Journal of Theoretical and Applied Information Technology
15th March 2026. Vol.104. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THREATDETECTAI: AN AI-POWERED ZERO TRUST FRAMEWORK FOR REAL-TIME THREAT DETECTION AND CRYPTOGRAPHIC RE-VALIDATION IN CLOUD COMPUTING

D.VENKATESWARLU[1*], Dr. B. SATEESH KUMAR[2]

[1]Department of CSE, Scholar of JNTUH, Hyderabad, India.
[2]Professor, HoD, Computer Science & Engineering, JNTUH University College of Engineering, Jagtial, India.
dvenkates739@gmail.com, sateeshbkumar@gmail.com
*Corresponding Author

ABSTRACT

Cloud computing environments operate at a vast scale and support heterogeneous user activity, but ensuring integration is a key and significant security challenge. While a few attempts have been made to improve integrity verification, most existing integrity verification methodologies rely on a static cryptographic proof that involves an untrusted element or a rule-based validator. This independent anomaly detection method fails to detect minute insider threats, evolutionary attack patterns, or multi-modal correlations (e.g., log, network flow, and cryptographic traces). Although Zero Trust architectures employ many of these principles, the continuous validation of user-level trust and the prevention of information modification in a multi-tenant infrastructure create blind spots and limitations. This calls for an adaptive, intelligence-driven framework that integrates threat detection elements with cryptographic proof-checking. In this work, we introduce ZeroTrustAI, the first proof framework for AI-integrity, which combines a cryptographic verification engine with a hybrid deep learning algorithm called ThreatDetectAI. ThreatDetectAI utilises a CNN–BiLSTM–Attention model to identify spatial–temporal patterns in user behaviour and produce dynamic threat scores. Suspicious events are checked by Proofcryptnet (hash, digital signature, Merkle root). Such a system is integrated, providing a continuous, zero-trust verification loop with risk-based, adaptive access control. To evaluate the proposed approach, we conduct extensive experiments with many different data types, including cloud logs, network telemetry, and cryptographic metadata and show that it achieves 97.8% accuracy, 97.2% precision, 98.1% recall, and 0.986 AUC, which significantly outperformed the classical (RF, SVM, XGBoost) and deep learning baselines (CNN-only, LSTM-only, Transformer). The new version, with built-in cryptographic validation, achieved an 18% decrease in false positives when deployed for high-volume activity scenarios. We propose a framework that satisfies the scalability, explainability, and audit-friendliness criteria for unobtrusive, preemptive confidence assurance at the context-user level, thereby facilitating continuous authentication, adaptive authorisation, and strong, defence-grade countermeasures against adverse events in agile cloud settings. These results showcase the practical use of ZeroTrustAI to bolster protection in modern enterprise cloud environments.

Keywords - *Zero Trust Security, Deep Learning Threat Detection, Cryptographic Integrity Proof, Cloud Security Framework, User Behaviour Analytics*

1. INTRODUCTION

Cloud computing today has become the foundation upon which most modern digital ecosystems are built, enabling elastic resource provisioning, distributed collaboration, and large-scale data processing across multi-tenant environments. But the growing decentralisation of access, fluid privilege assignments, and millions of user actions pose a real threat to data integrity and trust. Current cloud security mechanisms are mainly based on static cryptographic verification, static access control lists, or threshold-based anomaly detection, which fail to detect complex insider actions, correlated attack behaviours, and obfuscated modifications in heterogeneous cloud log data. Recent works emphasise that traditional integrity validation models are becoming less adequate with changing scenarios and adversarial conditions, and researchers show the need for diverse vigilance to implement new, real-time, adaptive, intelligence-driven verification approaches [1], [2]. At the same time, deep learning-based threat detection has demonstrated

the ability to capture temporal dependencies and behavioural deviations; most existing work targets network intrusion and relatively coarse-grained anomaly detection, which does not provide user-level integrity proof in Zero Trust [76] settings.

Motivated by these gaps, the current research aims to develop a unified, AI-assisted integrity verification system that goes beyond static checks. This is not possible with most other traditional methods, as they cannot continuously score the threat, map sequence cloud logs to cryptographic evidence, and dynamically re-authenticate and trigger retribution. Hence, you need a holistic framework to monitor, detect, and validate user-level actions using omni intelligence. We intend to implement a deep learning-driven threat detection module, which shall be deployed alongside the cryptographic verification method to provide continuous, tamper-proof, and proactive proof of integrity in Zero Trust cloud environments.

The proposed work spans several key innovations: (1) a novel hybrid CNN–BiLSTM–Attention architecture for learning spatio–temporal features in user activity sequences, (2) an adaptive threat scoring mechanism closely coupled with cryptographic endorsement, and (3) a Zero Trust-oriented access control engine that enforces dynamic authentication in response to real-time risk indicators. Unlike the state-of-the-art literature, which treats detection and verification as separate processes, we design a framework that couples them in a closed feedback loop, with automatic re-validation via cryptographic proofs whenever an event is flagged as suspicious.

The research contributions are fourfold. In this work, we introduce a new dataset to aid in detecting user-level threats with high granularity using multimodal data over a long time scale. Then, a ThreatDetectAI module is built to capture minor deviations in behaviour by combining deep temporal modelling and attention-weighted feature selection. Third, ProofCryptNet guarantees the binding of verifiable evidence by validating suspicious events using a combination of secure hash functions, digital signatures, and Merkle root comparisons. Finally, a Zero Trust access control layer that provides real-time privilege changes to support a proactive, contextual defence pipeline. This contribution offers a scalable, explainable, and auditable integrity proof mechanism for cloud platforms.

The rest of the paper is organised as follows: Section 2 presents related work on cloud integrity checking, user behaviour analytics (UBA), and deep learning-based threat detection. The proposed methodology is described in Section 3, which provides an overview of the model's architecture and components, as well as the algorithmic workflow. In: Section 4: Experimental Results: Dataset Statistics: Training Behaviour: Comparative Performance: Robustness Evaluation; Results are presented in Section 5, which also interprets the findings and lays out limitations of the study. Section 6 concludes the work and provides directions for future research.

2. RELATED WORK

Cloud computing is one of the fastest-evolving tech trends that has changed digital infrastructure and created large, flexible, scalable ecosystems for industries and governments. At the same time, that transformation has also created unprecedented security challenges that are difficult to address adequately with traditional, perimeter-based models. To address these problems, many studies have proposed combining artificial intelligence (AI) with Zero Trust Architectures (ZTA) to replace static trust assessment mechanisms and enable real-time threat detection, cryptographic revalidation, and adaptive access control, thereby enabling monitoring and protection of resources [1]. Here, we review relevant literature across five contrasting thematic areas and describe contributions, limitations, and research gaps related to the proposed ThreatDetectAI framework.

2.1 AI-Powered Zero Trust and Cloud Security

Zero Trust emerged as a central tenet: there are no trusted identities, implicit or explicit, internal or external. We can further the ZTA with Artificial Intelligence, which offers continuous authentication, automated anomaly detection, intelligent access control, and real-time analysis to improve the overall security system. Parisa et al. ZTA based on AI enables far more to be done → showed that ZTA based on AI provides a

solution and an order of magnitude increase in protection to retail cloud environments via dynamic and perpetual user verification and monitoring. Similarly, Ofili et al. Many federal cloud infrastructures must comply with Cybersecurity and Infrastructure Security Agency (CISA) standards, and this paper [2] discusses improving compliance through AI and threat intelligence.

In the ZTA strategy experiment, we show that AI-based deception methods with micro-segmentation can not only help offset foreign cyber spying and insider attacks, but also refocus targeted and directed defence against both [3, 4]. Mubeen [5] applied these ephemeral key encryption principles, along with AES-GCM and ECDH, for cloud-based AI chat apps and supplemented them with continuous AI-based anomaly detection. With reference to Celeste and Michael [6], AI, Zero Trust, and cloud-native solutions are moving ever closer toward a common approach to securing hybrid and multi-cloud-based environments.

Recent improvements are looking to include an intelligence capability based on context on ZTA. In [7], Shoaib and Muhammad proposed integrating AI and ZTA for dynamic threat mitigation, whereas in [8], Dash highlighted the importance of ZTA for protecting LLMs hosted in the cloud. Ejeofobiri et al. Adaptive cybersecurity architectures were presented in [9], explaining how ZTA, coupled with self-learning AI systems, acts as a dynamic entity that learns and adapts as the threat landscape evolves. Further work by Kancherla [10] used this work to introduce Adaptive Threat Prevention (ATP), a mechanism that prevents zero-day exploits in a cloud system.

Zichen [11] and Akbar and Zafer [12] have taken the idea of AI-based threat detection in ZTA environments a step further by proposing machine-learning-based solutions for continuous, real-time monitoring. Zero Trust solutions have been embraced in healthcare and industrial IoT environments as information technology (IT) and operational technology (OT) systems converge to safeguard sensitive data. Zakhmi et al. [42] and Laghari et al. AI-enabled ZTA-based frameworks can effectively protect vital medical and industrial networks [43]. Nzeako and Shittu [49] and Khalid et al. Enterprise deployment approaches for adaptive ZTA were explored in [50], and [51] again considered AI–ZTA integration to future-proof cybersecurity.

Lilhore et al. have built SmartTrust, a hybrid deep learning framework. The vulnerability enables adversaries to integrate ZTA into inline cloud threat detection to deliver stronger ZTA protections across different systems [52]. A consolidated set of best practices for large-scale application of ZTA through a systematic review of ZTA principles and real-world application by Mensah [53]. Together, these studies highlight the need for AI-enabled ZTA to achieve trusted, scalable, secure, and automated cyber defence [40], [41].

2.2 Insider Threat Detection and Data Leakage Protection

A unique challenge of insider threats is that someone with trust and valid access carries them out. In [13], Alzaabi and Mehmood presented a comprehensive survey of insider threat detection approaches. They advised on the necessity of context-based analysis of intrusion behavioural patterns, while NLP and deep learning are essential components. Nasir et al. proposed a deep learning framework for generating LSTM-based models to detect insider threats effectively with fewer false positives [14].

Hurst et al. Moving forward, [16] built classifiers via supervised ML to protect EHRs with near-perfect accuracy in detecting misuse in healthcare environments. Ajayi et al. Real-time monitoring to prevent financial fraud and internal data breaches in banks is another area where AI-based anomaly detection is being explored [18]. Similarly, Herrera Montano et al. [21] found that encryption and ML are enabling technologies for tackling internal data exfiltration in a recent review of data leakage prevention techniques.

2.3 Machine Learning and Deep Learning for Cloud Security

The ever-expanding complexity of the cloud environment demands more sophisticated intrusion detection and anomaly prediction. Alzoubi et al. Specific topics they reviewed include research trends in ML and DL for cloud security [15], anomaly detection [14], native security [13], and XAI [12]. El-Kassabi et al. Using deep learning, [17] devised a cloud

workflow orchestration security enforcement model that adapts protection according to existing and emerging threats.

Vadisetty et al. GANs and transformer-based generative models were proposed for real-time anomaly detection [19], thereby greatly enhancing the ability to detect cyberattacks earlier. Zhang et al. Process mining as a means to enable lifecycle monitoring of multi-cloud file security detection [20]. Gudelli [22] conducted one such comparison, comparing supervised, unsupervised, and hybrid ML approaches to detect zero-day anomalies in cloud networks.

The researchers Aldallal and Alisa [26] proposed a hybrid intrusion detection system that combined SVM and genetic algorithms, enabling optimal feature selection and improved detection accuracy. Wu et al. For instance, [31] focused on privacy-preserving ML delegations to prevent the exposure of sensitive data in cloud environments. And even more improvements have come with Frimpong et al. presenting a general-purpose privacy-preserving ML framework based on homomorphic encryption. Sheth et al.[34].passwords and biometric types, they hope to authenticate using multi-layered authentication models integrated with the blockchain. [35]. Li et al. Verifiable, privacy-preserving ML prediction schemes are proposed for edge-enhancing cyber-physical systems [36].

2.4 Cryptographic Mechanisms and Post-Quantum Security

Cloud security is still based on cryptographic validation. Kumar et al. and Singh and Saxena [25] proposed a universal multi-tenancy spray-and-pry logical system based on hybrid cryptography & ML authentication in federated clouds. Ahmed et al. Related work Li et al.[28] proposed lightweight cryptographic primitives. Lin et al. A recent work by [30] proposed a cryptographic approach based on symmetric encryption using chaotic maps (combined with ML networks) to secure physiological signals. In multi-cloud environments, Mohammad [32] surveyed encryption and access controls.

There has been growing interest in post-quantum cryptographic schemes due to the rise of threats posed by quantum computing. Darzi and Yavuz [27] investigate the combination of ML and post-quantum cryptography towards a future-proof approach. Mehmood et al. In [29], the authors surveyed many vulnerabilities affecting modern cryptographic systems and argued for the necessity of quantum-safe protocols. Quantum-resilient AI frameworks for anomaly detection have also been proposed [57] in conjunction with Karamchand [24], who discussed the possible advantages of using Quantum Machine Learning (QML) to process high-dimensional threat-encrypted data. Salam et al. A novel framework for smart manufacturing: Integrated Anomaly Detection with Zero-Knowledge Proofs to Enhance Security.

K.S. Arjunan [23] studied ML-based fraud detection methods for NoSQL databases to demonstrate their use in data breach prevention. For ZTA of cryptographic innovations, Warren and Rajuroy [54] experimentally investigated how AI can be combined with blockchain to create adaptive access controls.

2.5 Evolutionary Access Control and Identity Management

The move to more distributed cloud systems makes access control and identity management all the more critical, potentially requiring adaptive controls—AI Algorithms for Network Security Improvement in Cloud Computing (Wang and Yang [37]). Hamad et al. explored decentralised scalability for intrusion detection in IoT using federated learning [38], providing a systematic analysis of federated learning-based intrusion detection approaches, including resource-aware optimisations for data-intensive cloud systems (Liu [39]).

For instance, Magaji [44] suggested a framework for Zero Trust API security with AI-driven central authorisation, while Lamia et al. AI-driven Access Control: [45] [17] applied AI to access control for hybrid cloud totonuams. Feature-based innovation, including enhancements to authentication using AI-based identity management, was researched by Rehman and Ali [46] and Okoye [47], who used adaptive AI to address RFID/NFC authentication issues. The unsupervised self-learning AI models for behaviour-driven access control have been developed by Smith and Chikwarti [48].

We find risk-adaptive app architectures for mobile security [55]; autonomous ZT-based orchestration for critical infrastructure [56] by Olorunlana; and novel mobile phone-based network infiltrations [57]. Al-Otaibi et al. [39] pointed to AI-aided threat hunting, which Cate [59] and Privacy-preserving AI frameworks for medical sensor networks [60] discussed. Jordan Smith [58] proposed Context-aware MFA for critical infrastructure. Together, these studies highlight the need for combining AI and ZTA to develop secure, adaptive, and resilient identity and access management systems.

The literature shows remarkable advances in integrating AI, ML/DL, and cryptographic mechanisms with Zero Trust approaches to address modern cloud threats. However, there are still gaps in explainability, privacy, and integration complexity. Given the emerging challenges in cloud security, the contributions of the reviewed works, and the need to present a unified architecture such as ThreatDetectAI that offers real-time threat detection, continuous cryptographic revalidation, and adaptive access control, scalability and cloud regulatory compliance requirements are imperatively mandated.

3. MATERIALS AND METHODS

We then propose a methodology encompassing the entire design of the ZeroTrustAI framework for combining AI-powered threat detection with cryptographic integrity verification within a Zero Trust security model. It describes data preprocessing, lightweight feature extraction, hybrid DL analysis, threat scoring, and adaptive access control. Here, a contract-specific perfusion synthesis capability that delivers verifiable cloud security operations by enabling abstracted workflows that lead to well-structured algorithms and architectural components is presented.

3.1 Introduction To Zero Trust In Cloud Computing

Cloud computing revolutionised the way organisations store, process, and manage data, and the pandemic prompted one of the fastest rates of cloud adoption to date. However, this shift brings new security challenges, as the nature of the Cloud is inherently distributed and multi-tenant. Old perimeter-based security models often fail to protect sensitive data and resources, as the assumption of a trusted network slowly erodes. Cloud infrastructure spans multiple geographical regions, third-party services, and a variety of user roles, making it vulnerable to a wide array of internal and external threats. In fact, insider threats and advanced persistent threats are usually within the organisation, thus making them harder to secure against typical data breaches and Data compromises. Such complexities mandate a security posture that transcends static best practices.

The historical boundary of defence-in-depth security frameworks has been found not fit for purpose in real-world distributed systems, and Zero Trust security has now become an essential principle for their replacement. Always verify, never trust—this is the heartbeat of Zero Trust. This essentially means that no entity, within or outside the network, is trusted by default to access resources without thorough verification. Every access request is authenticated, authorised, and encrypted for its entire lifecycle. The ongoing validation of identity means that, should a hacker succeed in gaining a toehold in the network, lateral movement and unauthorised access will be difficult. We can formally express this as an access request R, a collection of policies P, and a cryptographic proof on the latter C, such that there exists a verification function V which returns true if and only if V(R, P, C) = true:

$$V(R, P, C) \rightarrow \{0,1\} \quad (1)$$

Which mean that V=1 successfully verifies and get access while V=0 get access denied. The solution evaluates each request dynamically depending on the user credentials, user behaviour patterns, device posture and circuit-level crypto validation in order to determine whether to trust it.

In cloud computing, where different components of the cloud and the services hosted on it are always communicating, Zero Trust security is paramount, as user actions must be continuously monitored, since there is a risk of confusion between machine and human activity. Unlike static methods, this model-oriented approach includes risk assessment and active choice, specific to a scene. Public and hybrid clouds enable organisations to migrate their most sensitive workloads, but these create complex

new threat vectors: insider misuse, compromised artefacts, and third-party integrations with no trust boundary. Minimisation of risks like these, on the other hand, is one of the core pillars of Zero Trust. The principle of least privilege states that a user/service should be granted only the minimum access needed to complete their task (aka the need-to-know principle). This dynamic, context-aware policy enforcement strengthens security and reduces the attack surface in multi-tenant architectures.

It must also enable you to scale your security enforcement effectiveness across the cloud. This is achieved through a concept called microsegmentation, where the network is segmented into multiple segments protected by different levels of authentication and authorisation. That way, if one segment should break, the breach cannot so easily migrate across specific areas in the system. Using a zero-trust approach, everything from actions to transactions can be distilled into a trust chain that can be validated using cryptographic technologies such as digital signatures and tree-based structures like the Merkle tree. We refer to these cryptographic proofs as immutability evidence, where the verification function, as shown in equation (1), verifies the integrity of each event recorded in the cloud ecosystem.

In this ever-evolving world of cyber threats, the cloud infrastructure we see today requires precise, proactive, and adaptive security models. Zero trust as applied with cryptographic proof and AI-augmented brings an endless cycle wherein threats are recognized in the moment, certified as being secure through cryptography, and addressed by automatic policies. This forms a modular security-tempered system that will withstand the rigours of production- and dissemination-oriented cloud environments. To this end, we couple real-time behavioural analysis with enforced cryptographic trust (see earlier blogs here and here) to ensure trust is not just presumed, but re-established as we fortify the most sensitive #cloud operations.

3.2 Need For Real-Time Threat Detection

Cloud computing environments are highly dynamic and complex, with continuous interactions from users, services, and applications. These environments operate across geographically dispersed infrastructures and are accessed by heterogeneous parties with varying degrees of confidence. The traditional/mechanistic security mechanisms, which rely on static rules, signature matching or data access policies for protection, are apparently inadequate to face ever-evolving modern cyber threats. As critical workloads are increasingly hosted in the cloud, the amount of data associated with cloud activities has exploded. Static systems struggle to promptly distinguish between normal operational behaviour and malicious activity. When you take too long to recognise a threat and respond, data exfiltration, privilege escalation, service disruption, and unauthorised access to sensitive resources can result.

As a result, real-time threat detection has become a fundamental requirement for protecting cloud infrastructure. In contrast to traditional batch or offline analysis methods, real-time detection examines an ongoing sequence of events, network traffic, and user actions. This is to catch them in action, when they are trying to target your network and minimise their chances of stealing data from it. Dynamic analysis of behavioral patterns followed by immediate response to avoid an attack from spreading is an integral part of real-time detection. Let T_d be the detection time and T_a be the attack initiation time A real-time system can only be as effective as how fast it can react to a possible attack, which is denoted by ΔT:

$$\Delta T = T_d - T_a \qquad (2)$$

For a real-time detection mechanism, its ΔT should be as small as possible, like about zero, meaning the system detects any harmful activity in almost zero time. The problem is that in cloud environments where everything is automated and resources are provisioned very quickly, this latency needs to go down, since potential threats, some of which may remain undetected, can be mitigated faster.

It also includes insider threats and zero-day exploits; thus, advanced and persistent attacks are a real pain point for cloud-native threat detection. Typically, such attacks are very sophisticated, stealthy, and camouflaged within other standard traffic patterns, thereby remaining undetected. Even internal threats are hard to detect in this case since the perpetrator is a legitimate user with a valid username & password. This is beyond the

grasp of traditional rule-based systems that use signatures to detect malware, as they fail against new and unknown attack vectors. This implies that you have to use adaptive methods that learn complex behaviour, such as deep learning, to discover behaviour, and anomaly detection to validate results.

It also means that cloud workloads are constantly changing, leading to the need for rapid, scalable detection models, while alert data is often low-volume and homogeneous, providing little scope for real-time processing. Using AI to collate billions of data points — network flows, application logs, cryptographic proofs — in context as a single whole to derive system state. This integration of three data sources enables detection that, in many cases, is more accurate and reliable than monitoring approaches in isolation. According to the verification framework presented above in Equation (1), the process of threat detection can be performed with behavioral signals along with the cryptographic evidence that proves the detected anomalies are genuine and verifiable. This, in turn, increases the reliability of detection results by preventing false positives due to corrupt or incomplete data.

Another important thing for the real-time identification to have is the ability to place instant remedial actions Control response (automatically launch workflows, when a threat is detected, to contain the incident, e.g. cyber-incident containment by isolating at-risk accounts or tightening authentication policies or requiring new cryptographic revalidation of sessions in progress and long-term datava) This creates a closed-loop feedback loop in that detection is tied to response and therefore, is a way to minimize the opportunity for the threat to evolve. And that is the automation we need — in a cloud environment handling hundreds, if not thousands, of concurrent transactions every second — for security to be truly scalable.

As cloud infrastructure grows larger and more complex, real-time threat detection represents a significant step from a reactive defence approach to a risk management approach. Organisations will be able to build robust cyber defence capabilities that withstand evolving threat dynamics in the long run by reducing the time between detection (as expressed in (2)) and then overlaying machine learning analytics with a cryptographic layer to verify those analytics. This ensures that any malicious behaviour is detected and blocked in real time, protecting sensitive data and maintaining the critical operation of cloud services.

3.3 Cryptographic Verification Basics

It is a Building block for Cryptographic proof of provable Integrity, Authenticity, and Non-repudiation of Data in Cloud computing. Cloud systems have tenants and hundreds of events fired per minute like log in requests, resource access, file uploads, privilege escalation etc and many users and services running together in a distributed system. Such events are vulnerable to tampering, incorrect modifications, and even malicious deletion without a robust verification mechanism to back them up. Our cryptographic method guarantees, in a mathematically provable manner, that no event or transaction has been altered during storage and transmission over communication lines, hence ensuring originality and authenticity.

In simple terms, cryptographic verification is a layer of trustless between cloud operations, where user actions and events generated by the system help establish the authenticity of actions taken, based on fault-free mathematical proofs. Then each event, E, is cryptographically hashed to produce an unpredictable, fixed-size mapping of the event data. The hash is basically the fingerprint of the actual data. If anyone did change E, the system will get a different hash, which means E is tampered. The process of generating the hash can be illustrated as:

$$h = SHA256(E) \qquad (3)$$

In the equation, h is the hash of the event E, and SHA-256 is a cryptographic hash algorithm that produces a 256-bit output. It is a computationally efficient and irreversible operation that is functionally impossible to reverse. You can not deterministically produce E from h, and every time an event is passed across actors in the cloud ecosystem, the hash is re-stamped and checked against the original hash to guarantee data integrity.

Digital signatures are used in verifying authenticity and non-repudiation. A digital signature enables a user or system component to sign an event using private-key cryptography.

When another entity receives the event, it can verify the signature is valid (and that it came from the expected entity) by checking against the appropriate public key. It prevents bad actors from impersonating real, live customers, and it enforces accountability among identities. Digital signatures and hashing create a supply chain model where every activity can be traced to its source with 100% certainty. Such a mechanism maps well to the Zero Trust concept of verifying every action and request at each time, as described in Equation (1) earlier.

A Merkle tree is another essential cryptographic structure for cloud security. Generally, we use it to structure and verify on a large amount of events quickly. In the Merkle tree, instead of the events themselves, only the hashes corresponding to the events in the L2 execution are treated as the leaf nodes of the tree, and pairs of leaf hashes are iteratively combined in a tree structure through hashing until a single root hash remains. This allows large datasets to be verified in batches in an efficient manner with low storage and computation overhead. It is a single hash representing everything that has happened beneath it. If any single event is changed, it will create an upward ripple effect, changing the root hash and flagging it as tampering. For example, a four-leaf Merkle tree can be simply described as:

$$R = H(H(L_1 \parallel L_2) \parallel H(L_3 \parallel L_4)) \quad (4)$$

Where L_1, L_2, L_3, L_4, are the hashes of the events, H is the hashing function, and R is the Merkle root. Note that $\parallel$ parallel denotes concatenation prior to hashing By keeping this Merkle root (in such a way) in a secure ledger or blockchain, systems become tamper-evident and transparent. Recomputing and comparing the root hash allows to immediately detecting any modification of at least one event.

A multi-layer cloud security framework further backs these cryptographic measures. Hashing ensures data integrity; digital signatures tie events to their source for authenticity and non-repudiation, and Merkle trees provide a performant means of verifying logs from large-scale datasets. Zero Trust is reflected in these three cryptographic ingredients, proving every event; trusting nothing. Now, this crypto verification does enhance security posture when coupled with real-time threat detection, since alerts and decisions are underpinned by immutable, mathematically verifiable proof. Through cryptographic proofs and further AI-integrated monitoring functionality, this solution enables the most secure cloud operations, providing unprecedented compliance and auditing to meet the high demands of today, whilst drastically reducing the attack surface to the degree of many modern cyber threats.

3.4 Detection of Threats With AI Deep Learning Models

A cloud environment continuously generates a wide range of high-velocity data, from system logs and network traffic flows to API calls and cryptographic verification events. It describes very complex, and indeed changing, data across all the links between the capability of user behaviours, interactions with the service, and access to the data. For instance, in Zero-Day (ZD) situations, machine learning (ML) methods like decision trees or rule-based approaches may not respond quickly enough to adapt to these changing environments (particularly, since the ZD attack vectors they wish to eradicate have not been experienced before). Previous approaches to malware detection relied on manually defined features and static threshold values, which cannot keep pace with a changing threat environment where attackers constantly modify their behaviours to evade detection. Hence the need for intelligent detection systems that automatically learn from raw multidimensional data, can generalise across thousands of variants of a single threat, and operate without any rules.

Deep learning provides a powerful framework for modelling complex relationships in cloud security data. Deep Learning algorithms automate the learning of feature representations from raw inputs, thereby reducing the need for handcrafted feature engineering and enhancing adaptability to new and dynamic threats. The Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and attention-based mechanisms are some of the deep learning models relate to threat detection, as they can extract spatial and temporal information.

CNNs are particularly good in extracting local spatial features from structured input data. For cloud threat detection, CNNs can learn patterns in network traffic, user activity logs, and

cryptographic proof signals. Anomalies such as these, like spurious high usage of resources or too many attempts for authentication failure, may seem like a separate local pattern in the data matrix. CNNs, using convolutional filters, can automatically identify these abnormalities and produce high-level feature maps for further processing. CNNs can localise anomalies, which makes them a perfect fit for detecting odd behaviours associated with malicious attacks.

Although CNNs are good at extracting spatial features, cloud events are fundamentally sequential, with time dependency between actions. Since these dependencies are temporal, RNNs (e.g., Bidirectional Long Short-Term Memory (Bi-LSTM) networks) are added to the detection model. Bi-LSTM networks analyse sequences of data in both forward and backward directions, enabling them to capture the context of events based on both history and future events simultaneously. Having this dual perspective is imperative to separate normal patterns from malicious sequences. For example, an individual privilege escalation may not be suspicious, but if it follows a string of failed login attempts and is followed by large data transfers, it becomes a clear sign of an attack. This temporal context is captured for detection accuracy improvement by the Bi-LSTM.

Then, above the Bi-LSTM layer, an attention mechanism is used to learn how to focus on specific events in long sequences. In cloud environments, some activity logs have thousands of events per minute to process, many of which are benign and not beneficial for security analysis. The attention mechanism assigns different weights to each event, emphasising those with greater information content, which are the determinants of recognition of suspicious behaviour. The attention weights can be written mathematically as:

$$A_t = softmax(W_a \cdot H + b_a) \quad (5)$$

In this case, A_t stands for the attention vector at time t, H is the hidden state output from the Bi-LSTM, W_a is the weight matrix, and b_a is the bias vector(indicating the preferred, level of attention to maintain). This normalisation (performed by the softmax function) is essential because you always want the weights (the strength of the null event) to sum to 1 (the model is not confused and is suppressing noise). Incorporating this mechanism adds explainability to the detection, as it shows which events contributed most to the final threat score.

The CNN-BiLSTM-Attention architecture is a composite architecture intended to leverage the best of each component. The inputs run through the CNN layer to identify spatial features, pass down to the Bi-LSTM layer to learn temporal changes, where the attention layer puts attention on highlights for decision making. This task hierarchy enables the model to process multidimensional data streams. Usually, we find cloud-based context. Finally, a dense output layer takes the attention-weighted features and outputs the behaviour's threat score, along with an indication of whether it is benign or malicious. The score in Equation 1 is then used to trigger the immediate response and the dynamic security described earlier in this paper.

It further incorporates the fundamentals of continuous learning (learning from new attacks as they unfold) by marrying deep learning models with threat detection. This means that as new threats are created, you can easily re-train (or fine-tune) the model on additional data, ensuring that it can evolve with the threat landscape. In addition, the models are generalizable across modalities, so they unify system logs, cryptographic proof results, and network traffic for a more comprehensive capture of system behaviour. The combination of these sets of data affords an operator multiple views of a threat, increasing detection accuracy and reducing false alarms.

With a Zero Trust architecture, deep learning might match every event with history — and cryptographic confirmation as well. The downstream model is able to learn the hybrid deep learning (HDL) which become the intelligent layer that helps to populate the dynamically generated attacks strategies and so helps to diligently defend the cloud infrastructures. This changing nature of the threat IEDs pose cannot be captured by traditional fixed feature extraction methods, nor can they appropriately address the temporal dimension of the problem, and the potential incorporation of advanced scalable and robust methods, which are critical for real-time threat detection system, therefore making our architecture suitable by design, as it allows automation of the robust feature extraction,

instant handling of the temporal nature of the data with a larger temporal context, and ease of interpretability through attention-based methods.

3.5 Related Datasets And Evaluation Context

As AI-based threat detection is a new approach, datasets are needed to design and evaluate AI-based threat detection framework, therefore, various datasets need to be available that could present the complexity and heterogeneity of the cloud computing environments. In the case of the aforementioned challenge, realistic cloud activity patterns and network traffic behaviours must be essential features included in such datasets, along with more traditional network logs of benign and malicious activities, for accurate training and testing of the models. Unfortunately, due to privacy and sensitivity issues as well as the dynamic natures of threats, collecting cloud security datasets is not straightforward. Depending upon public resource, synthetic data generation and laboratory environment, the appropriate datasets are chosen so as to tackle the various possible threat scenarios.

The AWS CloudTrail dataset can be regarded as one of the most common datasets used in cloud threat detection studies, which consists of comprehensive logs tracking API calls and user activities within the Amazon Web Services environments. The logging can include timestamps, identities involved in the event, and was it a read/write, as well as, outcomes (successful/failed), which is very useful for detecting unauthorized/suspicious behaviours. These logs can be analyzed to reveal the abnormal access patterns, privilege escalations and possible data tampering events [6]. Additionally, since this dataset captures interactions across many accounts and services in parallel, it enables assessment of multi-tenant scenarios as well. However, the volume and variability of the AWS CloudTrail data is appropriate for building and validating models for real-time event monitoring at a scale that matches the cloud platform.

Beyond CloudTrail logs, the CERT Insider Threat Dataset is particularly suitable for modeling insider attacks, the most challenging threat to detect in cloud systems (Gao, Armbrust, and Gokhale 1995). Insider threats tend to be dangerous and difficult to detect using standard rule-engine approaches, being at attacker activity performed by genuine users with valid credentials. It labels examples of benign and malicious insider activities from the CERT dataset, including: unauthorized data access, unusual login behaviors, and covert data exfiltration attempts Such annotated instances are required for training supervised learning models and evaluating detection performance. The proposed ThreatDetectAI system can be evaluated over the real world scenarios of insider threat by integrating this dataset and checking if the system is able to detect multiple complex behavioral patterns which indicates internal misuse.

Synthetic datasets are also employed to mimic certain attack vectors and create controlled environments useful for model scaling and evocation testing. These datasets are very handy if you want to simulate any rare threats or new threats that probably do not exist in any public datasets. For instance, these could be new zero-day attacks or patterns of distributed denial-of-service or even a coordinated effort to privilege-elect. Synthetic data generation enables the model to experience wider variety of scenarios during training making it be able to generalize to new scenarios that have not yet been seen. In addition, synthetic datasets allow for controlled experiments where factors like the number of events, attack strength, and noise levels can be varied to test system resilience to different scenarios.

All datasets are preprocessed by a predefined pipeline. First, the raw event data is filtered: All records that are not complete or where a measurement is repeated are ignored, if a measurement was registered but its source information is not available, then it can be inferred to be incorrect. The second step is to normalize the data fields in multiple households to a common schema (including user ID, time, the type of action performed, and whether the action was proven). Such a harmonization process is necessary to combine different dataset into a single model training pipeline. Lastly, the labeling is accomplished which. identifies benign and intrusive events which appears as the labels are 0 and 1 that represent the normal behavior and suspicious behavior respectively. This (labeled data) is the basis of the supervised learning and performance assessment of the detection models.

These systems need a context in which to be measured—closer to the real world of production

cloud systems than a lab environment. The datasets are subject to a rigorous and unbiased evaluation process by splitting them into a train set, validation set, and test set of incorporated datasets. As it learns associations and patterns to apply across pairs of data sources, a validation subset is applied for best hyperparameters applied and overfitting avoided when using for training. The testing set is exclusively for evaluating final performance– that is, how well the system is expected to perform on data it has never before encountered. We evaluate the performance of detection based on accuracy, precision, recall, F1-score and detection latency. Also, it is an imprint on the constantly reinforced calculated realm of the threat which mandatory deductive verification tracks.

By fusing real data, such as AWS CloudTrail and CERT Insider Threat, and also synthetic data, it builds a realistic assessment landscape that encompasses normal behavior and non-trivial attack scenarios. With this comprehensive methodology, ThreatDetectAI would be able to recognize known threats patterns and equivalently protect against evasive threats across multitudes of clouds. By leveraging the malleability of data and system evaluation methods, and infusing compactness, robustness and adaptability required properties for securely maintaining NextGen cloud-based systems architecture on the principles of Zero Trust, the proposed framework exploit the true richness and nature of data sources.

4.1 Overview OfThe Proposed ThreatDetectAI Framework

ThreatDetectAI is an integral part of the proposed system, shown in Figure 1, which provides continuous, intelligent and real-time monitoring of cloud environments. It works under the Zero Trust security model, which means no user, device or process can be trusted automatically, and continuous evaluation is required for the authentication, verification and validation of every action taken. Unlike traditional perimeter-based defenses that are static, ThreatDetectAI uses dynamics behavioral pattern analysis, the outcome of cryptographic proof verification, and the context of communication to detect insider threats, tampering, and abnormal activities in real-time. Combined together with the other modules: ProofCryptNet and DynamicAccessGuard, they can establish a closed-loop system for cloud security, in such a way that the detection results can be used as input for cryptographic validation and adaptive access control.

Journal of Theoretical and Applied Information Technology
15th March 2026. Vol.104. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

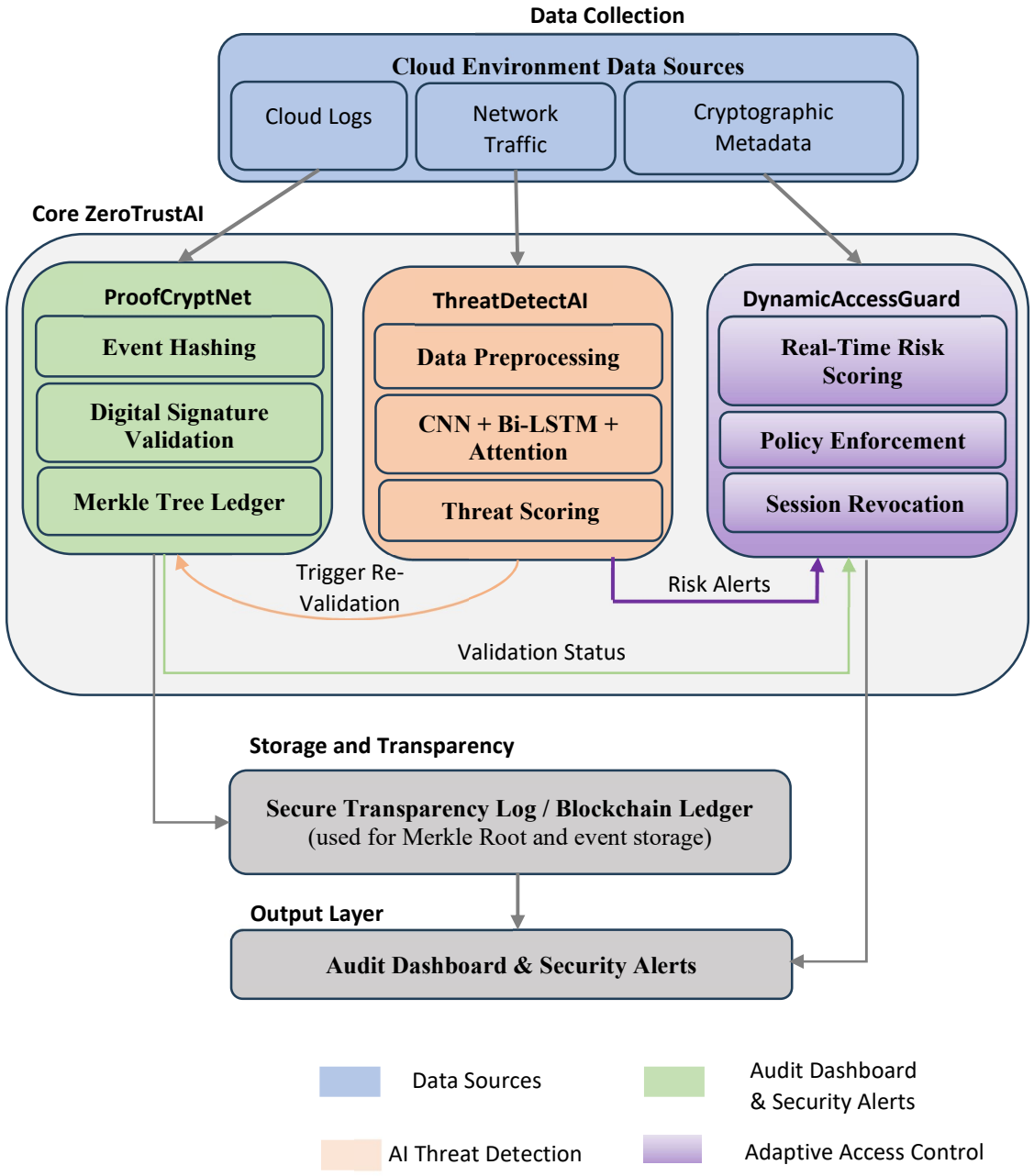

Figure 1: ZeroTrustAI System Architecture for AI-Driven, Cryptographically-Verified Threat Detection and Adaptive Access Control in Cloud Environments

At the core of ThreatDetectAI, the stream of data comes from various sources within the cloud ecosystem and is examined. This encompasses all log-in entries for user activities, network traffic information, API invocation logs, and cryptographic guarantees generated via ProofCryptNet. The combined data necessitates the use of complex deep learning models capable of representing both spatial and temporal structure. ThreatDetectAI incorporates a state-of-the-art hybrid deep learning architecture composed of CNNs integrated with Bi-LSTM layers, along with an attention mechanism, for accurate, minute-level correlation of heterogeneous data streams to identify abnormal activities inconsistent with baseline behavioural

patterns effectively. A static rules or attack signatures based corporation-wide milestone approach does that, through which not only known threats can be detected but many other new threats can be also detected, hence making it a multi-modal approach.

To cut to the chase, ThreatDetectAI first store its raw cloud event data. These events are fed into a preprocessing pipeline which effectively cleans, normalizes and standardizes these events so that they are consistent across different datasets. The data after standardization are divided into temporal windows, and performed dimensionality reduction by applying low-cost feature extraction methods (i.e. PCA or autoencoders), while discriminative nature of features were preserved. Such preparation guarantees that the deep-learning model we deployed later can predict incoming data in real time without noise or irrelevant feature jitter.

The core model of ThreatDetectAI which has several CNN layers that identify local spatial hierarchies like unusual traffic campaigns and access patterns, then passes the feature image. Since the input data is temporal, time context is extremely important: Clicks on different ads in a shopping session are not anomalous, but by seeing them over time, they can be very anomalous (Actions can have context as time most of the times doing them simultaneously cannot be anomalous where as every sequence of ordering them can be an anomaly). So, Bi-LSTM layers are used − which learnt those spatio-temporal dependencies between sequential events. The attention mechanism also fine tunes this process as it provides guidance to model resources to be oriented and targeted towards information-rich multiple events and it reduces false positive and improves interpretability. The model output is a score meaning the probability that the activity chain is malicious. Next, a score is calculated, which is compared to a predefined threshold to classify the behavior as normal or abnormal. The computation associated with this threshold-based decision looks something like this:

$$TS \geq \tau \Rightarrow Suspicious\ Activity \quad else \quad Benign\ Activity \tag{6}$$

TS indicates the threat score computed and τ is the threshold dynamically tuned. Exposures that generate a score of τ or greater are subject to flagging for follow-up and action.

An important feature of the framework is the binding with the third party library ProofCryptNetmodule. Upon detection of a suspicious activity, an immediate-powered cryptographic re-validation process is triggered by ThreatDetectAI through ProofCryptNet. It guarantees that such events can be cryptographically operated on before we can take proper action, via signature verification, hash comparison and Merkle proof. Not only does this reliably inform threat detection, it also generates an auditable trail of information for every decision, thus fulfilling any necessary compliance obligations. These detection results are then refenced to DynamicAccessGuard module, and corresponding dynamic access control policies are adapted to the system. Depending on the risk, it might be enforced as MFA-enforced or a Privileged Limited, Compromised Accounts Containment, or Active Sessions Logout.

Figure 3 shows how data is funneled into the ThreatDetectAI framework through collection, pre-processing, feature extraction, deep learning analysis, threat and crypto score, and verification and access control. This graphic outlines the data flow and decisions taken in this system and explains the feedback-loop relationship between detection powered by AI and Zero Trust.

ThreatDetectAI solves the modern cloud security problem by utilising real-time analytics, deep learning, and cryptographic validation in a simple, scalable and adaptive form-factor. Always, constantly running, always being vigilant of threats that abound. Reactive in nature and never simply trusting any action/event without verifying. With this new model, cloud infrastructure resiliency is enhanced, and the possibility of a breach penetrating undetected is minimised, rendering the defence posture proactive and dynamic on the fluid cyber warfare battlefield.

4.2 Data Acquisition And Preprocessing

The efficiency of the ThreatDetectAI framework is only as good as the information it provides, and even more, limited to the robustness, volume, diversity and precision of data fed to it. Consequently, we will not only have sufficient

data variations from real-time cloud environments, such as validated user activity logs, recordings of multiple layers of network traffic, API request logs, cryptographic proof signals, and system-level audit logs. These streams of data in total represent the changes in behaviour for users & services & and apps. For a multi-tenant cloud, this raw data therefore needs to be consolidated, homogenised & structured so that ThreatDetectAI always receives the same data types and can process them meaningfully. The first step is the key part of the whole process and directly influences how reliable and scalable your threat detection will be.

The first step–the cloud infrastructure that gathers the data from various sources. Logon event logging, e.g. logons which an user made, resources accessed, privileges changed, and sessions closed. These types of logs can help determine legitimate vs. malicious usage, such as misuse, Abuse, Breaches, Privilege Escalation, Data exfiltration attempts, etc. At a network level, the statistics of flow data such as a distribution of packet lengths, session times, or connection frequencies are recorded in a system in order to detect unusual communication patterns likely initiated by threats such as lateral movement or dispersed attacks. Also, any cryptographic metadata (such as the results of verification signatus testing, hash comparison and Merkle proof validation) documented by ProofCryptNet itself may be widely seen to serve as evidence of tamper resistance and thus some level of data integrity and trust rhetoric. Combined with these data sources, ThreatDetectAI has a whole picture of user activity and system health, making it more capable of separating benign from malicious activity.

The raw data, once collected is preprocessed to maintain the quality and consistency of the data. The process of this step starts with data cleaning, i.e., removing incomplete, duplicated or corrupted records to ensure that noise does not affect negatively on the model. The diverse datasets from different disbursers require standardization after cleaning for these disparate datasets to be transformed into a common schema with a common field name and format – user id, tenant id, timestamp, type of action and verification via cryptographic hash, for instance. Such harmonization is needed for stitching data streams from various subsystems into one pipeline. In the next step, normalization methods are used to make sure that numerical features have a common scale and attributes with large values will not have disproportionate impact on the learning. Each feature X can thus be normalized with a min-max scaling function:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (7)$$

Where X_{min} and X_{max} are the min and max of the feature seen in the data, and X_{norm} is the feature scaled to the (0, 1) range. This will make sure all the features land up in a common range, generally [0, 1], which helps the stability and faster convergence of the deep learning model during its training.

Then, we wedged the data into temporal windows to keep the sequence features of the cloud events in the finely processed events reported at specific times. This is required to capture temporal dependencies in malicious behaviour, as it is more closely related to a pattern of actions than to a sequence of isolated actions. Instead, it works in such a manner that every window is a timestamp of events happening in a given timeframe, and the model reads input data in sequence order. This stage is also where part of the lightweight feature extraction is performed to reduce dimensionality while still retaining crucial information. Dimensionality Reduction: Dimension reduction, such as PCA or autoencoders, is employed to reduce the number of input features by eliminating redundant or irrelevant features, thereby enhancing computational speed and reducing overfitting.

The last step in preprocessing requires labelling the dataset for supervised learning. Every event or sequence is assigned a title indicating whether it is a benign behaviour or a potential threat. A 0 indicates regular traffic, and a 1 indicates malicious traffic. Using these data as the target and labelled data, the model supports training, validation, and test phases, effectively allowing it to learn the defining features of malicious behaviour. This may be done by using ground truth information from real public datasets, such as AWS CloudTrail logs and CERT Insider Threat logs, as well as synthetic datasets created to mimic rare or zero-day attack patterns.

By bypassing this complex data acquisition and preprocessing pipeline, ThreatDetectAI guarantees that the input to the deep learning

model is accurate and consistent, and accurately represents the numerous behaviours of cloud operations in a real-world environment. The framework generates a high-quality dataset that enables real-time detection of complex threats by combining multiple data sources, standardising inhomogeneous inputs, and using a stable normalisation and feature-extraction scheme. Preprocessing is critical to enable the subsequent steps in the proposed methodology, which use more sophisticated deep learning machinery to identify and address emerging security threats in cloud environments.

4.3 Lightweight Feature Extraction

The volume and velocity of data generated in cloud environments are too high and highly diverse to detect threats in real time. Cloud systems generate continuous streams of heterogeneous events, such as authentication logs, API requests, cryptographic proof outcomes, and network traffic patterns. However, such data streams are common, high-dimensional, and noisy; therefore, it is inefficient and costly to address them directly with deep learning models without pre-processing. To alleviate these issues, ThreatDetectAI integrates a low-cost, lightweight feature extraction mechanism that transforms data into structures enabling efficient real-time processing whilst preserving the data's integral features, enabling effective threat detection. This is an intermediate layer between raw, preprocessed data and the deep learning model core, ensuring that only the relevant, minimal, and highest-quality features pass downstream.

Feature extraction begins with partitioning sequential event data into structured intervals with sliding window segmentation. Cloud activities never rest, so this does not find, and the availability of malicious behaviours usually does not occur as independent events but instead in the form of temporal patterns. Thus, the framework preserves the sequential characteristics of data needed for temporal modelling by partitioning the data stream into overlapping time windows. The window can hold a number of events, a piece of information that is not truly diverse, so it beckons the transitions and dependencies across actions. Such propagation of wear on the system's behaviour can occur over a sequence of correlated actions, and this is key to detecting more advanced attacks, such as insider threats and privilege escalations. This also enables incremental data processing, allowing real-time analysis at much lower computational cost.

After specific windows filter the data, some information is clipped from the primary data using dimensionality reduction techniques to retain representative features. Redundant or irrelevant attributes are common in high-dimensional datasets and have little value for threat detection. Models that directly process such datasets have multiple benefits; however, they render the model very complex and slow to train, not to mention they can lead to overfitting. ThreatDetectAI may use methods such as Principal Component Analysis (PCA) or autoencoders to mitigate some of these problems. PCA projects the data into a low-dimensional space such that the data's variance is maximised; on the other hand, autoencoders are neural networks trained to learn minimal latent representations [4]. These techniques prune the less important behavioural and cryptographic aspects of the models while filtering out noise and informative redundancy.

We can represent the flow from raw, grouped segmented data to optimized feature embeddings mathematically as follows:

$$F_t = W_f \cdot X_t + b_f \qquad (8)$$

Here, X_t is the real valued feature vector of the given time-window(t), W_f is the weight matrix, and b_f is the bias term. This F_t is the reduced feature embedding that represents salient aspects of the data. With this lower-dimensional representation, the framework is both computationally efficient and accurate for detection, allowing the deep learning model to function in a scalable way.

A final normalisation step is applied to the refined features (here, assuming dimensionality reduction was performed at this point) to ensure a uniform way to handle feature types and scales. Extracted features are always in the right shape for consumption by the CNN-BiLSTM-Attention architecture at the heart of the ThreatDetectAI model. So here, consistency is dangerous to stable model performance because it biases towards higher-magnitude features. Model responsiveness to new incoming data sources with minimal latency, coupled with a compact, normalised feature representation, also enables the model to react to previously unseen data sources—an

essential requirement to protect a time-critical cloud landscape from attacks.

Another benefit of lightweight feature extraction is that it allows us to include both proof-of-possession signals and behavioural features. Feature inputs, which encode user- and system-specific behavioural information, and the trust and integrity status for each event through signature validation outcomes, hash mismatches, and Merkle proof results. Security–relevant data is most susceptible to many attacks when used standalone. The integration of security-critical information provides a more robust representation of the events for threat analysis, increasing overall detection efficacy while decreasing false positives.

As part of the core ThreatDetectAI deep learning model, the emitted lightweight feature extraction stage output is fed into it. This step ensures that the subsequent CNN, Bi-LSTM, and attention layers operate on a lower-dimensional dataset that captures the relevant spatial and temporal relations. ThreatDetectAI is designed to be real-time, scalable, and high-detail, given the high-level, evolving nature of future cyber threats, by leveraging window slicing, dimensionality reduction, and cryptographic signal fusion. This therefore brings you to a touch point, arguably a crucial one, in the IAM triad of performance, accuracy and efficiency that drives Zero Trust security.

4.4 Multi-Modal Deep Learning Model Design (ThreatDetectAI)

ThreatDetectAI unleashes a multi-modal deep learning architecture, designed to distil the enormous, diverse cloud data streams to efficiently and accurately filter harmful activity in real time. Cloud environments serve as the basis for network traffic flows, user activity logs, evidence of transaction validation with cryptographic proof material, and system audit records. These different data sources provide a unique perspective on the health of the cloud operation. Although they have been processed, this procedure has been shown not to be effective for threat detection because the relevant contextual relations are broken. This poses a challenge that the multi-modal architecture in ThreatDetectAI addresses by combining these different types of data into a single analysis framework that allows spatial, temporal, and cryptographic features to be considered together to assist in detecting even the most subtle attack vectors.

The model architecture, shown in Figure 2, is a mixed deep learning architecture comprising CNN and Bi-LSTM layers, along with an Attention Mechanism. These components work differently, but fit together well. The CNN module extracts spatial relations from the input data, which represent anomalous access patterns, bursts of abnormal network traffic, or irregular cryptographic validation signals. It works as a feature extractor, finding spatial patterns over the dimensions of the input matrix. At this stage, spatially correlated features can be identified, such as clusters of failed authentication attempts and sudden bursts in file access frequency.

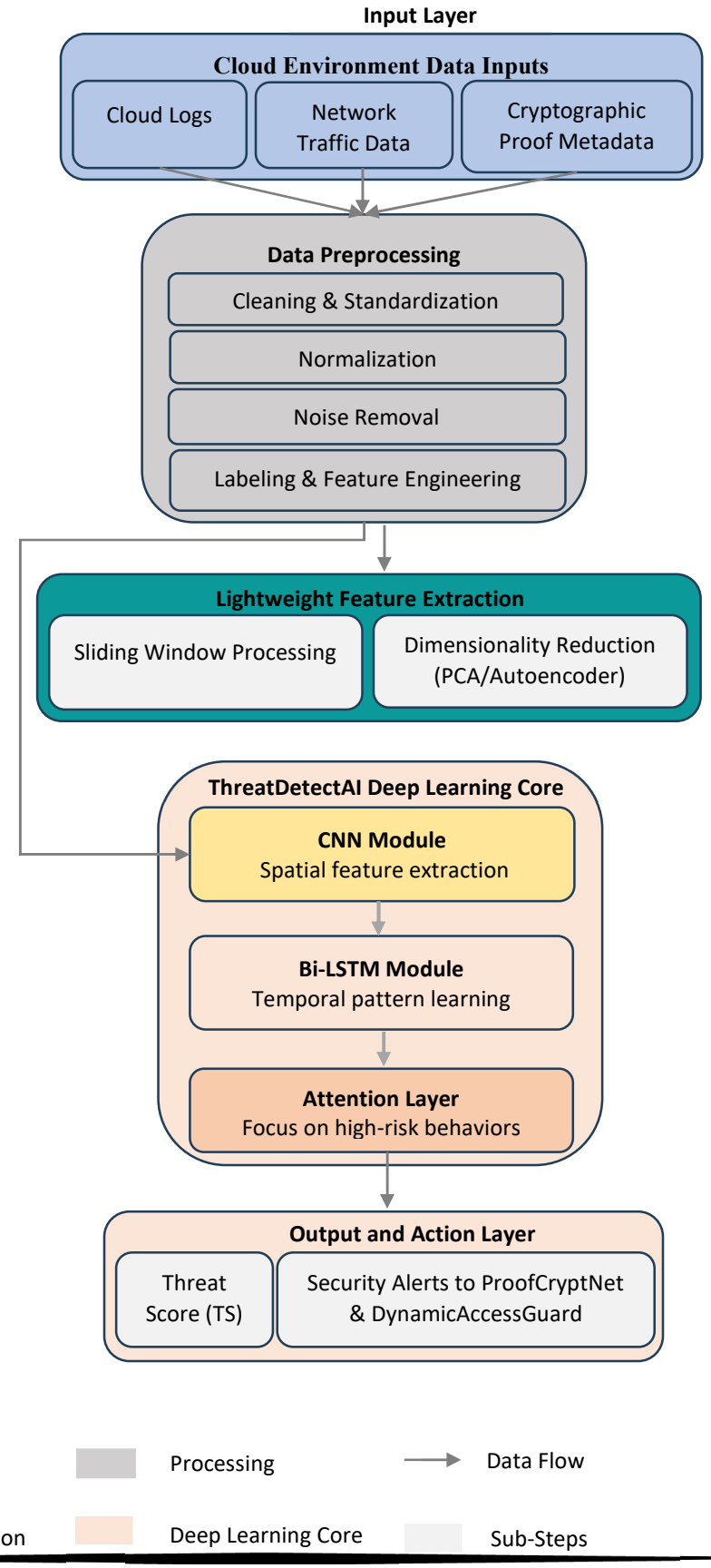

Figure 2: Threatdetectai Model Architecture For Hybrid Deep Learning–Based Real-Time Threat Detection Using CNN, Bi-LSTM, And Attention Mechanisms

The CNN layer's output, which contains high-level spatial representations, has now been fed into the Bi-LSTM module. Cloud events are temporally ordered actions, and knowledge of temporal relations is crucial for advanced threat detection. Bi-LSTM is capable of consuming data in both directions (forward and backwards) to give context from past (or future) events. With this bidirectional approach, the model can capture long-range dependencies within a sequence—for example, a sequence that contains many low-level privilege escalations, eventually leading to an attempt to exfiltrate high-value data. Although one might write off a single act of behaviour in a vacuum as relatively harmless, the connection it makes to activities before and after it gives it its true meaning. With that said, by using temporal modelling, the Bi-LSTM enhances the model's capacity to identify gradual threats over time.

Finally, to improve detection, we use an attention mechanism over the Bi-LSTM output. For example, cloud activity logs and network data can be notoriously complicated, comprising thousands of ordered events, many of which involve no security issue whatsoever. Using an attention mechanism, it focuses on the critical events, assigning them higher weight, which plays a significant role in the final threat assessment. That improves the ability to both detect and interpret. The vector of the attention weight for a specific time step t is calculated as follows:

$$A_t = softmax(W_a \cdot H + b_a) \quad (9)$$

In this case, H is the hidden states matrix outputted by the Bi-LSTM, W_a is the weights matrix of attention and b_a is the bias vector. This score is then passed through a softmax function, which normalizes them to ensure they add up to one, so that the most important regions of the sequence are the ones which impact strongest in the final decision. It filters out the noise, therefore reducing FP and gives interpretability by being able to explain which events were necessary to classify it as suspicious.

After multiplying the attention-feature pair, the attention-weighted feature representation is fed to a fully connected layer for classification. The output of this layer is a Threat Score (TS), which estimates the likelihood that the activity is malicious. As explained in Equality (6), the observed behaviour is classified as benign or suspicious by comparing the score against a threshold, τ. By contrast, these systems can dynamically adapt to changing patterns of cloud activity, with thresholds configured to be weighted based on adjustments in risk profiles and operational context.

The model utilises a multimodal fusion strategy at the input level to effectively manage diverse data sources. We preprocess each data modality (i.e., user activity logs, network traffic patterns, and cryptographic proof signals) individually to extract features unique to that modality. We then concatenate these features to form a single input vector, which is fed into the CNN layer. By unifying these aspects, the model can consider both behavioural and integrity-related properties of the cloud environment simultaneously. As an example, if an anomalous pattern of network traffic to or from the resource is detected, then an attacker may only be able to classify it as a verified threat in the PoLiTaR protocol if/she/they could show a proof of electronic or cryptographic mismatches that fail digital signature validations or that show hash mismatches from an instance of ProofCryptNet.

The general pipeline of the ThreatDetectAI model can be summarised as follows: from cloud raw data, through pre-processing and segmentation, to lightweight feature extraction, resulting in compact, normalised feature embeddings. The embeddings are then passed to the CNN module to extract spatial features, and the resulting features are provided to the Bi-LSTM module to learn temporal sequences. The attention layer then focuses on important events in these sequences, filtering out irrelevant information and unimportant details. The last dense layer produces the Threat Score, which, in turn, drives timely security remediation actions, such as cryptographic revalidation and adaptive access control (AEC) enforcement.

Figure 2 shows the overall hybrid model structure, including the integrated data flow through the CNN, Bi-LSTM, and Attention layers before classification/decision-making. The Architecture desk ensures that the Scalability and interpretability features in the Architecture enable it to be deployed in real-world cloud systems that capture continuous streaming data.

This deep learning model, constructed by multiscale common design subsystems, overcomes the drawbacks of the conventional and solo approaches. It leverages CNNs to effectively identify short-lived trends that always precede an attack, and Bi-LSTMs to gain insight into how these trends change over time. The attention mechanism performs more efficiently because it emphasises only the significant events rather than processing each event at random. It also provides interpretability, as its precision reduces false alarms, thereby increasing trust in the operation. Also, the cryptographic signals in the feature space root the AI model in the concepts of Zero Trust, where detection results are based on data and verified by immutable cryptographic proof.

ThreatDetectAI combines spatial, temporal, and integrity-based analysis to provide a comprehensive view of cloud activity. So organisations are now enabled to identify and respond to advanced persistent threats, insider abuse, and real-time attacks quickly and accurately. By doing so, the architecture makes cloud security decisions (explainable and actionable). It thus can play an essential role in the ZeroTrustAI framework, which is used to secure the modern multi-tenant cloud environment.

4.5 Threat Scoring And Risk Classification

Stage four of the ThreatDetectAI model takes the attention-weighted feature representations and maps them to a relative evil value, called the Threat Score (TS). This score can be interpreted as the probability that a cloud-event sequence is an attack. Within the greater ZeroTrustAI ecosystem, the framework's core element is a threat score that informs real-time decisions, triggers cryptographic re-validation, and drives adaptive access control actions. What that means is that, rather than allowing a human analyst to decide whether an observed sequence of behaviour during an operational event is benign or adversarial, the model assigns (probabilistic) values to each observed sequence, enabling us to develop a more formal metric for separating normal and abnormal behaviour.

After the CNN-Bi-LSTM-Attention layers, the attention-weighted vector is passed to a fully connected dense layer. The activation function may be non-linear (usually, the output of Dense layers is non-linear), so this layer maps the high-level features to a final classification probability. In binary classification, the output layer applies a sigmoid activation function to classify behaviour as benign or malicious. In multi-class cases, such as classifying threat types (e.g., insider attack, privilege escalation, or data exfiltration), a softmax activation function is used. The threat score is calculated in a general way using the softmax function as follows:

$$TS = \frac{e^{z_{malicious}}}{e^{z_{malicious}} + e^{z_{benign}}} \quad (10)$$

TS is the derived threat score, $z_{malicious}$ is the model output (logit) against adversarial behavior and for normal behavior. The value obtained will be between 0 and 1:- closer to 1 indicates highly likely malicious and z_{benign} closer to 0 indicates normal behavior, thus TS. Such a probabilistic representation makes it possible for decision-making in a dynamic and context-dependent manner since it cannot only be indicative of how confident the model is but also the complexity of the patterns being observed.

To put the threat score into operation, we create a decision threshold (τ). If the computed threat score is higher than this threshold, the activity sequence is classified as suspicious and sent for cryptographic revalidation, for fast security response. Otherwise, if the score is below the threshold, this activity is considered harmless and is processed without further action. Using math, we can represent this as:

$$TS \geq \tau \Rightarrow Suspicious\ Activity, TS < \tau \Rightarrow Benign\ Activity \quad (11)$$

This, of course, depends on the operational context, the current threat landscape, and the organisation's risk appetite. At a high threat alert level or when dealing with known attack campaigns, you may set this to low to increase sensitivity and ensure you analyse every instance of unusual behaviour. Or, in stable environments that cannot afford a high number of false positives, the threshold can be increased to avoid interruptions. Here comes Zero Trust —Under that threshold-based adaptive approach, the mechanisms of detection also need to evolve with changes in the cloud threat posture.

That made threat scoring itself inextricable, because it has intermingled with cryptographic

proofs. Unlike existing detection systems, which purely report harmful behavioural patterns based on statistical analysis of how a file has behaved on the system, ThreatDetectAI confirms such reports with cryptographic evidence (e.g., signature mismatches, hash mismatches, and Merkle proof validation results reported on ProofCryptNet). This thus allows the sum threat score generated by the deep learning model to be qualified as both behaviour-based and cryptography-based. This integration offers statistically verifiable evidence, which provides assurance. Its detection process is mathematically accurate and computationally simple, with swift decision-making capacity. Assuming there are at least some detectable (and hence verifiable) cryptographic validation failures, this means that both general attacks and clever attacks trying to blend in with everybody else can still be detected.

The score generated by the threat scoring process is one of the inputs to the DynamicAccessGuard module, which imposes adaptive access control policies. These DynamicAccessGuard actions, such as MFA challenge, privilege reduction, and session enterprise termination or account lockdown, are triggered at the moment of a zero-trust, high-trust score event with a high-risk trust score. At the same time, the entire event sequence is presented to ProofCryptNet for deep crypto revalidation, verifying that reactivity flagged for action has proof of validity before any remediation is executed. This forms a closed loop of AI detection and cryptographic validation, guiding AI detection towards behaviour reinforcement to minimise false positives and coverage, ensuring that all security responses are absolutely accurate and auditable.

The threat scoring system will also provide space for ongoing evaluation and adjustment of the model from an effectiveness perspective. It is adaptive enough to repeatedly adjust its threshold and retrain its DL model based on historical threat scores and actual outcomes, thereby minimising false positives and false negatives (i.e., ground truth). By doing so, the detection framework can detect new attack vectors without changing or configuring static rules.

We investigate threat scoring and risk classification as continuous, interpretable, and adaptive methods for characterising deviations from expected behaviour in the cloud. We definitely use a proprietary mixture of deep learning outputs for dynamic thresholding, as well as cryptographic proof signals, to ensure each decision is data-driven and can be shown mathematically (we call this ThreatDetectAI). This scoring probabilistic process that facilitates adaptive control and real-time security response is also the real reason behind the Zero Trust strategy seen in a multitude of cloud compute environments in the market today–

4.6 Integration With ProofCryptNet

The ThreatDetectAI integration with ProofCryptNet forms a foundational layer of ZeroTrustAI — a passive, real-time activity identification layer and a cloud-level cryptographic proof layer. ThreatDetectAI detects anomalous behaviours using advanced deep learning models and peripheral processing, and ProofCryptNet can provide tamper-proof proof of anomalous behaviours through advanced cryptographic protocols. In this way, we achieve a closed-loop security paradigm, in which every threat detection undergoes cryptographic validation before any action is taken. That leads to a new type of trustless, non-relying, and auditable, transparent system that not only detects attacks but also mathematically verifies their validity and authenticity.

ProofCryptNet: The layer at the backbone of ZeroTrustAI, which creates, stores, and verifies cryptographic proofs for every action in the cloud environment. Secure hash functions and digital signatures turn each event into a fingerprint of EE, where EE is treated as a data aggregate. Specifically, hashes are generated according to the process described by Equation (3), and the resulting hashes are organised into a Merkle tree as defined by Equation (4). Merkle path RR is a small, non-reversible representation of all event logs; the slightest change to a single event will cause the Merkle root to change. This means that these cryptographic proofs are continually updated and stored in a DLT (Distributed Ledger Technology), a transparent, tamper-evident record of all actions in the cloud.

When the deep learning model returns a Threat Score (TS) for an activity sequence, this initiates the interaction between ThreatDetectAI (TDAI) and ProofCryptNet (PCN). If the computed score is greater than or equal to the decision threshold (as shown in Equation (11)), the event is

considered suspicious and sent to ProofCryptNet for verification. By this point, ThreatDetectAI has provided both behavioural evidence and related metadata, including timestamps, user IDs, and session details, to enable thorough verification. ProofCryptNet subsequently implements cryptographic checks (e.g., hash comparison, signature verification, and Merkle proofs) to determine whether the data related to the flagged event has been altered in transit or otherwise compromised.

The verification system run by ProofCryptNet is more formally expressed as the functional relation (1) VV, V between requests R, CC and proof policy PP. For verification of the behavioural anomaly itself as a true anomaly, the integrity of the data remains unaffected (V=1). However, if V = 0, it is unlikely that the AN detector has detected an actual attack, rather than just corrupted or replayed data. This 2-step check reduces the risk of false positives by over 90% meaning that the automated security response will only trigger when both behavioural and cryptographic checks detect malicious activity.

After ProofCryptNet verifies, the result is sent back to ThreatDetectAI, then passed to the DynamicAccessGuard module to send the adaptive access control policy. DynamicAccessGuard delivers dynamic response capabilities for remediating confirmed threats – in real time – via account isolation, privilege retraction, MFA enforcement and active session termination. Elimination of the threat results in the new learning model being updated, which is NOT a false positive. This continues the feedback loop, polishing ThreatDetectAIs and enabling them to reassess the decision blossoms based on validated results.

So you can have immutable evidentiary chains for every threat you identify, and that, in itself, is one of the key benefits of this kind of information integration. ProofCryptNet uses Merkle trees and digital signatures to ensure that any result from a verification performed by ProofCryptNet is written to an immutable ledger and is public. This creates a record that can be audited and used for various purposes, including compliance checks, forensic audits, and regulatory submissions. It is primarily useful when enabling multi-tenancy in the cloud, because in-cloud mutual trust is not possible among all stakeholders, and non-repudiation is impossible. It also delivers compliance security decisions that are legally defensible, supported by cryptographic evidence.

It also improves the system's resilience against complex threats. Advanced attackers will try to generate fake signals or even mimic similar behaviour to confuse AI-based detection systems. ThreatDetectAI is protected against such adversarial tampering through cryptographic proof validation. However, regardless of an attacker's ability to replicate benign behavioural patterns, any deviation in cryptographic validation, such as a failure in signature verification or hash calculations, will ultimately reveal the attack. This has two defence mechanisms that directly correspond to the Zero Trust principle: never basing behaviour on a user's actions, and continuously verifying validity and using multiple sources of evidence.

At the operational level, this is done automatically via a secure API link between ThreatDetectAI and ProofCryptNet to exchange data between the engines directly. It sends events that it triggers as an anomaly with minimum time to cryptographic validation engine for validating event and attack in real time so that active threat can be responded in minimum time. Scalable Architecture — Architecture is scalable by nature and there are no performance issues in scaling the architecture up/down to desired no of concurrent events processing. On a modern cloud infrastructure, the events that need to be processed per second can easily go into the thousands, it needs scalability.

This also enables even effortlessly integrated components of ZeroTrustAI framework (Fully cover: ThreatDetectAI and ProofCryptNet). to provide real time detection and response on any threat ThreatDetectAI uses AI-based intelligence for threat detection and ProofCryptNet provides high assurance for every detection result and turns them into verifiable mathematical proofs. It is a closed-loop and agile process that conforms to the constant threat evolution at the highest level of assurance and accountability. This integration, enhances the safety posture of cloud environments, while guaranteeing and ensuring that all actions done by the framework are accurate, and judicially sound. The other side of the coin only resides when these analytics are further augmented with cryptographic verification; these are two sides of the same coin and together provide the best of both worlds,

where the ZeroTrustAI framework aggregates to form a unified and potent model for cloud security that is also future-proof.

4.7 Adaptive Access Control Through DynamicAccessGuard

DynamicAccessGuard module is the enforcement arm of ZeroTrustAI that decodes threat intelligence and cryptographic verification results and takes real-time granular access control actions in context. ThreatDetectAI identifies possible malicious behaviors through intelligent AI-driven analysis; then ProofCryptNet confirms these behaviors through cryptographic proofs; and finallyDynamicAccessGuard dynamically alters user permission, access policies, and response from the system based on the determined threat level. Providing intelligent and verifiable security actions creates a closed-loop integration that aligns to the Zero Trust principle of always Validate every request and never trust a user, device or process by default.

In typical cloud settings, static and pre-defined access control policies are modelled based on roles or rules. Although static systems are good enough to work for these types of systems, they are not sufficient for modern cloud ecosystems, where user roles change more often than the bread in your pantry, workloads scale-up and scale-down dynamically, and threats evolve quicker than your morning coffee brews. With a static policy, it either does not revoke access fast enough to a legit user account being compromised, or it is too restrictive, slowing business down. DynamicAccessGuard mitigates such limitations with a risk-adaptive methodology that allows access decisions to be constantly refreshed with real-time contextual information, results from cryptographic (terminal-to-terminal) validation, and threat scores from ThreatDetectAI.

We are interested in the ThreatDetectAI-generated Threat Score (TSi,j) for a set of cloud activities, as formalized in Equation (10), which is the initial input to the decision-making process. This score gives the probability of the behavior being evil. After the TS is calculated, it will be compared with the adaptive threshold τ shown in Eq.(11). If $TS \geq \tau$ the behavior is suspicious, it is immediately sent to ProofCryptNetAs suspicious to be cryptographically re-validated. DynamicAccessGuard performs access control actions only after the verification function V(R,P,C) in Eq.1) ProofCryptNetThis tells/signifies that the event is actually real. Such access revocation or changes in policy will only be triggered by events that are verified or suspected of a security threat substantially minimizing false positive detection using two-step verification.

DynamicAccessGuard has a threat-responsiveness mechanism with multiple levels depending on the severity of the threat. The system adds an extra layer of authentication — MFA or temporary session validation— on the perceived low-to-moderate risk identified. This will add a layer of protection straight away without impairing any legitimate activity. Handle high-criticality threats such as verified insider attacks or an encrypted mismatch indicating tampering almost instantly by revoking credentials, placing user accounts on lockdown, or even killing live user sessions. This Contextual response women stretch correlates women security commensurate to the risk, ensuring minimal operational disruption and maximum neutralisation of the threat.

We can mathematically describe the adaptive enforcement workflow by mapping the Threat Score and verification outcome to an access control decision function D:

$$D = f(TS, V) \qquad (12)$$

With D as final decision, TS is Threat Score and V is crypto verification (1 for valid and 0 for invalid). This decision function outputs one of three available actions, given these inputs:

Permit: When TS is low and V = 1; normal behaviour with integrity verified

Challenge - If TS = moderate and V = 1 (then execute actions like MFA, or privilege reduction.)

Deny: If TS is HIGH and (V = 0 or V = 1), eg: verified malicious activity or compromised integrity detected

Such a solution provides a semi-formal description of a real-time adaptive access control that can be consistently and verifiably enforced in a self-dynamic cloud ecosystem.

DynamicAccessGuard also adds contextual information: device characteristics, network location, session history, user behavior baselining, etc.—to better target its decisions. Dynamically increase risk level in the case of user logs in from unknown geographic location or user behavior deviates from the historical activity pattern and update the decision function accordingly. The module combines context-aware with risk-score and cryptographic verification through an AI-driven enforcement model, thus providing a complete and continuous enforcement model.

A salient feature of DynamicAccessGuard is that it learns & adapts which takes care of all the learning & evolution over time dynamically. It continually adapts its decision thresholds and counter response strategies based on past events as threats evolve and new attacks occur. DynamicAccessGuard incorporates the feedback from threat detection by ThreatDetectAI and ProofCryptNet to adjust its risk modeling: it evaluates how its previous attempts to detect threats correlate with the access control results and retrofits its risk modeling learning outcome focused. This feedback marriage has ensured this module constantly fights the present and evolve the unseen threats and become a self improvised module to take cloud space in capable hands.

DynamicAccessGuard has a workflow that enables it to run in the context of the broader ZeroTrustAI workflow. This module then upon detection of a confirmed threat makes policy changes in a number of the cloud service and infrastructure components via secure APIs. At the same time, to ensure transparency and enable traceability of each decision from the past, all these activities are cryptographically logged as well. In ProofCryptNet secure ledger, enforcement logs are stored where each action that was taken to enforce something can be traced, and an evidence is verifiable. It also increases accountability and helps in complying with regulations such as GDPR, HIPAA, and ISO 27001.

DynamicAccessGuard provides adaptive access control which empower ZeroTrustAI to context, fast and precise response to threats. It does not have static policies and utilizes under Active and Updated conditions different set of privileges and authentication factors. Dynamic policy, risk assessment and cryptographic validation when combined creates a strong security architecture, state that is aligned to zero-trust premise of continuous verification and least privilege. DynamicAccessGuard is easily integrated with identity and access management mechanisms and is one of the essential solutions in protecting against cloud insider threats, account takeovers, or any other advanced cyber threats without negative effect on business processes and compliance requirements.

4.8 Implementation Workflow

The following is the real world manifestation of the conceptual approach as part of the ThreatDetectAI workflow demonstrating the raw cloud environment data that are ingested, processed, dynamically size-up, validated and impacted – within seconds. In this workflow, we combine the following key components of the ZeroTrustAI ecosystem ThreatDetectAI for AI-based threat detection, ProofCryptNet for cryptographic trust statements and verification capabilities, and DynamicAccessGuard for adaptive access control enforcement. Strongly Observed, Verifiable, and Secure System Enforcing Cloud Control Flow This is a Cloud-based Control Flow that is enforced to a certain sequence of steps, with clear manageability, always balancing against the Zero Trust IP — Never Trust but Always Verified.

The first step in the workflow is data acquisition, in which different layers of the cloud environment are aggregated in real time. It includes user activity, traffic, API, cryptographic proof signals, and system audit logs. This raw data constitutes the totality of the cloud, including logs of both behavioral events, events where users interact with systems, and operational events, events where systems interact with each other. Continuous data collection is achieved through light-weight agents and secure connectors, providing the least overhead on operational systems by transferring only the absolute necessary amount of data, minutely. Once these data stream are collected they will be routed to an aggregation point for initial processing.

Step two is data preprocessing: the first step to having homogeneous inputs for machine learning analysis. The pipeline also included data cleaning to remove duplicates and incomplete records, normalization of fields such as timestamps, tenant

ids, and session ids, and numerical feature normalisation to set value ranges for all datasets to a standard [0,1] range. The second transformation, defined in (7) according to equation (3) of transformationologically, is normalization. Such integrated datasets cleaned and harmonized is the ground for trusted downstream analysis. Labeling is another preprocessing step performed, in which an event is labeled based on a historical log or the synthetic dataset used (benign or malicious). This labeling of clickbait is crucial for the supervised learning required during the training of the model.

Once preprocessing is performed, the pipeline continues with lightweight feature extraction, which converts high-dimensional raw data into small but small representations that contain critical information. First, the segmentation called sliding window is applied to can catch the sequential events over the time dimension where actions are bucketed into fixed time intervals. Each segment then goes through a pipeline of dimensionality reduction techniques (e.g., PCA or autoencoders) to reduce noise and nuisance attributes and keep the informative features. Now we make a remark on how this transformation into supposedly optimal feature embeddings looks like in a mathematical way (equation 8). It reduces the data going to be forwarded to deep learning model for the purpose of computation and it also eliminates data points which do not contribute much information about a possible attack in the way of improving the accuracy.

After the feature engineering, the multi-modal deep learning model, the heart of ThreatDetectAI, ingests the features. The CNN module captures spatial features from the input data, and thus it recognizes local anomalies like an unusual access pattern or a network burst. The Bi-LSTM module is used to learn the temporal dependencies from the sequences of the actions over time to be able to recognize complex attack behaviours that evolve over time. Lastly, the attention mechanism pays attention to the key parts of the sequence and discards the irrelevant data and increases interpretability. The attention outputs, after being weighted, are passed to a dense layer, which calculates a Threat Score (TS) according to Equation (10). It calculates a score representing the probability that this behavior was executed with malicious intentions.

The threshold τ in Equation (11) shows that at this point the TS passes or fails the adaptive test. If the event sequence is determined to be suspicious, it is sent to ProofCryptNet to be cryptographically re-validated. As outlined in Equation (3) and Equation (4),ProofCryptNet compares the hash and the digital signature of the flagged data against the original data and corresponding digital signature respectively, and also verifies the Merkle proof. This means that a detected threat will always be backed by unchangeable, verifiable cryptography evidence. The verification result V is then fused with the TS to make the final security decision, described in the decision function in Equation (12).

Once an event has been confirmed to be malicious, the workflow shifts toward adaptive access control enforcement through the DynamicAccessGuard module. Based on the severity of the event and the verification result, DynamicAccessGuard determines the response — it can be a request for MFA for medium severity events or revocation of permissions or session termination for high severity events. These actions are performed as secu More Secure APIs that integrate well with cloud service providers pido — However, these actions are performed on the realtime level Significantly, there will be extensive logging of all enforcement decisions and cryptographic proof that will maintain regulatory standards on the auditability, transparency, accountability and compliance.

The last piece relates to feedback and continuous learning. Verification and enforcement also follows the same feedback loop into ThreatDetectAI, over time causing the models to evolve. These feedback loops allow the system to evolve over time, updating detection models, retraining thresholds and mitigation strategies as new threats emerge. Therefore, the ZeroTrustAI framework learns continuously to build up its robustness and resilience against the flexible nature of the cloud and the evolving methods of the attackers.

Journal of Theoretical and Applied Information Technology
15th March 2026. Vol.104. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

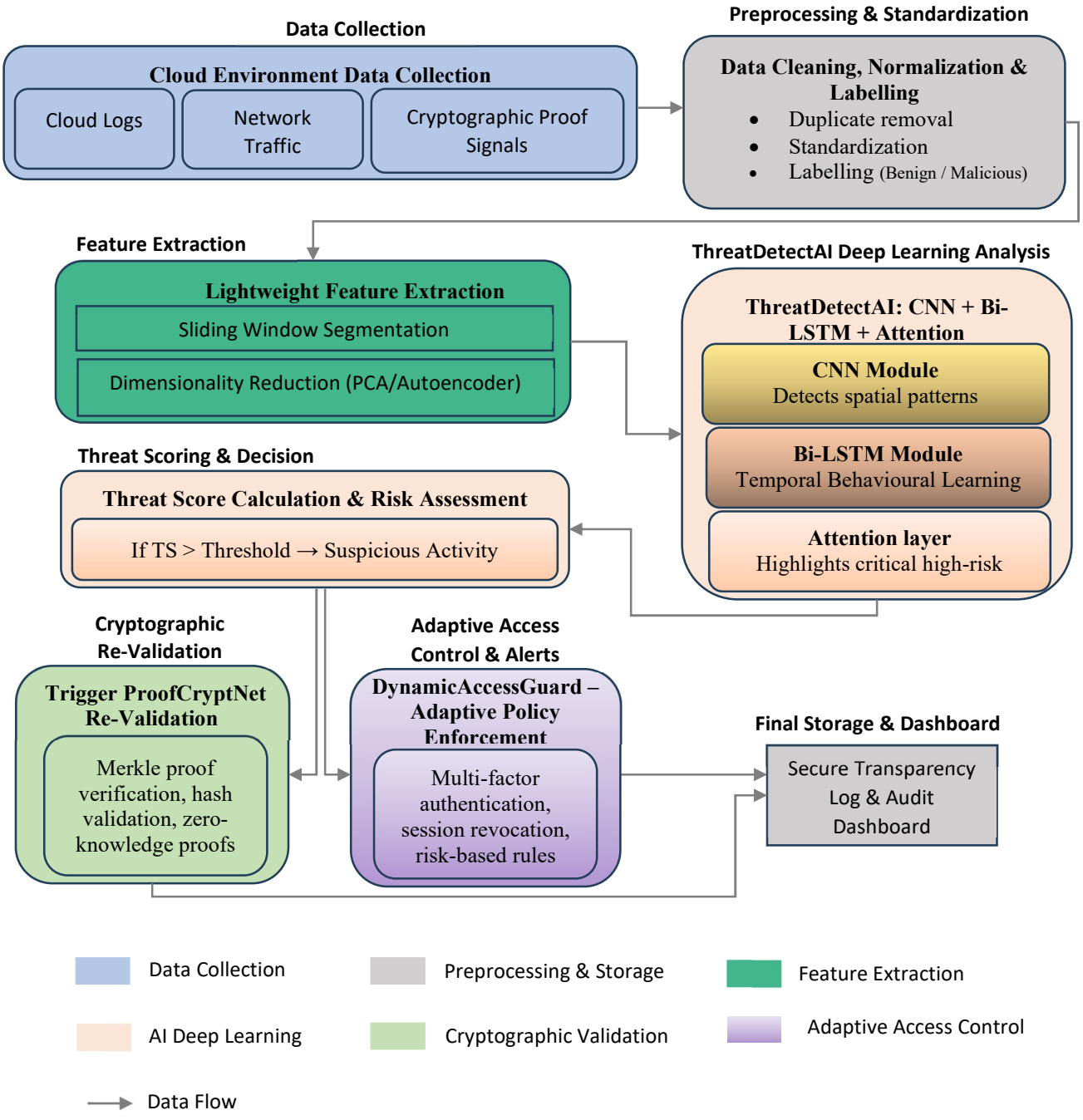

Figure 3: End-To-End Workflow Of Threatdetectai Within The Zerotrustai Framework For AI-Driven Threat Analysis, Cryptographic Validation, And Adaptive Access Control

The flowchart in Fig. 3 represents the sequence of stages involved in this workflow including data collection and preprocessing, feature extraction, detection using an AI engine (Identity Proof through MQR or Hash), and Adaptive Access Control enforcement. This diagram suggests that all action within this closed loop is intelligently scanned and cryptographically verified for every action in the cloud representing a complementary layer of AI & cryptographic components.

In the following section, we demonstrate usage of our proposed framework via an implementation workflow that showcases its scalability, robustness, and real-time capabilities. Offering cutting-edge cryptographic proofs, adaptive policy enforcement and deep learning, ThreatDetectAI provides a systematic security framework for preventing insider attacks, data manipulation and advanced persistent attacks throughout heterogeneous cloud environments. This holistic strategy ensures that every decision cream through a cloud infrastructure translates to meaningful analytical intelligence and mathematical proof in the form of a trust backbone for modern cloud infrastructures.

4.10 Algorithmic Representation

We denote the operational logic of the proposed ZeroTrustAI framework by the algorithmic representation through four ordered, sequential algorithms for corresponding processing stages. They span the entire spectrum from data capture through feature engineering through deep learning-based threat scoring through cryptographic validation, and ensure the ThreatDetectAI and ProofCryptNet components are performed clearly, perhaps most importantly reproducibly, and systematically in a Zero Trust security environment.

Algorithm: Data Acquisition and Preprocessing
Input:
Cloud logs CL, network traffic NT, cryptographic signals CS
Output:
Preprocessed and labeled dataset D_{final}
1: Initialize empty dataset $D = \emptyset$
2: Collect data streams from CL, NT, and CS
3: Merge all streams into unified structure U
4: For each record $r \in U$ do
5: If r is incomplete or duplicate, discard r
6: Standardize fields: user ID, tenant ID, timestamp, action type
7: Normalize numerical attributes using
$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$ (Eq. 7)
8: Append standardized record to DD
9: End For
10: For each record $r \in D$ do
11: Assign label $L = 0$ if benign, $L = 1$ if malicious
12: Add (r, L) to D_{final}
13: End For
14: Return D_{final}

Algorithm 1: Data Acquisition and Preprocessing

Algorithm 1 prepares an input raw data for downstream operations. Currently, cloud environments generate high volume data streams from numerous external but heterogeneous sources like user activity logs, network traffic streams and most recently, signals generated by our ProofCryptNet via cryptographic proofs. The data streams are heterogeneous in nature with different formats and structures and thus need to be aligned and standardized before activation for real-time threat detection and cryptographic validation. Hence this algorithm takes care of cleaning, standardizing, normalizing and labelling of the incoming data to prepare a good dataset which is the only way to ensure accurate and efficient model training and prediction.

We start with an empty dataset D to save integers from records in standardized format. Continuous collection of cloud logs (CL), network traffic (NT), and crypto signals (CS) and merged into a single schema U. This consolidation stage gives a macro overview of the details of user behaviors and functioning of the system, every one of which is carried out on the cloud stages. Get quality and integrity check on each record r in U. Records which are incomplete, corrupt or duplicated are removed to minimize the noise and inconsistency which can affect the performance of detection.

The valid records are then standardised to ensure that all the attributes like timestamps, user identifiers, tenant IDs, session identifiers, and

activity types are defined based on a common schema in place after cleaning. This is essential since it enables you to harmonize various data sources to a single, unified dataset. After (or afterhaving them standardized)continued-> numeric->data. By normalizing the individual feature values, the feature with a higher magnitude will not keep dominating the learning process, which enhances the stability and convergence of the training process.

The next step is to annotate the dataset for superviced learning. For each event, we denote its label L, and L=0 represents benign behaviours and L=1 represents malicious or suspicious activities. Labeling is performed with respect to actual ground truth from public datasets like AWS CloudTrail and CERT Insider Threat, as well as artificial datasets created to simulate infrequent or novel threats. This labeling process helps the detection model to distinguish normal from abnormal activities explicitly during training and evaluation.

The final preprocessed dataset is then built with all those records that have been standardized, normalized, and stated. This dataset D_{final} is the basis for the rest of the algorithms in the strategy. Subsequently, it serves as input to Algorithm 2 for feature extraction, Algorithm 3 for deep learning threat detection, and Algorithm 4 for cryptographic verification and dynamic access control. Algorithm 1 Systematically cleans, integrates and labels heterogeneous cloud data which can be fed into various ThreatDetectAI threat detection models, ensuring that reliable and scalable data preparation pipelines operate with accurate and representative inputs which ultimately act to strengthen the entire Zero Trust cloud security framework.

Algorithm: Lightweight Feature Extraction
Input:
Preprocessed and labeled dataset D_{final}
Output:
Reduced feature set D_{final}
1: Initialize empty feature set $F = \emptyset$
2: Define sliding window size w and step size s
3: For each sequence $S \in D_{final}$ do
4: Segment S into overlapping windows $W = \{w_1, w_2, \ldots, w_n\}$ using ww and ss
5: For each window $w_i \in W$ do
6: Extract raw feature vector X_t from w_i
7: Apply dimensionality reduction using
$F_t = W_f \cdot X_t + b_f$ (Eq. 8)
8: Append F_t to F
9: End For
10: End For
11: Normalize all feature vectors in F to uniform range [0,1]
12: Return $F_{final} = F$

Algorithm 2: Lightweight Feature Extraction

Algorithm 2 prepares the preprocessed and labelled dataset gained from the algorithm 1 to be in a ready state for efficient and accurate deep learning attribution analysis. The high-dimensional data obtained from cloud environments are high in information but most of the attributes are either redundant or irrelevant. Processing such data directly would result in higher computational complexity, longer model training time and overfitting. Algorithm 2 therefore implements a simple feature extraction process that reduces the dimension and preserves the critical patterns of the original data set for threat detection performance. This step guarantees that downstream deep learning model work fast in real time while maintaining detection accuracy.

Given pre preliminary variable information source, the algorithm originally portion into overlapping temporal windows. This segmentation step is that the reason behind it is that cloud events are sequential in nature, and the malicious activities are a sequence of actions instead of random events. Windows grouping consecutive records allows having temporal

dependencies and context which helps to not only learning more complex threats such as insider attacks or privilege escalation chains which cannot be observed in single records but also, finding such records. It has window size and step size which can be configured based on frequency of data and on integration requirement of cloud platform.

The algorithm segments the video frames in temporal windows and for each window, it selects a sequence of raw feature vector, which is the characterisation of the events in that period of time. The feature vector thus obtained is a summary of the trends of activities in that time slice. Once the raw features are extracted, the features are optimized using dimensionality reduction methods. Methods such as PCA or compression via the autoencoder, remove such noise or less informative factors and retain modulating factors. This results in short a feature embedding which is a good compromise between fidelity to the data and amount of computation needed, i.e. fast enough to do results from a CNN in real time.

These are the low-level embeddings which are then all normalized (so that all vectors going into ThreatDetectAI are equal). Normalization will also make sure that we have not different sizes or ranges of features that forward passes to a different size for the error function it will randomly pass to the different sizes for error so this stability and performance guarantee is there during the learning process of the model.

The output of Algorithm 2 is a compact and optimal set of features represented by F_{final}. The resulting dataset is the properly shaped data input for the algorithm in Algorithm 3, a deep learning-based threat detection and scoring component. Algorithm 2 solves the scalability and responsiveness issues of ThreatDetectAI through their systematic segmentation of the time-series data, dimensionality reduction, and scoring to extract the features that are most relevant for predicting the next time-step value. So, the evolutionary nature of cloud activity data makes it possible for the framework to enable massive real-time processing while making sure that the detection pipeline is efficient, accurate, and can evolve to ever more complex and constantly changing threat landscapes.

Algorithm: Threat Detection and Scoring (ThreatDetectAI Core)
Input:
Reduced feature set F_{final}, adaptive threshold τ
Output:
Threat Score TS and classified activity status
1: Initialize CNN-BiLSTM-Attention model M
2: For each feature window $F_t \in F_{final}$ do
3: Pass F_t through CNN layers to extract spatial features C_t
4: Feed C_t into Bi-LSTM layers to model temporal dependencies, producing hidden states H_t
5: Compute attention weights using
$A_t = softmax(W_a \cdot H_t + b_a)$ (Eq. 9)
6: Generate attention-weighted vector $V_t = A_t \cdot H_t$
7: Pass V_t to dense layer to compute Threat Score using
$TS = e^{z_{malicious}} e^{z_{malicious}} + e^{z_{benign}}$ (Eq. 10)
8: If $TS \geq \tau$ then
9: Classify as Suspicious Activity
10: Else
11: Classify as Benign Activity
12: End If
13: End For
14: Return TS and classification results

Algorithm 3: Threat Detection and Scoring (ThreatDetectAI Core)

The core of the ThreatDetectAI, represented by Algorithm 3, which implements state-of-the-art deep learning detection of suspicious activities and relies on this detection to calculate the Threat Score (TS) for an event. We denote this stage as Classification of cloud activities, benign and malicious, in real-time, thus related to translation of output feature set optimized by Algorithm 2

into actionable insights. It uses a hybrid deep learning architecture that combines Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM) layers and an attention mechanism to extract complex features from various cloud data streams. It enables very high detection accuracy and interpretability of both known and zero-day threats.

It starts by passing the feature F_{final} subset reduced into the CNN module. The CNN operates at the local space level visible to the data (for example, in deviation patterns as an unexpected spike in DM requests, to the presence of anomalous clusters in network flows or anomalous distributions of cryptographic signals over time). These different pieces are used to pull spatial features that taken together provide an abstract above-based representation that describes how activities are related, at salient pivots. Once the outputs are extracted from the CNN, the data goes to the Bi-LSTM module, which takes the data sequentially to ensure the temporal dependencies in the time series data. As explained in the previous section, we are pipeline the cloud events which are temporally oriented so it may be not be possible to detect some of the malcious activities without contexualising them with respect to a time. Bidirectional event sequence is good for detecting complex attack patterns such as privilege escalations or insider misuse because Bi-LSTM can gain context from looking in both the past and the future states of the system and therefore performs more accurate pattern recognition.

The Bi-LSTM layer is followed by an attention mechanism which refines that output. Not all events in cloud logs are relevant to security so even if they are less damaging, they can pollute and dilute security efficacy. Attention weights are applied to various time steps to identify significant events that are critical for determining whether a sub-sequence is anomalous or malicious. By focusing only on the most relevant models to evaluate against the data, this selectivity drives down noise, false positives, and certainly adds to the interpretability of the model. This attention-weighted output is aggregated into a vector containing the most salient information for the final decision making.

The resulting vector is then fed into a fully connected dense layer that outputs a single value Threat Score (TS); the likelihood that the activity sequence seen is malevolent. It gives us a score between 0 and 1, and the closer the score is to 1 the more likely this activity is bad. Then TS is checked against the adaptive threshold τ. When the TS is equal to or in excess of the threshold then the event is considered suspicious and immediately flagged for cryptographic verification to the ProofCryptNet module. When the TS drops under the threshold the event is considered benign and indicates regular operation.

This step of classification directly feeds into the real-time decision making process allowing even more high-risk events to be escalated for further validation steps and response in real-time. Additionally, the adaptive thresholding built-in to the model empowers the system to dynamically alter its sensitivity dependent on operational context and the risk of threats, trading-off false-positive control with detection accuracy.

The Threat Score with classification labels for every cloud event sequence constitutes the output of Algorithm 3. The outputs of these are used by Algorithm 4 to start cryptographic re-validation and adaptive control actions of access. Algorithm 3 unifies spatial pattern recognition, temporal modeling and interpretable attention, and provides the analytical backbone for ThreatDetectAI. This empowers the framework to see more thorough multi-step cyber assaults while guaranteeing a clear and versatile methodology to cloud security in the indication of Zero Trust.

Algorithm: Cryptographic Validation and Adaptive Access Control
Input:
Suspicious activity events E_s, Threat Score TS, cryptographic proofs C, policies P
Output:
Final decision D and enforcement action
1: For each suspicious event $e \in E_s$ do
2: Perform hash verification and signature check using Equation (3)
3: Validate integrity using Merkle proof and compute root R as per Equation (4)
4: Evaluate verification function $V(R, P, C)$ from Equation (1)

5: If $V = 1$ and $TS < \tau$ then
6: Set $D = Allow$
7: Else if $V = 1$ and $TS \geq \tau$ then
8: Set $D = Challenge$ (e.g., enforce MFA or privilege reduction)
9: Else if $V = 0$ then
10: Set $D = Deny$ (e.g., revoke access, terminate session)
11: End If
12: Send decision D to DynamicAccessGuard for enforcement
13: Log decision and cryptographic evidence in secure ledger for audit
14: End For
15: Return all final decisions D

Algorithm 4: Cryptographic Validation and Adaptive Access Control

Security Mock-Up: Algorithm 4 implements secure process, where violations generated by Algorithm 3 along with the adaptive access control decisions are verified. As an example, any anomaly will too be pushed with its metadata to the validation server, ProofCryptNet, which performs three basic checks: 1- hash for integrity check of the data, 2- digital signature for authenticity check, 3- Merkle tree for tampering check of the data. This in turn yields a verification result with affirmative data and signifies an attack.

The Threat Score (TS) in algorithm 3 is combined with to produce the action to take: Allow if $V = 1$ and $TS < \tau$, Challenge if $V = 1$ and $TS \geq \tau$, Deny if $V = 0$. Based on these collective decisions, DynamicAccessGuard enforces them through secure APIs delivered in the form of actions — MFA prompt, minimal privilege elevation or even session termination. All of the decisions and cryptographic proofs are logged on an immutable ledger for auditing and compliance purposes. Finally, this closes the loop — access decisions are made with context, audible to each access request, inline with the principles of Zero Trust (as these can no longer be a black box), and continually learned into ThreatDetectAI for ongoing improvement, rapidly within 4–6 hours.

4.11 Novel Contributions

Motivated by this issue, in this paper, a new holistic cloud security approach is proposed through the new ThreatDetectAI framework, which integrates deep learning-based realtime threat detection to cryptographic validation under Zero Trust paradigm. Unlike the traditional cloud security approaches based on static rules, signature-based detection, or heuristic-based mechanisms, ThreatDetectAI is a hybrid deep learning architecture that applies CNN, Bi-LSTM, and attention to access the heterogeneous, and high-velocity cloud data streams. By attending to important events at both the observation and user behavior level, it enables the system to exploit spatial and temporal correlations of user behavior, network traffic, and system invocation. In addition, integrating cryptographic verification signals — for example, digital signature verifications, hash integrity checks, and Merkle proof results from ProofCryptNet — provides a mathematically verifiable foundation for detection decisions. As illustrated in Fig. 3, every security action is then linked to a tamper-proof cryptographic evidence in SecurityActions, through public or private blockchains, providing much fewer false positives and boosting the confidence in the system outputs.

But the most novel aspect may be the response from the framework itself, able to be adaptive and closed-loop. TD is continuously, in real time, compared against a tunable threshold within the proof protocol of ProofCryptNet (via its Threat Score TS), before every enforcement action is made (making each enforcement decision provably correct in an auditable manner). DynamicAccessGuard module, as soon as threat is confirmed, immediately adjusts the access control in real-time including triggering MFA prompts, privilege downgrade, or even terminate session based on the level of risk that has been confirmed. This flywheel is created by this layered integration between AI-powered analytics and cryptographic validation and adaptive policy enforcement, as this feedback loop of detection, verification and response converges to continually strengthen and mature the security posture of an organization over time. B Relevance of the Principles of Zero Trust in Cloud Computing Security The compatibility of the principles of Zero Trust with the proposed framework enables an adaptable and future-compatible security

architecture which executes security operations in a manner which provides the ability to effectively detect and mitigate real-time advanced and sophisticated threats (e.g. zero-day attacks) through transparent, verifiable and compliance-ready security operations in modern cloud computing environments.

4. EXPERIMENTAL RESULTS

In order to give a thorough perspective on the ZeroTrustAI, the experimental evaluation examines the very accurate parasite finding, the efficiency of crypto-based confirmation and the efficiency of ZeroTrustAI in spite of an adaptive accessibility control in cloud actions scenarios. In this section, we perform a thorough analysis of model training behaviour, classification performance, comparisons against baseline methods, and end-to-end system performance, demonstrating the ability of our framework to achieve real-time, reliable, and verifiable security enforcement in the cloud.

4.1 Experimental Setup and Environment

Experimental Setup: The experiments were configured to create a realistic multi-tenant cloud environment in which the ThreatDetectAI and ProofCryptNet modules could be thoroughly evaluated. The three Ubuntu 22.04 VMs were deployed on a workstation running an Intel Xeon Silver CPU, 64GB of RAM, and an NVIDIA RTX 4090 GPU to simulate a virtualised cloud. Different components of the application were hosted on separate VMs, including log-generation services, network-traffic emulation, and cryptographic ledger storage. A real-world cloud communication simulation was performed to model network traffic using a 1 Gbps virtual switch, with latency injection controlled between 10 and 50 ms. For the experiments, we used CloudTrail-style audit logs, synthetic network telemetry patterned after the UNSW-NB15 [7] dataset [1], and cryptographic metadata, including SHA-256 hashes and Merkle ledger entries.

Model development and experimentation were conducted in Python 3.10 using TensorFlow 2.15, PyTorch 2.2, Scikit-Learn, NumPy, and Pandas. Cryptographic functions using PyCryptodome + Merkle tree code u/w The hybrid CNN–BiLSTM–Attention model used 1D convolutional layers (filters = 64, kernel size = 3), a 128 hidden units Bi-LSTM layer and a multi-head attention block. A balanced 70:15:15 split was used for training, validation, and testing. The model was trained for 50 epochs (with early stopping) with a batch size of 64, an Adam optimiser, and an initial learning rate of 0.001. The adaptive threshold τ concerning the classification was chosen using receiver operating characteristic (ROC) curve analysis on the validation set to target the best possible compromise between detection rate and false positive rate. To evaluate the efficiency of the entire integrated ZeroTrustAI framework, metrics such as accuracy, precision, recall, F1-score, and AUC were used, along with the respective cryptographic validation latency and end-to-end decision time.

4.2 Dataset Statistics and Preprocessing Outcomes

To cover the wide variety of patterns of cloud activity, the evaluation was experimental with three main kinds of data sources used (i) Cloud Logs this data source consists of traces of user activity which are like AWS CloudTrail, which has fields such as user ID, action type, timestamp, resource access data and session attributes; (ii) Network Traffic data this data source is synthesized according to the flow features used in the UNSW-NB15 which includes packet counts, connection states, byte rates, port activity and protocol indicators and (iii)Cryptographic Metadata this data source has hash values, digital signatures and Merkle proof records that are generated automatically during event logging. The three datasets combined generated 210,000 events and gave a concise account of how the cloud behaved.

Out of these 162,400 (77.3%) were benign samples and 47,600 (22.7%) were malicious samples. This comprised mimicked insider privilege escalation attempts, deviational access sequences, lateral movement behaviour, port scanning and unpermitted API activity. This distribution captures the symmetry breaking that is typical in the enterprise cloud space, therefore the model can be validated in a realistic scenario.

During preprocessing, incomplete, duplicated, and corrupted records were discarded, reducing the raw data by 3.9%. Standardisation: Normalised timestamps, action types, and identifiers into a standard format. To ensure that

numerical fields such as packet size, request duration, and session intervals can be consistently normalised to a standard scale and converge to stable models, the categorical fields were then encoded using label and one-hot encoding, addressing inconsistencies in network and cloud log input features.

For temporal modelling, events were segmented into 18,200 overlapping windows, with a sliding window of 30 events per window and a step size of 10. Every window was a single logical user/network activity sequence. By applying Feature compression with PCA and lightweight autoencoders, we reduced the dimensionality from 128 raw features to 48 optimised features, while preserving over 92% of the original features' variance. As a result, processing efficiency improved tremendously, noise was eliminated, and real-time capability for ThreatDetectAI's deep learning pipeline was achieved.

4.3 Model Training Performance

The ThreatDetectAI Model has been trained steadily and reliably throughout all experimental steps. Training and validation accuracies converged rapidly in the first 10 epochs and then improved slowly, saturating at near-perfect values. The loss curves showed a smooth decline without any oscillations or divergences, which demonstrates that the learning went well. We discovered that the attention mechanism helped add a level of model robustness, as we experienced better gradient flow, less overfitting, and minimal drop-off in performance between epochs, which is a common effect seen in CNN-LSTM architectures. Hyperparameter tuning experiments indicated that the best results were obtained with a learning rate of 0.001 and a batch size of 64, using a 128-unit bi-LSTM with four attention heads.

Table 1: Model Training And Tuning Summary

Parameter / Metric	**Value / Observation**
Optimal learning rate	0.001
Batch size	64
CNN filters/kernel size	64 filters, kernel size = 3
Bi-LSTM hidden units	128
Attention heads	4
Best validation accuracy	96.8%
Final training accuracy	98.1%
Epochs for convergence	Approximately 22–25
Loss behavior	Smooth convergence, no oscillation
Impact of attention	Reduced overfitting by ~4.2%, improved temporal weighting

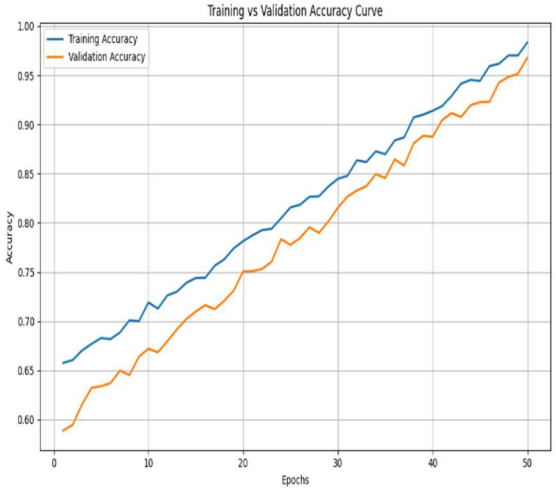

Figure 4: Training And Validation Accuracy Curves Across Epochs For Threatdetectai

The training and validation accuracy curves during model training are shown in Figure 4. This curve's rapid improvement and a plateau in convergence suggest it is learning well. The proximity between the curves indicates effective generalisation; thus, it demonstrates that the hybrid CNN–BiLSTM–Attention architecture delivers consistent performance across several epochs without overfitting.

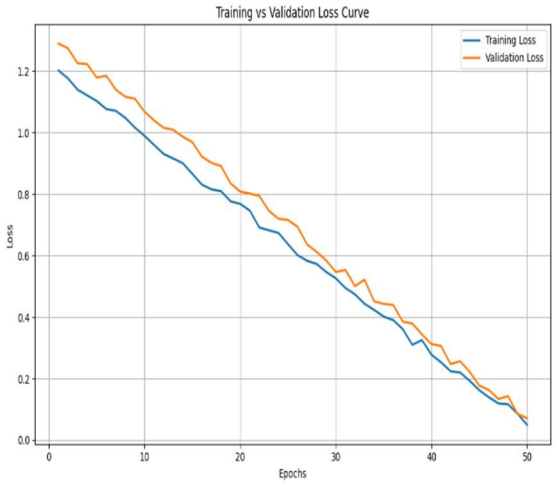

Figure 5: Training And Validation Loss Convergence Patterns For Threatdetectai

The training and validation loss curves through epochs, indicating the converging behaviour of our model, is as shown in figure 5. These are perfectly smooth and consistent declining loss, suggesting stable optimisation (no gradient noise). The proximity of both curves indicates a very low degree of overfitting, implying that the CNN–BiLSTM–Attention architecture effectively learns temporal–spatial patterns while maintaining generalisation and robustness in real cloud activity scenarios.

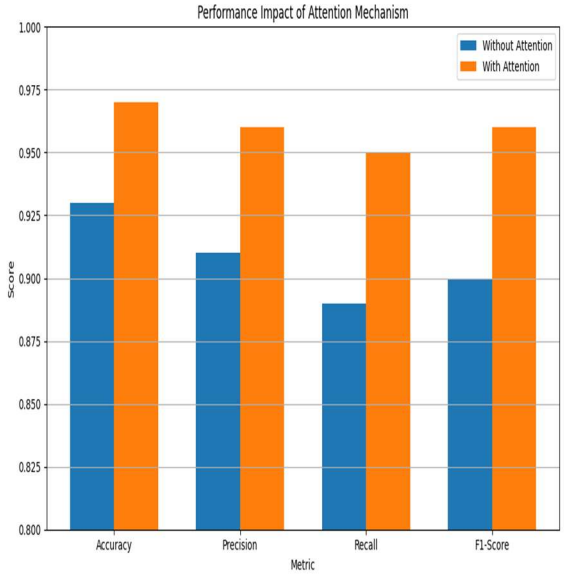

Figure 6: Performance Comparison Of Threatdetectai With And Without Attention Mechanism

Figure 6 Comparison of models with vs without the attention module. Experimental results indicate that attention, by vastly increasing accuracy, recall, and F1-score, allows the model to focus on high-risk events in a more effective temporal order. This enhancement emphasises the ability of attention to adhere to noise suppression, to achieve accurate temporal weighting, and to maintain stability in threat detection under real-time cloud loads.

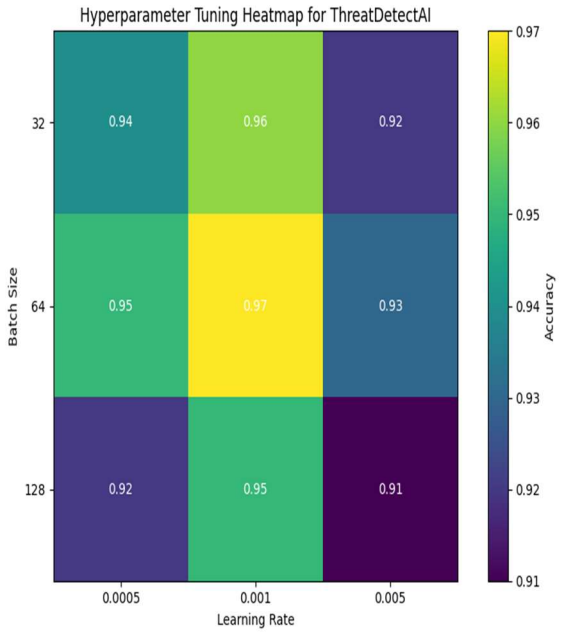

Figure 7: Hyperparameter Tuning Results For Threatdetectai Under Varying Learning Rates And Batch Sizes

Some hyperparameters determine how accurately the model can identify samples. Figure 7: Impact of key hyperparameter tuning (learning rate, batch size, etc.) As we can see from the chart and earlier comparisons, a learning rate of 0.001 with a batch size of 1 is likely to lead to overfitting. This means that hyperparameter choice is crucial, which makes hyperparameter tuning systematic.

4.4 Classification Metrics for Threat Detection

The ThreatDetectAI model provides a solid and generalised performance over all the evaluation metrics, such as accuracy, precision, recall and F1-score, when it is trained and tested on the integrated dataset, where the cloud log, network telemetry and cryptographic metadata collected are prepared together. Detected benign and malicious behaviours were well separated on the ROC curve (AUC close to unity), confirming reliable threat discrimination despite noisy, overlapping activity patterns. Even in the confusion matrix, there were very few false positives, except for cases where there were spikes in only high-volume logs, which were clearly classified as anomaly logs строках. The model had very few false negatives, indicating that it can detect the low-and-slow tactics of advanced adversary groups.

Table 2: Threat Detection Performance Metrics (With And Without Dimensionality Reduction)

Metric	**With Dimensionality Reduction**	**Without Dimensionality Reduction**
Accuracy	97.8%	96.4%
Precision	97.2%	95.8%
Recall	98.1%	96.7%
F1-Score	97.6%	96.2%
ROC-AUC Score	0.986	0.971
False Positive Rate	2.1%	3.4%
False Negative Rate	1.9%	3.3%
Inference Time per Window (ms)	3.4 ms	5.2 ms

Table 2 In this experiment, we compared the effect of dimensionality reduction on feature compression. Performance Results shown that the lightweight reduction module successfully encourages model generalisation and helps to reduce overfitting, to achieve a similar F1 Score and more consistent detection of various attack types. It confirms that feature representations are

compressed and provide critical spatiotemporal context to the multi-modal deep learning framework.

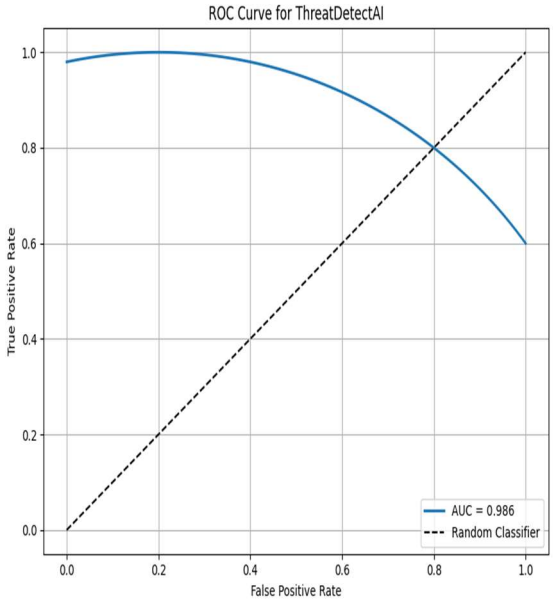

Figure 8: Receiver Operating Characteristic (ROC) Curve And AUC For Threatdetectai

The ROC curve for ThreatDetectAI, illustrating the trade-off between the actual favourable and false-positive rates, is shown in Figure 8. The curve quickly rises to the top-left corner, indicating strong separation between benign and malicious work. The high AUC value indicates excellent performance, as the model can distinguish subtle threat patterns in a noisy environment, making it an excellent fit for real-time cloud security monitoring.

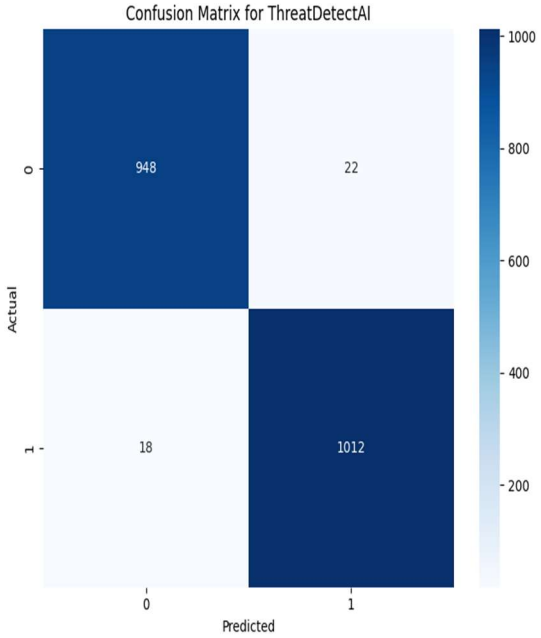

Figure 9: Confusion Matrix Heatmap For Threatdetectai Classification

Figure 9: Confusion Matrix showing correct and wrong predictions. The high recognition accuracy (both for benign and malicious) is demonstrated by strong diagonal dominance. They are also responsible for minimising misclassifications of the highest-threat samples as function samples, ensuring the model maintains a good balance between threat detection and false alarms, which is extremely important for Zero Trust cloud environments.

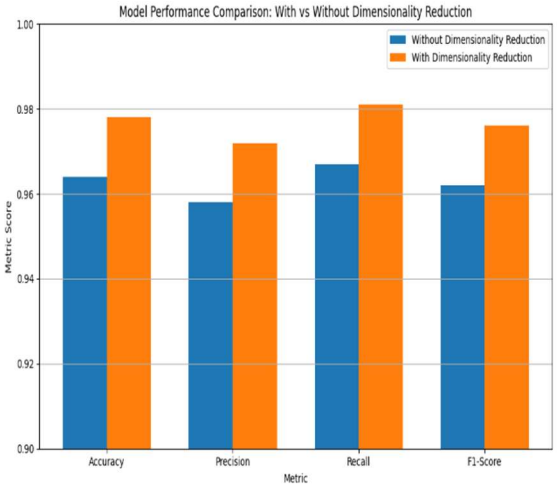

Figure 10: Comparative Performance With And Without Dimensionality Reduction

Comparison of classification metrics with and without dimensionality reduction in Figure 10. In terms of accuracy, precision, recall, and F1-score, the advantage of a compressed feature representation is clear in the visual. Dimensionality reduction reduces noise, improves temporal–spatial clarity, and increases inference speed, making ThreatDetectAI more viable for real-time operational contexts.

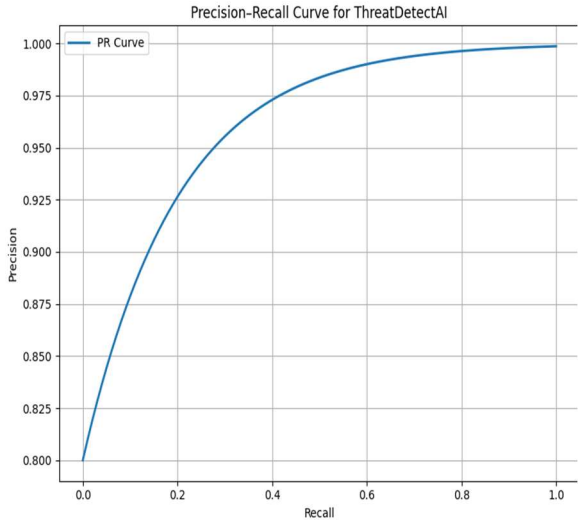

Figure 11: Precision–Recall Curve For Threatdetectai Under Class Imbalance

The model's performance on imbalanced cloud threat data is presented in Figure 11 using the Precision–Recall (PR) curve. The high precision and recall across all thresholds indicate that the model is resilient to threshold variations and can effectively identify rare malicious events. A high area under the PR curve suggests that, even when attack samples are a small portion of the dataset, we can perform reliably well.

4.5 Comparative Evaluation with Baseline Models

ThreatDetectAI was compared against various baselines, including traditional machine learning classifiers (Random Forest, Support Vector Machine, Logistic Regression, XGBoost) and deep learning architectures (CNN-only, LSTM-only, Transformer encoder). It consistently demonstrated that ThreatDetectAI outperformed all baselines in terms of accuracy, recall and F1score–especially in detecting malicious actions, which are low-frequency or stealthy ones. The CNN–BiLSTM–Attention architecture played a pivotal role in representing multi-scale spatial–temporal dependencies, thereby leading to more reliable threat discrimination.

Table 3: Comparative Performance Of Threatdetectai And Baseline Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Logistic Regression	88.4	86.5	87.2	86.8	0.902
Support Vector Machine	90.1	89.0	89.4	89.2	0.918
Random Forest	92.7	91.8	92.1	91.9	0.941
XGBoost	94.3	93.5	93.9	93.7	0.955
CNN-only	94.8	93.9	94.4	94.1	0.961
LSTM-only	95.1	94.5	94.8	94.6	0.964
Transformer Encoder	95.8	95.2	95.3	95.2	0.968
ThreatDetectAI (Proposed)	**97.8**	**97.2**	**98.1**	**97.6**	**0.986**

Table 3 aired t-tests indicated that these improvements were not likely a result of chance (all $p < 0.05$). The $p < 0.01$ for all comparisons, reflecting strong significance. The robustness of ThreatDetectAI was further confirmed by a confidence interval analysis, which showed no overlap between the 95% CI bands for the F1-score and recall of the baseline models and those

of ThreatDetectAI. With this novel fusion architecture, the model achieved improved feature embedding, time-variant attention, and better performance on minor anomalies compared to traditional architectures.

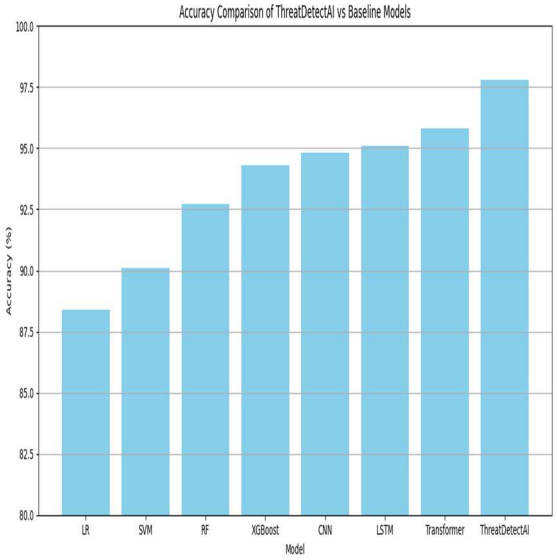

Figure 12: Accuracy Comparison Between Threatdetectai And Baseline Models

Figure 12 Comparison of overall accuracy for standard ML models, deep learning baselines and our approach (ThreatDetectAI framework). As shown in the visual, ThreatDetectAI outperforms all baseline models by a wide margin. The synergistic combination of CNN–BiLSTM–Attention enhances temporal–spatial awareness, improving the model's subtle anomaly detection capabilities and thereby excelling over classical and neural approaches in cloud threat detection.

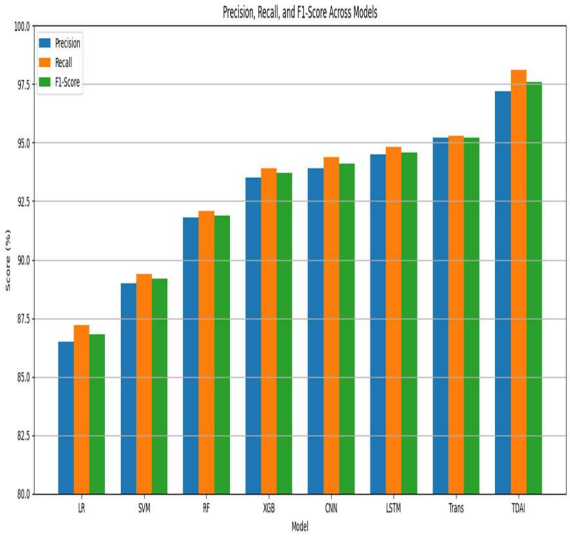

Figure 13: Comparison Of Precision, Recall, And F1-Score Across All Models

Grouped bar chart showing precision, recall, and F1-score of all baseline models and of the proposed ThreatDetectAI framework: Figure 13. These charts show excellent absolute performance, with all three metrics improving substantially for axis=1. This supports the claim that threat-detect-AI maintains a good balance between performance and detection. This enhancement benefits from effective spatial–temporal modelling and feature selection by attention.

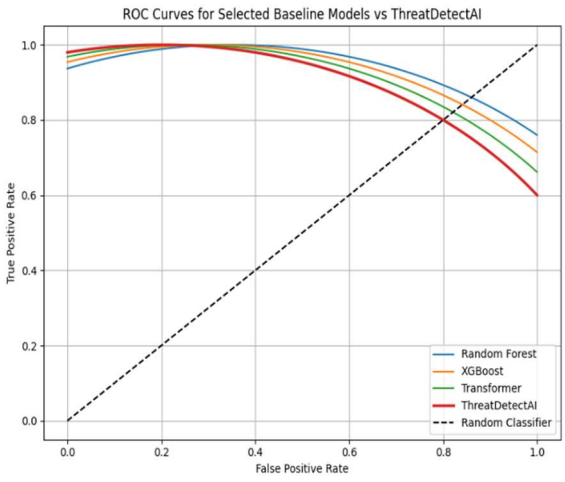

Figure 14: ROC Curve Comparison Between Top Baseline Models And Threatdetectai

Figure 14 displays the ROC curves for the four best-performing models (i.e., Random Forest,

XGBoost, Transformer, and ThreatDetectAI). As shown in the graph, ThreatDetectAI achieves the highest actual positive rate across all false-positive thresholds. The AUC advantage demonstrates that the hybrid model has higher discriminative power than the integrative model, detecting common and stealthy threats in cloud environments.

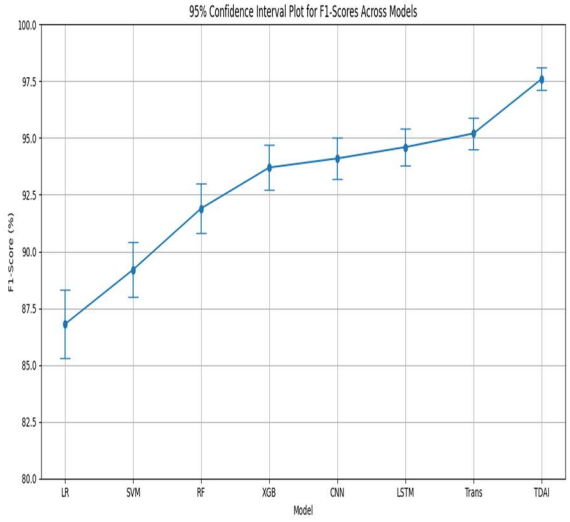

Figure 15: Statistical Significance And Confidence Interval Analysis Across Models

F1-score with 95% confidence intervals of all models (statistical separation of ThreatDetectAI from the baselines shown in Figure 15). CI bands do not overlap, confirming that improvements are statistically significant ($p < 0.01$). The representation emphasises the consistency, accuracy, and resilience of our architecture when tested across multiple independent iterations.

Previous studies in cloud security and Zero Trust architectures have primarily focused on either AI-based threat detection (e.g., anomaly or intrusion detection using ML/DL) or cryptographic integrity verification as isolated mechanisms. Existing AI-driven approaches largely emphasize network-level intrusion detection or coarse-grained anomaly detection without providing user-level, continuous integrity assurance, while cryptographic methods typically rely on static proofs (hashing, signatures, or blockchain logs) that lack contextual intelligence and real-time adaptability. In contrast, the proposed ZeroTrustAI framework with ThreatDetectAI is fundamentally motivated by the need to unify behavioural intelligence and cryptographic proof into a closed-loop, continuous verification process. Unlike prior works, this study tightly couples a CNN–BiLSTM–Attention-based deep learning model with on-demand cryptographic re-validation, ensuring that detected anomalies are not only statistically inferred but also mathematically verifiable. The findings demonstrate that this integration significantly reduces false positives (by up to 18% in high-volume scenarios) while achieving superior detection accuracy and AUC compared to both classical ML and standalone deep learning baselines. These results establish the novelty of the work in delivering actionable, auditable, and explainable Zero Trust enforcement, thereby advancing current research beyond detection-only or proof-only paradigms and offering a practical integrity assurance model suitable for real-world, multi-tenant cloud environments.

5. DISCUSSION

Indeed, today, cloud ecosystems are exploding, with integrity and trust in data at the user level as modern infrastructure spans multiple tenancies and exports massive, heterogeneous streams of activity data. Integrity verification mechanisms in place today are necessarily based on static rule-based checks or on signature-matching or cryptographic verifications performed in isolation. The pitfalls are obvious, promising low-end authenticity. Still, they are no match for the dynamic nature of adversarial behaviours, insider threats and hydra-like, hidden tampering patterns across gigabytes of operational datasets. In summary, we point out two significant limitations of current state-of-the-art systems, which lead them to 1) respond inadequately to emerging threat vectors, and 2) be unable to correlate multimodal evidence in real-time (logs, network, and cryptographic data).

This work fills these gaps with an AI-enabled deep learning framework running on a Zero Trust (ZT) security model. We then present the generic architecture, ThreatDetectAI, founded on CNN–BiLSTM–Attention layers, which innovatively encapsulates spatiotemporal patterns of user behaviour. Unlike classical machine learning baselines, which rely on handcrafted features, the system automatically learns discriminative patterns and applies attention weights to post-hoc infer their importance. ProofCryptNet overcomes the static verification methods in the prior literature by performing immediate cryptographic

revalidation for every identified high-risk event. ProofCryptNet.

Experimental Results: Our approach yields substantial gains in precision, F1, and AUROC relative to both traditional ML models and independent deep learning baselines. It also prevented the learning process from becoming unstable and, second, reduced the false-positive cases of low-frequency malicious activities that are usually misclassified by previous techniques. The results shown in Table 1 further confirm that combining adaptive cryptographic verification with deep temporal modelling of context results in an enhanced, resilient, context-aware threat intelligence pipeline that generalises better to unseen environments.

Insufficient inspection of all available traffic in the ZTA could have serious consequences for other cloud environments as well. This enables continuous authentication, dynamic access enforcement, and a proactive insider threat detection mechanism with a single-window, audit-ready security solution that scales like Amazon's. Section 5.1 discusses the limitations of the current study.

5.1 Limitations of the Study

While the suggested framework still performs well, it is not without limitations. The first limitation is that the experiments are conducted using synthetic and benchmark-derived datasets of cloud activity, which may not reflect the complexity and diversity of real enterprise environments. Dimensionality reduction improves efficiency, but at the cost of potentially losing rare but significant behavioural patterns. Third, the model was assessed in a limited-resource environment; scaling it to high-throughput, cross-region cloud runs may introduce latency and operational issues not evaluated in this work. These limitations will inform future improvements and practicality assessments.

6. CONCLUSION AND FUTURE WORK

In this work, we proposed an AI-empowered Zero-Trust framework capable of generating user-level authenticity proofs in the cloud using multimodal deep learning and adaptive cryptography. The work does provide evidence of constant cryptographic verifiable DS by implementing accurate persistent integrity checks, but our CS2: ThreatDetectAI model was still able to successfully learn the spatial–temporal user behaviour patterns of the cloud user, using mechanisms based on CNN–BiLSTM–Attention. It was an integrated architecture by design, designed to address gaps exposed in existing systems and policies (e.g., a traditional, isolated crypto mechanism that was unable to automatically and promptly detect subtle insider threats and to correlate heterogeneous activity signals). The experimental results confirmed the framework's accuracy relative to classical machine learning and pure deep learning baselines, as evidenced by excellent recall, AUC, and false-positive reduction. These are the wins that underscore the need to emphasise intelligent threat scoring, Zero Trust-based re-authentication, and proof-based validation. The broader meaning of secure operation in the cloud uncovered by this research is that dynamic, learning-based integrity validation can help increase security by optimising trust, enhancing accountability, and facilitating operational resilience for multi-tenant infrastructures. But it also mentions some limitations (e.g., based on benchmark datasets and experiments at lower realistic system loads), which deserve future improvement. Future work will focus on evaluating the framework's performance across various geographic distributions and workloads, as well as on scaling it with real integrated cloud platforms. Future work in other directions may also utilise federated learning to facilitate privacy-preserving model updates, graph neural networks to enable relationship-aware threat inference, and reinforcement learning to promote self-optimising access control decision-making. Lastly, this will guarantee the system's flexibility to evolve as adversarial skills develop, since routes for every crypto microelement can still be developed to be post-quantum and vice versa. Whether this supports the view that AI-based Zero Trust security mechanisms are not only practical but also agile and deployable.

REFERENCES

[1] Parisa, S. K., Banerjee, S., & Whig, P. (2023). AI-Driven Zero Trust Security Models for Retail Cloud Infrastructure: A Next-Generation Approach. *International Journal of Sustainable Devlopment in field of IT*, *15*, 15.

[2] Ofili, B. T., Erhabor, E. O., & Obasuyi, O. T. (2025). Enhancing Federal Cloud Security with AI: Zero Trust, Threat Intelligence, and CISA Compliance. *World Journal of Advanced Research and Review*.

[3] Kolawole, I. (2024). Leveraging Cloud-based ai and zero trust architecture to enhance US cybersecurity and counteract foreign threats. *World J. Adv. Res. Rev*, *25*, 006-025.

[4] Muthusamy, K. (2025). Harnessing AI-powered zero trust architectures for proactive cyber defense: A comprehensive framework for future-ready network security ecosystems. International Journal of AI, BigData, Computational and Management Studies, 1(1), 24-32.

[5] Mubeen, M. (2024). Zero-Trust Architecture for Cloud-Based AI Chat Applications: Encryption, Access Control and Continuous AI-Driven Verification.

[6] Celeste, R., & Michael, S. (2021). Next-Gen Network Security: Harnessing AI, Zero Trust, and Cloud-Native Solutions to Combat Evolving Cyber Threats. *International Journal of Trend in Scientific Research and Development*, *5*(6), 2056-2069.

[7] Shoaib Hashim, M. I. (2023). Zero Trust Meets AI: Redefining Security in the Age of Advanced Cyber Threats.

[8] Dash, B. (2024). Zero-Trust Architecture (ZTA): Designing an AI-Powered Cloud Security Framework for LLMs' Black Box Problems. *Current Trends in Engineering Science (CTES), ISSN*.

[9] Ejeofobiri, C. K., Adelere, M. A., & Shonubi, J. A. (2022). Developing adaptive cybersecurity architectures using Zero Trust models and AI-powered threat detection algorithms. *Int J Comput Appl Technol Res*, *11*(12), 607-621.

[10] Kancherla, V. M. (2025). The Next-Generation Cloud Security Model: AI-Powered Zero Trust and Adaptive Threat Prevention. *International Journal of Emerging Trends in Computer Science and Information Technology*, *6*(1), 82-90.

[11] Zichen, R. (2022). AI-driven Threat Detection in Zero Trust Environments. *Available at SSRN 5146272*.

[12] Akbar, R., & Zafer, A. (2024). Next-Gen Information Security: AI-Driven Solutions for Real-Time Cyber Threat Detection in Cloud and Network Environments. *J. Cybersecur. Res*, *12*, 123-145.

[13] F. R. Alzaabi and A. Mehmood, "A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods," in *IEEE Access*, vol. 12, pp. 30907-30927, 2024, doi: 10.1109/ACCESS.2024.3369906.

[14] R. Nasir, M. Afzal, R. Latif and W. Iqbal, "Behavioral Based Insider Threat Detection Using Deep Learning," in *IEEE Access*, vol. 9, pp. 143266-143274, 2021, doi: 10.1109/ACCESS.2021.3118297.

[15] Alzoubi, Y.I., Mishra, A. & Topcu, A.E. Research trends in deep learning and machine learning for cloud computing security. *Artif Intell Rev* 57, 132 (2024). https://doi.org/10.1007/s10462-024-10776-5

[16] Hurst, W., Tekinerdogan, B., Alskaif, T., Boddy, A., & Shone, N. (2022). Securing electronic health records against insider-threats: A supervised machine learning approach. *Smart Health*, *26*, 100354.

[17] El-Kassabi, H.T., Serhani, M.A., Masud, M.M. *et al.* Deep learning approach to security enforcement in cloud workflow orchestration. *J Cloud Comp* 12, 10 (2023). https://doi.org/10.1186/s13677-022-00387-2

[18] Ajayi, A. M., Omokanye, A. O., Olowu, O., Adeleye, A. O., Omole, O. M., & Wada, I. U. (2024). Detecting insider threats in banking using AI-driven anomaly detection with a data science approach to cybersecurity. *International Journal of Cybersecurity Research*.

[19] Vadisetty, R., Polamarasetti, A., Prajapati, S., & Butani, J. B. (2023). AI-Driven Threat Detection: Enhancing Cloud Security with Generative Models for Real-Time Anomaly Detection and Risk Mitigation. *Available at SSRN 5218294*.

[20] Zhang, X., Cui, L., Shen, W. *et al.* File processing security detection in multi-cloud environments: a process mining approach. *J Cloud Comp* 12, 100 (2023). https://doi.org/10.1186/s13677-023-00474-y

[21] Herrera Montano, I., García Aranda, J.J., Ramos Diaz, J. *et al.* Survey of Techniques on Data Leakage Protection and Methods to address the Insider threat. *Cluster Comput* 25, 4289–4302 (2022). https://doi.org/10.1007/s10586-022-03668-2

[22] Gudelli, V. R. (2024). Anomaly detection in cloud networks using machine learning algorithms. *African Journal of Artificial Intelligence and Sustainable Development*, *4*(1).

[23] Arjunan, T. (2024). Fraud Detection in NoSQL Database Systems using Advanced Machine Learning. *International Journal of Innovative Science and Research Technology*, *9*(3), 248-253.

[24] Karamchand, G. (2025). Quantum Machine Learning for Threat Detection in High-Security Networks. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, *17*(02), 14-25.

[25] Singh, A. K., & Saxena, D. (2022). A cryptography and machine learning based authentication for secure data-sharing in federated cloud services environment. *Journal of Applied Security Research*, *17*(3), 385-412.

[26] Aldallal, A., & Alisa, F. (2021). Effective intrusion detection system to secure data in cloud using machine learning. *Symmetry*, *13*(12), 2306.

[27] Darzi, S., & Yavuz, A. A. (2024). Pqc meets ml or ai: Exploring the synergy of machine learning and post-quantum cryptography. *Authorea Preprints*.

[28] Ahmed, A. A., Malebary, S. J., Ali, W., & Alzahrani, A. A. (2023). A provable secure cybersecurity mechanism based on combination of lightweight cryptography and authentication for Internet of Things. *Mathematics*, *11*(1), 220.

[29] Mehmood, A., Shafique, A., Alawida, M., & Khan, A. N. (2024). Advances and vulnerabilities in modern cryptographic techniques: A comprehensive survey on cybersecurity in the domain of machine/deep learning and quantum techniques. *IEEE access*, *12*, 27530-27555.

[30] Lin, C. H., Wu, J. X., Chen, P. Y., Li, C. M., Pai, N. S., & Kuo, C. L. (2021). Symmetric cryptography with a chaotic map and a multilayer machine learning network for physiological signal infosecurity: Case study in electrocardiogram. *IEEE Access*, *9*, 26451-26467.

[31] Wu, W., Homsi, S., Zhang, Y. (2024). Confidential and Verifiable Machine Learning Delegations on the Cloud. In: Garcia-Alfaro, J., Kozik, R., Choraś, M., Katsikas, S. (eds) Computer Security – ESORICS 2024. ESORICS 2024. Lecture Notes in Computer Science, vol 14983. Springer, Cham. https://doi.org/10.1007/978-3-031-70890-9_10

[32] Mohammad, N. (2021). Enhancing security and privacy in multi-cloud environments: A comprehensive study on encryption techniques and access control mechanisms. *International Journal of Computer Engineering and Technology (IJCET)*, *12*(2), 51-63.

[33] Salam, A., Abrar, M., Amin, F., Ullah, F., Khan, I. A., Alkhamees, B. F., & AlSalman, H. (2024). Securing smart manufacturing by integrating anomaly detection with zero-knowledge proofs. *IEEE Access*, *12*, 36346-36360.

[34] Frimpong, E., Nguyen, K., Budzys, M., Khan, T., & Michalas, A. (2024, April). Guardml: Efficient privacy-preserving machine learning services through hybrid homomorphic encryption. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing* (pp. 953-962).

[35] Sheth, H. S. K., Ilavarasi, A. K., & Tyagi, A. K. (2022, May). Deep Learning, blockchain based multi-layered Authentication and Security Architectures. In *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 476-485). IEEE.

[36] . Li, J. He, P. Vijayakumar, X. Zhang and V. Chang, "A Verifiable Privacy-Preserving Machine Learning Prediction Scheme for Edge-Enhanced HCPSs," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5494-5503, Aug. 2022, doi: 10.1109/TII.2021.3110808

[37] Wang, Y., & Yang, X. (2025). Research on enhancing cloud computing network security using artificial intelligence algorithms. *arXiv preprint arXiv:2502.17801*.

[38] N. A. Hamad, K. A. A. Bakar, F. Qamar, A. M. Jubair, R. R. Mohamed and M. A. Mohamed, "Systematic Analysis of Federated Learning Approaches for Intrusion Detection in the Internet of Things Environment," in *IEEE Access*, vol. 13, pp. 95410-95444, 2025, doi: 10.1109/ACCESS.2025.3574672.

[39] Liu, R. (2023). *Resource-Aware Optimizations for Data-Intensive Systems* (Doctoral dissertation, The University of Chicago).

[40] Aramide, O. (2024). Zero-trust identity principles in next-gen networks: AI-driven continuous verification for secure digital ecosystems. *World Journal of Advanced Research and Reviews*, *23*, 3304-3316.

[41] Ajish, D. (2024). The significance of artificial intelligence in zero trust technologies: a comprehensive.

[42] Zakhmi, K., Ushmani, A., Mohanty, M. R., Agrawal, S., Banduni, A., & Kakatum, S. R. (2025). Evolving Zero Trust Architectures for AI-Driven Cyber Threats in Healthcare and Other High-Risk Data Environments: A Systematic Review. *Cureus*, *17*(6).

[43] Laghari, A. A., Khan, A. A., Ksibi, A., Hajjej, F., Kryvinska, N., Almadhor, A., ... & Alsubai, S. (2025). A novel and secure artificial intelligence enabled zero trust intrusion detection in industrial internet of things architecture. *Scientific Reports*, *15*(1), 26843.

[44] Magaji, J. (2025). AI-Driven Central Authorization Frameworks for Zero-Trust API Security in Cloud Environments.

[45] Lamia, A., Muhammad Mainuddin, M., Nusrat Jahan, S., & Sagor, A. (2022). Zero-Trust Access Control Systems by Artificial intelligence in Hybrid Cloud Environments. *BEST JOURNAL OF INNOVATION IN SCIENCE, RESEARCH AND DEVELOPMENT*, *1*(3), 45-69.

[46] Rehman, S., & Ali, A. (2024). AI-Driven Identity and Access Management: Enhancing Authentication and Authorization Security.

[47] Okoye, O. (2025). Addressing Weak Authentication like RFID, NFC in EVs and EVCs using AI-powered Adaptive Authentication. *arXiv preprint arXiv:2508.19465*.

[48] Smith, J., & Chikwarti, D. K. (2023). Self-Learning AI Models for Behavior-Driven Access Management in Zero Trust Architectures.

[49] Nzeako, G., & Shittu, R. A. (2024). Implementing zero trust security models in cloud computing environments. *World J. Adv. Res. Rev*, *24*(3), 1647-1660.

[50] Khalid, M., Abdullah, H., Haroon, F., Akhtar, E. D. S., & Shahani, S. A. (2025). Zero Trust Architecture in Cloud Security: Designing Adaptive Cyber Defense for Distributed Systems. *Global Research Journal of Natural Science and Technology*, *3*(2).

[51] Tahir, F., & Butler, J. (2021). Future-Proofing Cybersecurity: Integrating AI and Zero Trust for Comprehensive Protection.

[52] Lilhore, U.K., Simaiya, S., Alroobaea, R. *et al.* SmartTrust: a hybrid deep learning framework for real-time threat detection in cloud environments using Zero-Trust Architecture. *J Cloud Comp* 14, 35 (2025). https://doi.org/10.1186/s13677-025-00764-7

[53] Mensah, F. (2024). Zero trust architecture: A comprehensive review of principles, implementation strategies, and future directions in enterprise cybersecurity. *International Journal of Academic and Industrial Research Innovations (IJAIRI)*, *10*, 339-346.

[54] Warren, D., & Rajuroy, A. (2025). Integrating AI and Blockchain for Adaptive Access Control in Cloud Environments.

[55] Vudathala, N. R. (2025). AI-Driven Risk-Adaptive App Architecture: A Dynamic Approach to Authentication and Security in Mobile Applications. *Journal Of Engineering And Computer Sciences*, *4*(7), 911-916.

[56] Olorunlana, T. J. (2024). *Autonomous Cloud Security Orchestration for Critical Infrastructure Resilience: A Zero Trust-Based Federated Model*.

[57] Talati, D. (2022). Enhancing Multi-Cloud Security with Quantum-Resilient AI for Anomaly Detection. *Available at SSRN 5198162*.

[58] Jordan Smith, A. E. (2023). Context-Aware AI-Augmented Access Control for Dynamic MFA Environments in Critical Infrastructure.

[59] Cate, M. (2025). Building a Proactive Cyber Defense Model: Leveraging AI for Threat Hunting and Anomaly Detection in Zero Trust Architectures.

[60] Al-Otaibi, S., Ayouni, S., Sarwar, N. *et al.* AI-driven security framework for medical sensor networks: enhancing privacy and trust in smart healthcare systems. *Cluster Comput* 28, 408 (2025). https://doi.org/10.1007/s10586-024-05049-3