SCIENTIFIC Publishing Limited

Digital Technologies Research and Applications

http://ojs.ukscip.com/index.php/dtra

Review

Exploring Machine Learning Algorithms to Enhance Cloud Computing Security

Mircea Ţălu [1,2]

[1] Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Cluj 400027, Romania

[2] SC ACCESA IT SYSTEMS SRL, Cluj-Napoca, Cluj 400158, Romania

[*] Correspondence: talu.s.mircea@gmail.com; Tel.: 004-0264-401-200

Received: 26 May 2025; **Revised:** 6 June 2025; **Accepted:** 10 June 2025; **Published:** 3 July 2025

Abstract: The increasing adoption of cloud computing (CC) has introduced significant security and privacy concerns, demanding intelligent and adaptive solutions. This review explores the application of machine learning (ML) algorithms—both supervised and unsupervised—in addressing these challenges within cloud environments. A total of 87 peer-reviewed studies published between 2014 and 2025 were analyzed to assess the effectiveness of various ML techniques. Supervised Machine Learning (SML) algorithms such as Artificial Neural Networks (ANNs), Support Vector Machines (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, and C4.5 Decision Trees are examined for their effectiveness in intrusion detection, anomaly classification, and threat mitigation. Concurrently, Unsupervised Machine Learning (UML) algorithms, including Unsupervised Neural Networks (UNNs), K-Means clustering, and Singular Value Decomposition (SVD), are analyzed for their capacity to detect unknown threats and extract latent patterns from unlabeled data. Key trends reveal a growing preference for hybrid models, the superior accuracy of deep learning in anomaly detection, and the emerging use of context-aware frameworks. The review shows a comparative analysis of these approaches, highlighting their advantages, limitations, and application scenarios in cloud security. Future research directions are proposed, emphasizing hybrid learning models, enhanced datasets, and context-aware security frameworks. The findings underscore the transformative potential of ML in fortifying cloud infrastructures against evolving cyber threats.

Keywords: Cloud Computing; Cloud Security; Machine Learning; Security Threats; Storage-Based Attacks; VM-Based Attacks; Machine Learning Algorithms

1. Introduction

The term “cloud computing” first emerged in the mid-2000s and marked a significant turning point in the evolution of information technology. It introduced a new paradigm for delivering computing services—such as servers, storage, databases, networking, software, analytics, and intelligence—over the internet, commonly referred to as “the cloud.” This innovation fundamentally transformed how organizations and individuals access, manage, and scale computing resources [1–3]. This paradigm shift towards cloud-based solutions has been propelled by the growing demand for cost-effective, scalable, and flexible infrastructure to accommodate individuals’ and businesses’ increasing data needs. By providing standardized devices and shared resources through the internet, cloud computing (CC) minimizes costs and meets the diverse needs of its users. With the increase of CC, these resources became available on-demand, allowing users to pay only for what they use, scale services dynamically, and access

powerful computing capabilities from virtually anywhere. The flexibility, scalability, and cost-efficiency offered by cloud computing led to its rapid adoption across industries, empowering everything from startups and small businesses to large enterprises to innovate faster, improve operational efficiency, and deliver digital services with unprecedented speed. Today, CC is a cornerstone of modern digital infrastructure, underpinning technologies such as artificial intelligence, big data, Internet of Things (IoT), and mobile applications. Furthermore, the services provided by Cloud Service Providers (CSPs), such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS), cater to various requirements and support a broad spectrum of applications [4–6]. Among these services, cloud storage plays a pivotal role in the architecture of cloud computing, facilitating data sharing and storage for consumers.

Central to the notion of cloud security is data protection, which encompasses a range of practices aimed at securing data within the cloud environment. This involves storing data in secure data centers and ensuring that only authorized users gain access [5]. Data protection strategies include the integration of policies, procedures, and technological solutions to secure cloud-based applications and systems, alongside the data they house and the user access that governs them. One common method for strengthening security is through authentication processes, which verify the identities of users and entities within the cloud [6]. Furthermore, trust evaluation is an essential aspect of cloud security, particularly when processing data in a decentralized manner. This evaluation helps users select trustworthy CSPs by assessing the risk associated with particular providers, with trust levels directly correlating to the perceived degree of risk. Despite its numerous advantages, CC introduces several security concerns that can impede its widespread adoption, including vulnerabilities related to user privacy, data integrity, and network security [7–9]. These concerns are exacerbated by the non-transparent and distributed nature of cloud environments, which complicates the protection of sensitive information. The very reliance on internet connectivity for cloud services makes them susceptible to various threats, including malware injection, data breaches, and data losses. Given the critical nature of security in cloud environments, both (CSPs) and users must prioritize safeguarding their systems and data against these risks [10]. To address these challenges, various security mechanisms have been proposed, such as access control, data protection, attack mitigation, and trust delegation [11]. These solutions aim to enhance the security framework of CC, fostering a more resilient and trustworthy ecosystem [12]. Recent studies provide critical insights into the security and performance aspects of cloud-native applications and emerging technologies in CC [6,13–18].

As cloud environments continue to evolve in complexity and cyberattacks become increasingly sophisticated, the adoption of machine learning (ML) techniques has emerged as a promising approach to bolstering cloud security. ML algorithms, which empower systems to learn from data and adapt over time, are particularly effective in detecting anomalies, predicting emerging threats, and automating responses to security incidents in real-time [7].

This review provides an in-depth exploration of the role of ML algorithms in enhancing CC security. We examine the various security threats and challenges that cloud environments face, alongside the potential solutions offered by ML techniques. Furthermore, we conduct a comparative analysis of different ML algorithms, assessing their effectiveness within the context of cloud security and outlining the strengths and limitations of each approach.

2. Research Methodology

This review employs a systematic literature review to explore and analyze the application of ML algorithms in enhancing cloud computing (CC) security. The methodology is structured into the following stages: (a) formulation of focused research questions; (b) identification and collection of relevant scholarly literature; (c) critical evaluation of the quality and scope of the selected studies; (d) synthesis of extracted data; and (e) interpretation of findings to derive key insights and trends.

To ensure a comprehensive and up-to-date analysis, a literature review was conducted on 87 peer-reviewed articles published between 2014 and 2025. Articles and books were sourced from prominent academic databases including IEEE Xplore, SpringerLink, ScienceDirect, MDPI, ACM Digital Library, and Google Scholar. The literature search was guided by a Boolean query string integrating relevant keywords and phrases, such as “machine learning in cloud security,” “supervised learning for intrusion detection,” “unsupervised anomaly detection,” “cloud computing threat mitigation,” “ML-based data classification,” “deep learning in CC,” and “privacy-preserving ML algorithms.”

To ensure transparency and methodological rigor, the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework was adopted as a guiding structure for study selection, screening, and synthe-

sis.

The inclusion criteria were defined as follows: (i) publications must be peer-reviewed, (ii) published in English, (iii) fall within the 2014–2025 time frame, and (iv) directly address the application of machine learning techniques in cloud security.

Studies were excluded if they lacked scientific rigor, were not peer-reviewed (e.g., opinion articles, editorials, or white papers), or addressed ML outside the scope of CC-related cybersecurity challenges.

By applying these rigorous inclusion and exclusion criteria, the final selection of studies offers a comprehensive and representative foundation for evaluating the capabilities, advantages, and limitations of various ML techniques. These studies underpin the comparative analysis of supervised and unsupervised learning methods, providing valuable insights into current challenges and helping to identify promising avenues for future research in cloud security.

3. Cloud Computing Architecture: Service Models and Deployment Models

A CC architecture is structured to offer adaptable and scalable IT services via the internet. It consists of multiple interconnected layers and functional components, which collectively ensure the reliable, efficient, and secure delivery of computing resources. These components work in harmony to manage data storage, processing power, networking, and application deployment without the need for direct hardware management by the end user (**Figure 1**).

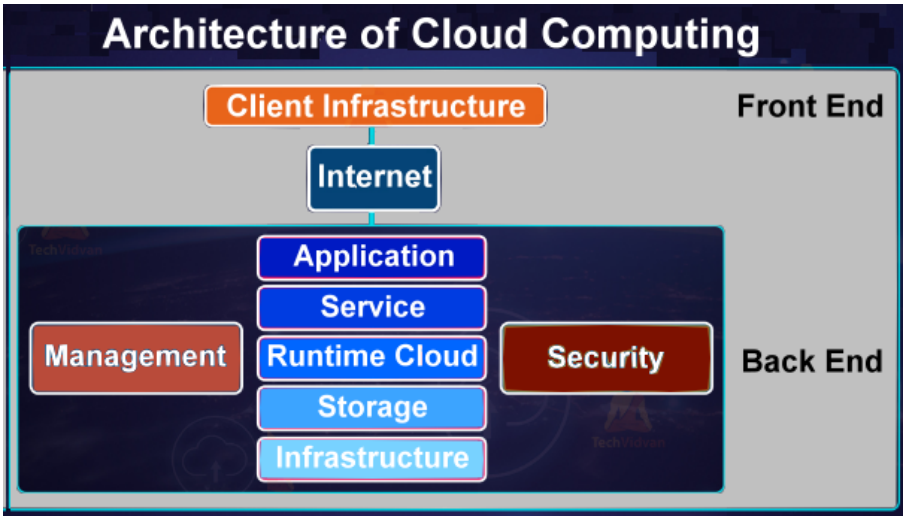

Figure 1. The Architecture of Cloud Computing.

With the rapid expansion of the Internet of Things (IoT), a growing number of critical infrastructure systems—such as those in energy, transportation, healthcare, and manufacturing—are transitioning to cloud-based environments. This shift enables organizations to streamline operations, increase flexibility, and scale more effectively in response to evolving demands and data loads.

CC architecture is structured into two main parts: the front end, which includes user-facing interfaces like web browsers, mobile devices, and thin or fat clients; and the back end, which is managed by service providers and handles data storage, virtual machines, deployment models, security, traffic control, and servers. These two ends communicate through the internet.

The architecture comprises several key components. Client infrastructure forms the front end, offering a GUI (Graphical User Interface) for interaction. Applications are the software platforms users access, while services—delivered as SaaS, PaaS, or IaaS—determine the type of cloud functionality provided. Runtime cloud offers execution environments for virtual machines, and storage provides scalable space for data management. The infrastructure includes hardware and software like servers, networking devices, and virtualization tools. Management oversees coordination and operation of all components, while security ensures backend protection. The internet acts as the medium linking front and back ends, enabling seamless communication and service delivery.

CC architecture also incorporates Elastic Resource Management, a critical capability that dynamically allocates computing, storage, and networking resources to various applications. This allocation aims to meet the performance objectives of cloud applications, cloud service providers (CSPs), and end users. The primary goal for CSPs is to ensure efficient and effective utilization of available resources while adhering to the constraints defined by Service Level Agreements (SLAs). To support this objective, virtualization technologies are employed within the infrastructure layer, enabling statistical multiplexing of physical resources across multiple customers and applications. Furthermore, workload execution data is continuously collected and maintained in a historical workload database, which serves as the foundation for training predictive models. These workload predictors estimate future resource demands and usage patterns, providing critical insights for energy-efficient resource allocation and intelligent load balancing decisions. As a result, cloud platforms become more adaptive and capable of sustaining diverse and dynamic application requirements, especially in high-demand environments such as IoT-enabled infrastructure systems.

The Resource Management System (RMS) is implemented in the data center to handle application requests from cloud clients and provide appropriate responses by allocating the necessary resources, typically in the form of virtual machines (VMs), based on the demands of the application (**Figure 2**). The RMS is comprised of two main components: the VM Management Unit (VMU) and the Task Management Unit (TMU). The VMU is responsible for scheduling VMs, determining their placement on physical machines (PMs), and managing VM migration when a PM experiences overloading or underutilization. The TMU handles incoming application requests from cloud clients or users, breaks them down into smaller tasks, schedules these tasks, and assigns them to the selected VMs for execution. The system tracks the execution data of workloads, which is stored in a historical workload database. This data is then used to train a workload prediction model, which estimates future workload patterns and resource usage. The predictions made by this model are instrumental in making decisions about energy-efficient resource allocation and optimizing load balancing within the system.

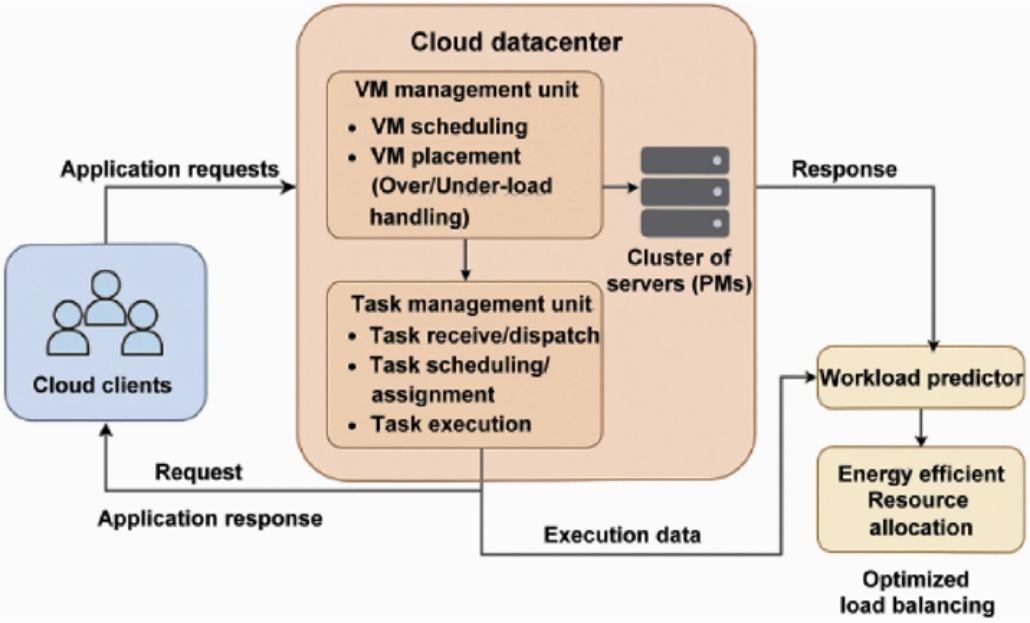

Figure 2. The Conceptual Framework for Resource Management in a Cloud Environment.

3.1. Cloud Service Models

CC is underpinned by a service-oriented delivery framework, which is typically categorized into three primary models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (**Figure 3**). Each model represents a distinct abstraction layer in the cloud ecosystem, offering varying degrees of control, flexibility, and management responsibilities to the user [6].

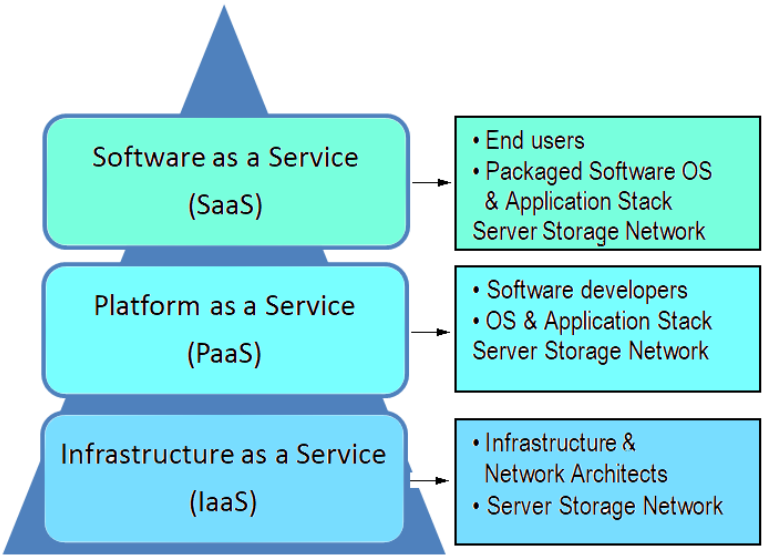

Figure 3. Cloud Computing Service Models.

- Infrastructure as a Service (IaaS) constitutes the foundational layer of cloud services, delivering virtualized computing resources—such as servers, storage systems, and networking infrastructure—over the internet. This model facilitates the provisioning of scalable and on-demand computational capacity, enabling users to deploy and manage operating systems, applications, and storage with minimal hardware dependencies. IaaS supports a utility-based consumption model, wherein users access computing resources dynamically, as needed, without incurring the overhead of maintaining physical infrastructure.
- Platform as a Service (PaaS) builds upon the IaaS layer by abstracting hardware and operating system complexities, and instead, providing a comprehensive development and deployment environment. This includes integrated development frameworks, application hosting platforms, databases, middleware, and essential development tools. PaaS empowers developers to build, test, and manage cloud-native applications without the operational burden of configuring or maintaining the underlying infrastructure. Furthermore, it enables rapid application deployment and scalability through standardized APIs, automated provisioning, and support for continuous integration/continuous deployment (CI/CD) pipelines.
- Software as a Service (SaaS) represents the highest abstraction level in the cloud service hierarchy, delivering fully operational, user-facing software applications over the internet. SaaS offerings typically operate on a subscription-based or pay-per-use model, allowing users to access and utilize software through web interfaces without requiring local installation or maintenance. Common examples include email platforms, customer relationship management (CRM) systems, and enterprise collaboration tools. SaaS simplifies software accessibility, enhances cross-platform compatibility, and reduces time-to-deployment, while centralizing updates, data management, and security controls within the service provider's domain.

3.2. Cloud Deployment Models

Cloud deployment models can be categorized into the following five types, each tailored to meet specific organizational requirements (**Figure 4**). A comparative analysis of the cloud deployment models based on key parameters is shown in **Table 1**.

Figure 4. Types of Cloud Computing Deployment Models.

Table 1. Comparative Analysis of the Cloud Deployment Models.

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud	Multi-Cloud
Host	Service provider	Enterprise (or third party)	Enterprise (or third party)	Community (or third party)	Multiple cloud providers
Users	General public	Selected users	Selected users	Community members	Multiple organizations
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN	Internet, VPN
Owner	Cloud service provider	Enterprise	Enterprise	Community	Multiple organizations
Cost Model	Pay-per-usage	Infrastructure investment	Mixed/variable	Shared among members	Variable, usage-based
Security	Provider-managed	High control and customization	Varied (depends on configuration)	Varied, community-defined	Varied, provider-dependent
Scalability	Highly scalable	Limited to internal resources	Scalable (via public + private integration)	Scalable within shared community resources	Scalable across providers
Customization	Limited	High	Depends on deployment model	Depends on community needs	Limited per provider
Resource Sharing	Not shared	Not shared	Variable (depends on design)	Shared among community members	Shared across providers
Examples	AWS, Microsoft Azure, Google Cloud	OpenStack, VMware-based private cloud	AWS Outposts, Azure Stack	Government or Healthcare Community Cloud	AWS + Azure + GCP combinations
Advantages	Cost-effective, scalable, minimal maintenance	Enhanced control, data security	Flexibility, balanced cost-security model	Cost sharing, regulatory alignment	Resilience, vendor independence, optimized performance
Disadvantages	Limited control, lower security	High cost, maintenance responsibility	Complex management, vendor dependency	Not suitable without collaboration	Complex integration, governance challenges

4. Cloud Computing Threats

This section examines the principal threats to CC, categorized using the foundational principles of the CIA Triad (Confidentiality, Integrity, Availability) [19–21], alongside a taxonomy of attacks targeting key cloud infrastructure components.

1. Confidentiality threats primarily concern unauthorized access to sensitive data, whether through internal misuse or external breaches. Three critical risks dominate this domain:

- Insider threats: One of the most insidious and difficult-to-detect threats arises from malicious insiders — individuals with authorized access who exploit their privileges to compromise customer data. Such threats are particularly alarming given the broad administrative access many CSPs possess, often without adequate monitoring mechanisms.
- External attacks: The openness and interconnected nature of cloud environments render them susceptible to external adversarial attacks, including but not limited to distributed denial-of-service (DDoS), remote code execution, and credential theft. These attacks often exploit insecure network channels, unpatched vulnerabilities, or misconfigured resources.
- Information leakage: Inadvertent data exposure—resulting from human error, misconfigured access controls, or inadequate security tools—remains a pervasive concern. The sheer scale of data processing and storage in cloud systems exacerbates the potential impact of such leakages.

2. Data integrity in CC is compromised when unauthorized modifications occur, whether during storage, transmission, or processing. The following issues undermine integrity assurances:

- Data isolation failures: Cloud systems frequently rely on virtualization technologies to allocate shared resources among multiple tenants. Misconfigurations in virtual machine (VM) environments or hypervisors may lead to overlapping access boundaries, thereby enabling one tenant to tamper with or access another's data.
- Inadequate access controls: Weak identity and access management (IAM) systems can permit attackers to impersonate legitimate users or escalate privileges, thereby facilitating unauthorized data alterations. Poorly implemented or inconsistently enforced IAM policies are a primary contributor to such vulnerabilities.

- Data quality threats: Flawed data ingestion pipelines, lack of validation mechanisms, or synchronization errors across distributed systems can introduce erroneous or corrupted data, undermining trust in cloud-hosted information assets.

3. Availability threats in CC is defined as the continuous and reliable access to cloud services and resources, is threatened by both operational disruptions and malicious activities. Key threat vectors include:

- Service downtime due to system changes: Changes in the cloud infrastructure — such as software upgrades or hardware reconfigurations — can inadvertently lead to service outages if not properly managed. The dynamic nature of cloud environments complicates the assurance of seamless service delivery.
- Infrastructure and network failures: External disruptions, including failures in domain name system (DNS), bandwidth exhaustion, or data center outages, can render services inaccessible to end-users. These disruptions may stem from cyberattacks or natural disasters affecting the physical infrastructure.
- Physical attacks and intrusions: Physical damage to servers, data centers, or transmission lines — whether due to sabotage, accidents, or environmental factors — poses a direct threat to cloud availability.
- Inefficient disaster recovery mechanisms: Inadequate or poorly tested disaster recovery plans hinder the timely restoration of services following an incident. This results in prolonged downtimes and significant losses in operational continuity.

4.1. Classification of Cloud-Based Attacks

CC is susceptible to specific attack vectors classified according to the architectural layer they target: network-based, VM-based, storage-based, and application-based attacks. Each category is outlined below with representative threats.

1. Network-based attacks compromise cloud communication channels and can act as precursors to deeper system intrusions:

- Port Scanning: This reconnaissance technique enables attackers to identify open ports and services, which may then be exploited using targeted attacks. While port scans are often overlooked, they provide crucial intelligence for subsequent exploitation.
- Botnets: Composed of compromised internet-connected devices, botnets are orchestrated to execute coordinated attacks, including spam distribution and DDoS campaigns, often with significant destructive capacity.
- Spoofing attacks: In spoofing scenarios, malicious actors impersonate legitimate entities to deceive systems or users, thereby gaining unauthorized access or exfiltrating data. This includes Internet Protocol (IP) spoofing, Address Resolution Protocol (ARP) spoofing, and Domain Name System (DNS) spoofing, among others.

2. Virtual Machine (VM)-based attacks compromise the fundamental isolation and security mechanisms of virtualization in cloud computing.

- Side-channel attacks: These sophisticated attacks extract sensitive information by analyzing physical or behavioral patterns—such as timing, cache access, or electromagnetic emissions—rather than exploiting software flaws.
- Malicious VM images: Compromised or intentionally manipulated VM images can introduce persistent threats into the cloud environment, especially if reused or shared without proper validation and scanning.
- VM escape and hypervisor attacks: Attackers exploit vulnerabilities in the hypervisor layer to break the isolation between VMs, gaining unauthorized access to underlying system resources or co-hosted VMs.

3. Storage-based attacks in CC compromise the confidentiality, integrity, and availability of data stored in cloud environments.

- Data scavenging: Even when data is deleted, residual traces may persist on storage media, enabling attackers to recover sensitive information through forensic techniques.

- Data deduplication exploits: Attackers may infer or gain access to files based on storage deduplication mechanisms, especially in public cloud scenarios where deduplication is shared across tenants.
- Unauthorized access to backup data: Improperly secured backups or shadow copies can serve as an alternative entry point for attackers seeking sensitive historical data.

4. Application-based attacks compromise the functionality, confidentiality, integrity, and trustworthiness of cloud-based software services. These attacks target the software layer — particularly SaaS platforms, Application Programming Interfaces (APIs), and web services — exploiting weaknesses in application design, deployment, or configuration.

- Malware injection and steganographic attacks: Adversaries may embed malicious payloads into cloud-hosted applications or conceal them using steganographic methods, compromising system integrity without immediate detection.
- API abuse and web service exploits: Weakly secured APIs can be leveraged for privilege escalation, data exfiltration, or denial of service attacks. Common vulnerabilities include insufficient authentication, rate limiting, and input sanitization.
- Shared technology vulnerabilities: Exploiting common components or libraries used across cloud services can allow attackers to simultaneously affect multiple applications or tenants.

5. Machine Learning Algorithms and Cloud Computing Security

Machine Learning (ML), a subset of artificial intelligence, has emerged as a critical enabler of intelligent cloud security, offering data-driven models for threat detection, risk assessment, and anomaly identification. By analyzing vast amounts of data, ML algorithms can identify patterns, predict malicious activities, and mitigate risks in real-time. Classification tasks can be addressed through both supervised and unsupervised ML approaches, each offering distinct methodologies and applications. This section discusses the role of supervised and unsupervised ML algorithms in CC, highlighting their objectives, techniques, advantages, and challenges through recent scholarly contributions.

5.1. Supervised Machine Learning Algorithms

Supervised Machine Learning (SML) refers to a type of ML where the model is trained using labeled data. In the context of CC, SML algorithms are widely used to detect patterns, predict outcomes, and automate decision-making in cloud-based applications such as intrusion detection, threat mitigation, resource allocation, and data classification. SML algorithms work by learning from a training set and using the patterns identified to make predictions on new, unseen data.

Types of SML algorithms in CC are:

- Artificial Neural Networks (ANNs). An ANN is composed of layers of interconnected nodes (neurons), where each neuron performs a weighted sum of its inputs followed by a non-linear activation function. ANNs are used for complex pattern recognition, anomaly detection, and deep learning tasks such as cloud-based image and speech recognition. However, their computational cost can be high, as training deep neural networks requires substantial processing power. Furthermore, the interpretability of ANN models can be limited, which may pose challenges in decision-making scenarios where transparency is critical.
- Support Vector Machines (SVM). SVM are used for binary classification by finding the optimal hyperplane that maximally separates two classes in a high-dimensional space. SVM are excellent for classification tasks like intrusion detection, especially when the dataset is high-dimensional. However, one significant drawback of SVM is its computational complexity, especially in large-scale cloud environments. Additionally, while SVM performs well in binary classification tasks, it may encounter difficulties when extended to multi-class problems, requiring additional modifications and tuning.
- K-Nearest Neighbors (K-NN). K-NN is used for classification (and regression) and assigns the label to a new data point based on the majority class of its k closest neighbors in the feature space. K-NN is ideal for classification tasks, such as categorizing cloud traffic or user behavior based on labeled historical data. However, K-NN can

be computationally expensive when dealing with large datasets, particularly when the number of dimensions is high. Furthermore, its performance heavily depends on the choice of the 'k' parameter and the distance metric used, which can be challenging to optimize in dynamic cloud environments.

- Naive Bayes. Naive Bayes is a probabilistic classifier based on Bayes' Theorem, assuming that the features are conditionally independent given the class label. It is particularly well-suited for tasks like spam filtering in email systems or categorization of cloud traffic. Naive Bayes is commonly used for spam filtering and other classification tasks in cloud environments. However, the strong independence assumption often does not hold in real-world datasets, which can lead to suboptimal performance, especially when features are highly correlated. Additionally, Naive Bayes may struggle to capture complex relationships within the data, making it less effective in situations where deep learning or more sophisticated models are required.
- C4.5 Decision Trees. C4.5 is an algorithm for generating a decision tree from a dataset. It works by recursively splitting the data into subsets based on feature values. The key idea is to choose the best feature that maximizes information gain (or reduces entropy) for classification tasks. C4.5 is used for decision support and classifying large datasets, common in resource allocation in CC. However, decision trees like those generated by C4.5 can suffer from overfitting, especially when the tree grows too large or the dataset is noisy. To mitigate this issue, pruning techniques are often applied, but this adds additional computational overhead. Moreover, decision trees may struggle to model complex relationships, limiting their effectiveness when applied to high-dimensional data.

(a) SML algorithms offer several advantages for CC. They provide high accuracy when trained with large, labeled datasets, making them ideal for critical tasks like security threat detection. These algorithms also offer clear decision-making insights, especially with models like decision trees. SML models are scalable, capable of handling vast amounts of cloud data, and can automate predictions, reducing human intervention. Furthermore, they are versatile, applicable to various CC tasks, from network traffic analysis to resource management.

(b) SML algorithms in CC have several disadvantages. Obtaining accurate data labels, especially for novel threats, can be challenging, limiting the initial effectiveness of SML models. Additionally, cloud environments are dynamic, and SML models may struggle to adapt to changing patterns unless retrained regularly with new labeled data. These models also require periodic retraining and maintenance, which can be resource-intensive. Moreover, if the training data is biased or unrepresentative, SML models may perpetuate biases, leading to unfair or inaccurate predictions.

(c) SML algorithms have several limitations. They rely heavily on large amounts of labeled data, which can be resource-intensive and time-consuming to acquire in cloud environments. SML models may also suffer from overfitting if not properly tuned, leading to poor generalization on unseen data, particularly in dynamic cloud settings. Training complex models, like deep neural networks, requires significant computational resources, driving up costs in cloud environments. Furthermore, the accuracy of SML models is dependent on the quality of labeled data; inaccurate or biased data can degrade model performance and lead to incorrect predictions.

Table 2 shows a comparative overview of supervised learning techniques used for CC. While each algorithm has distinct advantages and areas of application, understanding the trade-offs is critical for selecting the appropriate approach for specific cloud security tasks.

Table 2. A Comparative Overview of Supervised Learning Techniques Used for Cloud Computing.

Objective	Technique	Advantages	Disadvantages
Public/private cloud workload protection	ANN	High data privacy	Requires specialized applications
Secure cryptosystems	SVM	Improved security	Storage/network errors
Attack detection	ANN	High accuracy	Time and storage intensive
Intrusion detection	ANN	Effective dataset testing	Accuracy not reported
Resource provisioning	K-NN + Data Mining	Simple, scalable	High memory, time-consuming
Privacy preservation	K-NN	Time-efficient	Accuracy not reported
Cloud protection	C4.5 + Signature Detection	Handles noise, supports various data types	Overfitting, unstable trees

Table 2. *Cont.*

Objective	Technique	Advantages	Disadvantages
Web pre-fetching	Naive Bayes	Efficient data handling	Time and storage issues
Intrusion detection	Naive Bayes	Compatible	Accuracy not reported
Intrusion detection	SVM + Naive Bayes	High accuracy	Limited environment
Cloud authentication	ANN + Delphi	Better data analysis	Low detection precision, unpredictable behavior
Cloud-level attack mitigation	ANN/NN	Parallel processing	High computational cost

5.2. Unsupervised Machine Learning Algorithms

Unsupervised Machine Learning (UML) refers to a class of machine learning algorithms designed to analyze and interpret datasets without labeled responses. In the context of CC, UML is crucial for uncovering hidden patterns, groupings, and structures within large-scale, unclassified data. These algorithms are widely applied in tasks such as anomaly detection, intrusion detection, user behavior analysis, trust modeling, and dimensionality reduction -often serving as the foundation for exploratory data analysis. UML algorithms learn the inherent structure of the data by identifying clusters or reducing dimensionality to reveal critical patterns. As these algorithms do not require labeled data, they are highly valuable in dynamic or rapidly evolving cloud environments where labeled datasets may not be readily available.

Types of UML algorithms in CC include:

- Unsupervised Neural Networks (UNNs): These networks operate without prior knowledge of data outputs. They classify data based on internal similarities and can detect correlations between diverse data sources, making them suitable for anomaly detection and pattern recognition in unstructured cloud traffic. By examining the intrinsic features of the data, UNNs can uncover hidden relationships and detect unusual patterns that may indicate security breaches, such as DDoS attacks or unauthorized access. Their ability to adapt to evolving data makes them highly valuable in rapidly changing cloud environments, although their computational complexity can be a challenge in large-scale systems.
- K-Means Clustering: A simple yet powerful clustering technique, K-Means identifies a predefined number of centroids (clusters) and iteratively assigns data points to the nearest cluster. It is frequently used for workload segmentation, intrusion detection, and user behavior modeling in cloud infrastructures. However, K-Means requires the number of clusters to be predefined, which can be challenging in dynamic environments where the optimal number of clusters may not be immediately obvious.
- Singular Value Decomposition (SVD): A robust dimensionality reduction algorithm, SVD is central to recommendation systems and compressed representation of high-dimensional cloud data. It aids in uncovering latent features, optimizing storage, and enhancing data interpretability. SVD also aids in improving the performance of ML models by eliminating noise and irrelevant information, making it easier to detect important security-related patterns in cloud infrastructures. However, like other dimensionality reduction techniques, SVD may lose some information during the reduction process, which can impact the accuracy of certain analyses.

(a) UML algorithms offer several advantages for CC. UML algorithms can effectively analyze unlabeled or semi-structured data, offering scalability and flexibility in data-driven tasks. These models reduce the dependency on human-labeled datasets, allowing automation in scenarios like anomaly detection or trust evaluation. Moreover, techniques like SVD help reduce computational load by minimizing redundant information in high-dimensional cloud datasets.

(b) UML algorithms come with notable disadvantages. The absence of labeled data makes evaluation of model accuracy difficult. Interpretability is also a challenge, as the clustering and dimensionality reduction processes may not yield clearly actionable insights. Additionally, some techniques (e.g., deep unsupervised models) are computationally intensive and sensitive to parameter selection, affecting model stability and scalability in complex cloud environments.

(c) UML algorithms have several limitations. UML algorithms struggle with managing dynamic cloud data

where patterns frequently shift. Their reliance on internal metrics for clustering or decomposition may yield sub-optimal results when data distributions are uneven or noisy. Moreover, without labeled feedback, models cannot self-correct misclassifications, necessitating careful tuning and monitoring.

Table 3 shows a comparative overview of unsupervised learning techniques used for CC. While each algorithm has distinct strengths, their effectiveness depends on the specific security tasks at hand. However, the lack of labeled data can pose challenges in evaluating model performance, and interpretability remains an ongoing concern for many unsupervised algorithms.

Table 3. A Comparative Overview of Unsupervised Learning Techniques Used for Cloud Computing.

Objective	Technique	Advantages	Disadvantages
ML capability for secure cryptosystems	K-Means, ANN	Ensures high data privacy; protects cloud workloads	Requires specialized client-server applications for proper functionality
Trust evaluation strategy to predict trust values for users/resources	SVD	Efficient access control; strong privacy protection	Impacts network performance; security vulnerabilities
Analyze encrypted mobile traffic using deep learning	CNN, Deep Learning	Enables fast data transfer; strong security	May produce runtime errors
Operationalization of ML-based security detections	K-Means, Intrusion Detection	High privacy, consistency, and information restriction	Difficulties managing dynamic cloud data
Intrusion detection in cloud environments	K-Means	High accuracy and consistency	Limited model comparability
User privacy preservation	SVD	High accuracy in privacy-sensitive applications	Tested on a single model; lacks generalization
Dimensionality reduction for cloud datasets	SVD	Achieves high accuracy in compressed representation	Model comparability issues

6. Performance Metrics of ML Algorithms in Cloud Computing Security

The performance metrics are defined as [22]:

- Accuracy measures the proportion of correctly classified instances (both attacks and normal) out of the total, and it ranges from 0 to 1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where: TP (True Positive): Correctly predicted positive cases (e.g., actual attacks correctly identified); TN (True Negative): Correctly predicted negative cases (e.g., normal traffic correctly identified); FP (False Positive): Incorrectly predicted positive cases (e.g., normal traffic flagged as attack); FN (False Negative): Incorrectly predicted negative cases (e.g., attack missed as normal).

- Precision (Positive Predictive Value) quantifies the proportion of predicted attacks that are truly attacks.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- Recall (also known as Sensitivity or True Positive Rate) quantifies the proportion of actual attacks that are correctly identified.

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

- The F1-Score, defined as the harmonic mean of precision and recall, provides a balance between false positives and false negatives.

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

- Specificity (True Negative Rate) indicates the model's ability to correctly identify negative instances, thereby avoiding false positives.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5}$$

- False Positive Rate (FPR) measures the proportion of negative instances incorrectly classified as positive, highlighting the importance of minimizing false alarms to prevent alert fatigue.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{6}$$

- False Negative Rate (FNR) measures the proportion of actual positive instances incorrectly classified as negative, and a lower FNR ensures that real attacks are not overlooked, preventing security breaches from going undetected.

$$\text{FNR} = \frac{FN}{FN + TP} \tag{7}$$

Table 4 highlights the key performance metrics for each of the ML algorithms, reflecting typical ranges observed in academic and industry research applied to CC security [23].

Table 4. Performance Metrics of ML Algorithms in Cloud Computing Security.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	FPR (%)	FNR (%)
ANN (Artificial Neural Networks)	92	91	93	92	90	10	7
SVM (Support Vector Machines)	89	88	90	89	87	13	10
K-NN (K-Nearest Neighbors)	85	84	86	85	82	18	14
Naive Bayes	80	79	82	80	78	22	18
C4.5 (Decision Trees)	88	86	87	86	85	15	13
UNN (Unsupervised Neural Networks)	87	85	88	86	84	16	12
K-Means Clustering	84	82	85	83	80	20	15
SVD (Singular Value Decomposition)	83	80	84	82	79	21	16

The performance metrics shown in **Table 4** highlight significant differences between the various ML algorithms used in cloud computing (CC) security. ANN and SVM stand out with the highest overall performance across key indicators—accuracy (92% and 89%), precision (91% and 88%), recall (93% and 90%), and F1-score (92% and 89%), respectively. These high values suggest that ANN and SVM are particularly effective in correctly identifying both normal and malicious activity, making them well-suited for critical applications such as real-time intrusion detection, DDoS mitigation, and adaptive threat response systems. Their relatively low False Positive Rates (FPR: 10% for ANN, 13% for SVM) and False Negative Rates (FNR: 7% for ANN, 10% for SVM) further reinforce their reliability, minimizing both missed attacks and false alerts that could lead to alert fatigue or system downtime.

K-Nearest Neighbors (K-NN) and C4.5 decision trees demonstrate moderate performance, with accuracy scores of 85% and 88%, respectively. While K-NN offers simplicity and ease of implementation, its FPR (18%) and FNR (14%) suggest potential drawbacks in high-volume or noise-prone environments. C4.5 fares slightly better in specificity (85%) and recall (87%), making it a viable choice for classification tasks where interpretability and logical rule generation are essential, such as access control or compliance monitoring systems.

Naive Bayes, while computationally efficient, shows the weakest performance across most metrics, with accuracy (80%), precision (79%), and a relatively high FPR (22%) and FNR (18%). This indicates a higher likelihood of both missed detections and false alarms, limiting its use in scenarios demanding high assurance. Nevertheless, its low computational footprint may still make it useful in edge computing scenarios with limited processing power.

UNN and K-Means Clustering offer notable potential in anomaly detection and unsupervised threat discovery, particularly in dynamic or zero-day environments. While their accuracy (87% for UNN, 84% for K-Means) and

F1-scores (86% and 83%) are slightly lower than top performers, they maintain a balance between detection capability and generalization. However, their higher FPRs (16% and 20%) indicate a tendency to over-alert, which may necessitate post-processing or hybrid refinement with supervised models.

Lastly, SVD demonstrates moderate effectiveness in terms of accuracy (83%) and recall (84%), but its precision (80%) and F1-score (82%) suggest that it may struggle with distinguishing subtle threats. While SVD is advantageous for dimensionality reduction and noise filtering, its use as a standalone predictor in threat detection scenarios is limited, and it is best employed as a preprocessing step to enhance the performance of other algorithms.

In conclusion, high-performing models like ANN and SVM are preferable for mission-critical environments requiring high accuracy and low latency, while hybrid models combining dimensionality reduction, anomaly detection, and classification may offer improved flexibility and robustness in evolving threat landscapes.

7. Future Research Directions

Recent studies have identified several pivotal areas for future research:

- Hybrid unsupervised and semi-supervised learning models: Integrating unsupervised and semi-supervised learning approaches can enhance the detection of specific attack types and zero-day vulnerabilities. This fusion leverages the strengths of both methods, aiming to improve detection accuracy without extensive labeled datasets.
- Crowdsourcing for data annotation in security applications: Utilizing crowdsourcing for data annotation presents opportunities to enrich training datasets for ML models. However, challenges such as ensuring data quality, maintaining security, and managing the integration of diverse data sources remain critical areas for further investigation.
- Artificial Intelligence in hybrid cloud security: The application of AI-driven automation in hybrid cloud environments addresses issues of security fragmentation. By unifying security policies and enhancing visibility, AI can mitigate risks associated with disparate on-premises and cloud infrastructures.
- Location-Based Services (LBS) integration: The expansion of LBS offers avenues for context-aware security measures in cloud computing. Anticipated trends and technological advancements in LBS can be harnessed to bolster security protocols, particularly in mobile and geographically distributed networks.

8. Conclusions

This review has examined the role of both supervised and unsupervised ML algorithms in enhancing cloud security by enabling real-time threat detection, intrusion prevention, trust evaluation, and data privacy preservation. Supervised learning techniques, such as ANNs, SVM, K-NN, Naive Bayes, and C4.5 decision trees, have demonstrated strong performance in identifying known threats and automating classification tasks within cloud environments. Their ability to learn from labeled datasets makes them particularly suitable for structured security problems such as intrusion detection and attack classification. In parallel, unsupervised learning approaches, including UNNs, K-Means clustering, and SVD, offer valuable capabilities for uncovering hidden patterns in unstructured or unlabeled data. These methods are critical for anomaly detection, feature extraction, and identifying previously unknown attack vectors in evolving cloud ecosystems. Despite their respective strengths, both paradigms face limitations, such as data dependency, computational overhead, and difficulties in managing high-dimensional or noisy datasets. Addressing these limitations calls for innovative research in hybrid learning models, more robust datasets, and scalable architectures tailored to dynamic cloud infrastructures. Future research should focus on enhancing model interpretability, leveraging real-time threat intelligence, and ensuring privacy-preserving learning across distributed cloud platforms.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

All data are shown in the authors' published papers cited in this paper.

Acknowledgments

Not applicable.

Conflicts of Interest

The author declares no conflict of interest.

References

1. Antonopoulos, N.; Gillam, L. *Cloud Computing: Principles, Systems and Applications*, 1st ed.; Springer: Cham, Switzerland, 2017.
2. Wegener, A. *Cloud Computing: Systems and Technologies*, 1st ed.; Clanrye International: New York, NY, USA, 2019; pp. 210–220.
3. Comer, D. *The Cloud Computing Book*, 1st ed.; CRC Press, Taylor & Francis Group: Boca Raton, FL, USA, 2021; pp. 5–12.
4. Khalil, I.M.; Khreishah, A.; Azeem, M. Cloud computing security: A survey. *Computers* **2014**, *3*, 1–35. [CrossRef]
5. Butt, U.A.; Mehmood, M.; Shah, S.B.H.; et al. A review of machine learning algorithms for cloud computing security. *Electronics* **2020**, *9*, 1379. [CrossRef]
6. Almutairi, M.; Sheldon, F.T. IoT–cloud integration security: A survey of challenges, solutions, and directions. *Electronics* **2025**, *14*, 1394, 1–28. [CrossRef]
7. Vacca, J.R. *Cloud Computing Security: Foundations and Challenges*, 2nd ed.; CRC Press, Taylor & Francis Group: Boca Raton, FL, USA, 2020; pp. 64–70. [CrossRef]
8. Harkut, D.G. *Cloud Computing Security: Concepts and Practice*; IntechOpen: London, UK, 2020; pp. 6–10. [CrossRef]
9. Achari, A. *Cybersecurity in Cloud Computing*; Educohack Press: Delhi, India, 2025; pp. 18–22.
10. Abdulsalam, Y.S.; Hedabou, M. Security and privacy in cloud computing: Technical review. *Future Internet* **2022**, *14*, 11. [CrossRef]
11. Chauhan, M.; Shiaeles, S. An analysis of cloud security frameworks, problems and proposed solutions. *Network* **2023**, *3*, 422–450. [CrossRef]
12. Khan, M.A.; Khan, S.M.; Subramaniam, S.K. A systematic literature review on security issues in cloud computing using edge computing and blockchain: Threat, mitigation, and future trends. *Malays. J. Comput. Sci.* **2023**, *36*, 347–367. [CrossRef]
13. Ahmad, W.; Rasool, A.; Javed, A.R.; et al. Cybersecurity in IoT-based cloud computing: A comprehensive survey. *Electronics* **2022**, *11*, 16. [CrossRef]
14. Țălu, M. A review of vulnerability discovery in WebAssembly binaries: Insights from static, dynamic, and hybrid analysis. *Acta Tech. Corviniensis Bull. Eng.* **2024**, *17*, 13–22.
15. Țălu, M. A review of advanced techniques for data protection in WebAssembly. *Ann. Fac. Eng. Hunedoara Int. J. Eng.* **2024**, *22*, 131–136.
16. Țălu, M. A comparative study of WebAssembly runtimes: Performance metrics, integration challenges, application domains, and security features. *Arch. Adv. Eng. Sci.* **2025**, 1–13. [CrossRef]
17. Țălu, M. Security and privacy in the IIoT: Threats, possible security countermeasures, and future challenges. *Comput. AI Connect* **2025**, *2*, 1–10. [CrossRef]
18. Țălu, M. Cyberattacks and cybersecurity: Concepts, current challenges, and future research directions. *Digit. Technol. Res. Appl.* **2025**, *4*, 44–60. [CrossRef]
19. Parast, F.K.; Sindhav, C.; Nikam, S.; et al. Cloud computing security: A survey of service-based models. *Comput. Secur.* **2022**, *114*, 102580. [CrossRef]

20. Subramanian, N.; Jeyaraj, A. Recent security challenges in cloud computing. *Comput. Electr. Eng.* **2018**, *71*, 28–42. [CrossRef]
21. Pavithra, B.; Vinola, C.; Mishra, N.; et al. Cloud security analysis using machine learning algorithms. In Proceedings of the Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 23–25 August 2023; pp. 704–708. [CrossRef]
22. Rainio, O.; Teuho, J.; Klén, R.B. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **2024**, *14*, 6086. [CrossRef]
23. Nassif, A.B.; Talib, M.A.; Nasir, Q.; et al. Machine learning for cloud security: A systematic review. *IEEE Access* **2021**, *9*, 20717–20735. [CrossRef]