaram, T. Sethukarasi, M. Sindhu, and H. Jeyamohan, "A summary on privacy and security in cloud data using various approaches," in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2023, pp. 1746–1752.

[20] F. Martinelli, F. Marulli, F. Mercaldo, S. Marrone, and A. Santone, "Enhanced privacy and data protection using natural language processing and artificial intelligence," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020, pp. 1–8.

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

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

[23] Z. Aslanyan and M. S. Boesgaard, "Privacy analysis of format-preserving data-masking techniques," in *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*, Copenhagen, Denmark, 2019, pp. 1–6.

[24] A. S. Al-Ahmad and H. Kahtan, "Cloud computing review: Features and issues," in *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, Shah Alam, Malaysia, 2018, pp. 1–5.

[25] H. Yang, Y. Li, and Y. Zhang, "Machine learning-driven data protection strategies for cloud storage," *Sensors*, vol. 22, no. 8, p. 2875, 2022.

[26] P. Patel, D. Gupta, and A. Sharma, "A survey on machine learning techniques for cloud data security," *Journal of Information Security and Applications*, vol. 64, p. 103075, 2022.

[27] H. Chen, Y. Li, and W. Wang, "A comprehensive review of data security and privacy in cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3307–3319, 2022.

[28] J. Sun, X. Liu, and Y. Zhang, "Privacy-preserving machine learning for cloud-based data security: A review," *IEEE Access*, vol. 10, pp. 149 798–149 814, 2022.

[29] H. Zhou, X. Liu, and W. Zhang, "Efficient and secure data storage using machine learning in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 1356–1369, 2022.

[30] J. Wu, M. Yu, and Z. Zhang, "Secure data storage and privacy-preserving for cloud computing using machine learning," *Security and Privacy*, vol. 5, no. 4, p. e189, 2022.

[31] R. Khan, S. U. Islam, and W. Iqbal, "A machine learning approach to secure personal data in cloud storage," *Applied Sciences*, vol. 13, no. 5, p. 2567, 2023.

[32] Y. Li, H. Yang, and Y. Li, "Advanced data security techniques in cloud storage based on machine learning," *Sensors*, vol. 23, no. 2, p. 987, 2023.

[33] R. Patil, A. Desai, and S. Kumar, "Machine learning techniques for ensuring data privacy in cloud storage," *Journal of Cloud Computing*, vol. 12, no. 3, pp. 45–60, 2023.

[34] S. H. Kim, J. H. Park, and M. S. Kang, "Data security and privacy management in cloud storage using machine learning," *IEEE Access*, vol. 12, pp. 11 234–11 250, 2024.

[35] C. A. B. D. Carvalho, M. F. D. Castro, and R. M. D. C. Andrade, "Secure cloud storage service for detection of security violations," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, Spain, 2017, pp. 715–718.

[36] 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," *IEEE Access*, vol. 12, pp. 26 334–26 343, 2024.

[37] O. A. Khashan, "Secure outsourcing and sharing of cloud data using a user-side encrypted file system," *IEEE Access*, vol. 8, pp. 210 855–210 867, 2020.

[38] S. Wang, X. Wang, and Y. Zhang, "A secure cloud storage framework with access control based on blockchain," *IEEE Access*, vol. 7, pp. 112 713–112 725, 2019.

[39] 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," *Proceedings of the IEEE*, 2022.

[40] 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.

[41] 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.

[42] Z. Li and J. Wang, "Security storage of sensitive information in cloud computing data center," *International Journal of Performability Engineering*, 2019.

[43] 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," *IEEE Access*, vol. 8, pp. 68 449–68 459, 2020.

[44] 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.