ISSN: 2959-6386 (Online), Volume 2, Issue 2

Journal of Knowledge Learning and Science Technology

Journal homepage: https://jklst.org/index.php/home

Streamlining Compliance: Orchestrating Automated Checks for Cloud-based AI/ML Workflows

Samir Vinayak Bayani[1], Ravish Tillu[2], Jawaharbabu Jeyaraman[3]

[1]Broadcom Inc, USA.

[2]RBC Capital Markets, USA.

[3]TransUnion, USA.

Abstract

Ensuring security and safeguarding data privacy within cloud workflows has garnered considerable attention in research circles. For instance, protecting the confidentiality of patients' private data managed within a cloud-deployed workflow is crucial, as is ensuring secure communication of such sensitive information among various stakeholders. In light of this, our paper proposes an architecture and a formal model for enforcing security within cloud workflow orchestration. The proposed architecture underscores the importance of monitoring cloud resources, workflow tasks, and data to identify and anticipate anomalies in cloud workflow orchestration. To achieve this, we advocate a multi-modal approach combining deep learning, one-class classification, and clustering techniques. In summary, our proposed architecture offers a comprehensive solution to security enforcement within cloud workflow orchestration, leveraging advanced techniques like deep learning for anomaly detection and prediction, particularly pertinent in critical domains such as healthcare during unprecedented times like the COVID-19 pandemic.

Keywords: Artificial intelligence (AI), disabilities, algorithmic prejudice, bias, ethical concerns

Article Information:

Article history: Received: 01/08/2023 Accepted:10/08/2023 Online: 30/08/2023 Published: 30/08/2023

DOI: https://doi.org/10.60087/jklst.vol2.n3.p435

[i] ***Correspondence author:*** Samir Vinayak Bayani

Introduction

Cloud computing has emerged as a potent paradigm for managing and delivering computations, applications, and services over the Internet [1].

The computational prowess offered by the cloud encompasses a broad spectrum of services, spanning storage, processing, and application services. This computational muscle has empowered researchers to conduct a plethora of computationally intensive and scientific workflows, facilitating extensive experiments that were previously unattainable using local servers. This shift has markedly reduced the overall costs associated with related software systems [1], establishing itself as a promising design paradigm for workflow deployment, processing, and orchestration. A typical large-scale scientific workflow entails a set of interconnected tasks characterized by complexity, fault tolerance, and dynamic execution and orchestration aimed at producing scientific results. However, a cloud workflow specifically refers to a workflow deployed and executed on the cloud. Cloud workflows boast features such as transparency, scalability, multi-tenancy, and real-time monitoring [2].

With virtually limitless computing resources, cloud computing caters to the demands of complex scientific data-intensive workflow tasks and liberates cloud workflows from the onus of resource provisioning planning. Nevertheless, several research challenges must be addressed before fully realizing this potential. These challenges encompass security threats in the cloud domain, including integrity, authorization, availability, reliability, and trust. These challenges extend to workflow security and privacy enforcement in the cloud environment, characterized by complexity, dynamism, and multi-dimensional aspects. Ensuring secure access, deployment, execution, and management of workflows across cloud platforms is paramount for both cloud providers and consumers. Providers must ensure that the resources allocated for workflow execution remain safeguarded against hacking, misuse, or

damage. Similarly, workflow consumers must be assured of the security and protection of their workflows and associated data from external attacks.

Numerous cloud computing security threats have been identified [3] and extensively scrutinized [4–8]. These threats, encompassing data breaches, leaks, loss, denial of service (DoS), and malicious insiders, stem from issues such as multi-tenancy, loss of data control, and breaches of trust. Upholding security in a dynamic environment across different platforms, stakeholders, and processes necessitates the involvement of various entities beyond cloud providers. A holistic security solution encompassing security enforcement, trust chains in clouds, and adherence to policies and regulations is imperative to ensure security and privacy across multiple participants and heterogeneous environments.

The emergence of security challenges within cloud workflows has spurred researchers to delve into various research domains pertaining to security enforcement in dynamically executed and orchestrated cloud workflows. However, existing research on workflow management has predominantly focused on aspects such as anomaly and error detection [9], workflow task scheduling [10], and autonomic workflow resource provisioning and management [11]. These studies primarily aimed to prevent failure or resource contention and ensure the efficient deployment, execution, and performance guarantee of workflows [12]. Unfortunately, such initiatives often overlook the critical aspect of cloud workflow security enforcement, which could enhance the aforementioned aspects and bridge the gap in handling security and data integrity in dynamic cloud workflows. Few studies have exclusively addressed the security aspects of cloud workflow orchestration, management, and enforcement [13]. These studies typically center on anomaly detection and prediction using techniques like HTM [14], statistical clustering [8, 15, 16], regression [17, 18], and unsupervised machine learning (ML) [19]. However, they often lack clear definitions of security attributes and fail to specify the characteristics of cloud workflows, which are resource-aware, time-series, and highly dynamic. Moreover, they tend to focus solely on a specific security dimension, such as data, tasks, or resources, while neglecting other dimensions that could lead to enhanced security enforcement when integrated. Furthermore, the anomaly prediction and detection schemes proposed in most of these studies rely solely on dynamic features of cloud workflows, disregarding static features that typically capture vital workflow information and its hosting environment. Finally, previous attempts have primarily concentrated on security/anomaly detection

and prediction, neglecting the necessity for resource and workflow adaptation strategies to mitigate security threats and potential attacks.

To address some of the limitations of previous studies, we emphasize security anomaly/attack detection and prediction in a cloud workflow orchestration environment. We propose an adaptation scheme to address possible vulnerabilities and mitigate their effects on cloud workflow execution. In such an environment, attacks could target various entities and components, including workflow data, tasks, resources, monitoring, and adaptation. Our proposed model contributes to the state-of-the-art literature on cloud workflow security by incorporating the following:

A multi-dimensional security enforcement framework emphasizing cloud workflow security across various levels: tasks, data, resources, and monitoring schemes.

A scheme integrating static and dynamic features for predicting anomalies/attacks, offering a unique approach to feature modeling that enhances anomaly detection by capturing stakeholders' needs and all aspects of the cloud workflow. This scheme utilizes deep learning autoencoder-based dimensionality reduction for dynamic data, resulting in improved characterization of workflow tasks and consequently better attack prediction.

An unsupervised learning technique requiring no class labeling, additional work, or manual intervention from experts, thus offering convenience and realism in dealing with unknown anomalies.

An adaptation model that accommodates flexible representation and planning of resource requirements over time and throughout the various phases of the cloud workflow execution cycle.

The remainder of this paper is structured as follows: Section 2 discusses related work, compares it, and identifies limitations. Section 3 presents a case study utilizing a workflow based on a COVID-19 dataset. Section 4 proposes an architecture for enforcing end-to-end security in cloud workflow orchestration. In Section 5, a detailed formulation of the cloud workflow security enforcement model and associated learning pipeline algorithms is provided. Section 6 covers implementation details, conducted scenarios, experiments, and a discussion of the results obtained. Finally, conclusions and suggestions for future work are presented in Section 7.

Previous Research

As the prominence of cloud services surged, the realm of security within the cloud computing ecosystem garnered significant attention, with threats and vulnerabilities evolving alongside the exponential growth of cloud services. To advance state-of-the-art security solutions, it is crucial to first identify related risks inherent in emerging cloud services. Such security risks typically stem from three primary attack vectors: external users, internal users, and cloud providers [20, 21]. Over time, security threats within cloud service environments have continued to evolve, with common threats including data breaches, data loss, denial-of-service (DoS) attacks, malicious insiders, service traffic hijacking, vulnerabilities in shared technologies, malware, cyber-attacks, network intrusions, threats at the virtual machine (VM) level, and issues pertaining to data transparency.

Recent advancements in deep learning have prompted researchers to explore its potential in addressing cloud security threats. However, existing approaches often fall short of providing a comprehensive solution encompassing all security threats. Many of these approaches tend to focus solely on detecting and addressing patterns associated with a specific threat within a single deployment scenario. For instance, researchers in employed a multi-layer neural network to detect and recognize malicious behaviors exhibited by users. They transformed user behavior data into a comprehensible format and subsequently classified malicious behavior for detection and recognition purposes.

In their work the authors introduced PredictDeep, a security analytical framework designed for both known and unknown anomaly detection and prediction within Big Data systems. PredictDeep is conceived as a service catering to cloud users, comprising three core modules: a graph model designer, a feature extractor, and an anomaly predictor. Renowned for its scalability, PredictDeep operates effectively in dynamic environments, facilitating real-time anomaly monitoring. However, an assumption made by PredictDeep is the accuracy of all log files, with no injected fake data that could compromise the prediction model's accuracy. Additionally, the proposed approach relies on the integrity of the deployment infrastructure.

Intrusion detection systems (IDS) play a crucial role in monitoring networks, services, and workflows for violations or malicious activities within cloud services orchestration. Detecting novel attacks in such scenarios poses a formidable challenge. Deep learning-based intrusion detection techniques have shown promise in predicting unknown attacks and detecting malicious activities. In the authors introduced an IDS utilizing a deep reinforcement

learning-based architecture capable of addressing and classifying new and complex attacks. They implemented a reward vector to incentivize classifiers yielding identical results, thereby enhancing classification accuracy. Furthermore, in the authors tackled multi-cloud cooperative intrusion detection and improved decision-making in real-time environments using a deep neural network (DNN) model. Leveraging historical feedback data, they employed a DNN model to predict suspicious intrusions. Additionally, a deep learning-based IDS proposed in aimed to detect suspicious attacks within a cloud computing environment by monitoring network traffic. This system utilized a self-adaptive genetic algorithm (SAGA) to automatically generate a DNN-based anomaly network IDS, demonstrating high detection rates, accuracy, and low false alarm rates.

Other research endeavors have delved into enhancing cloud workflow security enforcement, with numerous researchers and industries proposing solutions to bolster the security of cloud services and workflow orchestration. For instance, in developers involved in the European-funded ASCLEPIOS (Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare) project have leveraged various cryptographic and access control techniques to safeguard user data privacy and fortify protection against security breaches within a cloud-based eHealth framework. One of the project's objectives is to facilitate secure data sharing among healthcare stakeholders while upholding participant privacy. The proposed architecture comprises seven layers aimed at providing security and analytic features to support data privacy and access control. However, the incorporation of machine learning techniques for detecting, evaluating, and mitigating potential anomalies in data or attacks on system components such as available resources was not integrated into the proposed architecture.

In the PICASO project a framework was introduced to facilitate cross-organization sharing of electronic health records through a cloud-based solution. This project endeavors to implement the requisite security and privacy measures, in addition to service orchestration and data capture and management. Security features were implemented via separate subsystems to ensure the privacy of patient records, user authentication, traceability of transaction information, and enforcement of access control policies. However, the project did not incorporate mechanisms for detecting anomalies within the proposed framework.

Authors in conducted a literature review on security in Function as a Service (FaaS) orchestration systems, categorizing existing works based on criteria such as the protected asset, the cause of threats, and the protection

approach. They observed that while most works focus on data confidentiality, data integrity is often overlooked. Moreover, function flows and platform misconfigurations are prevalent considerations in the reviewed works. Additionally, authors in provided another classification of existing works employing machine learning and deep learning techniques for online malware detection in the cloud. They categorized malware detection approaches into static analysis, which operates offline without monitoring, and dynamic analysis, which necessitates real-time monitoring and utilizes neural networks to predict potential virtual machine infections, a more suitable approach for the cloud environment. Experimentation revealed that deep learning techniques offer high accuracy in detecting malware. However, this work did not analyze end-to-end security enforcement for workflows within a cloud environment.

Case Study: Cloud Workflow Management during the COVID-19 Pandemic

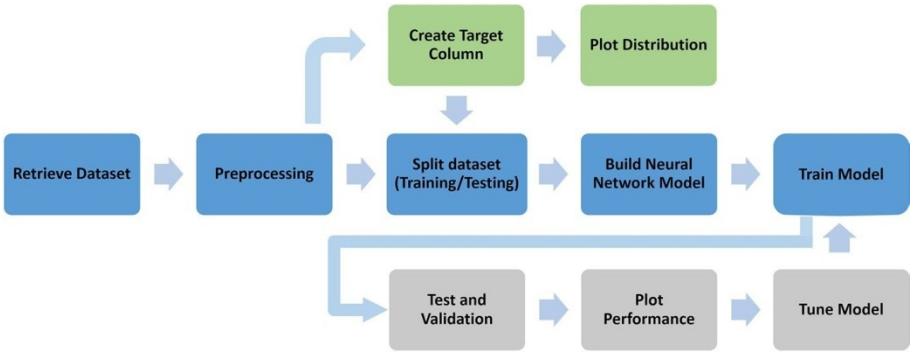
The global healthcare landscape has faced unprecedented challenges with the onset of the Novel Coronavirus (COVID-19) outbreak, declared a pandemic by the World Health Organization in March 2020. The emergence of novel virus strains has surpassed expectations, putting immense strain on healthcare systems worldwide. This pandemic has tested the resilience of various stakeholders, including healthcare providers, governmental agencies, and research institutions, as they collaborate to develop effective treatments or vaccines. Amidst these efforts, it is paramount to safeguard facilities, confidential data, and workflows from potential malicious attacks that could jeopardize the integrity of the entire process.

Throughout the COVID-19 pandemic, the reliance on online resources and cloud-based infrastructure systems has surged dramatically. Lockdown measures, contact-tracing applications, and the widespread adoption of remote working and distance-learning platforms have contributed to this increased reliance. However, this surge has also resulted in a notable uptick in cyber-attacks and breaches of data confidentiality and integrity.

To illustrate the applicability and efficacy of a security enforcement architecture and to identify primary security threats in cloud workflow orchestration, we present a case study centered around a cloud workflow tailored to manage a COVID-19 dataset.

Cloud Workflow and COVID-19 Dataset

Figure 1 illustrates the health monitoring cloud workflow developed utilizing epidemiological data from a COVID-19 outbreak dataset, employing a deep learning model to predict the length of hospital stay for COVID-19 patients. The dataset was meticulously collected and curated from national, provincial, and municipal health reports, alongside other online sources. Geocoded data encompass symptoms, key dates (onset, admission, confirmation), and travel histories of individual patients. The dataset, encompassing 2,500,000 records, each representing a unique patient case, includes 33 columns such as patient ID, age, gender, onset symptoms date, hospital admission date, confirmation date, additional information, chronic disease indicators, specific symptoms, and patient outcomes. Detailed explanations for each field are provided. We utilized this cloud workflow example to identify and assess various security breaches that may be encountered.



Security Threats

The literature on cloud-based infrastructures has extensively addressed various security issues, including insider attacks, data loss, and Denial of Service (DoS) attacks. In this section, we delve into anomaly detection within a cloud workflow orchestration setting, where attacks may target different entities and components, including workflow data, tasks, resources, monitoring, and adaptation components. Below, we outline a few examples of security breaches in a cloud workflow environment.

Cloud Workflow Data Attack

Data attacks encompass various malicious activities, such as data injection intended to corrupt datasets or compromise them through suspicious sharing or downloads. Unauthorized data access and anomalous admin user activities are also common anomalies. For instance, within our cloud workflow, an attacker might inject redundant or fabricated data to disrupt the training and prediction processes, compromising the quality of the prediction model. Such tampering could lead to critical issues, such as patient harm, or overload the Machine Learning (ML) training process, erroneously triggering Quality of Service (QoS) degradation and unnecessary workflow adaptation.

Cloud Workflow Task Attack

A cloud workflow comprises multiple tasks that may run in parallel or sequentially with varying dependency levels. Task attacks encompass a broad spectrum of anomalies, including malware infections, query injections, and DoS attacks. Moreover, attackers may strategically target sensitive processes or tasks that serve as dependencies for numerous other tasks, amplifying the potential damage.

Resource Attack

Various resources within the cloud environment, such as Virtual Machines (VMs), CPUs, memory, and networks, are susceptible to different types of attacks. These attacks may involve unauthorized resource access or overwhelming service requests. False reporting of resource overload or overutilization in monitoring logs could trigger compromised nodes to initiate unnecessary and costly workflow adaptation processes.

Monitoring and Adaptation Component Attack

Monitoring and adaptation components play a crucial role in any cloud workflow orchestration environment, as they are vital for resource management and performance optimization. Within this workflow example, an attack against a monitoring system could coerce the compromised monitoring task into generating false resource underutilization logs, thereby evading necessary adaptation and resulting in performance degradation leading to a DoS attack. Another instance of such an attack involves automatic system reconfiguration, where a compromised node falsely identifies a problem and triggers unnecessary adaptation actions.

These types of attacks have a detrimental impact on the performance and integrity of a cloud workflow orchestration system. In this work, we concentrate on anomaly detection in cloud workflow data, resources, tasks, and monitoring components. Therefore, we propose monitoring resources such as CPU utilization, memory usage, I/O operations, and network activity, as well as task profiles and performance metrics. In the following section, we present our proposed security enforcement mechanism for cloud workflow orchestration.

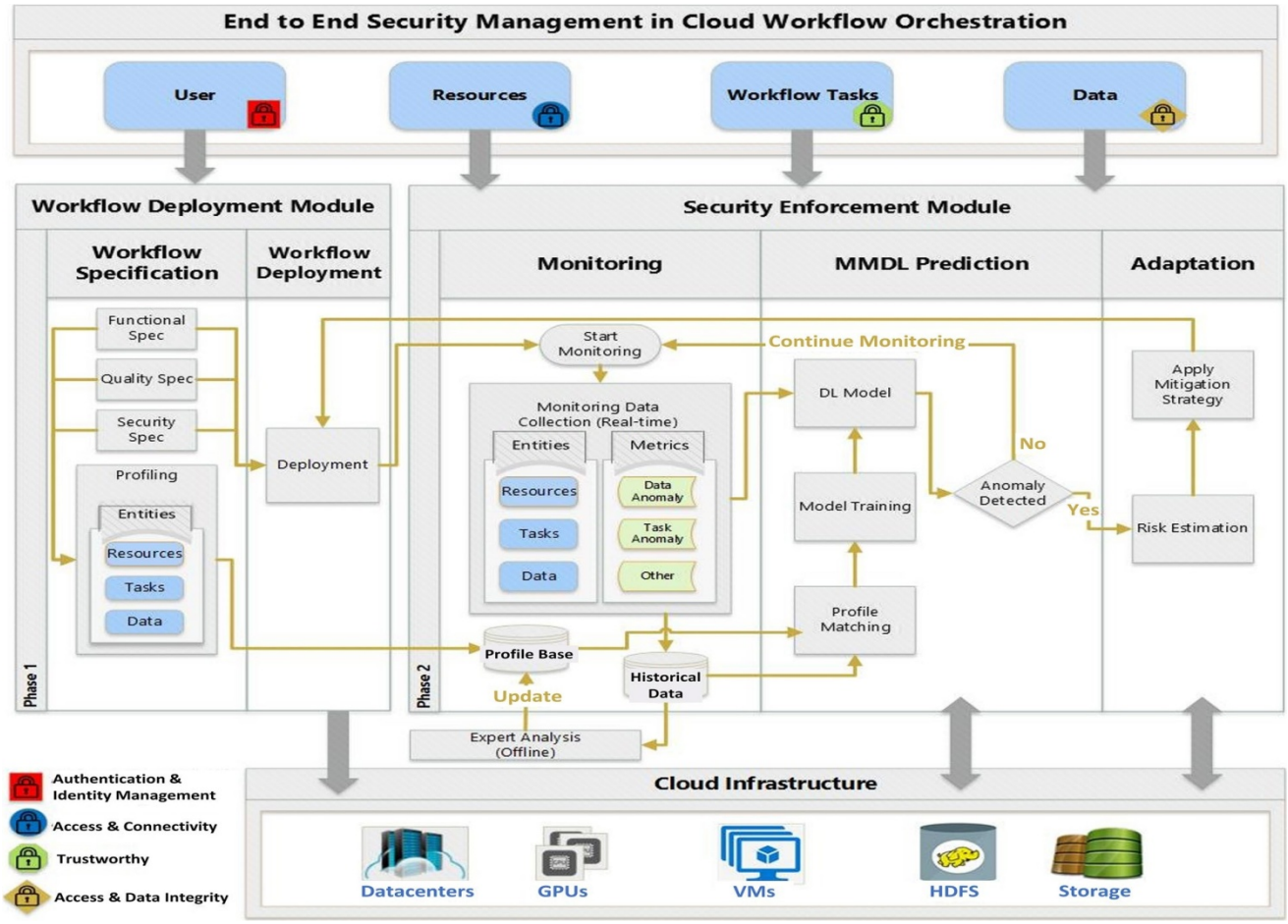
End-to-End Security Enforcement in Cloud Workflow Orchestration

In this section, we outline and describe our end-to-end security enforcement architecture, illustrated in Figure 2. It comprises two main modules: a workflow deployment module and a security enforcement module. Both modules utilize the underlying processing and storage resources (e.g., VMs, GPUs, Storage) from the cloud infrastructure to execute various storage and processing tasks. Security enforcement measures implemented within our architecture are applied to four main entities: users, resources, workflow tasks, and data.

Subsequently, we provide a detailed description of each component of the architecture, highlighting the security features that enhance security, data integrity, and authentication.

Entities

Entities engage with the two modules of the architecture to establish different security boundaries, encompassing authentication and identity management for users interacting with the architecture, access and connectivity management of utilized resources, security enforcement of cloud workflow tasks, and ensuring access and integrity of workflow data.



Workflow Deployment Module

This module comprises two sub-components: the workflow specification and the workflow deployment components. The workflow specification component outlines the functional and non-functional requirements, including quality and security parameters, for the workflow. It also creates profiles for entities such as tasks, data, and resources. On the other hand, the workflow deployment component oversees the deployment and execution lifecycle of the workflow over the cloud infrastructure. The output of this module is a functioning workflow monitored by the security enhancement module, which is responsible for detecting and/or predicting encountered security threats and initiating necessary adaptation actions to mitigate them.

Security Enforcement Module

The security enforcement module consists of three sub-components: monitoring, Multi-Modal Deep Learning Autoencoder (MMDLA)-based prediction, and adaptation sub-modules. These sub-modules work collaboratively to achieve comprehensive cloud workflow monitoring, anomaly detection, and prediction. Finally, these sub-modules implement an adaptation strategy to mitigate risks identified through various anomaly evaluations.

Monitoring Sub-Module

This sub-module is tasked with continuous data collection and monitoring. It collects various runtime data and logs from monitored entities, including tasks, data, and resources. The collected data serve purposes of training and prediction, and they are stored in a historical database for further analysis.

MMDLA Sub-Module

Utilizing the data gathered by the monitoring sub-module, this module trains a multi-modal deep learning autoencoder model for dimensionality reduction. Additionally, it trains a profile matching classification model using the dimensionally reduced data to predict anomalies. The training process of the MMDLA model combines input data generated from the entities profiling module (static) with real-time logs data from monitoring (dynamic). The resulting MMDLA model reduces data dimensionality to enhance efficiency and effectiveness, providing reduced dimensional data as input to an anomaly detection machine learning algorithm for anomaly detection.

Upon detection of an anomaly, the anomaly evaluation process identifies the type and threat level of the anomaly. Subsequently, the anomaly evaluation information is forwarded to the risk estimation process and eventually stored in a database for expert validation, such as identifying suspicious user behavior. A detailed description and implementation of the key features of this module's components are provided in subsequent sections.

Cloud Infrastructure

This component fulfills the architecture's resource requirements concerning the various resources necessary for processing and storing data. Processing tasks encompass activities such as MMDLA model training for dimension reduction, training and classification of anomaly detection models, and monitoring of data storage.

Cloud Workflow Security Enforcement Module

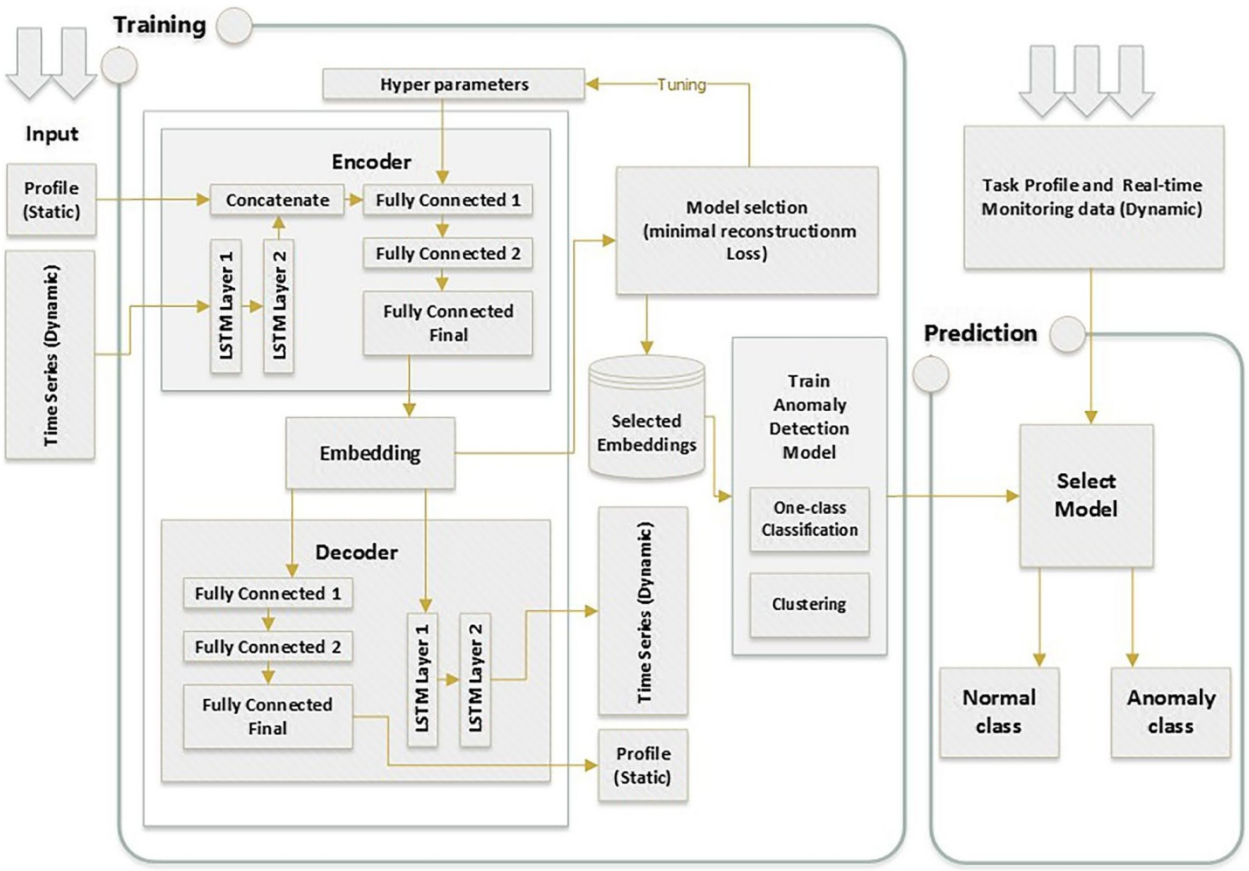
In this section, we delve into the operational framework of the MMDLA prediction-based security enforcement module. Initially, we provide definitions of key terminologies crucial for understanding the prediction model. Subsequently, we address the problem formulation. Lastly, we elucidate the learning pipeline algorithms employed for the proposed solution approach.

Feature Reduction Using Deep Autoencoder

As previously discussed, the feature vector for each task comprises four static features and six time-series features. To train a model utilizing two-dimensional feature vectors, we must flatten the time-series feature matrix into a one-dimensional feature vector and merge it with the static features. However, this process can result in the creation of a high-dimensional feature vector. Specifically, the total number of features in the vector would be $4 + 6k_i$, where k_i is the number of observations of the dynamic features. For instance, if $k_i = 100$, the total number of flattened features would be 604. Hence, a feature reduction technique becomes necessary.

We opt for the AutoEncoder technique for two main reasons:

1. AutoEncoder enables unsupervised feature reduction, which aligns with a crucial aspect of our proposed model.
2. We propose a multi-modal deep learning (MMDLA) based AutoEncoder model by integrating Long Short-Term Memory (LSTM) – a specific type of recurrent neural network (RNN) – with a Deep FeedForward network (DFN). This MMDLA model streamlines the feature reduction process by learning from the temporal relationships among time-series features.



Features and Combine Them with Static Features, Rather Than Implementing a Feature Reduction Process That Flattens All the Time-Series Features and Loses the Temporal Information Contained in the Feature Set.

Figure 3 illustrates the high-level architecture of this feature reduction, training, and prediction technique.

Here, we elaborate on the proposed AutoEncoder-based model, which will be referred to as MMDLA. It comprises two main components: the Encoder and the Decoder, detailed as follows. The Encoder comprises two LSTM layers, a concatenation layer, and three fully connected layers, as shown in Fig. 3.

Implementation and Experimentation

Environment Setup

In this section, we delineate the experimental environment. We established a Docker Swarm Cluster comprising one master node and four worker nodes. The cloud workflow described in Section 3 was deployed over a workstation

running Linux Ubuntu 18.04, equipped with 24 CPU cores, two NVIDIA GeForce GTX 1080 Ti GPUs with 11GB GDDR5X memory each, a 1-TB HDD, and 64-GB RAM. Each task in the cloud workflow was instantiated as a Docker container, executed with varying data input sizes. The Docker swarm cluster was managed by a master node, orchestrating the required cluster state, while the worker nodes received and executed tasks dispatched by the master node. Deploying a workflow to a swarm necessitates providing service definitions to the master node, which then dispatches units of work, known as tasks, to the worker nodes. Throughout the workflow execution, we captured a live data stream to execute task containers and monitor various performance metrics, elaborated upon in detail in the subsequent section. Additionally, we ran mock containers to overload nodes in the cluster, simulating a real-world environment. The experimental setup is depicted in Fig. 4.

We implemented the proposed algorithms in Jupyter Notebook using Python 3.6. The AutoEncoder and SVM algorithms were developed utilizing Pytorch and Scikit-learn, respectively, which are open-source Python implementations of machine learning and deep learning neural networks. The experiments were conducted on a Mac computer with OS X Catalina 10.15.4 operating system, featuring a 2.8-GHz Quad-Core Intel Core i7 processor and 16-GB 1600-MHz DDR3 RAM.

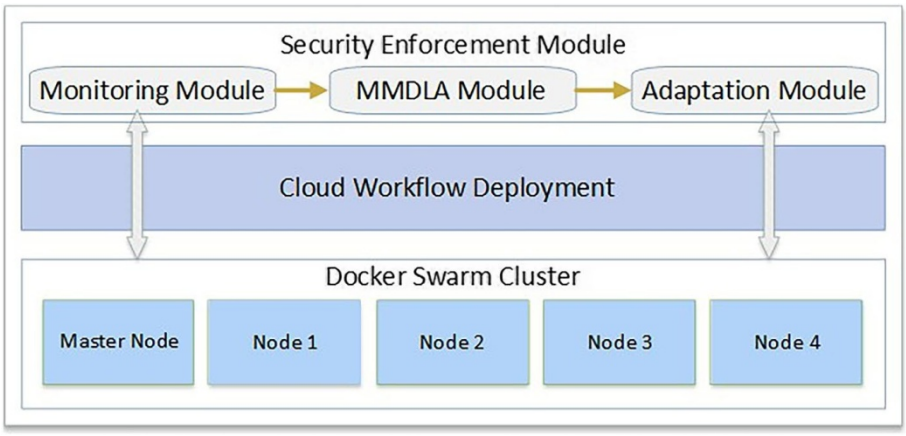
Dataset

Description of Dataset

In our experiments, we amalgamated two types of data for each running task. Initially, we defined a static task profile, which encompassed various types of information such as task duration, data input size, data output size, and task category (pre-processing, training, or evaluation). The dynamic data comprised live stream data of performance monitoring metrics for each running task. This data consists of time-series records, including CPU and memory usage by the container, total memory used by the container, size of data sent and received by the container over the underlying network, and the size of read/write data by the container from block devices on the host.

Data Preparation and Preprocessing

The preprocessing activities primarily revolved around converting Docker's generated monitoring statistics. Initially, the Docker stats were formatted with a flag to output the exact required container statistics. Subsequently, the output file was parsed and cleaned using regex to appropriately split the column headers. The data was then converted to a Pandas DataFrame, and proper data types were assigned to each column (e.g., the timestamp column used the datetime datatype). Additionally, the units of the memory utilization columns were standardized to bytes.



Deep Learning Approach for Training and Anomaly Detection

To detect anomalies, we initially trained our dataset using a reconstruction AutoEncoder model to reduce the data dimension into a 30-dimensional embedding. Subsequently, we inputted the output generated by the AutoEncoder model into an anomaly detection model. The following sequence of steps outlines our implementation:

First, we split the dataset into two sets: static profile data and dynamic time-series performance monitoring information. The architecture of the encoder-decoder neural network developed for feature learning is depicted in Figure 3.

The dynamic part of the data is fed into two layers of a time-series RNN model encoder. This model takes batch size, number of records, and number of features as inputs and returns outputs in the form of a (1, 30) vector, which represents the final hidden state. The output is concatenated with the static data portion and fed into three fully connected layers to produce the output shape of a (1, 30) vector.

The decoder, on the other hand, uses the (1, 30) vector and passes it to two separate layer sequences: three fully connected layers and two RNN layers. The fully connected layers decode the static part of the input, while the RNN layers produce the dynamic time-series part. An essential aspect here is that the encoder always provides the data input length, enabling the decoder to determine how many time-series data points to produce.

The output of the encoder was trained over an anomaly detection model, such as a one-class classification or clustering algorithm. The one-class classification algorithms are unsupervised learning algorithms trained using only non-anomaly data, i.e., the reduced feature set resulting from the aforementioned AutoEncoder algorithm. These algorithms include one-class SVM, Isolation Forest, Elliptic Envelope, and Local Outlier Factor. Additionally, we used different clustering algorithms, i.e., unsupervised learning algorithms trained using both anomaly and normal data. Among these, we utilized k-means, Mini Batch k-means, MeanShift, and Birch. All models predicted two classes or clusters, i.e., normal or anomaly. However, the performance of each model varied in terms of accuracy, precision, recall, and F1 score. We then selected the best-performing model based on the calculated performance metrics for our real-time security enforcement.

Experimental Scenarios, Evaluation Criteria, and Fault Injection Scheme

We conducted several experiments to evaluate our proposed security enforcement and anomaly detection framework. In these experiments, we aimed to evaluate the anomaly detection scheme by investigating the performance of different anomaly detection algorithms and models. Additionally, we conducted different experiments to evaluate the performance of the cloud workflow within the adopted proposed security enforcement model. We benchmarked the cloud workflow performance based on the previously proposed adaptation strategies. In these experiments, we ran our designed hospital length-of-stay prediction workflows several times with different patient dataset sizes. The performance of the cloud workflow was continuously monitored, and adaptation strategies were executed when necessary, depending on the decision taken by the adaptation module.

Scenarios

We designed two scenarios for testing the proposed security enforcement model. The first scenario focused on testing the performance and accuracy of our anomaly detection and prediction model, and the second scenario evaluated the overall performance of the cloud workflow.

In the first scenario, implemented in two stages, we used the Deep Learning AutoEncoder to reduce the dimension of the dataset containing encodings, which were then fed to the anomaly detection module. The latter implements different ML algorithms, including one-class classification and clustering algorithms. Each algorithm was evaluated and compared in terms of four different performance measures, including accuracy, precision, recall, and F1 scores, after applying cross-fold with k-fold values of 3, 5, and 10.

In the second scenario, where we evaluated the overall cloud workflow performance, we considered the CPU utilization, memory usage, network I/O bound, and disk space usage features. The cloud workflow was executed over the implemented Docker swarm environment with different resource load capacities.

Conclusion

In conclusion, our study has highlighted the critical importance of streamlining compliance through orchestrating automated checks for cloud-based AI/ML workflows. As organizations increasingly rely on cloud infrastructure to deploy AI and ML solutions, ensuring compliance with regulatory standards and internal policies becomes paramount.

Through our proposed approach, we have demonstrated the effectiveness of integrating automated compliance checks into cloud-based AI/ML workflows. By leveraging orchestration tools and incorporating compliance checks at various stages of the workflow, organizations can proactively identify and address compliance issues, thereby reducing the risk of regulatory violations and reputational damage.

Moreover, our study underscores the benefits of automation in enhancing efficiency and scalability while maintaining compliance. By automating compliance checks, organizations can streamline their workflow processes, minimize manual intervention, and ensure consistency in adherence to regulatory requirements.

Moving forward, further research and development in this area are warranted to explore additional strategies for enhancing compliance in cloud-based AI/ML workflows. This includes investigating advanced techniques for automated compliance monitoring, integrating real-time anomaly detection, and leveraging machine learning algorithms for predictive compliance analysis.

Overall, our study contributes to the ongoing efforts to strengthen compliance practices in cloud-based AI/ML workflows, ultimately enabling organizations to harness the full potential of these technologies while mitigating associated risks.

References :

[1]. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU (2015) The rise of 'big data' on cloud computing: review and open research issues. Information Systems 47:98–115.

[2]. Huang H, Zhang YL, Zhang M (2013) A survey of cloud workflow. Advanced Materials Research 765:1343–1348.

[3]. Puthal D, Sahoo BPS, Mishra S, Swain S (2015) Cloud computing features, issues, and challenges: a big picture. In: 2015 International conference on computational intelligence and networks, pp 116–123.

[4]. Halabi T, Bellaiche M, Abusitta A (2018) Online allocation of cloud resources based on security satisfaction. Proceedings - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. pp 379–384. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00063

[5]. Halabi AAT, Bellaiche M (2018) Cloud security up for auction: a DSIConline mechanism for secure iaas resource allocation. 2nd Cyber Security in Networking Conference, CSNet, vol 2018. pp 1–8.

[6]. Halabi T, Bellaiche M, Abusitta A (2019) Toward secure resource allocation in mobile cloud computing: a matching game. In: Int. Conf. Comput. Netw. Commun. ICNC 2019, pp 370–374. https://doi.org/10.1109/ICCNC.2019.8685509

[7]. Abusitta A, Bellaiche M, Dagenais M (2018) An SVM-based framework for detecting DoS attacks in virtualized clouds under challenging.

[8]. Pillai, A. S. (2023). Advancements in Natural Language Processing for Automotive Virtual Assistants Enhancing User Experience and Safety. *Journal of Computational Intelligence and Robotics*, *3*(1), 27-36. https://thesciencebrigade.com/jcir/article/view/161

[9]. Sarker, M. (2022). Towards Precision Medicine for Cancer Patient Stratification by Classifying Cancer By Using Machine Learning. *Journal of Science & Technology*, *3*(3), 1-30. DOI: https://doi.org/10.55662/JST.2022.3301

[10]. Manoharan, A., & Sarker, M. REVOLUTIONIZING CYBERSECURITY: UNLEASHING THE POWER OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR NEXT-GENERATION THREAT DETECTION. *DOI: https://www. doi. org/10.56726/IRJMETS32644*.

DOI : https://www.doi.org/10.56726/IRJMETS32644

[11]. Bappy, M. A., & Ahmed, M. (2023). ASSESSMENT OF DATA COLLECTION TECHNIQUES IN MANUFACTURING AND MECHANICAL ENGINEERING THROUGH MACHINE LEARNING MODELS. *Global Mainstream Journal of Business, Economics, Development & Project Management*, *2*(04), 15-26.

DOI: https://doi.org/10.62304/jbedpm.v2i04.67

[12]. Hossain, M. I., Bappy, M. A., & Sathi, M. A. (2023). WATER QUALITY MODELLING AND ASSESSMENT OF THE BURIGANGA RIVER USING QUAL2K. *Global Mainstream Journal of Innovation, Engineering & Emerging Technology*, *2*(03), 01-11. DOI: https://doi.org/10.62304/jieet.v2i03.64

[13]. Sharma, Y. K., & Harish, P. (2018). Critical study of software models used cloud application development. *International Journal of Engineering & Technology, E-ISSN*, 514-518.

https://scholar.google.com/citations?view_op=view_citation&hl=en&user=Fxv3elcAAAAJ&citation_for_view=Fxv3elcAAAAJ:d1gkVwhDpl0C

[14]. Padmanaban, H. (2023). Navigating the intricacies of regulations: Leveraging AI/ML for Accurate Reporting. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *2*(3), 401-412.

DOI: https://doi.org/10.60087/jklst.vol2.n3.p412

[15]. Padmanaban, P. H., & Sharma, Y. K. (2019). Implication of Artificial Intelligence in Software Development Life Cycle: A state of the art review. *vol*, *6*, 93-98.

https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Implication+of+Artificial+Intelligence+in+Software+Development+Life+Cycle%3A+A+state+of+the+art+review&btnG=

[16]. PC, H. P., Mohammed, A., & RAHIM, N. A. (2023). *U.S. Patent No. 11,762,755*. Washington, DC: U.S. Patent and Trademark Office.

https://patents.google.com/patent/US11762755B2/en

[17]. Miah, S., Rahaman, M. H., Saha, S., Khan, M. A. T., Islam, M. A., Islam, M. N., ... & Ahsan, M. H. (2013). Study of the internal structure of electronic components RAM DDR-2 and motherboard of nokia-3120 by using neutron radiography technique. *International Journal of Modern Engineering Research (IJMER)*, *3*(60), 3429-3432

https://shorturl.at/nCJOQ

[18]. Rahaman, M. H., Faruque, S. B., Khan, M. A. T., Miah, S., & Islam, M. A. (2013). Comparison of General Relativity and Brans-Dicke Theory using Gravitomagnetic clock effect. *International Journal of Modern Engineering Research*, *3*, 3517-3520.
https://shorturl.at/hjm37

[19]. Miah, M. H., & Miah, S. (2015). The Investigation of the Effects of Blackberry Dye as a Sensitizer in TiO2 Nano Particle Based Dye Sensitized Solar Cell. *Asian Journal of Applied Sciences*, *3*(4).
https://shorturl.at/iyJQV

[20]. Miah, S., Miah, M. H., Hossain, M. S., & Ahsan, M. H. (2018). Study of the Homogeneity of Glass Fiber Reinforced Polymer Composite by using Neutron Radiography. *Am. J. Constr. Build. Mater*, *2*, 22-28.
https://shorturl.at/joDKZ

[21]. Miah, S., Islam, G. J., Das, S. K., Islam, S., Islam, M., & Islam, K. K. (2019). Internet of Things (IoT) based automatic electrical energy meter billing system. *IOSR Journal of Electronics and Communication Engineering*, *14*(4 (I)), 39-50.

[22]. Nadia, A., Hossain, M. S., Hasan, M. M., Islam, K. Z., & Miah, S. (2021). Quantifying TRM by modified DCQ load flow method. *European Journal of Electrical Engineering*, *23*(2), 157-163.

https://shorturl.at/csuO3

[23]. Miah, S., Raihan, S. R., Sagor, M. M. H., Hasan, M. M., Talukdar, D., Sajib, S., ... & Suaiba, U. (2022). Rooftop Garden and Lighting Automation by the Internet of Things (IoT). *European Journal of Engineering and Technology Research*, *7*(1), 37-43.
DOI: https://doi.org/10.24018/ejeng.2022.7.1.2700

[24]. Prasad, A. B., Singh, S., Miah, S., Singh, A., & Gonzales-Yanac, T. A Comparative Study on Effects of Work Culture on employee satisfaction in Public & Private Sector Bank with special reference to SBI and ICICI Bank.

[25]. Ravichandra, T. (2022). A Study On Women Empowerment Of Self-Help Group With Reference To Indian Context.

https://www.webology.org/data-cms/articles/20220203075142pmwebology%2019%20(1)%20-%2053.pdf

[26]. Kumar, H., Aoudni, Y., Ortiz, G. G. R., Jindal, L., Miah, S., & Tripathi, R. (2022). Light weighted CNN model to detect DDoS attack over distributed scenario. *Security and Communication Networks*, *2022*.
https://doi.org/10.1155/2022/7585457

[27]. Ma, R., Kareem, S. W., Kalra, A., Doewes, R. I., Kumar, P., & Miah, S. (2022). Optimization of electric automation control model based on artificial intelligence algorithm. *Wireless Communications and Mobile Computing*, *2022*.
https://doi.org/10.1155/2022/7762493

[28]. Devi, O. R., Webber, J., Mehbodniya, A., Chaitanya, M., Jawarkar, P. S., Soni, M., & Miah, S. (2022). The Future Development Direction of Cloud-Associated Edge-Computing Security in the Era of 5G as Edge Intelligence. *Scientific Programming*, *2022*.
https://doi.org/10.1155/2022/1473901

[29]. Al Noman, M. A., Zhai, L., Almukhtar, F. H., Rahaman, M. F., Omarov, B., Ray, S., ... & Wang, C. (2023). A computer vision-based lane detection technique using gradient threshold and hue-lightness-saturation value for an autonomous vehicle. *International Journal of Electrical and Computer Engineering*, *13*(1), 347.
https://shorturl.at/ceoyJ

[30]. Patidar, M., Shrivastava, A., Miah, S., Kumar, Y., & Sivaraman, A. K. (2022). An energy efficient high-speed quantum-dot based full adder design and parity gate for nano application. *Materials Today: Proceedings*, *62*, 4880-4890.

https://doi.org/10.1016/j.matpr.2022.03.532