International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

MFAAMDTL: An Efficient Multimodal Feature Analysis Model to Mitigation Cloud Attacks using Transfer Learning Operations

Anagha Raich[1], Vijay Gadicha*[2]

Submitted: 22/12/2023 **Revised**: 28/01/2024 **Accepted**: 08/02/2024

Abstract: The persistent issue is that cloud applications fortified post-deployment with security patches remain susceptible to sophisticated attack vectors. In response to this, the discourse introduces an innovative, lightweight header layer designed to preemptively filter incoming requests prior to their processing by Cloud Virtual Machines (CVMs). Leveraging a combination of instantaneous and temporal analytics, this layer is adept at the early detection and neutralization of a broad spectrum of both active and passive cybersecurity threats, significantly bolstering the resilience of cloud deployments against malicious endeavors. To operationalize this defense mechanism, the system deploys an advanced logging framework capable of high-velocity data capture, triggered by an array of header-level events such as authentication attempts, access requests, and the temporal intervals between successive requests. This granular data collection strategy equips the system with a comprehensive dataset, derived from continuous user interactions, which is subsequently subjected to an intricate post-processing regimen aimed at the extraction of multimodal features. This process involves the manual tagging of request-response pairs by a curated group of users, facilitating the identification of diverse threat signatures such as temporal attack probabilities, IP-based attack typologies, user access patterns, and anomalies in request-response dynamics. At the heart of this model lies a sophisticated deep transfer learning framework, integrating the nuanced capabilities of Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU)-based Recurrent Neural Networks (RNNs), trained on an extensive corpus of user-generated data. This hybrid RNN methodology enables the model to discern and classify a wide array of attack vectors with remarkable accuracy. An incremental learning module further refines the model's efficacy, enabling dynamic adaptation and continuous improvement in its predictive accuracy, precision, and recall metrics across various attack scenarios, including but not limited to Distributed Denial of Service (DDoS), brute force, cross-site scripting, SQL injection, as well as more passive threats like access control breaches and restricted ownership transfer attempts. Empirical evaluations of this model underscore its superior performance, achieving notable accuracy rates in detecting authentication attacks (99.3%), unauthorized access attempts (97.1%), DDoS and similar request-pattern aberrations (99.1%), and Man in the Middle (MITM) attacks (99.2%). When benchmarked against contemporary models, this innovative approach demonstrated a performance uplift of 6.5%, underscoring its viability for real-time deployment and scalability across diverse cloud networking scenarios.

Keywords: *Cloud, pattern, feature, classification, attack, detection, accuracy, online, offline, machine, LSTM, GRU, RNN, temporal, analysis*

1. Introduction

The design and fortification of secure cloud networks constitute a complex, multi-faceted endeavor that mandates a holistic integration of various computational disciplines and methodologies. At the heart of this endeavor lies the orchestration of advanced data management and analytical processes, encompassing efficient data collection, meticulous pre-processing, sophisticated feature extraction and selection, precise attack classification, and comprehensive post-processing techniques. The seamless integration of these elements empowers cloud architects to construct and maintain cloud infrastructures that are not only fortified against a broad spectrum of cyber threats but are also optimized for peak performance.

[1] *G.H.Raisoni University, INDIA*
ORCID ID : 0000-0002-0067-9821
[2] *G.H.Raisoni University, INDIA*
ORCID ID : 0000-0002-7497-2289
** Corresponding Author Email: anagha.raich21@gmail..com*

To actualize this vision, cloud designers are tasked with the development of robust data collection frameworks that are embedded at every juncture within the cloud infrastructure. These frameworks are instrumental in facilitating the generation of high-density logs and detailed documentation, capturing a granular snapshot of cloud interactions and activities. This wealth of data serves as the foundational layer upon which security and performance optimization strategies are built and refined.

Once the data is amassed, it undergoes a rigorous pre-processing phase. This phase is critical for enhancing the quality and relevance of the data by eliminating anomalies and extraneous information that could potentially skew the analysis. A quintessential aspect of this phase involves the implementation of rule-based models designed for outlier detection and removal. For instance, an exemplary rule might be delineated for the exclusion of logs pertaining to requests originating from local machine IP addresses. Such rules are meticulously crafted and applied to ensure the

integrity and utility of the dataset, paving the way for more accurate and meaningful insights.

Equation 1, as referenced, symbolizes a specific instance of these rule-based models, offering a mathematical representation of the criteria employed to identify and eliminate irrelevant or misleading log entries. This step is paramount in streamlining the dataset, ensuring that subsequent stages of feature extraction, selection, and attack classification are based on data that accurately reflects genuine cloud interactions, devoid of noise and irrelevant artifacts,

$$R_{local} = Logs_{All} \bigcap Logs_{IP} \dots (1)$$

Where, R_{local} represents rules to remove local machine logs, while $Logs_{All}$ & $Logs_{IP}$ represents all logs & logs from a particular IP respectively. Within the sophisticated landscape of cloud security, the extraction and analysis of logs from cloud deployments embody a critical step in identifying and mitigating potential threats. Leveraging advanced feature extraction models, these processes delve into the nuanced, context-sensitive temporal and instantaneous aspects of cloud interactions. Such models adeptly discern between benign and nefarious requests by evaluating a constellation of distinctive features. These features might include, but are not limited to, the average duration of access sessions, the volume of requests within specific temporal windows, and the presence or absence of particular character sequences within the requests themselves. This meticulous analysis illuminates the intricate behavioral patterns that differentiate legitimate user activities from potentially malicious endeavors.

Following the extraction phase, a sophisticated feature selection layer is employed to refine the dataset, emphasizing the elimination of superfluous data through a methodical examination of class-level statistical attributes. An exemplar of this approach is the utilization of variance-based feature selection mechanisms, as illustrated in equation 2. This technique employs the average variance of identified features as a benchmark, systematically excluding those with variance levels falling below this threshold. The objective here is twofold: to streamline the dataset by excising redundant or non-informative features and to enhance the efficacy of the subsequent classification processes.

The strategic culling of data not only simplifies the dataset but also significantly amplifies the accuracy of the classifier block. By focusing on features with greater variance and, by extension, higher discriminative power, the system is better positioned to distinguish between malicious and normal requests with heightened precision. This nuanced approach to data analysis and feature selection underpins the effectiveness of cloud security measures, ensuring that protective mechanisms are both robust and targeted. Through this refined lens, cloud deployments can achieve an optimal balance between accessibility and security, safeguarding against the myriad threats that pervade the digital domains,

$$Var = \frac{\sum_{i=1}^{N_f} f_i - \sum_{j=1}^{N_f} \frac{f_j}{N_f}}{N_f} \dots (2)$$

Researchers have proposed a diverse array of classification models to effectively distinguish between normal and malicious requests. The following section of this text provides a survey of these models [3, 4], exploring their intricacies, advantages, drawbacks, and prospects for future research. This examination reveals that deep learning methods, such as Recurrent Neural Networks (RNNs) and their variations, contribute to high-performance classification. However, these models are primarily utilized within existing cloud infrastructures, limiting their adaptability. Furthermore, they tend to be intricate and necessitate frequent reconfiguration when interfacing with established cloud deployments.

To confront these challenges, Section 3 introduces a novel approach called Multimodal Feature Analysis for mitigating both internal and external attacks in cloud environments, leveraging the power of Deep Transfer Learning. This method, referred to as MFAAMDTL, capitalizes on a hybrid integration of LSTM and GRU architectures to construct an RNN-based classifier, proficiently discerning normal from malicious requests.

Subsequently, in Section 4, an empirical assessment of the model's capabilities unfolds through the introduction of multiple attack scenarios into a simulated cloud deployment. The evaluation process hinges on a comprehensive array of metrics, encompassing attack detection accuracy, precision, recall, and the area under the curve (AUC). The model's performance is meticulously scrutinized, drawing comparisons with various state-of-the-art counterparts. These evaluations affirm its potential for scalability and deployment in real-world scenarios.

2. Literature Review

Numerous security models in the realm of cloud computing have been proposed by researchers, all geared towards mitigating the risk of attacks in real-time cloud deployments. For instance, research efforts [5, 6] have introduced concepts like N version programming (NVP) and dispersed convergent encryption (DCE) as countermeasures against data-level attacks. These approaches revolve around the modification of data communication packets to enhance attack detection, reducing unauthorized access or tampering by malicious actors. However, these models are often intricate and resource-intensive.

In contrast, a more streamlined approach is presented in

[7], which advocates for Memory introspection-based Malware Detection. This method demonstrates efficiency and lower complexity in identifying Virtual Machine (VM) malwares. Extensions to this model are further elaborated in [8, 9], introducing enhanced authentication for Remote Data access and learning-driven detection mitigation (LEDEM) techniques. These strategies prioritize simplicity and adaptability when deployed in real-time cloud environments. Similar avenues are explored in [10, 11], discussing the adoption of Bayesian Q-Learning Game and threshold voting-based N version programming (NVP) to counter various attack types. These approaches reduce the likelihood of attacks by predicting diverse user inputs, thus bolstering cloud security. However, they have limitations when it comes to adversarial attacks, limiting their practicality in real-time scenarios.

To address the challenge of adversarial attacks, researchers have proposed methods like Federated Models for Defense against adversarial attacks (FDA3) [12], Group data sharing with anonymous and traceable cryptanalysis [13], queueing theory for mitigating low-rate DoS attacks [14], and a Dynamic pricing-based Resilient Model [15]. These approaches excel at extracting high-density features from user requests for classification into one or multiple attack types. Extensions to these approaches are discussed in [16, 17, 18], covering topics like Optimal Load Distribution, scale inside out for DDoS detection, and block Design-based key agreement for Group Data sharing to streamline data communication within the network, reducing overhead. However, these models require prior knowledge about requesting nodes, limiting their applicability in ad-hoc environments.

To overcome this limitation, researchers propose the adoption of Software Defined Network (SDN), ad-hoc Machine Learning Models, and an enhanced history-based IP filtering scheme in [19, 20, 21]. These solutions conduct temporal data assessments and make decisions based on these evaluations. Similar models are presented in [22, 23, 24], where researchers employ enhanced cryptographic techniques, untrusted on-chip security deployments, and dynamically traceable Ciphertext-policy attribute-based encryption (DT CP ABE) to enhance privacy and security at the application layers. These models simplify computational requirements by replacing complex computations with more straightforward alternatives, thus enhancing both speed and security performance.

Building upon these models, researchers have introduced specialized solutions such as Secure Software-Defined Networks (SecSDN) [25], Mixture Localization-based Outliers (MLO) [26], division and replication of data cloud for optimal performance and security (DROPS) [27], a Naïve Bayes-based Attack Resilient cloud-assisted IoT model [28], and the implementation of strict transport security [29, 30]. While these models enhance attack detection capabilities for specific application-specific deployments, they may not be suitable for general-purpose application scenarios.

To address these limitations and design a faster and more accurate attack detection model, the subsequent section introduces the development of a Multimodal Feature Analysis method for internal and external Attack Mitigation in cloud environments through Deep Transfer Learning. This model has undergone rigorous evaluation, assessing accuracy and delay across various attack detection scenarios. These parameters are then compared against different state-of-the-art methods, providing a comprehensive performance assessment across diverse deployment scenarios.

3. Proposed Multimodal Feature Analysis Method for Internal & External Attack Mitigation for Clouds via Deep Transfer Learning

The literature review reveals a plethora of cloud security models proposed by researchers, with most of them tailored to specific deployment contexts. While some of these models find utility in general-purpose public cloud deployments, they tend to offer only moderate levels of security and quality of service (QoS) performance. Furthermore, the integration of these methods into existing cloud deployments brings about a host of security and privacy concerns inherent to these models. Consequently, cloud applications that incorporate post-deployment security patches may not achieve complete security, leaving room for attackers to launch aggregated and distributed attacks.

To address these shortcomings, this section introduces a novel Multimodal Feature Analysis approach for mitigating both internal and external attacks through a Deep Transfer Learning process. The model's overall workflow is illustrated in Figure 1, where a lightweight header-level checker is employed to assess request patterns. These patterns undergo both online and offline processing phases, facilitating comprehensive attack detection. Notably, the model operates at the request level, making it independent of the underlying cloud deployment infrastructure. This characteristic renders the model deployable across a wide array of cloud applications, requiring minimal reconfiguration efforts for activation.

The model incorporates an online checking layer, responsible for evaluating request periodicity, user access, authentication, and request parameter values. These inputs are amalgamated to generate a request status, aiding the cloud system in determining the authenticity or malicious nature of incoming requests. The outcomes of this

decision-making process are stored in a database, contributing to the training of the offline layer.

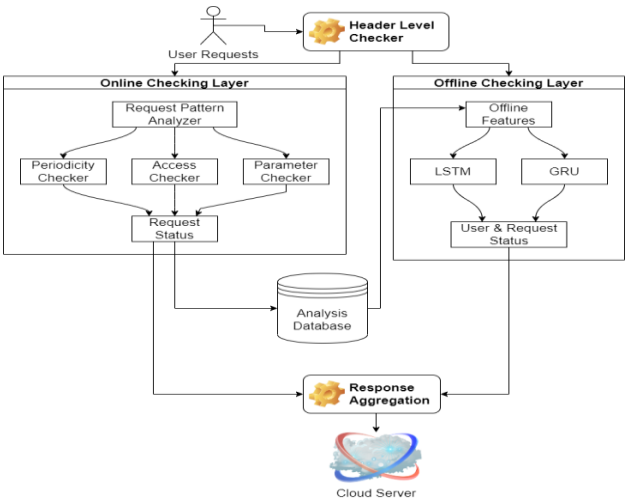

Fig 1. Overall flow of the proposed model

The offline layer plays a crucial role in the evaluation of various offline features, encompassing temporal attack probability, attack types per IP, average page access time per IP, total attacks per IP, page access patterns, request patterns, and temporal response patterns. These patterns undergo processing using a hybrid LSTM and GRU model, which aids in feature augmentation to train a highly efficient Recurrent Neural Network (RNN) model. The outcomes from both the online and offline phases are merged through an aggregation layer, facilitating the identification of authentic and malicious requests. Authenticated requests proceed to the cloud server, while malicious ones are reintegrated into the database for incremental learning purposes. The design of the proposed model is segmented into multiple sub-parts, each of which is elaborated upon in different sub-sections of this text. This segmentation facilitates researchers in deploying these models either partially or in their entirety, depending on their specific application requirements.

3.1 Design of the Header-Level Checker

All incoming requests are processed through a header-level checker, which employs lightweight parallel logging and rule-based request evaluations. This approach ensures swift decision-making within the model, minimizing any adverse impact on the cloud's response time. The design of this engine can be described through the following steps:

Step 1: Users are required to log in using standard key-value pairing mechanisms.

- All inputs undergo the following checks, aiding in the detection of SQL Injection and Cross Site Scripting (XSS) attacks:
- If the input contains single quotes, double dashes (--), or hypertext tags (<tag>), these characters are replaced with blanks.
- Scripting constants such as mocha, view-source, vbs, livescript, wscript, Javascript, jar, applescript, jscript, and vbscript found in the input are replaced with blank characters.
- In case any of the conditions are met, the following parameters are used to generate an event log: Attack type (SQLi or XSS), a timestamp instance, and the IP address requesting the instance.
- Requests failing the login process are logged with the following parameters: Attack Type (Inadequate Verification), IP address requesting, the current timestamp instance, and the variables passed during authentication.
- The server retains the identity of the current user as an immutable session variable for subsequent requests, controlling access to different cloud areas and verifying authentication.
- The utilization of this session variable enables the header-level check layer.
- All input variables passed by the user, including POST, GET, DELETE, and PUT request types, are scanned for further evaluation.
- The following information is logged whenever a user attempts to view a page for which access is denied ,Attack Type (o): Incorrect Access , IP address request; Current Timestamp instance; Authentication variables; oURL of the page being viewed, and After checking the logs, a temporal validity score is evaluated for each user IP via equation 3,

$$T_{score_{IP}} = \left[\frac{\sum_{T=t_1}^{t_2} R_{invalid_{IP}}}{\sum_{T=t_1}^{t_2} R_{valid_{IP}} + R_{invalid_{IP}}} \right] \dots (3)$$

Where, $T_{valid_{IP}}$ represents the temporal validity score between time occurrences t1 and t2, and $R_{valid_{T_{IP}}}$ and $R_{invalid_{T_{IP}}}$ denote the total number of valid and invalid requests found between the specified time instances. More than 10% of all requests are invalid if $T_{score_{IP}}$>0.1. When this occurs, the user logs out and reports the activity through the logging layer with the following information:

o Attack Type: Inappropriate Temporal Access o IP addres s request o Current Timestamp instance o Authentication v ariables passed o URL of the page being accessed

Based on these operations, a large number of logs are stored on the cloud, and are processed via both offline & offline phases. These phases assist in detection & mitigation of multiple application-level attacks before they are processed by the processing cloud.

3.2 Design of Online Checking Layer

The logs stored on cloud are initially processed via an online checking layer, which assists in making instantaneous decisions based on real-time requests. This layer stores linked log data on the cloud using the process indicated in figure 2, wherein the following information is stored at user & client level,

- User identifier & their IP address
- Types of attacks detected by the logging layer for this IP & user
- Timestamp of these attacks
- URI on which these attacks are detected
- Request variables passed by user or IP
- Values of these variables passed by user or IP
- Authorization & session information
- Meta data about the requests

Based on these logs, both user-level & IP-level metrics are evaluated. These metrics include, access control score (AC_{score}), authentication score (AU_{score}), page-level access control score (AC_{page}), request periodicity ($R_{periodicity}$), & pattern attack score (P_{score}), and are evaluated via equations 4, 5, 6, 7 & 8 as follows,

$$AC_{score}(IP) = \frac{\sum_{i=1}^{N_{req}} B(IP)_i}{N_{req}} \dots (4.1)$$

$$AC_{score}(User) = \frac{\sum_{i=1}^{N_{req}} B(User)_i}{N_{req}} \dots (4.2)$$

Where, $B(IP)$ & $B(User)$ represents number of blocked requests for the given IP & the given user respectively, & N_{req} represents number of requests sent by the IP or user for which this metric is being evaluated.

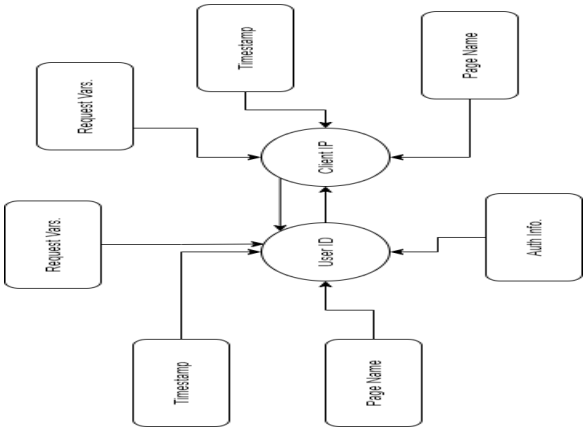

Fig 2. Generated graph on a per-user and per-IP basis

Similarly, authentication score is evaluated via equation 5 as follows,

$$AU_{score}(IP) = \frac{\sum_{i=1}^{N_{req_{auth}}} NonAuth(IP)_i}{N_{req_{auth}}} \dots (5)$$

Where, $NonAuth(IP)$ represents number of authorization requests which are blocked requests for the given IP, & $N_{req_{auth}}$ represents number of authorization requests sent by the IP which this metric is being calculated. The page level access control score is evaluated on a per-page & per user/IP basis via equation 6 as follows,

$$AC_{page}(IP) = \frac{\sum_{i=1}^{N_{req_{page}}} B(IP)_i}{N_{req_{page}}} \dots (6.1)$$

$$AC_{page}(User) = \frac{\sum_{i=1}^{N_{req_{page}}} B(User)_i}{N_{req_{page}}} \dots (6.2)$$

Where, $N_{req_{page}}$ represents number of requests sent by the IP or user for which this metric is being evaluated on this page, and is updated continuously. Similarly, request periodicity is evaluated via equation 7 as follows,

$$R_{periodicity}(IP \mid User) = \frac{\left[\sum_{i=1}^{N_{req}-1} t_{i+1} - t_i\right]}{N_{req} - 1} \dots (7)$$

Where, t_i represents time instant at which this request is sent by the user or IP for which periodicity is being evaluated. Similarly, pattern attack score is evaluated via equation 8 as follows,

$$P_{score}(User \mid IP) = \frac{\sum_{i=1}^{N_{req}} B_{pattern_i}}{N_{req}} \dots (8)$$

Where, $B_{pattern}$ represents number of blocked pattern requests were removed from by the model as SQL Injection & XSS attack types. If any of these score values is above a particular threshold, then request from user or IP address are blocked, & any further access instances are reported to the cloud designers. The evaluated metrics are used during the offline phase, which is discussed in the next section of this text.

3.3. Design of Offline Checking Layer with hybrid combination of LSTM & GRU models

Numerous metrics are assessed at both the user level and IP level during the online phase, with their scope expanded through the estimation of Multimodal parameters. These parameters encompass various aspects, including:

- Temporal attack probability (P_{TA})
- Type of attacks originating from the IP (T_{att})
- Average page access time per IP (D_{access})
- Probability of total attacks performed by the IP (Att_{IP})
- Page access patterns ($Acc_{pattern}$)
- Request patterns ($Req_{pattern}$)
- Temporal response patterns ($Resp_{pattern}$)

These parameters are evaluated via equations 9, 10, 11, 12, 13, 14, and 15 as described as follows,

$$P_{TA}(r) = \frac{AC_{score}(r) + AU_{score}(r) + AC_{page}(r) + P_{score}(r)}{4} \dots (9)$$

In this context, "r" signifies the request originator type, which may be either IP or User, the evaluation of the type of attacks originating from the IP is conducted via equation 10, outlined as follows,

$$T_{att}(r) = Max\left(\bigcup_{i=1}^{N_{attacks}} P_{TA}(r)\right) \dots (10)$$

Where, $N_{attacks}$ represents number of attacks injected by the user or the IP address. The average page access time is evaluated via equation 11,

$$D_{access} = \frac{\left[\sum_{i=1}^{N_{req}} t_{resp_i} - t\_req_i\right]}{N_{req}} \dots (11)$$

Where, t_{resp} & t_{req} represents response timestamp & request timestamp for the given user or IP address. This assists in estimating the delay with which requests are being responded for the given requesting device. Similarly, probability of total attacks performed by the IP or User are evaluated via equation 12 as follows,

$$P(Att_p) = 4 * P_{TA} * \log\left(\frac{1}{P_{TA}}\right) \dots (12)$$

Page access patterns are evaluated via equation 13, wherein time instances for accessing particular pages are estimated for each IP and user as follows,

$$Acc_{pattern} = \sum_{i=1}^{N_{pages}} \frac{t_{resp_i} - t_{req_i}}{N_{pages}} \dots (13)$$

Where, t_{resp} & t_{req} represents response timestamp & request timestamp for page i, while N_{pages} represents number of pages or access areas on the cloud deployment. Similarly, request & response patterns are evaluated via equation 14 & 15 as follows,

$$Req_{pattern} = \frac{\sum_{i=2}^{N_{req}} t_i - t_{i-1}}{N_{req}} \dots (14)$$

$$Resp_{pattern} = \frac{\sum_{i=2}^{N_{resp}} t_i - t_{i-1}}{N_{resp}} \dots (15)$$

In the context of this analysis, Nreq and Nresp are variables denoting the quantities of requests and responses attributed to specific IP addresses and users. These particular characteristics, pertaining to the request-response interactions, are systematically compiled and subsequently furnished as input to a classification model, which adopts a hybrid architecture comprising GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory) components. The structural intricacies of this hybrid GRU-LSTM based classification model can be observed in Figure 3. Within this framework, the feature sets initially undergo a processing stage via the GRU component, followed by further refinement through the utilization of LSTM techniques. This entails the application of a GRU layer to assess the more general attributes of the input features, as delineated by equations 16 through 21. During this phase, specific constants associated with the GRU layer play a pivotal role in enhancing the features under consideration. Subsequently, the augmented features are subjected to the LSTM layer, contributing to the overall classification process, which aims to categorize requests into two distinct groups: Normal and Malicious.

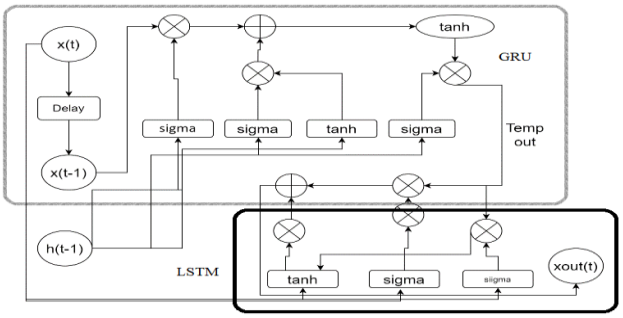

Fig 3. Combination of GRU & LSTM for better feature augmentation

The intensity of one feature set directly depends on the intensity of features extracted via earlier layers when these features are evaluated in a cascade manner. Equation 16 is first used to evaluate an intermediate feature vector I using input features xin,

$$i = var(x_{in} * U^i + h_{t-1} * W^i) \dots (16)$$

Similarly, frequent features & omni features are evaluated via equation 17 & 18 as follows,

$$f = var\left(x_{in} * U^f + h_{t-1} * W^f\right) \dots (17)$$

$$o = var(x_{in} * U^o + h_{t-1} * W^o) \dots (18)$$

These feature sets are combined, and an initial convolutional & temporal feature set is evaluated via equation 19 & 20 respectively,

$$C_t' = \tanh(x_{in} * U^g + h_{t-1} * W^g) \dots (19)$$

$$T_{out} = var(f_t * x_{in}(t-1) + i * C_t') \dots (20)$$

Using these feature sets, the final output of GRU layer is estimated via equation 21 as follows,

$$h_{out} = \tanh(T_{out}) * o \dots (21)$$

Where, suffixes of U & W represents GRU constants, and are evaluated via the RNN model by continuously tuning the internal layers. The extracted features are processed via

a LSTM layer, which results in augmented feature sets. These feature sets are evaluated via equations 22 through 25 as follows,

$$z = \text{var}(W_z * [h_{out} * T_{out}]) \dots (22)$$

$$r = \text{var}(W_r * [h_{out} * C'_t]) \dots (23)$$

Based on these intermediate feature set, the output filter metric h_t and output feature metric x_{out} are evaluated via equations 24 & 25 as follows,

$$h'_t = \tanh(W * [r * h_{out} * T_{out}]) \dots (24)$$

$$x_{out} = (1 - z) * h'_t + z * h_{out} \dots (25)$$

In this context, the variable W represents a constant associated with the LSTM component, and its determination is a critical step within the RNN model to optimize accuracy metrics. The subsequent stages of feature processing involve the utilization of a Recurrent Neural Network (RNN), as evident in Figure 4. Within this architectural framework, the combined features derived from the GRU and LSTM layers of one level are harnessed to compute higher-density features for the subsequent layer.

This model operates by extracting and evaluating multiple features within each layer and then transmitting these features to the subsequent layer to facilitate more robust feature augmentations. The amalgamation of these features culminates in the creation of an augmented recurrent metric, which plays a pivotal role in the final classification of user requests into either the 'normal' or 'malicious' categories.

The determination of the output class within the RNN model is carried out through the application of equation 26. This equation incorporates a pure-linear activation function, specifically tailored for the purpose of request type classification. This multifaceted approach, combining LSTM constants, recurrent feature processing, and linear activation functions, synergistically contributes to the classification task's accuracy optimization.

$$C_{out} = \text{purelin}\left(\sum_{i=1}^{N} x_{out_i} * W_i\right) \dots (26)$$

Where, W_i represents weight for given feature set, while C_{out} represents output probability for the given input features to be in either malicious or normal categories.

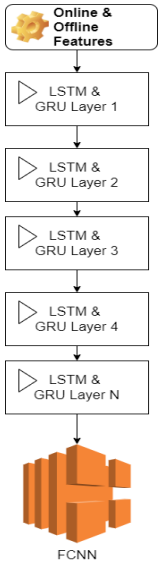

Fig 4. Design of the hybrid GRU & LSTM based RNN model

The output generated by the RNN model is synergistically integrated with the outcomes produced by the online request check layer, and together, they collectively form a comprehensive response aggregation layer. This aggregation layer plays a pivotal role in shaping the final decision-making process concerning user requests. Detailed insights into the intricacies of this response aggregation layer's design are expounded upon in the subsequent section of this text.

3.4. Design of Response Aggregation Layer

The consolidation of responses emanating from both the online and offline learning layers is achieved through a weighted sum methodology. This approach effectively amalgamates the accuracy of offline learning with the real-time adaptability of online learning, culminating in the formulation of definitive access decisions.

The evaluation of the final decision is executed by applying equation 27, where the outputs from the offline learning classification and the online learning's determination of malicious status are harmoniously integrated. This fusion of information encapsulated in equation 27 constitutes a crucial step in the decision-making process, combining the strengths of both learning layers to reach a comprehensive and informed verdicts for different input sets.

$$f_{acc} = w_{offline} * d_{offline} + w_{online} * d_{online} \dots (27)$$

Where, $w_{offline}$ & w_{online} are different weights of the given offline & online models which are evaluated from equation 28,

$$w_i = \frac{N_c}{N_t} \dots (28)$$

Within this context, Nc represents the count of requests that have been accurately classified, while Nt stands for the

total number of requests that have undergone processing within this phase. It's worth noting that requests that have been rejected by the online phase are excluded from processing by the offline phase. Consequently, the sum of $w_{offline}$ and w_{online} will invariably be less than 1, eliminating the necessity for quantization of these values for different use cases.

The variables d_{online} and $d_{offline}$ denote the decisions made by the online and offline phases regarding the acceptance or rejection of requests. Those requests deemed acceptable by these phases are directly routed to the cloud infrastructure for further handling, whereas the discarded requests are allocated for the purpose of retraining the RNN model. This retraining process facilitates incremental enhancements in the model's performance over temporal instance sets. The evaluation of performance is carried out by considering several key parameters, including but not limited to:

- Authorization Attack Detection Accuracy (AADA)
- Invalid Access Detection Accuracy (IADA)
- Accuracy for Detection of SQL, XSS, and DDoS Attacks (ASDX)
- Accuracy for Detection of Man-in-the-Middle Attacks (AMITM)
- Authorization Attack Detection Delay (DAA)
- Invalid Access Detection Delay (DIA)
- Delay for Detection of SQL, XSS, and DDoS Attacks (DSDX)
- Delay for Detection of Man-in-the-Middle Attacks (DMITM)

These performance metrics serve as critical benchmarks for evaluating the effectiveness and efficiency of the proposed security model. A comparative analysis of these parameters with various state-of-the-art security models is presented in the subsequent section of this document, shedding light on the model's strengths and capabilities.

4. Comparative Analysis of Results for Proposed Model in different Scenarios

The proposed model employs a dual-phase approach, incorporating both online and offline components, to enhance the effectiveness of attack detection mechanisms. To evaluate the performance of this model, a Dropbox-like application was developed, encompassing the following key functionalities:

- User registration and login via Email and Password combination.
- File upload and download capabilities.
- File sharing with read-only access for other users.
- File sharing with modify access for other users.

This model was deployed on the Apache Cloud infrastructure, and various types of cyberattacks were simulated to assess its robustness. The simulated attack scenarios included Brute Force attacks, Distributed Denial of Service (DDoS) attacks, Man-in-the-Middle (MITM) attacks, Cross-Site Scripting (XSS) attacks, SQL Injection attempts, improper authorization, and inadequate access control measures.

The creation of attack scenarios followed a systematic process:

- For SQL Injection attacks, attack packets containing sample queries (e.g., single quotes, double dashes) were generated and transmitted to the server for evaluation sets.
- XSS requests involving redirections and session hijacking were initiated, and their responses were analyzed under different circumstances.
- Packets with high-frequency patterns were sent from various IP addresses to the server, and their responses were assessed in various use cases.

To structure these attack scenarios, attack trees were employed, providing a graphical representation to illustrate potential threats to the system. Attack trees are designed on the premise that each potential attacker possesses unique goals and skill sets. These trees help in identifying possible targets, routes to success, and the potential impact of vulnerabilities. Preventative actions are then determined based on the sensitivity and impact of the threat.

In this environment, nearly 9 million requests were generated from malicious nodes, and the model's performance was assessed using several key metrics, including:

- Authorization Attack Detection Accuracy (AADA)
- Invalid Access Detection Accuracy (IADA)
- Accuracy for Detection of SQL, XSS, and DDoS Attacks (ASDX)
- Accuracy for Detection of Man-in-the-Middle Attacks (AMITM)
- Authorization Attack Detection Delay (DAA)
- Invalid Access Detection Delay (DIA)
- Delay for Detection of SQL, XSS, and DDoS Attacks (DSDX)
- Delay for Detection of Man-in-the-Middle Attacks (DMITM)

These metrics were also calculated for other security models such as LE DEM [9], FDA3 [12], and DT CP ABE [24], which operate in similar execution environments. The LEDEM model emphasizes internal attack detection mechanisms, FDA3 focuses on authentication and access control models, and DTCPABE enhances security against data attacks. Consequently, these three models were selected for comparative purposes.

Based on the analysis and Figure 5, it becomes evident that the MFAAMDTL model outperforms its counterparts with a 4.5% higher accuracy in authorization attack detection than LE DEM [9], a 4.3% higher accuracy than FDA3 [12], and a 6.8% higher accuracy than DT CP ABE [24]. This improvement in accuracy can be attributed to the hybrid approach that combines online and offline phases, reducing detection errors, especially for a substantial volume of authorization requests. Similar trends were observed in the delay of authorization checks (DIA), as detailed in Table 2, where the delay is compared across different models concerning the number of requests processed by the cloud model from various requesting users.

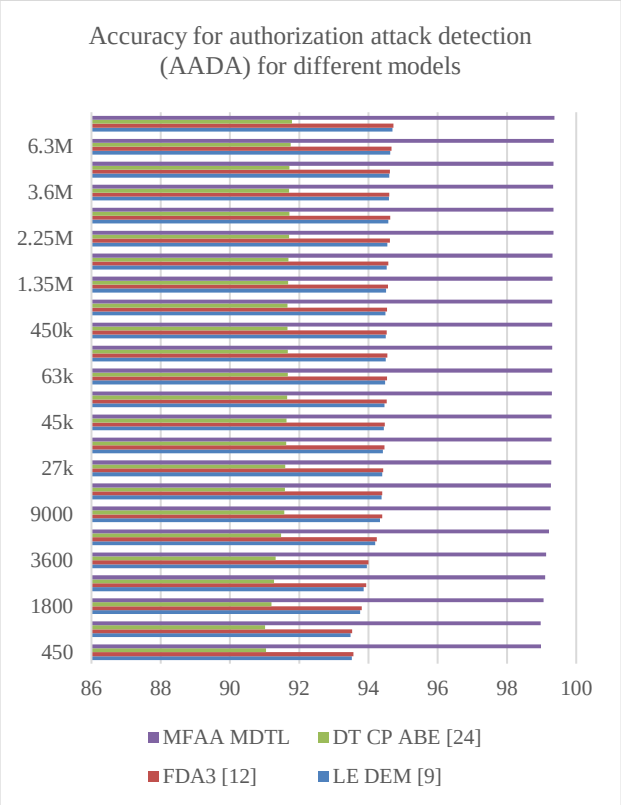

Fig 5. Accuracy for authorization attack detection (AADA) for different models

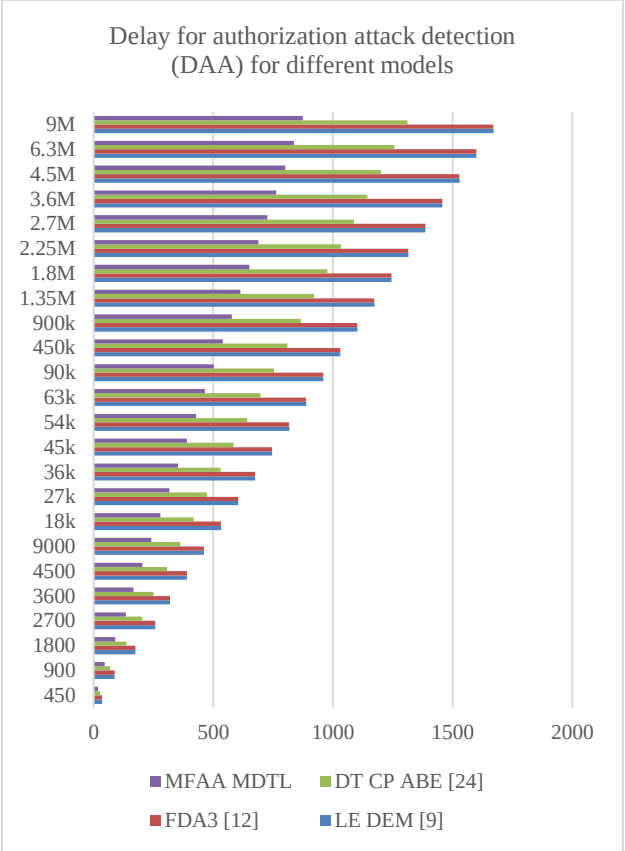

Fig 6 Delay for authorization attack detection (DAA) for different models

Based on this analysis & figure 6, it can be observed that the MFAAMDTL model has 15.6% lower delay than LE DEM [9], 15.5% lower delay than FDA3 [12], and 8.5% lower delay than DT CP ABE [24] for authorization attack detection, thereby showcasing its superior performance for real-time deployments. The reason for this reduction in delay is deployment of light-weight header level checking layer, which assists in reducing computational overheads for checking malicious requests. Similar observations were made for invalid access detection accuracy (IADA).

Table 1: Percentage Improvement of Mfaamdtl Model with Respect to. Other Security Models

Metrics	LE DEM [9]	FDA3 [12]	DTABE [24]	Remarks
AADA	4.05%	3.87 %	6.12%	Higher Efficiency
DAA	14.04%	13.95 %	7.65%	Fast
IADA	37.35%	35.64 %	19.71%	Higher Efficiency
DIA	21.51%	21.51 %	14.85%	Fast

ASDX	5.31%	4.86 %	6.12%	Higher Efficiency
DSDX	13.86%	1.71 %	4.77%	Fast
AMITM	4.86%	4.41 %	6.48%	Higher Efficiency
DMITM	7.65%	0.45 %	5.31%	Fast

These benefits make the suggested approach suitable for a wide range of cloud applications that need more security with lower overhead.

5. Conclusion and Future Scope

The MFAAMDTL model, as put forward, initiates the extraction of an extensive array of user-specific and IP-specific attributes. These attributes encompass factors such as temporal attack probabilities, attack types, average page access durations, cumulative attack counts, access and request patterns, as well as temporal response profiles. The treatment of these patterns takes place in both online and offline modes, which contributes to achieving a low-overhead attack detection capability with heightened accuracy when juxtaposed with various existing models.

Noteworthy is the observation that the accuracy of this proposed model exhibits an incremental trend in relation to the volume of requests processed. This trend is attributed to the presence of a response aggregation layer, a dynamic component that continuously updates the database contingent upon the classification of attacks. Consequently, the proposed model attains an impressive accuracy rate of 99.3% in detecting authentication attacks, 97.1% in identifying invalid access attempts, 99.1% in countering DDoS and other request-pattern attacks, and 99.2% when dealing with Man-in-the-Middle (MITM) attack scenarios. This performance superiority over various existing methods renders the proposed model highly suitable for deployment across a spectrum of real-time scenarios.

Furthermore, the model introduces the concept of a streamlined header-level request verification layer, effectively curtailing the computational overhead associated with security measures. As a consequence, the proposed model manages to reduce the time delay required for various attack detections by more than 15% when juxtaposed with preexisting models. These advantages position the proposed model as an ideal choice for deployment in high-speed and high-accuracy real-time cloud applications.

For future prospects, researchers are encouraged to subject the proposed model to rigorous evaluation under diverse cloud-based scenarios. This scrutiny may unveil potential shortcomings that warrant attention before considering large-scale deployment. Additionally, researchers can explore the substitution of the existing RNN model with alternative techniques such as CNN, Q-Learning, or deep reinforcement learning. Such experimentation will facilitate the assessment of performance variations, ultimately leading to the identification of the most optimal models tailored to specific deployment scenarios.

References

[1] A. Bhardwaj, V. Mangat and R. Vig, "Hyperband Tuned Deep Neural Network With Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud," in IEEE Access, vol. 8, pp. 181916-181929, 2020, doi: 10.1109/ACCESS.2020.3028690.

[2] H. Yang, S. Ju, Y. Xia and J. Zhang, "Predictive Cloud Control for Networked Multiagent Systems With Quantized Signals Under DoS Attacks," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 2, pp. 1345-1353, Feb. 2021, doi: 10.1109/TSMC.2019.2896087.

[3] Kushwah, G.S., Ranga, V. Detecting DDoS Attacks in Cloud Computing Using Extreme Learning Machine and Adaptive Differential Evolution. Wireless Pers Commun (2022). https://doi.org/10.1007/s11277-022-09481-9.

[4] X. Gong et al., "Defense-Resistant Backdoor Attacks Against Deep Neural Networks in Outsourced Cloud Environment," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2617-2631, Aug. 2021, doi: 10.1109/JSAC.2021.3087237.

[5] G. Xu, S. Xu, J. Ma, J. Ning and X. Huang, "An Adaptively Secure and Efficient Data Sharing System for Dynamic User Groups in Cloud," in IEEE Transactions on Information Forensics and Security, vol. 18, pp. 5171-5185, 2023, doi: 10.1109/TIFS.2023.3305870.keywords: {Security; Cloud computing; Resistance; Access control; Heuristic algorithms; Encryption; Standards; Dynamic user groups; adaptive security; fine-grained access control},

[6] J. Deng et al., "A Survey on Vehicular Cloud Network Security," in IEEE Access, vol. 11, pp. 136741-136757, 2023, doi: 10.1109/ACCESS.2023.3339192.keywords: {Security; Cloud computing; Reliability; Roads; Network security; Authentication; Surveys; Edge computing; Vehicular ad hoc networks; Cloud computing; edge computing; security; VANETs; vehicular cloud network},

[7] K. Muniasamy, R. Chadha, P. Calyam and M. Sethumadhavan, "Analyzing Component Composability of Cloud Security Configurations," in IEEE Access, vol. 11, pp. 139935-139951, 2023, doi: 10.1109/ACCESS.2023.3340690.keywords: {Security; Cognition; Databases; Cloud computing security; Symbols; Large-scale systems; Buildings; Formal concept analysis; Cloud security; composability; formal analysis; policy-based verification},

[8] J. Zhang, T. Li, Z. Ying and J. Ma, "Trust-Based Secure Multi-Cloud Collaboration Framework in Cloud-Fog-Assisted IoT," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1546-1561, 1 April-June 2023, doi: 10.1109/TCC.2022.3147226.keywords: {Security; Cloud computing; Access control; Collaboration; Internet of Things; Authentication; Clouds; Cloud computing; multi-cloud service composition; secure collaboration; single sign-on; role-based access control},

[9] A. Wu, A. Yang, W. Luo and J. Wen, "Enabling Traceable and Verifiable Multi-User Forward Secure Searchable Encryption in Hybrid Cloud," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1886-1898, 1 April-June 2023, doi: 10.1109/TCC.2022.3170362.keywords: {Cloud computing; Servers; Security; Encryption; Indexes; Cryptography; Hash functions; Forward secure searchable encryption; multi-user; verifiability; traceability; revocation},

[10] C. Wang, Z. Yuan, P. Zhou, Z. Xu, R. Li and D. O. Wu, "The Security and Privacy of Mobile-Edge Computing: An Artificial Intelligence Perspective," in IEEE Internet of Things Journal, vol. 10, no. 24, pp. 22008-22032, 15 Dec.15, 2023, doi: 10.1109/JIOT.2023.3304318. keywords: {Security;Privacy;Artificial intelligence;Internet of Things;Cloud computing;Data privacy;Computer architecture;Artificial intelligence (AI);fifth generation (5G);Internet of Things (IoT);machine learning (ML);mobile-edge computing (MEC);security and privacy;software-defined network (SDN) security;virtual machine security},

[11] A. Bagheri and A. Shameli-Sendi, "Automating the Translation of Cloud Users' High-Level Security Needs to an Optimal Placement Model in the Cloud Infrastructure," in IEEE Transactions on Services Computing, vol. 16, no. 6, pp. 4580-4590, Nov.-Dec. 2023, doi: 10.1109/TSC.2023.3327632.

[12] keywords: {Security;Cloud computing;Servers;Energy consumption; Data centers; Computational modeling; Quality of service; Automation;cloud computing; NFV; network security defence patterns;security function placement},

[13] Q. Wang, Z. Wang and W. Wang, "Research on Secure Cloud Networking Plan Based on Industry-Specific Cloud Platform," in IEEE Access, vol. 11, pp. 51848-51860, 2023, doi: 10.1109/ACCESS.2023.3279409. keywords: {Switches; Security; Cloud computing; Firewalls (computing);Servers;Safety;Maintenance engineering; Commercial security service; IDC; industry-specific cloud; network security;SDN},

[14] Y. Zhang, T. Zhu, R. Guo, S. Xu, H. Cui and J. Cao, "Multi-Keyword Searchable and Verifiable Attribute-Based Encryption Over Cloud Data," in IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 971-983, 1 Jan.-March 2023, doi: 10.1109/TCC.2021.3119407.keywords: {Cloud computing; Encryption; Cryptography; Keyword search; Servers; Data models; Security; Multi-keyword; attribute-based searchable encryption; verification; shared multi-owner mechanism},

[15] Z. Li, H. Jin, D. Zou and B. Yuan, "Exploring New Opportunities to Defeat Low-Rate DDoS Attack in Container-Based Cloud Environment," in IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 3, pp. 695-706, 1 March 2020, doi: 10.1109/TPDS.2019.2942591.

[16] H. Yuan, Y. Xia, M. Lin, H. Yang and R. Gao, "Dynamic Pricing-Based Resilient Strategy Design for Cloud Control System Under Jamming Attack," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 111-122, Jan. 2020, doi: 10.1109/TSMC.2019.2952467.

[17] O. A. Wahab, J. Bentahar, H. Otrok and A. Mourad, "Optimal Load Distribution for the Detection of VM-Based DDoS Attacks in the Cloud," in IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 114-129, 1 Jan.-Feb. 2020, doi: 10.1109/TSC.2017.2694426.

[18] G. Somani, M. S. Gaur, D. Sanghi, M. Conti and M. Rajarajan, "Scale Inside-Out: Rapid Mitigation of Cloud DDoS Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 6, pp. 959-973, 1 Nov.-Dec. 2018, doi: 10.1109/TDSC.2017.2763160.

[19] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun and Y. Xiang, "Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 6, pp. 996-1010, 1 Nov.-Dec. 2019, doi:

10.1109/TDSC.2017.2725953.

[20] S. Dong, K. Abbas and R. Jain, "A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments," in IEEE Access, vol. 7, pp. 80813-80828, 2019, doi: 10.1109/ACCESS.2019.2922196.

[21] N. Agrawal and S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3769-3795, Fourthquarter 2019, doi: 10.1109/COMST.2019.2934468.

[22] T. V. Phan and M. Park, "Efficient Distributed Denial-of-Service Attack Defense in SDN-Based Cloud," in IEEE Access, vol. 7, pp. 18701-18714, 2019, doi: 10.1109/ACCESS.2019.2896783.

[23] I. Khan, Z. Anwar, B. Bordbar, E. Ritter and H. Rehman, "A Protocol for Preventing Insider Attacks in Untrusted Infrastructure-as-a-Service Clouds," in IEEE Transactions on Cloud Computing, vol. 6, no. 4, pp. 942-954, 1 Oct.-Dec. 2018, doi: 10.1109/TCC.2016.2560161.

[24] L. Xu et al., "Architectural Protection of Application Privacy against Software and Physical Attacks in Untrusted Cloud Environment," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 478-491, 1 April-June 2018, doi: 10.1109/TCC.2015.2511728.

[25] Premkamal, PK, Pasupuleti, SK, Alphonse, PJA. Dynamic traceable CP-ABE with revocation for outsourced big data in cloud storage. Int J Commun Syst. 2021; 34:e4351. https://doi.org/10.1002/dac.4351.

[26] I. H. Abdulqadder, D. Zou, I. T. Aziz, B. Yuan and W. Li, "SecSDN-Cloud: Defeating Vulnerable Attacks Through Secure Software-Defined Networks," in IEEE Access, vol. 6, pp. 8292-8301, 2018, doi: 10.1109/ACCESS.2018.2797214.

[27] O. AlKadi, N. Moustafa, B. Turnbull and K. -K. R. Choo, "Mixture Localization-Based Outliers Models for securing Data Migration in Cloud Centers," in IEEE Access, vol. 7, pp. 114607-114618, 2019, doi: 10.1109/ACCESS.2019.2935142.

[28] M. Ali, K. Bilal, S. U. Khan, B. Veeravalli, K. Li and A. Y. Zomaya, "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security," in IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 303-315, 1 April-June 2018, doi: 10.1109/TCC.2015.2400460.

[29] S. Javaid, H. Afzal, M. Babar, F. Arif, Z. Tan and M. Ahmad Jan, "ARCA-IoT: An Attack-Resilient Cloud-Assisted IoT System," in IEEE Access, vol. 7, pp. 19616-19630, 2019, doi: 10.1109/ACCESS.2019.2897095.

[30] B. G. Raúl and A. M. L. Sevillano, "Services cloud under HSTS, Strengths and weakness before an attack of man in the middle MITM," 2017 Congreso Internacional de Innovacion y Tendencias en Ingenieria (CONIITI), 2017, pp. 1-5, doi: 10.1109/CONIITI.2017.8273322.

[31] V. B. Gadicha and A. S. Alvi, "A review towards enhancing authentication scheme using Image Fusion and multishared Cryptography," 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 0654-0657, doi: 10.1109/ICCSP.2015.7322570.