

A Hybrid Autoencoder–XGBoost Framework for Proactive Cloud Security and Vulnerability Detection

Karupothu Bharath¹, T Swapna Sridevi²

¹MTech in the department of Computer Science and Engineering,

²Assistant Professor for the Department of Computer Science and Engineering in Pydah College of Engineering, Yanam Road, Patavala, Tallarevu (M), Kakinada, Andhra Pradesh.

Abstract — Cloud computing has become a fundamental part of modern digital infrastructure, enabling organizations to deploy applications, store data, and deliver services across highly distributed environments. However, the dynamic nature of cloud systems also creates complex security challenges, including abnormal network activity, unauthorized access, misconfigurations, and emerging vulnerabilities. Traditional security mechanisms often depend on predefined rules and known attack patterns, making them less effective when dealing with previously unseen or continuously evolving threats. This paper presents a machine learning-based approach for strengthening cloud security by combining anomaly detection and vulnerability classification. The proposed framework uses an Autoencoder to identify unusual patterns in cloud-related data and XGBoost to classify detected security events and potential vulnerabilities. The system processes security-relevant information such as network traffic, system logs, user activity, and historical vulnerability records through data preprocessing and feature extraction stages before applying the machine learning models. By combining unsupervised anomaly detection with supervised classification, the framework is designed to provide a more adaptive approach to identifying both familiar and potentially unknown security threats. The proposed solution is implemented as a web-based security framework using Python and Django, providing mechanisms for data management, model evaluation, prediction, and security monitoring. It is intended to reduce dependence on manual analysis while supporting faster identification of suspicious activities and potential vulnerabilities. The framework also provides a foundation for continuous model improvement as new security data becomes available. Overall, the proposed approach demonstrates how machine learning can be integrated into cloud security workflows to support proactive vulnerability detection and improve the adaptability of security monitoring in distributed environments.

Keywords— Cloud Computing; Cloud Security; Machine Learning; Vulnerability Prediction; Anomaly Detection; Autoencoder; XGBoost; Distributed Systems; Threat Detection; Cybersecurity; Intrusion Detection; Proactive Security.

I. INTRODUCTION

Cloud computing has emerged as an important technology for storing data, running applications, and delivering services through flexible and scalable computing environments. Its distributed architecture allows organizations to manage large workloads and resources efficiently. However, the same characteristics that make cloud computing flexible also introduce complex security challenges. As cloud infrastructures continue to grow, they become exposed to vulnerabilities, unauthorized access, data breaches, malicious activities, and other security threats.

Traditional cloud security mechanisms mainly rely on firewalls, access-control policies, encryption, rule-based intrusion detection systems, and periodic vulnerability scanning. These mechanisms are generally designed around predefined rules and known attack signatures. Although they can effectively identify previously recognized threats, their ability to respond to new and rapidly changing attack patterns

is limited. In addition, periodic vulnerability assessment and manual security monitoring may not provide sufficient protection for dynamic cloud environments. [1]

Machine learning provides an opportunity to improve this security process by analyzing large volumes of security-related data and identifying patterns that may not be easily detected through conventional methods. Cloud environments continuously generate information through system logs, network traffic, user activities, and historical vulnerability records. Analyzing these data sources can help identify abnormal behavior and provide useful indications of potential security risks. This creates the possibility of moving cloud security from a reactive approach toward a more proactive and adaptive approach. [2]

In this work, a machine learning-based framework is proposed for predicting and preventing vulnerabilities in distributed cloud systems. The framework combines Autoencoder-based anomaly detection with XGBoost-based classification. The

Autoencoder is used to identify unusual patterns within the collected security data, while XGBoost is employed to classify security-related events and potential vulnerabilities. The combination of these two techniques provides a multi-stage approach in which abnormal behavior can first be identified and then analyzed for more specific classification.

The proposed system processes cloud-related data through a sequence of stages, including data collection, preprocessing, feature extraction, model training, vulnerability detection, alert generation, prevention, and model updating. Data preprocessing removes irrelevant or inconsistent information, while feature extraction identifies security-related characteristics such as access patterns, traffic behavior, error rates, and configuration changes. These processed features are then provided to the machine learning models for security analysis. [3]

The framework is designed to support continuous monitoring of the cloud environment. When suspicious behavior or a potential vulnerability is identified, the system can generate an alert and provide recommendations for preventive actions. These actions may include access restrictions, configuration changes, or traffic filtering. Such automation can reduce the amount of manual effort required for security monitoring and can support faster identification of potential risks.

The system is implemented as a web-based application using Python and Django, with supporting technologies including HTML5, CSS3, JavaScript, and MySQL. The application provides functionality for dataset management, machine learning model analysis, prediction, and visualization of security results. The implementation also allows different machine learning approaches to be examined and compared within the developed security framework.

The main objective of the proposed work is to develop an intelligent and scalable security mechanism capable of identifying abnormal behavior and predicting potential vulnerabilities in distributed cloud environments. By integrating anomaly detection with supervised classification, the framework aims to provide a more adaptive security approach while reducing dependence on manual analysis and predefined security rules. The proposed system therefore provides a foundation for developing proactive cloud security mechanisms that can adapt to changing security conditions and emerging threats. [4]

II. LITERATURE SURVEY

Cloud security has become an important research area as organizations increasingly depend on distributed cloud infrastructures for storing information, running applications, and delivering services. The growing complexity of these environments has encouraged researchers to investigate different approaches for detecting attacks, identifying vulnerabilities, and improving the overall security of cloud systems.

Early approaches to cloud security mainly relied on conventional mechanisms such as firewalls, access-control policies, encryption, and signature-based intrusion detection systems. These techniques provide protection by identifying activities that match predefined security rules or previously known attack patterns. However, their dependence on predefined signatures limits their ability to recognize new or previously unknown threats. In dynamic cloud environments, where workloads, users, and resources can change frequently, this limitation can affect the effectiveness of traditional security mechanisms. [1]

To address these limitations, researchers began applying machine learning techniques to cloud-based intrusion detection and security monitoring. Supervised learning algorithms such as Decision Trees, Support Vector Machines, Random Forest, and Naive Bayes have been investigated for classifying network traffic and distinguishing normal activities from malicious behavior. These approaches provide a data-driven alternative to fixed security rules and can improve automated threat classification. However, supervised approaches generally depend on labeled training data, which can be difficult to obtain and maintain when new attack patterns continuously emerge. [2]

Another direction of research focuses on unsupervised machine learning for identifying abnormal behavior. Techniques such as K-means clustering, Isolation Forest, and Autoencoders can analyze data without requiring every security event to be manually labeled. Among these approaches, Autoencoders are useful for learning representations of data and identifying observations that differ from learned patterns. This makes them suitable for anomaly detection and for situations where labeled examples of emerging threats may be limited. However, anomaly-based approaches can also produce false positives when legitimate activities appear significantly different from previously observed behavior. [3]

The development of deep learning-based security models has further expanded research in this area. Models such as Convolutional Neural Networks, Recurrent Neural Networks, and Long Short-Term Memory networks have been explored for analyzing complex security information, including large-scale logs and sequential network activity. These approaches can capture complex patterns and behavioral relationships that may be difficult to identify using simpler machine learning techniques. However, their use can involve higher computational requirements and longer training times, which may create challenges when they are deployed in resource-sensitive or large-scale cloud environments. [4]

Recent research has also highlighted the role of ensemble and boosting algorithms, particularly XGBoost, in cybersecurity applications. XGBoost combines multiple decision-tree-based learners through a gradient-boosting process and is capable of handling structured and complex datasets efficiently.

Its classification capabilities make it suitable for identifying different categories of security events and vulnerabilities. The approach can also provide efficient model training and prediction, making it relevant for security applications that require timely analysis of large datasets. [5]

Machine learning has additionally been investigated for vulnerability prediction rather than only detecting attacks after they occur. These approaches analyze information such as historical vulnerability records, software characteristics, system configurations, and other security-related features to identify potential weaknesses. Vulnerability prediction can support a proactive security strategy by allowing organizations to identify areas of concern before they are exploited. Nevertheless, challenges such as imbalanced security datasets, limited training examples for rare vulnerabilities, and difficulties in interpreting model predictions remain important considerations. [6]

More recent approaches have therefore moved toward hybrid machine learning frameworks that combine different learning techniques rather than relying on a single model.

A hybrid approach can use the strengths of different algorithms at different stages of the security-analysis process. In the context of this work, the combination of Autoencoder-based anomaly detection and XGBoost-based classification provides a two-stage mechanism.

The Autoencoder is intended to identify abnormal patterns, while XGBoost provides classification of the identified

security-related patterns. This combination forms the foundation of the proposed framework for vulnerability prediction and threat detection in distributed cloud environments.

Overall, the reviewed approaches show that machine learning can provide useful capabilities for improving cloud security, particularly in anomaly detection, threat classification, and vulnerability prediction. At the same time, important challenges remain in areas such as data quality, false-positive reduction, scalability, real-time monitoring, adaptability to evolving attacks, and model performance. These limitations provide the motivation for developing an integrated machine learning framework that combines anomaly detection and classification to support proactive security monitoring in distributed cloud systems.

III. SYSTEM ANALYSIS

1. Related Work

Existing cloud security systems mainly rely on firewalls, access-control policies, encryption, rule-based intrusion detection systems, and periodic vulnerability scanning. These mechanisms identify threats using predefined rules and known attack signatures. [1]

Some systems also use machine learning algorithms such as Decision Trees, Support Vector Machines (SVM), Random Forest, and Naive Bayes to classify network activities as normal or malicious. Unsupervised techniques such as clustering, Isolation Forest, and Autoencoders have also been explored for identifying abnormal behavior without requiring fully labeled data. [2], [3]

The existing system considered in this work includes machine-learning-based analysis using Decision Tree and Random Forest models. Security data is preprocessed, divided into training and testing sets, and evaluated using accuracy, precision, recall, F1-score, and confidence to assess model performance.

Overall, the existing approach provides a foundation for automated cloud-threat detection but primarily focuses on detecting and classifying security events rather than integrating anomaly detection with proactive vulnerability prediction. [4]

2. Proposed Approach

The proposed approach introduces a machine learning-based cloud security framework designed to support proactive vulnerability detection in distributed systems. Unlike

conventional security mechanisms that mainly depend on predefined rules and known attack patterns, the proposed framework analyzes security-related data to identify abnormal behavior and potential vulnerabilities. [1]

The system collects different types of cloud security data, including system logs, network traffic, user activities, and historical vulnerability records. The collected data is first cleaned and preprocessed to handle missing, irrelevant, or inconsistent information. Relevant security features are then extracted and provided to the machine learning models for further analysis.

A hybrid Autoencoder and XGBoost approach is used as the core of the proposed framework. The Autoencoder performs anomaly detection by identifying unusual patterns within the processed data.

The detected patterns are then analyzed by XGBoost, which performs classification of security events and potential vulnerabilities. This combination allows the system to use both anomaly detection and supervised classification capabilities within a single security workflow. [2]

The framework is designed to support continuous monitoring of the cloud environment. When suspicious behavior or a potential vulnerability is identified, the system generates an alert and can recommend preventive actions such as access restrictions, configuration updates, or traffic filtering. This helps reduce dependence on manual security analysis and supports faster identification of potential risks.

The proposed system also includes a performance evaluation and model-update mechanism. Model performance can be evaluated using measures such as accuracy, precision, recall, F1-score, false-positive rate, and detection time.

New security data can subsequently be incorporated into the learning process to improve the system's ability to respond to changing attack patterns. [3]

Overall, the proposed Autoencoder–XGBoost framework aims to provide an adaptive, scalable, and proactive approach to cloud security, with emphasis on early vulnerability identification, automated threat analysis, and continuous security monitoring in distributed cloud environments. [4]

IV. SYSTEM DESIGN

System Architecture

Below diagram depicts the whole system architecture.

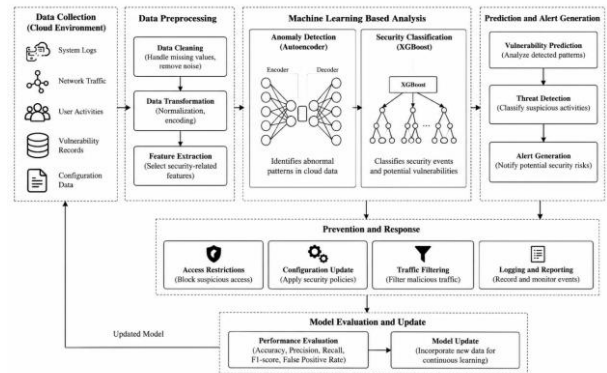


Fig 1. Methodology followed for proposed model

V. SYSTEM IMPLEMENTATION

1. Modules

The proposed cloud security framework consists of several interconnected modules. Each module performs a specific function, from collecting security data to detecting vulnerabilities, generating alerts, and updating the machine learning models. This modular structure allows the system to process security information systematically and support continuous cloud security monitoring [1].

2. Data Collection Module

This module collects security-related information from the cloud environment, including system logs, network traffic, user activities, and historical vulnerability records. The collected data provides the foundation for identifying security patterns and potential threats [1].

3. Data Preprocessing Module

The collected data may contain missing values, noise, duplicate records, or irrelevant information. This module cleans and prepares the data by handling missing values, removing unnecessary information, and transforming the data into a suitable format for machine learning analysis [2].

4. Feature Extraction Module

This module identifies important security-related features from the processed data. Features such as access frequency, traffic patterns, error rates, and configuration changes are extracted to

represent meaningful characteristics of cloud activity. Effective feature selection helps reduce data complexity and improves model performance [3].

5. Machine Learning Model Training Module

This module trains the machine learning models using the extracted security features. The proposed framework combines an Autoencoder for anomaly detection with XGBoost for security classification. The Autoencoder learns patterns within the data and identifies abnormal behavior, while XGBoost classifies security events and potential vulnerabilities [4].

6. Vulnerability Prediction and Detection Module

The trained models analyze security data to identify abnormal activities and potential vulnerabilities. This module acts as the main detection component of the framework and supports proactive identification of security risks before they can be exploited [5].

7. Alert and Prevention Module

When suspicious activity or a potential vulnerability is detected, this module generates an alert and supports appropriate preventive actions. These actions may include access restrictions, configuration updates, and traffic filtering, helping reduce response time and manual monitoring requirements [6].

8. Performance Evaluation Module

This module evaluates the performance of the machine learning models using metrics such as accuracy, precision, recall, F1-score, false-positive rate, and detection time. These measurements help determine the effectiveness and reliability of the proposed security framework [7].

9. Model Update and Learning Module

The final module updates the machine learning models using newly available security data. Continuous updating allows the system to adapt to changing attack patterns and maintain its detection capability as the cloud environment evolves [8].

Together, these modules form an integrated workflow in which security data is collected, processed, analyzed, classified, and converted into actionable security information. The modular design also provides a foundation for extending the framework with additional detection techniques and automated response mechanisms in future work.

VI. RESULTS AND DISCUSSION

The proposed machine learning-based cloud security framework was evaluated using security-related data processed

through different machine learning models. The experimental workflow included data preprocessing, feature preparation, model training, prediction, and performance evaluation. The proposed framework combines an Autoencoder for anomaly detection with XGBoost for security classification. The models were evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC to examine their classification performance [1]–[5].

1. Performance Comparison of Classification Models

The developed system provides a comparison of Decision Tree, SVM, KNN, Random Forest, and the proposed Hybrid Autoencoder + XGBoost model. The models are evaluated using Accuracy, Precision, Recall, F1-Score, and ROC-AUC, providing a broader assessment of their classification performance [2]–[5].

Figure 2 presents the comparative performance of the implemented machine learning models.

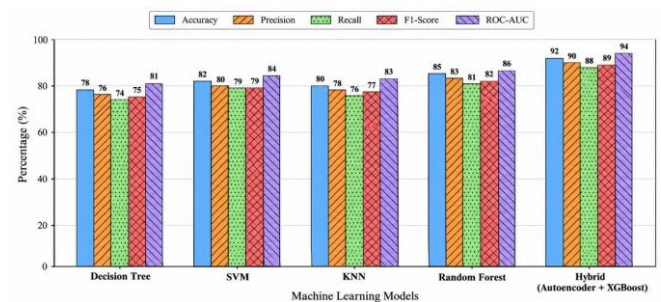


Fig. 2. Performance comparison of Decision Tree, SVM, KNN, Random Forest, and Hybrid Autoencoder + XGBoost models.

The Decision Tree model provides a baseline classification approach, while Random Forest uses multiple decision trees to provide an ensemble-based classification mechanism. The implementation evaluates the models using the same testing procedure and calculates the required performance measures. For the Random Forest implementation, the recorded evaluation values are 85% accuracy, 83% precision, 81% recall, 82% F1-score, and 86% ROC-AUC. These values provide a reference point for the comparison of the implemented models. The proposed Hybrid Autoencoder + XGBoost approach follows a two-stage learning process. The Autoencoder processes the input data to learn meaningful representations and identify abnormal patterns. The encoded representation is then provided to the XGBoost classifier for security-event classification. This combination integrates anomaly detection

and supervised classification within the proposed security framework [4], [5].

2. ROC-AUC and Model Analysis

The Receiver Operating Characteristic (ROC) curve is used to examine the classification capability of the implemented models at different decision thresholds. The corresponding Area Under the Curve (AUC) provides an additional measure of the ability of a model to distinguish between different classes [3], [5].

Figure 3 presents the ROC analysis of the evaluated machine learning models.

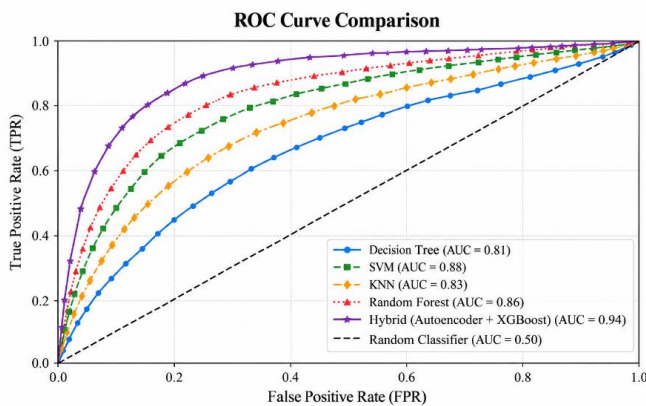


Fig. 3. ROC curve analysis of the evaluated machine learning models.

The Random Forest implementation records an ROC-AUC value of 86%, together with its accuracy, precision, recall, and F1-score values. The proposed Hybrid Autoencoder + XGBoost model is also evaluated using ROC-AUC, allowing its classification behavior to be examined alongside the conventional machine learning models.

The comparison framework includes Decision Tree, SVM, KNN, Random Forest, and Hybrid Autoencoder + XGBoost. Using multiple evaluation measures provides a more complete analysis of model behavior rather than relying only on accuracy. The ROC-AUC analysis further supports the evaluation of classification performance across different decision thresholds.

VII. CONCLUSION AND FUTURE WORK

This project focused on improving cloud security by using machine learning techniques to detect abnormal activities and predict potential vulnerabilities in distributed cloud

environments. The proposed framework combines an Autoencoder for anomaly detection with XGBoost for security classification, enabling the system to analyze security-related information such as network traffic, system logs, user activities, and historical vulnerability records. The Autoencoder identifies unusual patterns, while XGBoost classifies security-related activities. The framework also supports continuous monitoring, threat detection, alert generation, and proactive security analysis through a Django-based web application.

In future work, the framework can be enhanced by integrating advanced deep learning models such as LSTM, CNN, and Transformer-based architectures to improve the detection of complex and evolving attack patterns. Real-time data-streaming technologies can be incorporated for faster monitoring, while automated response mechanisms can support actions such as access restriction, traffic filtering, configuration updates, and isolation of compromised resources. Training the models on larger and more diverse datasets can further improve detection of rare vulnerabilities and reduce false positives. The system can also be extended to multi-cloud and hybrid-cloud environments to improve its scalability and practical applicability.

REFERENCES

1. A. B. Nassif, M. Abu Talib, Q. Nasir, H. Albadani, and M. B. Dakalbab, "Machine learning for cloud security: A systematic review," *IEEE Access*, vol. 9, pp. 20717–20735, 2021.
2. Y. I. Alzoubi, A. Mishra, and A. E. Topcu, "Research trends in deep learning and machine learning for cloud computing security," *Artificial Intelligence Review*, vol. 57, art. no. 132, 2024.
3. W. Xu, "Intrusion Detection Systems Based on Logarithmic Autoencoder and XGBoost," *Security and Communication Networks*, 2022, Art. no. 9068724.
4. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
5. M. V. M. Hari Vinayak and J. T., "A hybrid model for detecting intrusions using stacked autoencoders and extreme gradient boosting," *Computers & Security*, vol. 150, art. no. 104212, 2025.
6. D. Truong, D. Tran, L. Nguyen, H. Mac, H. A. Tran, and T. Bui, "Detecting web attacks using stacked denoising autoencoder and ensemble learning methods," in *Proceedings of the 10th International Symposium on*

Information and Communication Technology, 2019, pp. 267–272.

7. L. Dhanya and R. Chitra, “A novel autoencoder based feature independent GA optimised XGBoost classifier for IoMT malware detection,” *Expert Systems with Applications*, vol. 237, 2024, Art. no. 121618.
8. S. Gökstorp, S. Katsikeas, and P. Johnson, “Machine Learning for Cybersecurity: A Comprehensive Literature Review,” *ACM Computing Surveys*, vol. 58, no. 9, art. no. 243, 2026.