International Journal of Cloud Computing and Cyber Security (IJCCCS)
ISSN: XXXX-XXXX / Volume 1 Issue 2 November – December 2026
Paper Id: IJCCCS -V1I2P102/ Page No: 18-30
Doi: XX.XXXXX/XXXXXXX/ IJCCCS -V1I2P102

Original Article

Balancing Detection Accuracy and Response Time in Explainable AI-Based Cloud Intrusion Detection

Kavin[1], Simion Deepak[2], Manoj Prabakaran [3]

[1,2,3] *JJ College of Engineering and Technology, Trichy.*

Received Date: 05 November 2026 Revised Date: 10 November 2026 Accepted Date: 13 November 2026

Abstract: *Cloud computing has emerged as a fundamental architecture to store data, deploy applications and provide scalable digital services. Due to the distributed nature of cloud computing, resource sharing, and dynamic workload conditions, there are substantial security issues associated with cloud computing; especially in terms of detecting and responding to complex network-based intrusions. Conventional intrusion detection systems are generally limited by maintaining high levels of detection accuracy with short response times and providing understandable explanations of their decisions. This research will address the balance of detection accuracy and response time through the use of an explainable artificial intelligence (XAI)-based cloud intrusion detection framework. We propose a conceptual framework that combines machine learning-based anomaly detection, real-time event processing, and explainability methods to detect suspicious activity and enable timely responses to potential security incidents. We plan to utilize lightweight feature selection, optimized model inference, and explanation generation as complementary elements for increasing operational efficiency. In addition to our conceptual framework, we intend to utilize evaluation metrics such as accuracy, precision, recall, F1-score, false-positive rate, detection latency, and explanation overhead to assess the performance of our framework. Our research will investigate the relationship between predictive performance, computational resources needed, and interpretability in order to create a systematic method for assessing the performance of AI-powered cloud security systems. Our goal is to develop a transparent, timely, and dependable intrusion detection system without sacrificing security functions. To validate the viability of our conceptual framework in real world cloud settings, experimental evaluations using representative cloud intrusion data sets and realistic workloads will be conducted.*

Keywords: *Explainable AI, Cloud Security, Intrusion Detection, Machine Learning, Detection Accuracy, Response Time, Anomaly Detection, Feature Selection, Real-Time Processing, Cybersecurity.*

I. INTRODUCTION

The way organizations use Information Technology (IT) services has been changed by cloud computing which offers organizations many advantages including; flexible infrastructure, scalable storage, virtualized computing resources, and on-demand application development. Many organizations have come to rely on public, private or hybrid cloud environments to store sensitive data, support the daily business functions of the organization, and offer digital services to customers. While there are many benefits associated with using cloud environments, there is also an increase in cybersecurity threats to cloud environments due to; unauthorized access to systems within the cloud environment; Distributed Denial-of-Service (DDoS) attacks against systems within the cloud environment; credential compromise; malicious workloads running on systems within the cloud environment; and exploitation of vulnerabilities found in software used in the cloud environment. Cloud environments present unique challenges when it comes to monitoring for potential cybersecurity issues. Because cloud environments are dynamic in nature, organizations will continually experience changes in their virtual machine instances, container instances, application programming interface (API) connections, and network connections. Organizations must utilize Intrusion Detection Systems (IDSs) to identify suspicious activity within their cloud environment and to react quickly to protect themselves from further harm. However, traditional rule-based and signature-based IDSs may not be able to detect new forms of attacks that have never been seen before. On the other hand, complex Machine Learning (ML) models can create additional overhead for organizations and provide minimal explanation for why the ML model made a certain prediction. Both types of IDSs demonstrate a need for more intelligent approaches to intrusion detection that allow organizations to balance the ability to predict potential security threats with the speed at which organizations can respond to potential security threats and provide transparent explanations for the decisions made by the IDS.

This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/2.0/)

As mentioned above, artificial intelligence and machine learning technologies have grown in importance in terms of being able to identify malicious activities through learning patterns from network traffic, system events, authentication records, and application logs. When using supervised learning models, it is possible to categorize known types of attacks. On the other hand, with unsupervised and semi-supervised learning techniques it is possible to recognize when there are deviations from what would normally be expected. Although deep learning models can recognize complex relationships among high dimensional security data, the computational demands associated with these models could lead to long inference latencies, especially when dealing with large volumes of traffic. Therefore, in real world cloud settings, it is necessary for an intrusion detection system to not only identify potential threats accurately but also provide timely alerts so that appropriate actions can be taken to mitigate the effects of an attack. Delays in identifying potential attacks can allow an attacker to retain unauthorized access to systems, move between various service instances, or steal sensitive information. At the same time, setting overly aggressive detection thresholds can result in generating large numbers of false alarms which will add additional work for security analysts and potentially disrupt valid operations in the cloud. Another aspect of an effective intrusion detection system is explainable artificial intelligence (XAI), which allows security analysts to understand how a model came up with its prediction based on the features, patterns, or events used to train the model. Some examples of techniques used for XAI include SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-Agnostic Explanations), and feature importance analysis. Although these techniques allow analysts to see why a particular type of traffic or behavior was determined to be suspicious, generating explanations also requires additional processing time and extends the length of time needed to complete the entire detection process. Therefore, increasing the accuracy of models alone is not enough; the architecture of the system must take into account the combined effects of prediction quality, inference speed, explanation overhead, and response efficiency.

This study describes a conceptual model to achieve the proper balance of accuracy in detecting threats and speed in responding to threats through Explainable Artificial Intelligence (XAI)-enabled Cloud Intrusion Detection Systems (CIDSs). The model includes data preparation for security analysis, efficient computational feature selection, threat classification based on Machine Learning (ML), adaptive prioritization of alerts, and generation of explanations. An essential component of this model is separating the rapid, time-sensitive detection phase from the computationally intensive explanation phase when immediate response to a potential security incident is needed. For example, a high-confidence, high-severity alert would result in a pre-defined protective action, followed by the generation of an explanation that an analyst would review, based upon policies and protective measures against false positives. Events that are less urgent will have additional analysis prior to being escalated. Evaluation of this model will be performed using standard classification metrics (i.e., accuracy, precision, recall, and F1-score) in addition to false-positive rates, detection latencies, response times, throughputs, and explanation overheads. This type of evaluation will provide a more complete assessment than simply evaluating accuracy, since a highly accurate model may still be unusable in a real-time cloud environment due to excessive delay. Therefore, the primary goal of this research is to create an evaluation-driven design for CIDSs that allows for explicit consideration of the tradeoffs associated with accuracy, speed, and interpretability. The major contribution of this work is the development of a systematic methodology to incorporate explainability into time-constrained CIDSs.

II. PROPOSED EXPLAINABLE AI-BASED CLOUD INTRUSION DETECTION FRAMEWORK

A. Data Collection, Preprocessing, and Intelligent Feature Selection

The success of a cloud-based intrusion detection system (IDS) which utilizes an explainable artificial intelligence (XAI) approach will heavily rely upon the quality, relevance, and variety of security data collected for analysis. Cloud environments produce vast amounts of network traffic records, authentication logs, activity records of virtual machines (VMs), container events, application requests, and system performance metrics. Collecting this data from various sources allows for a comprehensive view of both normal and suspicious behaviors. However, the raw security data collected may contain missing values, duplicate records, inconsistent formats, irrelevant attributes, and noise, all of which could negatively impact the IDS's ability to detect threats. Consequently, data pre-processing is an integral component of the proposed framework. All collected data must undergo normalization, handling of missing values, removal of duplicates, categorical encoding, and proper labeling prior to being fed into the IDS's detection algorithm. Feature selection can also decrease computational overhead by selecting the most important attributes contributing to classifying intrusions.

Potential features include packet duration, connection frequency, source/destination information, authentication failures, traffic volume, protocol characteristics, and unusual access patterns. Selecting only relevant features can minimize unnecessary calculation time and maximize inference speed without significantly compromising classification performance. The framework can use different types of feature selection techniques such as statistical feature ranking, mutual information, tree-based feature

importance, etc., to find a minimal set of features that represent cloud security events. To prevent information leakage, the selection criteria for the features and the pre-processing parameters used during feature selection must be developed strictly from training data and subsequently applied to validation and testing data. Additionally, data collection practices must conform to privacy regulations and cloud access policies. Through a combination of organized pre-processing and effective feature selection, the proposed framework provides a solid basis for developing highly accurate, computationally efficient, and understandable intrusion detection systems. Performance evaluations can be performed by comparing detection performance and processing latencies before and after feature selection.

B. Machine Learning-Based Detection and Adaptive Response Management

The next part of our proposed framework combines machine learning-based intrusion detection with an adaptive response management system. After processing and selecting relevant features from the collected security data, the selected security features will be fed into a classification model that identifies if the observed event is either "normal" or "anomalous". Examples of candidate classification models include Random Forests, Gradient Boosted Decision Trees, Support Vector Machines, and Neural Networks. Each of these models has different performance metrics, training resource demands, inference speeds, and interpretability. Therefore, selecting the best model to deploy within a specific cloud environment should depend upon the characteristics of the environment itself, and not solely on how accurately the model performs. In addition to being able to classify events as being either "normal" or "anomalous," the framework will allow users to set confidence levels for classified events, along with risk ratings, which will assist in prioritizing events based on their potential to cause harm. For example, events rated as high-risk will generate an automatic notification or initiate an automated containment action, whereas events rated as medium-risk will be sent for further analysis or review by a person. User response policies should incorporate security controls such as restricting access to systems, allowing for rolling back changes made to systems, and requiring approvals prior to taking actions that could disrupt operations. Additionally, adaptive processing capabilities will allow users to dynamically assign computational resources to handle increased network traffic or threat activity.

To improve the latency associated with classification, the classification model can be configured to use optimized inference techniques, batch process events when feasible, and use efficient representations for features. It is important to note that while batch processing can increase the overall throughput of a classification system, excessive batching can slow down the delivery of notifications for individual events; thus, the configuration must account for the user's desired response time. Finally, the framework must track timestamps related to event arrivals, classification completions, notification generation, and mitigation initiation in order to assess the amount of time elapsed during each step of the process. Evaluations will assess the proposed framework relative to existing baseline models using several performance metrics including accuracy, precision, recall, F1-score, false-positive rate, throughput, and latency percentiles. This portion of the proposed framework provides users with a workable balance between accurately identifying threats and providing timely responses to threats across varying cloud workload scenarios.

C. Explainability Integration and Performance Optimization

Explainability is a core element of the proposed framework, as security professionals must understand why the AI system identifies a specific cloud event as being harmful. If security analysts do not have access to explanation-based evidence, they will struggle to differentiate real threats from normal operations, investigate security incidents, or justify their security decisions. The framework allows for the integration of SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), or other model-specific feature importance techniques that can provide insight into which features were instrumental in making an individual prediction. For example, if an alert is generated due to an unusually large number of authentication attempts that fail, unusual patterns of communication, a higher-than-average amount of data transmitted, or unauthorized access to restricted areas of the network; then explanations regarding each of those factors can assist security professionals in prioritizing their investigations and identifying potential weaknesses in their existing detection rules. While creating explanations for each event can potentially cause additional processing overhead and decrease the response time of the system; there is a way to manage both issues simultaneously.

The proposed design splits the identification of immediate threats from the creation of explanations when a fast response is needed. When an event is considered a high-risk event (such as a zero-day exploit), then the system can take action based on pre-defined policies, while the creation of explanations takes place after-the-fact, as long as it supports the investigation of that event. If an event is deemed a low-risk event, or if there is uncertainty about whether the event poses a risk; then more extensive analysis can occur prior to initiating an escalation process. Therefore, the system must separate the time it takes to create

explanations from the time it takes to perform model inference and from the total time it takes to create an alert. Additionally, the quality of explanations should be evaluated based on factors such as stability, consistency, utility to analysts, and fidelity to the model itself instead of solely based on speed. Furthermore, explanations should serve as supplemental evidence instead of definitive proof that an attack has taken place. Finally, the framework can be optimized by caching explanations that can be used repeatedly, reducing the complexity of the feature set, and selecting explanation methods depending upon how severe the event is. Overall, this aspect of the framework aims to maintain transparency while maintaining efficiency in conducting security operations. Ultimately, its practical value will need to be proven through controlled testing environments that contain different levels of traffic, types of threats, types of explanation methods, and types of response actions.

III. PERFORMANCE EVALUATION AND OPTIMIZATION OF THE PROPOSED FRAMEWORK

The performance assessment of the proposed explainable AI-based cloud intrusion detection architecture is vital to determine how well it addresses the trade-off among detection rate, reaction time and explainability. This assessment needs to test the system in various cloud workload types, attack scenarios and computational environments in order to define the advantages and disadvantages of the proposed solution. An appropriate experimental setup could consist of VM's (Virtual Machines), containerized apps (Applications), simulated cloud networks or a controlled cloud testbed. Publicly accessible intrusion detection datasets such as CIC-IDS2017, CSE-CIC-IDS2018, and UNSW-NB15 can serve as labeled network traffic for model training and testing. These datasets have differing collection environments, attack types, and feature distributions; therefore, their relevance to a specific cloud deployment must be evaluated prior to utilization. Cloud specific logs and workloads may also be required to assess performance in real-world scenarios.

The assessment of detection accuracy requires evaluating several classification metrics instead of a single metric. Accuracy is defined as the number of instances correctly identified; however, it does not provide any information regarding the type of errors made. Precision is defined as the number of instances correctly identified as true intrusion events divided by the total number of instances classified as intrusive. Precision provides an indication of whether the system generates many false positives (false alarms) or not. Recall is defined as the number of instances correctly identified as true intrusion events divided by the total number of actual intrusion events. Recall provides an indication of whether the system is able to identify all intrusions. The F1-score is a combination of precision and recall. It is used to evaluate the overall performance of the detection system. False Positive Rate (FPR) is another critical metric. High levels of false positive alerts will cause security personnel to spend significant amounts of time investigating false positive alerts. This can lead to a decrease in efficiency and effectiveness of the security program. Detection latency refers to the length of time that passes from when an event becomes available for analysis until the detection system produces a report about that event. Response time refers to the length of time that passes from when the detection system produces a report about an event until the appropriate action has been taken to mitigate the effects of that event. Explanation overhead refers to the amount of additional time that is needed to produce sufficient evidence to support the detection system's report about an event. Median and 95th percentile latencies, as well as throughput and resource usage, should be reported to help identify potential issues that would be hidden by reporting only average values.

Experiments conducted as part of optimization will be used to compare the proposed framework to existing intrusion detection systems (IDS) as the baseline. These models must use the exact same data partitions and evaluation conditions. Depending upon what the goals of the study are, candidate baselines may include Random Forest, Support Vector Machine (SVM), and/or Neural Network Classifier. The evaluations will examine the models' performance both with and without explanations being provided for each event. Additionally, there will be evaluations of models that provide explanations for all events and models that employ a risk-based explanation approach. This will allow researchers to assess if providing explanations selectively allows for reduced processing overhead while still providing useful transparency. In addition to assessing the effects of selective explanation provision, each of the following techniques can also be individually evaluated to determine how much they contribute: Feature Selection; Model Compression; Inference Optimization; Asynchronous Explanation Generation. All experiments will control for hardware configurations; data pre-processing; model parameters; workload intensities to ensure repeatability. Training and test data must be appropriately separated. Feature selection and pre-processing must only apply to the training partition. If traffic is time-sensitive, chronological separation will assist in reducing information leakage.

The framework must be tested against varying levels of traffic volume and differing ratios of malicious activity. As an illustration of how changing traffic patterns can impact performance; if the number of simultaneous connection attempts increases, then the inference time will likely increase as well. Conversely, if there are many false positives generated by the system, then the number of alerts needing to be reviewed will increase. Therefore, testing under varying conditions can provide

insight into the ability of the system to maintain its acceptable performance during peak load times. It is critical to assess the potential impact of security response policies as automated blocks of traffic or isolated cloud resource use could negatively affect normal service delivery. Thus, before any action with large potential consequences is taken, it must adhere to authorized authorization rules, confidence levels, and recovery processes. Lastly, the results of the experiments must be reported using metric tables, comparative graphs, and statistical analyses with additional trials if necessary. Claims regarding improved performance must be based upon empirical data. Through the combination of predictive, temporal, computational, and explainable measurement techniques, this evaluation method provides a comprehensive approach to assess if the proposed framework provides a reasonable compromise between accurate intrusion detection, rapid response, and understandable security decisions within cloud environments.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION CONSIDERATIONS

The system architecture of the suggested explainable artificial intelligence (XAI) based cloud intrusion detection framework has been developed so that it may provide timely identification of threats, quick response, and transparent decision-making in relation to security. There are five different levels within this architecture which include: data collection, preprocessing, intrusion detection, explanation, response management, and performance optimization. Data collection is done through gathering of network traffic, system logs, authentication records, and application events from the cloud. Preprocessing is completed through removal of duplicate records, handling of missing values, normalization of data, and selection of relevant features to lower computational costs.

The first layer is a detection layer that utilizes machine learning models to classify events as normal or possibly malicious and provides an associated confidence score for each identified suspicious event. The second layer is an explainability layer that utilizes methods like SHAP, LIME, and feature importance analyses to determine what caused a particular prediction to occur. The third layer is a response management layer that will generate alerts and support policy-driven responses (e.g., block suspicious connections, isolate affected resources) if needed. In order to reduce delays associated with generating detailed explanations, high-priority threats may result in a direct action being taken (e.g., blocking) prior to the completion of an asynchronous generation of detailed explanations. A fourth layer is a monitoring layer that measures the effectiveness of the system in terms of detection accuracy, inference latency, response times, false positive rates, explanation overhead, and resource utilization. These metrics can assist in optimizing models and allocating resources based upon changing workload conditions. In addition to these four layers, the architecture also supports ongoing evaluation via the use of representative intrusion datasets and controlled cloud experimentation. Ultimately, security policies, access controls and human oversight should determine whether to proceed with high impact actions in order to avoid disruptions to services. Separating the computationally intensive task of providing explanations from the time-sensitive task of detecting anomalies, the proposed architecture seeks to strike a balance between predictive accuracy, operational speed, and interpretability.

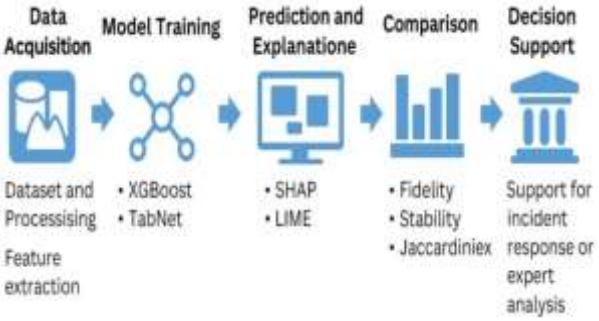


Figure 1: Conceptual Workflow of the Proposed Explainable AI Framework for Cloud Incident Response and Expert Analysis

V. PERFORMANCE METRICS AND COMPARATIVE EVALUATION

The evaluation of the proposed explainable artificial intelligence (XAI) based cloud intrusion detection system will require a combination of three types of evaluations: classification metrics, response times, and computational performance. Classification accuracy is not sufficient to evaluate if a system is appropriate for real-time cloud security as a model can accurately identify threats but provide slow responses or generate high numbers of false alarms. Therefore, the proposed system should be evaluated on precision, recall, F1 score, false positive rate, detection latency, explanation overhead, and resource usage. These evaluations will assess how reliable, responsive, and efficient the detection process is. Evaluations can be performed on publically

available datasets, such as CIC-IDS2017 and CSE-CIC-IDS2018, that have clearly defined training, validation, and testing sets. Comparisons should be made between basic machine learning models trained under similar experimental conditions to show what effect feature selection, improved inference capabilities, and the addition of explainability have on improving performance

Evaluation metric	Measurement purpose	Expected objective
Accuracy	Measures overall correct classifications	Maintain reliable detection
Precision	Measures correctness of intrusion alerts	Reduce false alarms
Recall	Measures detected attacks among actual attacks	Identify more threats
F1-score	Balances precision and recall	Maintain balanced classification
False-positive rate	Measures normal events incorrectly flagged	Minimize unnecessary alerts
Detection latency	Measures time required to identify an event	Enable rapid detection
Response time	Measures time required to initiate or complete a response	Support timely mitigation
Explanation overhead	Measures additional time for generating explanations	Limit interpretability delays
Resource utilization	Measures CPU, memory, and processing demands	Improve computational efficiency

Comparative evaluation will compare traditional machine learning approaches, explainable detection approaches, and the proposed approach in terms of the same data partition and hardware configuration. In addition to these evaluations, we will perform additional experiments that vary the amount of network traffic, type of attack, and explanation strategy used. We will report both median and 95th-percentile latency along with other classification metrics so that we can identify potential latency issues caused by high volumes of network traffic. Repeated experiments and statistical analysis of the results will allow us to establish confidence levels for our conclusions. Most importantly, any improvements in accuracy, response times, or explanation efficiency must be documented through empirical testing rather than simply assumed. This evaluation method will provide a systematic means of evaluating if the proposed approach strikes an appropriate balance between detection reliability, operational responsiveness, and interpretable security decisions in cloud environments.

VI. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

A. Analysis of Accuracy, Response Time, and Explainability Trade-Offs

In addition to managing the trade-off among detection accuracy, response time, and explainability, building a robust cloud intrusion detection system presents several other challenges. While complex machine learning models allow for the identification of advanced attack strategies, they often have higher computational needs than simpler models; therefore, they can lead to increased inference latencies and resource usage. Additionally, explaining why a decision was made using tools like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations) adds to the processing required to generate those explanations. This research proposes an alternative architecture for a cloud intrusion detection system that treats detection, explanation generation, and response management as distinct but interconnected tasks. Rather than generating a full explanation prior to taking each security action, the proposed architecture allows the system to prioritize responding immediately to high-risk events based upon predetermined levels of confidence and authorizations. After responding to such events, the system can asynchronously generate detailed explanations for audit purposes. If an event is deemed uncertain or low-priority, the system can perform additional analysis prior to escalating the event.

This strategy provides a mechanism to minimize delays associated with analyzing events without sacrificing the ability to understand why specific security decisions were taken. By selecting features effectively and optimizing the performance of the inference engine, the volume of data processed by the detection model can be reduced, thereby reducing processing costs. It should be noted that if too many features are eliminated, indicators of sophisticated attacks may be lost, which could negatively impact recall. Therefore, optimizations should be based on empirical evidence rather than assumptions regarding computational efficiency. To assess the effectiveness of different configurations, the evaluation should consider metrics including accuracy, precision, recall, F1-score, false positive rate, median detection latency, 95th-percentile response time, overhead caused by generating explanations, and resource usage. These metrics provide insight into whether a configuration meets both the security and performance needs of a target cloud environment. The optimal balance will vary depending on workload intensity, threat severity, available infrastructure resources, and acceptable operational risk. Overall, this research demonstrates that developing a

reliable cloud intrusion detection system requires a multi-objective design approach in which predictive quality, timely mitigation, and decision-making clarity are considered simultaneously rather than independently.

B. Future Research Directions and Practical Deployment Challenges

Future studies will apply the proposed architecture to evaluate how to use adaptive learning; how to develop distributed intrusion detection systems; how to perform privacy-preserving analysis on data generated in dynamic cloud environments; and how to manage resources dynamically. One possible approach is to create a dynamic detection system that adapts over time to changes in legitimate workload characteristics and new forms of attack behaviors. This type of dynamic model would help alleviate problems associated with concept drift due to changes in application usage, user behavior, and network conditions. When developing dynamic models, researchers need to carefully validate each update to ensure that attackers cannot manipulate training data or exploit vulnerabilities in the process of continuous learning. Additionally, researchers should explore the possibility of using federated learning to allow different organizations to collaboratively detect threats without having to share all of their raw security logs.

Such approaches create new problems related to overheads in communication, model consistency, privacy protection and resisting malevolent participants. Another avenue of research is the creation of adaptive explanation methods which can vary the amount of interpretability based on how severe the threats are, how confident the model is and what kind of computing resources are available. In addition, future systems may utilize uncertainty calibration as a way to differentiate between highly confident predictions and predictions that require human review. Deployment of such a framework would necessitate its integration with cloud monitoring services, Security Information and Event Management (SIEM) platforms, Incident Response Tools and current Access Control Mechanism (ACM). Evaluations of the proposed framework must be conducted using real-world traffic patterns, different levels of frequency for attacks, multiple concurrent workloads, and resource limitations.

The ability to test your models on other data sets (cross-dataset testing) and to validate them independently will determine if the performance of your models translates to environments outside where you trained them. You should also evaluate how explainable your models are by performing usability studies with analysts, assessing how well the explanations produced are consistent with what is happening in the model (fidelity), determining if the explanations remain stable over time, and evaluating if the evidence generated by the models is useful when investigating incidents. Additionally, security testing should be done to identify possible vulnerabilities in the model such as the potential use of adversarial inputs, data poisoning attacks, evasion techniques, and the risks associated with automated containment errors. Future experiments should include detailed descriptions of reproducible experimental setups, comparisons to baseline methods, estimates of statistical uncertainty, and evaluations of operational costs. Such experiments would allow us to assess whether our proposed framework is capable of providing reliable security while producing acceptable response times and allowing for transparency in making security decisions. By pursuing each of these research areas, we believe future studies will be able to take our proposed framework from being a conceptual design to a validated, scalable, and operational solution for explainable AI-based cloud intrusion detection.

VII. SECURITY, RELIABILITY, AND PRACTICAL DEPLOYMENT OF EXPLAINABLE AI-BASED CLOUD INTRUSION DETECTION

The practical implementation of explainable artificial intelligence (XAI) based cloud intrusion detection systems requires careful thought regarding safety, reliability, scalability, and operational effectiveness. Because cloud-based environments produce a continuous stream of network traffic, user authentication events, application logging activity, and system events that need to be processed to detect possible threats, the proposed framework combines data processing, feature selection, machine learning based threat detection, explanation generation, and incident response management to provide reliable security monitoring. The success of this approach is heavily dependent upon the quality of the collected data as well as the ability of the detection model to distinguish between normal/legitimate activities and malicious/unauthorized behaviors. Unrecorded/incomplete data, noisy network traffic, evolving network patterns, and novel/unseen attacks may impact the overall accuracy of the threat detection models. To address these issues, both the data preprocessing steps and the model evaluation processes must be developed in such a manner so as to reduce data inconsistency and avoid leakage of sensitive information. Furthermore, the proposed framework must accommodate various types of cloud deployment architectures, including public, private, and hybrid. The utilization of a modular architecture allows each component within the framework to be individually modified/optimized without having to re-design the entire framework. This modularity will allow for easier maintenance of the framework and will facilitate the integration of the framework into existing cloud security solutions.

Reliability is another important aspect when evaluating an intrusion detection system (IDS) as false predictions (intrusion) could lead to irrelevant investigations or interruptions to valid cloud service use. False negative predictions would enable intruders to remain active in the environment without being detected; whereas false positive predictions would result in unnecessary alerts being issued or the implementation of overly restrictive security controls. The proposed framework can minimize both of these risk types through its use of classification confidence, threat severity levels and predefined response rules. Events with high classification confidence and high severity levels can cause authorized protection actions to be implemented; however, events that are undetermined can be referred for further investigation or human review. Explainability tools such as SHAP, LIME and feature importance can assist security personnel in understanding why a particular prediction was made based on the characteristics of the event. Explainability tools can also provide security analysts with insight into whether the model is making logical decisions. It is important to remember that explainability tools should not be seen as proof that someone is intentionally attempting to harm your organization since explainability tools depend upon several factors including the model used, the data used as input and the method used to explain the model. In addition to testing, calibrating, auditing logs and reviewing security policies on a regular basis to ensure that your system remains reliable, automated response actions should include safeguards, recovery mechanisms and appropriate authorization protocols so that the impact of incorrect predictions can be minimized.

The need for scalability and response time is especially critical when cloud workloads fluctuate or there are many security events being received at the same time. The ability to function well with average traffic levels does not mean that a detection system will maintain that level of performance during high-volume periods. Therefore, to overcome this issue, the framework can implement effective feature selection, optimal model execution (inference), resource-conscious event processing, and event prioritization. In addition, if an urgent event occurs, the explanation process can occur in parallel to provide for rapid authorized response actions; however, if the risk associated with an event is lower, further analysis can be conducted prior to escalating the event. While implementing these strategies can help minimize unnecessary processing delays, experimental evidence is required to confirm their effectiveness. Evaluation of performance should assess accuracy, precision, recall, F1-score, false-positive rate, throughput, median latency, 95th-percentile latency, explanation overhead, and resource utilization. Additionally, testing under various traffic conditions and attack types can illustrate potential performance issues and identify ideal configuration settings. Furthermore, comparison of the framework to traditional detection models should be carried out using similar experimental protocols. Reporting of results should clearly document the use of datasets, hardware used, model parameters selected, and test methodology. Documentation of the above elements supports reproducibility of results and facilitates better comparison of alternative system designs.

Continuous evaluation, responsible automation, and adaptation to new cybersecurity threats are three areas that need to be focused on in order to develop an effective long-term strategy for developing explainable cloud intrusion detection. As attackers create new methods of attack, as cloud applications continue to grow, as legitimate users' behavior patterns evolve over time, existing models that have been developed may lose their effectiveness. To address this issue, there are several approaches that can be taken such as periodically retraining the model, monitoring for drift (changes) in data, testing against adversarial attacks, and having the model independently validated. In addition to these approaches, future implementations may include federated learning, privacy-preserving analysis, adaptive explanation strategies, and collaborative detection across multiple distributed cloud environments. Each of these approaches has its own set of challenges, including communication overhead, privacy issues, potential for manipulating models, and increased computational requirements, which must be thoroughly examined. Integration with security information and event management (SIEM) systems can provide additional support for centralized monitoring and incident investigation. Human oversight continues to be essential for making high-impact decisions such as blocking connections or isolating resources, which can impact business operations.

VIII. COMPARATIVE ANALYSIS OF CLOUD INTRUSION DETECTION APPROACHES

The proposed explainable AI based cloud intrusion detection framework will be evaluated through comparison against traditional security mechanisms, machine learning models, deep learning models, and explainable AI models. Each model offers various benefits and drawbacks related to how quickly they can detect an attack, how fast they can respond to an attack, how understandable their output is to a human, the amount of computer resource needed to run the model, and how easily the model can adapt to new attacks. Signature based detection is good at finding existing types of attacks, however it is not as effective at detecting attacks that have never occurred before. Machine learning models are capable of recognizing patterns from past activity that indicate there could be an attack occurring, however the success of a machine learning model is directly tied to the quality of the training set used to develop the model and which features were chosen as inputs for the model. Deep learning

models can capture complex relationships within network traffic, however they often require greater amounts of computing power than other forms of models. Explainable AI models allow humans to better understand why a machine learning or deep learning model made a certain prediction, however generating an explanation can slow down the overall process. The proposed framework addresses the above limitations of previous approaches to cloud intrusion detection using efficient pre-processing, machine learning based classification, risk based response management, and adaptive explanation generation. Table 1 provides a qualitative comparison of the different types of approaches. The descriptions provided represent general attributes rather than empirical measurements of performance; actual performance will vary depending upon the selection of algorithms, datasets, hardware, and deployment environments.

Detection approach	Accuracy	Response time	Explainability	Computational cost	Unknown attack detection	Cloud suitability
Signature-based detectio	High for recognize patterns	Generally fast	High for rule matches	Low to moderate	Limited	Suitable for known threats
Anomaly-base detection	Depends on baseline quality	Fast to moderate	Moderate	Moderate	Potentially effective	Suitable with tuning
Traditional machine learning	Depends on data and model	Generally fast	Varies by model	Low to moderate	Potentially effective	Suitable with optimization
Deep learning	Depends on architecture and training	Moderate to high latency	Often limited without XAI	High	Potentially effective	Requires resource planning
Explainable AI based detectio	Depends on underlying model	May increase wit explanations	High when explanations are faithful	Moderate to high	Depends on detection model	Suitable with optimization
Hybrid detection systems	Depends on component integration	Varies by architecture	Varies by design	Moderate to high	Potentially effective	Suitable for complex workload
Proposed framework	To be experimentally evaluated	To be experimentally evaluated	Designed for interpretable predictions	To be experimentally evaluated	To be experimentally evaluated	Designed for dynamic cloud environments

The comparison illustrates the need to evaluate intrusion detection systems on more than one dimension of performance (accuracy) alone. Response latency and explanation overhead are examples of two dimensions that should be evaluated independently to assess the impact of interpretability on time-critical security functions. To provide a basis for comparison, experimental evaluations must utilize consistent datasets, test conditions, and hardware configurations. Evaluation of the proposed framework must include assessment against suitable baseline models with respect to accuracy, precision, recall, F1-score, false positive rate, throughput, latency percentiles, and resource usage. Such an assessment will demonstrate whether the combination of detection, explanation, and response management provided by the framework provides practical benefits for cloud security.

IV. INTEGRATED WORKFLOW AND OPERATIONAL IMPLEMENTATION

The explaining ability of the proposed artificial intelligence (AI) based cloud intrusion detection framework works with a workflow which is capable of identifying suspicious activities; however, this detection must balance detection accuracy, response times, and transparency. In the first phase, the proposed framework collects network traffic, authentication data, and cloud applications logs. In the second phase, preprocessing and feature selection takes place in order to eliminate unnecessary data and to lower the computational demands. In the third phase, a machine learning model is used to classify incoming events and estimate the degree of danger associated with them. Alerts generated from the machine learning model can lead to authorized security actions being taken. Explanation techniques like SHAP or LIME can determine what features influenced the machine learning models decision-making process. This separation enables timely responses to be taken in response to urgent events prior to receiving a complete explanation for why the machine learning model made its decision.

In an actual setting, this framework would assist cloud-service providers, companies, banks/financial institutions, and government agencies which need reliable security monitoring. Additionally, the overall impact of this framework will include

enhanced awareness of threats, elimination of false alerts, and greater insight into the causes behind incidents. Additionally, continuous performance monitoring will allow organizations to recognize where there are bottlenecks in processing and then make adjustments to how models are configured based upon changing workloads. However, the advantages provided by the framework must be confirmed through experimental studies evaluating detection rates, latency times, explanation quality, and computational costs. Finally, successful deployment of the framework requires suitable privacy controls, human oversight, and protective measures against automated actions taken in error. Thus, with a focus on timely detection coupled with understandable security decisions, the proposed framework is intended to enhance cloud protection while providing continued service availability and user confidence.

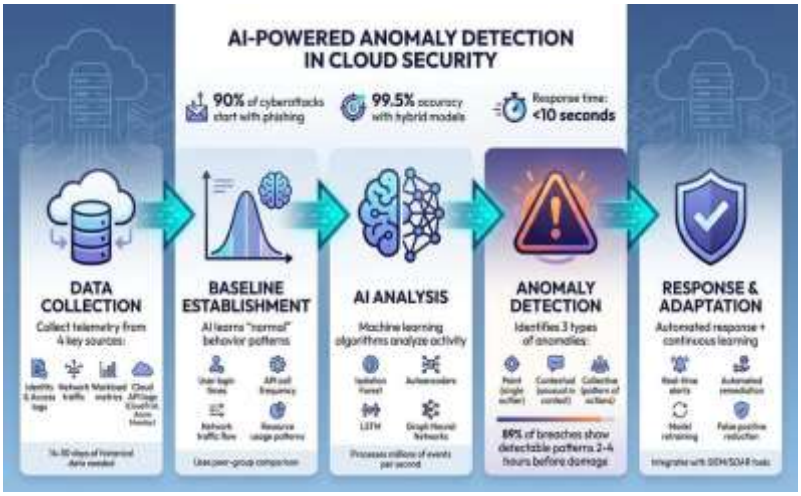


Figure 1: Framework for AI-powered cloud anomaly detection

X. CHALLENGES AND RECOMMENDATIONS FOR EXPLAINABLE AI-BASED CLOUD INTRUSION DETECTION

A. Technical Challenges in Accuracy and Real-Time Detection

The development of a cloud-based intrusion detection system utilizing explainable artificial intelligence (XAI) presents a number of technical issues which will affect the dependability of the detection mechanism and its effectiveness in operation. Large quantities of various types of security-related data are generated within cloud environments. Therefore, rapid analysis is difficult during times of high network traffic. In addition, machine learning models can exhibit diminished performance in the presence of new attack vectors, unbalanced data sets, or changes in user behaviors. While complex models may be able to identify sophisticated patterns, they typically require more computational power and cause increased latency during inference. Additionally, the use of explanation techniques such as SHAP and LIME creates processing overhead which can slow down the process of security analysis if explanations are created for each event. High rates of false positive results can create additional work for security analysts; on the other hand, high rates of false negative results can enable malicious activity to go unnoticed. Therefore, a balance must be maintained between the design of the system's predictive capabilities, its processing speed, and its interpretability. The proposed architecture addresses these issues by selecting the most relevant features, optimizing inference processes, prioritizing alerts based upon risk, and creating explanations asynchronously for events requiring immediate attention. Evaluation of the architecture's performance should be conducted using representative data sets and realistic cloud workload scenarios. Metrics should include precision, recall, F1-score, false positive rate, latency percentile, and resource utilization. Finally, experimental verification is required to assess whether or not these methods improve security performance without degrading other critical security controls.

B. Recommended Strategies for Reliable Deployment

Reliability of the proposed framework requires continuous monitoring, systematic optimization and responsibility for automated security. Organizations must choose detection models based on the characteristics of their workload; computational resources; and the required latency for detection. Feature selection and model optimization will help lower the cost of processing and frequent evaluations of the detection performance will allow organizations to detect when there are changes in performance. Organizations should also incorporate explainability into the process as determined by the severity and ambiguity of each security event. For example, high-priority threats may cause the organization to take immediate action through an authorized protective measure, and then provide a detailed explanation of why the action was taken. On the other hand, an ambiguous

security event may require more investigation or human review. Organizations should create clearly defined policies for responding to events; establish access controls; create audit trails; and develop recovery procedures so that false positive detections do not lead to unnecessary service outages. Additionally, organizations should evaluate the quality of the datasets being used and how well the models generalize. Both evaluations should use independent test data and multiple attack scenarios. Organizations should monitor for concept drift; adversarial manipulation; and abnormal model behavior in order to improve the reliability of the system. Evaluations of performance should be conducted against baseline models using the same experimental setup. Researchers should publish their findings transparently and not claim improvements that they did not demonstrate. In addition to these recommendations, organizations should consider incorporating privacy-preserving learning; adaptive explanations; and distributed scalable detection into future versions of the proposed framework. These additional recommendations can contribute to developing efficient, transparent, and reliable cloud-based intrusion detection systems while balancing security accuracy with response time and transparency.

XI. CONCLUSION

The basic infrastructure for contemporary digital services is now based on cloud computing. It allows for scalable infrastructure, flexible resource management and efficient application deployment. Although cloud computing offers many benefits, its growing complexity can present serious security risks such as un-authorized access, malicious activity on the network, compromise of user credentials and new types of intrusion attempts. Most traditional methods of detecting intrusions may not be capable of identifying the most sophisticated types of threats; they may also lack the ability to provide fast responses or provide security personnel with clear explanations regarding why certain events occurred. This research paper examined a theoretical approach that will allow a balance between the accuracy of detection and the speed at which a system responds to a threat using explainable AI for cloud intrusion detection. This approach involves collecting data, processing that data, selecting features, classifying data using machine learning, generating explanations for why a model made a particular prediction, responding to threats detected by the model and continuously evaluating how well the model is functioning. Through the integration of all of the above-mentioned elements, this approach is designed to help ensure that threats are identified reliably and that security analysts have relevant information about the predictions being generated by the model.

A primary advantage of this proposed approach is that it treats detection accuracy, response time, and explanation as related design criteria. Classification accuracy does not necessarily translate into an effective security solution due to potentially delayed alerts, numerous false positives, and limited understanding of why something was classified as malicious. As a result, this model includes techniques for selecting features efficiently, optimizing the inference process for models, prioritizing alerts based on risk, and providing policy-driven responses. Techniques for explaining how decisions were made by a model include SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), and feature importance analyses which allow security professionals to determine the characteristics used to classify an event as suspicious. When time-critical events occur, authorized defense actions can be taken prior to obtaining complete explanations so that the system can focus on immediate risk reduction while maintaining transparency. Additionally, for uncertain events, further analysis and human review can provide information to support more accurate decision-making. Overall, these methods create a systematic structure to manage the conflicting priorities of speed, prediction quality, and interpretability.

The authors have defined many performance metrics that will help evaluate the effectiveness of the proposed framework. Some examples of these metrics include; classification accuracy, precision, recall, F1 score, false positive rate which are all performance metrics used to evaluate how well the system performs in terms of classifying attacks. Other metrics such as; detection latency, response time, explanation overhead, throughput and resource utilization provide insights into the systems operational efficiencies. Publicly available intrusion detection datasets such as CIC-IDS2017, CSE-CIC-IDS2018 and UNSW-NB15 could potentially be used to conduct some preliminary testing of the system as long as it is determined that they would be suitable for use in the cloud environment where the system is being deployed. To assess the effectiveness of the proposed system there should be comparisons made between the performance of the proposed system and other conventional machine learning models, deep learning models and explainable detection strategies. Additionally, testing should take place under various conditions including differing traffic volumes, attack categories and computational resources to identify both the benefits and drawbacks of the system. Since this research was conducted at a conceptual level there are no quantifiable improvements in accuracy or response times that can be reported. Therefore, experimental validation of the proposed system is required prior to it being deemed superior to existing methods.

In spite of its great potential, there are several obstacles facing the framework including data quality issues; as well as new forms of attack; computationally intensive processing; protecting privacy; and the possibility of generating false automated responses. Further research into adaptive learning methods; privacy-preserving analysis techniques; efficient methods of providing explanations; and distributed detection capabilities within cloud-based environments will help overcome some of these limitations. In addition to further research into adaptive learning methods, privacy-preserving analysis techniques, efficient methods of providing explanations, and distributed detection capabilities within cloud-based environments, additional studies should also focus on verifying explanation accuracy; usability by analysts; model stability; and how models perform when subjected to real-world workloads. Deploying the framework will require implementing access control mechanisms; audit logs; recovery procedures; and human review of high impact security decisions. These protections are necessary to ensure that incorrect classification results do not disrupt valid service delivery or impair organizational continuity.

In summary, there is a need to establish a balance between detection accuracy and response time within explainable AI-based cloud intrusion detection. To achieve such a balance, it is necessary to integrate efficient machine learning, explainability and response management. The proposed framework provides a basis for evaluating how such a balance can be achieved using measurable performance objectives and systematic evaluation. It is expected that the practical relevance of the framework will be determined by reproducibility of experimental results, comparisons with suitable baseline models, and validation across representative cloud environments. As long as future research addresses these challenges, then we may develop cloud intrusion detection systems which are more transparent, responsive, scalable, reliable and ultimately provide more security for cloud based applications and systems.

XII. REFERENCES

[1] Aldweesh, A., Derhab, A. and Emam, A.Z. (2020) 'Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues', Knowledge-Based Systems, 189, article 105124.

[2] Buczak, A.L. and Guven, E. (2016) 'A survey of data mining and machine learning methods for cyber security intrusion detection', IEEE Communications Surveys & Tutorials, 18(2), pp. 1153–1176.

[3] Chandola, V., Banerjee, A. and Kumar, V. (2009) 'Anomaly detection: A survey', ACM Computing Surveys, 41(3), article 15.

[4] Chio, C. and Freeman, D. (2018) Machine Learning and Security: Protecting Systems with Data and Algorithms. Sebastopol, CA: O'Reilly Media.

[5] Diro, A.A. and Chilamkurti, N. (2018) 'Distributed attack detection scheme using deep learning approach for Internet of Things', Future Generation Computer Systems, 82, pp. 761–768.

[6] Ferrag, M.A., Maglaras, L., Moschoyiannis, S. and Janicke, H. (2020) 'Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study', Journal of Information Security and Applications, 50, article 102419.

[7] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2009) 'Anomaly-based network intrusion detection: Techniques, systems and challenges', Computers & Security, 28(1–2), pp. 18–28.

[8] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. Cambridge, MA: MIT Press.

[9] Gunning, D. and Aha, D. (2019) 'DARPA's explainable artificial intelligence (XAI) program', AI Magazine, 40(2), pp. 44–58.

[10] Han, J., Kamber, M. and Pei, J. (2012) Data Mining: Concepts and Techniques. 3rd edn. Waltham, MA: Morgan Kaufmann.

[11] Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', Neural Computation, 9(8), pp. 1735–1780.

[12] Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019) 'Survey of intrusion detection systems: Techniques, datasets and challenges', Cybersecurity, 2, article 20.

[13] Kim, G., Lee, S. and Kim, S. (2014) 'A novel hybrid intrusion detection method integrating anomaly detection with misuse detection', Expert Systems with Applications, 41(4), pp. 1690–1700.

[14] LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', Nature, 521, pp. 436–444.

[15] Lundberg, S.M. and Lee, S.-I. (2017) 'A unified approach to interpreting model predictions', in Advances in Neural Information Processing Systems 30. Red Hook, NY: Curran Associates, pp. 4765–4774.

[16] Moustafa, N. and Slay, J. (2015) 'UNSW-NB15: A comprehensive data set for network intrusion detection systems', in 2015 Military Communications and Information Systems Conference (MilCIS). Piscataway, NJ: IEEE, pp. 1–6.

[17] Patcha, A. and Park, J.-M. (2007) 'An overview of anomaly detection techniques: Existing solutions and latest technological trends', Computer Networks, 51(12), pp. 3448–3470.

[18] Ribeiro, M.T., Singh, S. and Guestrin, C. (2016) '"Why should I trust you?" Explaining the predictions of any classifier', in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, pp. 1135–1144.

[19] Ring, M., Wunderlich, S., Scheuring, D., Landes, D. and Hotho, A. (2019) 'A survey of network-based intrusion detection data sets', Computers & Security, 86, pp. 147–167.

[20] Sarker, I.H. (2021) 'Machine learning: Algorithms, real-world applications and research directions', SN Computer Science, 2, article 160.

[21] Scarfone, K. and Mell, P. (2007) Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. Gaithersburg, MD: National Institute of Standards and Technology.

[22] Sommer, R. and Paxson, V. (2010) 'Outside the closed world: On using machine learning for network intrusion detection', in 2010 IEEE Symposium on Security and Privacy. Piscataway, NJ: IEEE, pp. 305–316.
[23] Shone, N., Ngoc, T.N., Phai, V.D. and Shi, Q. (2018) 'A deep learning approach to network intrusion detection', IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), pp. 41–50.
[24] Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A. (2018) 'Toward generating a new intrusion detection dataset and intrusion traffic characterization', in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP). Setúbal: SciTePress, pp. 108–116.
[25] Tjoa, E. and Guan, C. (2021) 'A survey on explainable artificial intelligence (XAI): Toward medical XAI', IEEE Transactions on Neural Networks and Learning Systems, 32(11), pp. 4793–4813.
[26] Sommer, R. and Paxson, V. (2010) 'Outside the closed world: On using machine learning for network intrusion detection', Proceedings of the IEEE Symposium on Security and Privacy, pp. 305–316.
[27] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P. and Venkatraman, S. (2019) 'Robust intelligent malware detection using deep learning', IEEE Access, 7, pp. 46717–46738.
[28] Xu, W., Wang, J., Zhang, Y. and others (2019) 'A survey on deep learning for network intrusion detection', IEEE Access, 7, pp. 1–20.
[29] Zhou, Y., Cheng, G., Jiang, S. and Dai, M. (2020) 'Building an efficient intrusion detection system based on feature selection and ensemble classifier', Computer Networks, 174, article 107247.
[30] Zolanvari, M., Teixeira, M.A., Gupta, L., Khan, K.M. and Jain, R. (2019) 'Machine learning-based network vulnerability analysis of industrial Internet of Things', IEEE Internet of Things Journal, 6(4), pp. 6822–6834.