2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC) | 979-8-3503-6013-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCWC60891.2024.10427776

Hierarchical Software Framework for Safe Unmanned Aerial Systems Integration into National Airspace (NAS)

Jaya Preethi Mohan*, Prakash Ranganathan, Hassan Reza
School of Electrical Engineering and Computer Science (SEECS)
University of North Dakota
jayapreethi.mohan@und.edu

Unmanned aerial systems (UAS) have diverse applications and rely on GPS technology for providing information on aircraft location and position. However, a number of factors, including air conditions, satellite configuration, and interference from other sources, can impact GPS signals. GPS is also vulnerable to cyber security attacks and threats, including jamming, spoofing, and false data injection. The threats to GPS are depend on various factors and the integrity of timestamps could be compromised by cyber-attacks. This paper provides a conceptual software framework that integrates machine learning workflow, zero-trust and reinforcement learning to mitigate interference-free environment for UAS navigation. The framework is structured as a hierarchical design: sensing, communication, and control layers to host cyber physical and computing elements.

Keywords—Unmanned Aerial Systems, software framework and zero trust.

I. Introduction

Cellular and electromagnetic interference can potentially affect aircraft security and it is critical to mitigate such threats for GPS systems in the future. GPS interference detection and mitigation is an important concept in the GPS industry, as jamming and spoofing attacks can severely impact the performance of GPS systems. To address this issue, software architecture plays a crucial role in the development of GPS interference detection systems. The software architecture for GPS interference detection typically involves a combination of signal processing algorithms, machine learning models, and data analysis techniques. One approach is to use open-source software-defined architecture (SDA) to implement these techniques. This approach allows for greater flexibility and developers to customize the software to meet specific requirements. In addition, the software architecture should incorporate efficient algorithms for detecting and mitigating GPS interference. For example, a frequency analysis algorithm can be used to detect frequency-hopping attacks [1], while advanced machine learning models and reinforcement learning models can be used to detect more complex interference patterns [2]. An onboard software architecture executes a cinematographic shooting and navigation plan, covering multi-event scenarios from ground stations to visual analysis[3]. A control algorithm concept was considered for real time aerial prototypes through a software architecture[4]. Furthermore, the software architecture should also incorporate a secure and robust communication infrastructure to ensure that the GPS interference detection system can communicate effectively with other components of the GPS system. This infrastructure should be designed to protect sensitive data and prove that the system can operate even under adverse conditions on any type of UAS system.

Fig. 1. 5G signal interference and cyber-attack with UAS types.

Fig 1 is the representation of cyber security attacks and 5G signal on different types (small, mid and large scale) of UAS. Table 1 shows the related works on software framework for UAS and real time machine learning. Related works on UAS real-time software architecture influence the design and optimization of systems for enhanced data processing, reliability and adaptive responses in dynamic aerial operation environments.

979-8-3503-6013-4/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

TABLE 1. RELATED WORKS ON REAL TIME MACHINE LEARNING FRAMEWORK

S.No	Year	Research Focus	Type of architecture	Techniques used	Reference
1.	2012	Real-time software for anti-interference from signals on GPS/WAAS sensors	Pipeline/Pipe and Filter	Software-defined radio on MATLAB, Beamforming algorithm	[5]
2.	2017	Trajectory tracking in the GPS denied areas	Event-driven/ Implicit Invocation	Lazy Theta and Monte Carlo Localization	[6]
3.	2018	Autonomous navigation with obstacle avoidance	Peer to Peer	Sensor Fusion, 2D Camera Mapping	[7]
4.	2021	Simulation for real-time sensor input functions	Model View Controller	Gazebo, Path Planning Algorithm	[8]
5.	2022	Collision avoidance and localization of the autonomous micro vehicle	Event-driven	Kalman Filter for Sensor Fusion and Time Estimation	[9]
6.	2022	Identification of teaming aspects for runtime monitoring AI collaboration in smart manufacturing.	Layer/Event-driven	Knowledge Graph, Graph-based ML Engine	[10]
7.	2021	Edge-controller-based architecture for cellular networks of a major U.S. operator	Layer	Clustering and Cloud Implementation	[11]
8.	2021	Machine learning workflows of top-run libraries with continuous data streaming	Distributed and Layer	Supervised and Unsupervised	[12]
9.	2020	To detect attacks made by malicious devices	Service oriented	Regression for Cyber-attack Detection	[13]
10.	2019	To run distributed machine learning implementation with reinforcement learning	Agent-based model	Reinforcement Learning, Machine Learning, and Amazon Web Services	[14]
11.	2020	Development of deep learning techniques with safety and security.	Layer	DNN, Hypervisor Functions, HPC Simulation	[15]
12.	2020	Integration of artificial intelligence and machine learning for deployments	Microservices	Data Modeling, On-Premises and Service Premise Code, CI/CD Platform	[16]
13.	2019	Troubleshooting machine learning systems for stability	Layer	Problem Localization, Rollback on CI/CD Platform	[17]
14.	2022	Self-adaptive edge controllers for machine learning operations.	Layer	Reinforcement Learning and IoT Edge Software	[18]
15.	2021	Identification of best practices for machine learning based software architecture	Client- Server	Mathematical Models, Probabilistic Measure	[19]
16.	2020	Detection of anomalies in Industry 4.0 data	Layer	Event Detection, Feedback Control, and Mobile/Web Application	[20]
17.	2019	Network latency reduction in Health Care	Tier architecture	Fuzzy Logic, Reinforcement Learning, and Fog Computing	[21]

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

In [22], the authors demonstrate how the performance of MIMO systems outperform over systems with receive-diversity only under noisy conditions. The differences and absolute numbers decrease when co-channel interference from neighboring cells plays a major role. In 1999, the neural networks were used to predict the aircraft trajectory that is shown in [23]. The use of an autoencoder model has been implemented in the ECG signals dataset for the prediction of anomalies in [24] that proves sequential data anomaly prediction. The scope of feature extraction methods based on signal processing parameters is limited. It is due to heavy reliance on manual feature selection and prior knowledge, which changes with different work tasks. In recent times, deep learning-based intelligence methods have received extensive research on anomaly detection[25]. The current anomaly detection techniques predict with the training set contains only normal samples. Identifying the small number of heterogeneous anomalies in real time data is challenging since they frequently coexist with many normal samples. As a result, there are frequently noisy examples that are mislabeled as opposite categories in practice, and present approaches are particularly vulnerable in such scenarios.

II. Software architecture on UAS research

The research problem is the need for a software architecture that performs multiple machine learning methodologies in real time to support the UAS trajectory interference, anomaly detection and risk classification. In [26], the authors have designed a software architecture to detect radiological points with RPAS Mission Management Architecture (RIMA) on DJI drones. National Science Foundation's research [27] "An Experimental Research Platform Architecture for UAS-Enabled Wireless Networks" describes the architectural design of a research platform for conducting experiments related to UAS-enabled wireless networks. These are just two examples of academic research manuscripts related to UAS software architecture. There are many more articles and publications that discuss this topic, so it is important to conduct further research and consult additional sources for a comprehensive understanding of the research problem. There are several research problems of UAS that were addressed by different software architecture styles and methodologies. This paper shows the software architecture in an incident of GPS interference and 5G network interference with aircraft.

The software framework facilitates the deployment of machine learning models, enabling real-time analysis of GPS data and real-time analysis across the system components and layers of the UAS. The design of the software architecture supports simulation, machine learning, mapping, cloud technologies, sensor fusion, and localization techniques. The most recent publications of real-time machine learning models have discussed uses of mathematical models, artificial intelligence, data modeling fuzzy logic and image analysis.

III. Components in sensing, communication and control layers

Requirement elicitation for hierarchical UAS anomaly detection and navigational framework in layered software architecture with zero trust security is mentioned as follows.

A. Characterizing and Modeling: Sensing or Physical Layer

- Components: Battery, IMU, GPS, altimeter, and others.

B. Characterizing and Modeling: Communication Layer

- Communication Link, 2.4GHz/5.6GHz; ground and drone interactions; drone and commercial Flights
- Interactions: Drone → commercial airlines → 5G → GCS → cloud
- Visualization: Component view, system view, systems of system view

C. Characterizing and Modeling: Control Layer

- Actuators: motor control, battery power adjustments, fail-safe landing.

IV. Machine learning workflow

A. Data acquisition network

The mapping process in the sensor uses the positioning sensor for UAS navigation. It involves data acquisition, and position estimation with geospatial data[28]. The data acquisition network of this software architecture extracts the time feature from the cloud containing UAS sensors data. Time feature is extracted for the data preprocessing, apply machine learning methods for GPS anomaly detection, and risk classification. UAS position information or detection time is basically recorded in coordinated Universal Time (UTC) which requires preprocessing. The data preprocessing stage converts the default time to UNIX time format for machine learning. The data acquisition network interacts with the intrusion diagnostic system that orchestrates the supervised and unsupervised machine learning models.

B. Intrusion diagnostic system

The intrusion diagnostic system of the hierarchical framework is the main software component that operates GPS anomaly detection and risk classification. The GPS anomaly detection is a sub-component that enables multiple unsupervised machine learning models such as DBSCAN, OPTICS, Gaussian mixture models, and Hierarchical clustering for detecting the anomalies in the dataset. In this analysis, GPS anomalies are the indicator of the inconsistent time difference between the GPS messages. The messages with abnormal time difference values are considered as anomalies and it is detected by the clustering models shown in Fig.2. From the analysis, it is proven that OPTICS performs well in different scenarios that produce low to high time difference values on the GPS messages.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

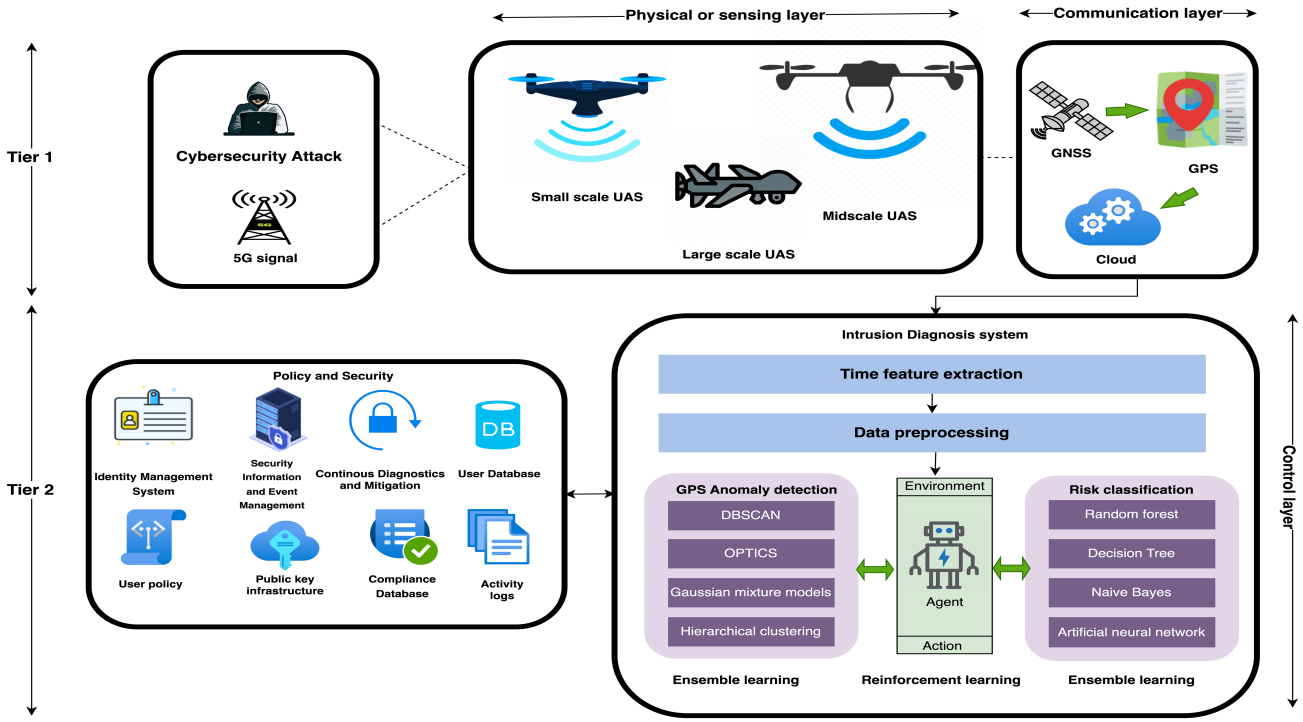

Fig. 2. Layered software architecture for UAS

In the anomaly detection research, the OPTICS clustering result was verified with False Data Injection (FDI) scenarios. In case, a model attains underperformance or failed anomaly detection, another model restores the anomaly detection capabilities and produces the result with ensemble learning.

The detected anomalies from the GPS anomaly detection system interact with a reinforcement learning agent and indicate the risk level of the detected anomalies. The reinforcement learning agent acts as an intermediate agent that inquiries about the anomaly detection results and sends the results to the risk classification stage of the intrusion diagnostic system. The risk classification system stage consists of multiple supervised machine-learning algorithms.

The classification algorithms such as random forest, decision tree, naïve bayes, and artificial neural network were experimented. All the time difference values are grouped into different labels with a threshold level of low, medium, high, lost, and zero. The difference in time with the least anomalies rate would be categorized as low and the high number of anomalies would be categorized as high risk or lost labels.

C. Parameters

Several parameters are recorded in the UAS dataset that provides UAS position, navigation, hardware component-related informants like battery temperature, voltage, speed, and more. The most common and essential data features are time, latitude, longitude and altitude. These features support various research results like interference time, location, cyber-attack time, and location from the aircraft trajectory. This influenced the selection of features for the intrusion diagnostic system. The recent development in the technologies used in the architecture considers various factors in real-time scenarios.

CI/CD workflows are efficient in cloud computing environments, data processing and real-time analysis. The research influences the use of machine learning models for real-time scenarios related to GPS and UAS. The software architecture of the system must be secured and reliable for cybersecurity applications. The components of the architecture must be designed in a way to avoid interference issues. The architecture could be designed to support CI/CD workflows for maintaining the UAS trajectory software. Multiple layers of the system or the components need to interact with the application interface, cloud for storage and maintenance. This CI/CD represents multiple technologies and applications with the help of cloud-based services such as location tracking through map-based services and cloud repositories.

V. Layered Software Architecture

Layered architecture is an architecture style that breaks the software components into multiple layers for different functions. The layer comprises the presentation layer, business layer, persistence layer, and database layer. Each layer is distinct from the other layer in the system architecture [29]. The software architecture in Fig. 2. shows the sensing, communication, and control layer connected for the intrusion diagnostic system.The layered architecture of a system supports scalability, reliability, and availability. It is shown that layer architecture is majorly used in kernel-based operations, web applications, and cloud computing [30]. The. deep learning framework use layer architecture to improve dense layer networks for classification and prediction problems [31].This architecture is used for the interoperability improvement of the software through multiple layers. The alternate layer in the layered architecture allows easy integration[32].

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

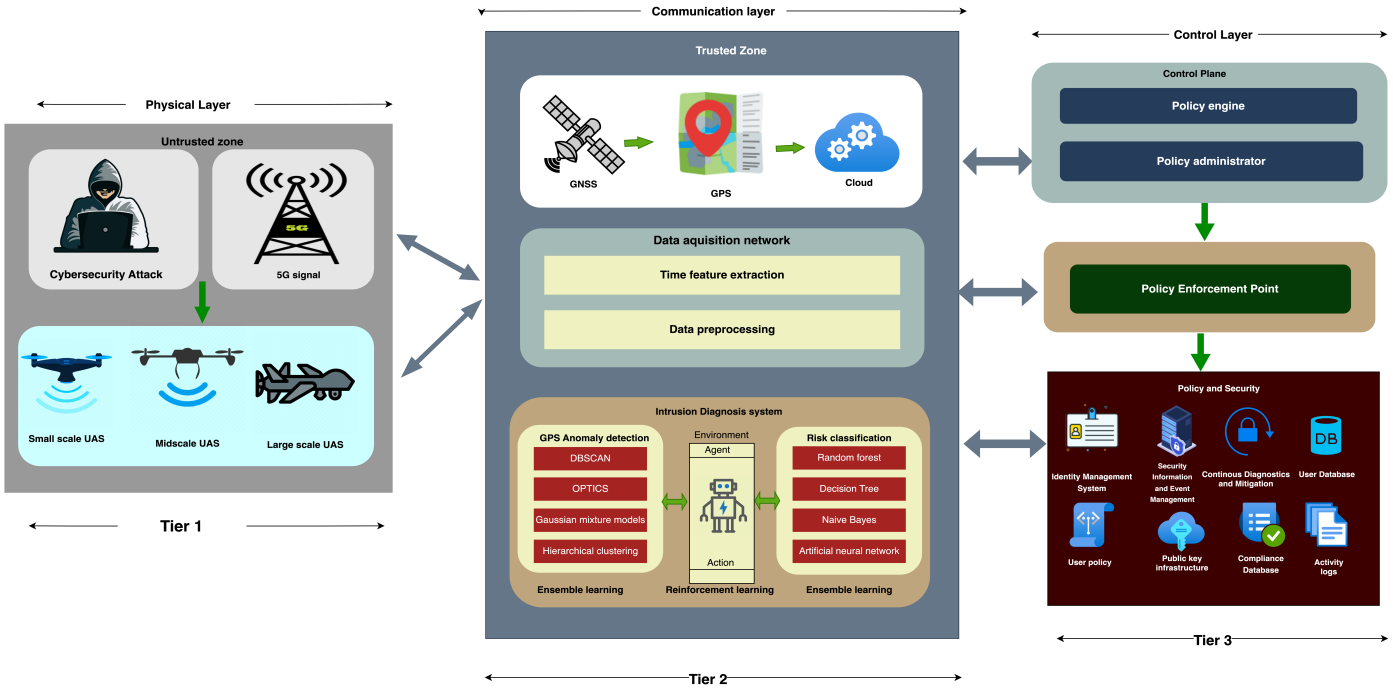

Fig. 3. Modified layered architecture with Zero trust

- In Tier 1, 5G signal and cyber attackers disrupt the different types of UAS behavior and it modifies the data transmitted in the communication layer.
- In Tier 2, the intrusion diagnostic system uses machine learning, reinforcement learning methodology for GPS anomaly detection and risk classification on the control layer.
- Policy and security components are ensured from the control layer as it can impact the real-time operation of GPS anomaly detection and risk classification.

A. Characteristics of the hierarchical framework

- The use of a zero-trust architecture ensures that every access request to the system is verified and authenticated, minimizing the risk of unauthorized access or data breaches.
- The additional layers of the modified layered architecture can be used to detect anomalies in the UAS flight logs, such as unexpected deviations from the planned flight path, altitude changes, or erratic movements.
- The modified layered architecture can also be used to classify risks based on the detected anomalies. The classification of risks can help prioritize actions and responses to potential safety hazards shown in Fig.3.

B. Zero trust architecture

The zero-trust architecture is a security standard framework that assumes zero trust between entities or systems and verifies each access request before granting access. Zero trust principles are followed in the safety-critical systems to ensure the safety and security of the cyber-physical systems. Continuous authentication can be implemented in the UAS. For example, the delivery system with UAS can use a zero-trust security framework[33]. 5G/6G analysis with machine learning practices zero-trust architecture on the O-RAN [34].

- A modified layered architecture, combined with a zero-trust architecture, can provide significant benefits over a simple layered architecture for UAS anomaly detection and risk classification with UAS flight logs.
- Each layer is designed to perform a specific task, but with additional layers to support security, privacy, and authentication.
- The use of a modified layered architecture and zero-trust architecture provides flexibility in terms of the choice of tools and technologies used in each layer, enabling developers to choose the best solutions for each task.
- The modular design of the modified layered architecture makes maintenance easier and less time-consuming, even when dealing with sensitive data such as UAS flight logs.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

C. Limitations

Layered architecture provides modularity, separation of concerns, flexibility, and simplified maintenance. It remains complex and involves latency, impact on data integrity, and performance overhead.

- Latency: The use of multiple layers can introduce latency or delays in data processing and analysis.
- Data Integrity: Each layer of the architecture must ensure data integrity and consistency.
- Integrity can be challenging, especially when dealing with large datasets of UAS flight logs.
- Performance overhead, as each layer must perform its specific operations before passing data to the next layer. The use of multiple layers can affect the performance.

UAS data is required to pass the chi-square test to be qualified for GPS anomaly detection analysis. Offline learning ensures anomaly exclusion with clustering models and online learning classifies the risk based on the anomaly detection results.

D. Use case

The architecture applies to two different use cases in real-time scenarios. The first use case authenticates the drone by zero-trust authentication and sets the trajectory planning for a safe landing. The flowchart depicted in the Fig.4. is a comprehensive guide for the management and operational flow of drone activities, adhering to regulatory and security protocols. The process begins with a 'Facilitator' component that checks the drone's registration status, which is crucial for lawful operation. If the drone is not registered, the pathway leads to a 'Communication failure' that is logged by the FAA, indicating a breach in compliance.

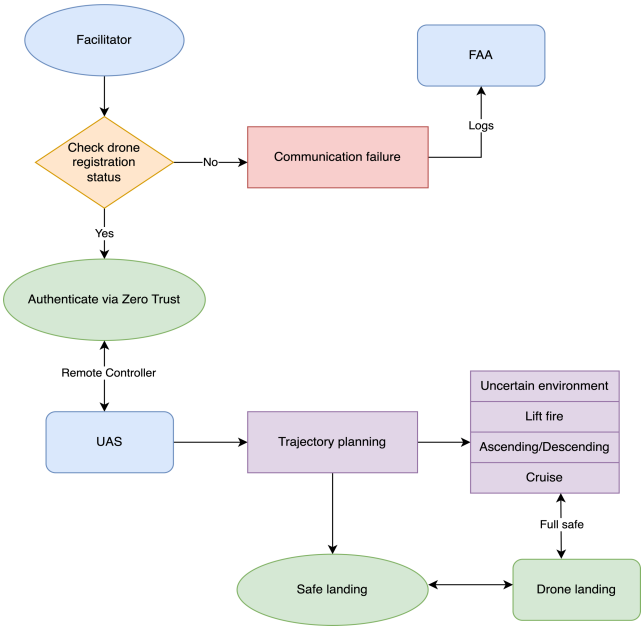

Fig. 4. Use case 1: Drone Operation Workflow

Conversely, a registered drone advances to a stringent 'Authentication via Zero Trust' framework, emphasizing cybersecurity in the control process. Subsequently, the 'Remote Controller' engages, commanding the 'UAS' (Unmanned Aircraft System). This stage is critical for 'Trajectory planning', incorporating strategic considerations for navigation in potentially unpredictable conditions. The planning considers environmental uncertainties, including 'Lift fire', 'Ascending/Descending', and 'Cruise' phases, ensuring the drone's performance is optimized for safety. navigated to land securely, marking the end of the operational cycle with the 'Drone landing'.

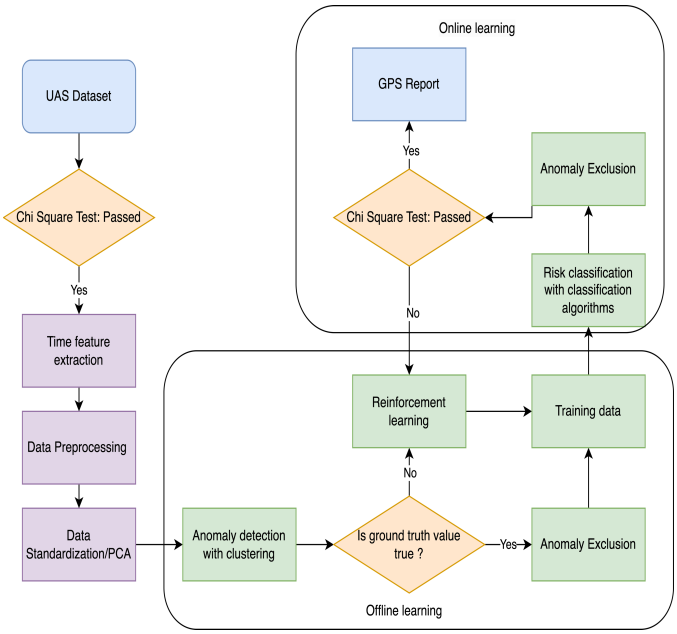

Fig. 5. Use case 2: Online and Offline Machine Learning for UAS

The use case 2 is a dual-phase machine learning process for drone data analysis, integrating online and offline learning with a Chi-square test as a pivotal decision point. Initially, the 'UAS Dataset' undergoes a 'Chi Square Test'; if passed, it moves to 'Time feature extraction' and 'Data Preprocessing', which includes 'Data Standardization/PCA'. Concurrently, in the online learning phase, a 'GPS Report' informs the Chi-square test; a pass leads to 'Anomaly Exclusion', while a failure directs the data towards 'Risk classification with classification algorithms'. Offline learning involves 'Anomaly detection with clustering'; if the 'ground truth value' is true, it circles back as 'Training data' for 'Reinforcement learning'. If false, 'Anomaly Exclusion' occurs, influencing the online learning trajectory. This recursive feedback loop between the online and offline phases fosters a robust, self-improving system where the 'Training data' refined by offline learning is cycled back to enhance the online process. Such a comprehensive system ensures that the drone's navigational and operational data is analyzed, anomalies are addressed and the overall data management is continuously optimized for accuracy and reliability.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

E. Evaluation

Table 2 represents the quality attribute of the simple layered architecture and modified layered with zero trust environment and the aimed outcomes on the modified layered architecture over simple layered architecture.

TABLE 2. COMPARISON TABLE OF MODIFIED LAYERED ARCHITECTURE

Attribute	Layered	Modified Layered with zero trust	References
Reuse	✓	✓	[35][36]
Performance	×	✓	[37][38]
Usability	✓	✓	[39][40]
Maintainability	×	✓	[41][42]
Adoptability	✓	✓	[43][44]

- The basic layered architecture lacks the performance and maintainability of the hierarchical layers.
- Modified layered architecture is at ease with the performance and maintainability attributes.
- The proposed- modified layered architecture with zero trust can be adopted for safe secured navigation and path planning.

VI. POTENTIAL SOFTWARE ARCHITECTURE TYPES

Based on the observations from Table 2, the potential software architecture types are consolidated to analyze the capabilities and limitations. The review of each architecture type would assist in choosing the most appropriate architecture type for GPS anomaly detection and 5G interference detection with aircraft. This section discusses the architecture types used in the various research implementations proposals oriented to real-time analysis, machine learning workflows and systems infrastructure.

A. Event-driven architecture

Event-driven architecture is adaptable for small and large applications and their scalability. There are two topologies for event-driven architecture which are a mediator and a broker. The mediator is an orchestrator of multiple steps in an event whereas the broker topology integrates multiple events together[29]. The event-driven architecture supports decision-making systems in real-time on handling complex events streams such as sensor-based traffic control systems. The real-time decision systems are processed through sequential event hierarchy steps[45]. It has high availability and high data volume for cloud applications in continuous streaming and analysis. It is capable of handling complex applications through microservices architecture and REST API services. It involves data retrieval from multiple services with data consistency across multiple services [46].

B. Microservice architecture

Microservice architecture is an evolving architecture in the industry that uses service components. Microservice architecture enhances agility, scalability, and reliability for systems such as e-commerce. The functionalities of microservices architecture involve vertical decomposition, scalability, fault tolerance DevOps, and deployment [33]. This architecture incorporates static and dynamic analysis for reconstruction view and run time. It works based on the decentralization methods and system-centric view. The system-centric view can be parted into four different views such as domain view, technology view, service view, and operation view[34].

C. Peer-to-Peer architecture

Peer-to-Peer (P2P) architecture is a network that contains the system shareable resources such as network elements, storage capacity other hardware resources like network link capacity, printer, and processing power. This architecture is known as distributed network architecture. It is directly accessible to other peer components without any intermediate entities. The peers are services-content and services-requestors[49]. By using the P2P architecture pattern, the system could be scalable, reliable, and performance-oriented for dynamic operation. It is used for building a topology for overlay networks and routing mechanisms on the power-law structure[50]. Recently, the P2P architecture is adopted for edge and fog computing that incorporates scheduling algorithms and distributed computing environment. It can be built on the docker containers for multiple services of edge and fog computing testing and deployment[51].

D. Client-Server architecture

Client-Server architecture processes the service of client requests and provides services for the client requests. The service is operated with the help of resources in any format like data, files, CPU, a display device and others. This architecture is independent of the hardware components of the system. The client-server architecture system is scalable horizontally and vertically. The new client workstation adding or removing on low performance with horizontal scaling and file transfer is limited with vertical scaling [52]. Client Server architecture is restricted to accessing multiple service requests at a time and is suitable for a single request. The major issues with client-server architecture are requesting services from multiple servers either directly or indirectly, managing services from multiple servers (implicit/explicit), and blocking or non-blocking responses to service requests[53]. The combination of mathematical models on information systems resulted in the ultimate reliability with client-server architecture type. The reliability feature identifies the failure components and indulges in the restoration process[54].

CONCLUSION

UAS systems are vulnerable to various threats, including interference from 5G signals and cyber-attacks. The potential threats to GPS will increase as these systems become more integrated and connected to each other components in different

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

layers of the systems. It is critical to address these vulnerabilities through effective security implementation and measures to protect against these threats. Thus, the paper discussed a hierarchical software framework for UAS on sensing, communication, and control Layers. The framework is compared with the layered architecture styles and reviews and its advantages and limitations are presented. Architectural and Analysis and Design Language(AADL) could be used as a descriptive language to model the dependencies of a various subcomponents of UAS. The modified layered architecture or proposed hierarchical framework ensures safe communication and data transmission with a threat detection mechanism.

REFERENCES

[1] C. Lavaud, R. Gerzaguet, M. Gautier, O. Berder, E. Nogues, and S. Molton, "Interception of Frequency-Hopping Signals for TEMPEST Attacks." [Online]. Available: https://inria.hal.science/hal-03027537

[2] J. Zhang and X. Wu, "RL-Based Frequency Hopping With Block-Shifted Patterns: Balancing Between Anti-Jamming Performance and Synchronization Overhead," *IEEE Trans Veh Technol*, 2023, doi: 10.1109/TVT.2023.3307341.

[3] I. Mademlis *et al.*, "A multiple-UAV architecture for autonomous media production," *Multimed Tools Appl*, vol. 82, no. 2, pp. 1905–1934, Jan. 2023, doi: 10.1007/s11042-022-13319-8.

[4] A. Offermann, J. De Miras, and P. Castillo, "Software architecture for controlling in real time aerial prototypes," in *2023 International Conference on Unmanned Aircraft Systems, ICUAS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 493–498. doi: 10.1109/ICUAS57906.2023.10156203.

[5] Y. H. Chen *et al.*, "Design and implementation of real-time software radio for anti-interference GPS/WAAS sensors," *Sensors (Switzerland)*, vol. 12, no. 10, pp. 13417–13440, Oct. 2012, doi: 10.3390/s121013417.

[6] F. J. Perez-Grau, R. Ragel, F. Caballero, A. Viguria, and A. Ollero, "An architecture for robust UAV navigation in GPS-denied areas," *J Field Robot*, vol. 35, no. 1, pp. 121–145, Jan. 2018, doi: 10.1002/rob.21757.

[7] K. Mohta *et al.*, "Fast, autonomous flight in GPS-denied and cluttered environments," *J Field Robot*, vol. 35, no. 1, pp. 101–120, Jan. 2018, doi: 10.1002/rob.21774.

[8] A. Guou, J. Gu, H. Hu, L. Wang, and D. Sun, "Gazebo-Based Simulation Environment Integration for UAS," in *AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2021. doi: 10.2514/6.2021-3011.

[9] H. Lu, H. Shen, B. Tian, X. Zhang, Z. Yang, and Q. Zong, "Flight in GPS-denied environment: Autonomous navigation system for micro-aerial vehicle," *Aerosp Sci Technol*, vol. 124, May 2022, doi: 10.1016/j.ast.2022.107521.

[10] P. Haindl, G. Buchgeher, M. Khan, and B. Moser, "Towards a reference software architecture for human-AI teaming in smart manufacturing," Association for Computing Machinery (ACM), May 2022, pp. 96–100. doi: 10.1145/3510455.3512788.

[11] M. Polese, R. Jana, V. Kounev, K. Zhang, S. Deb, and M. Zorzi, "Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks," *IEEE Trans Mob Comput*, vol. 20, no. 12, pp. 3367–3382, Dec. 2021, doi: 10.1109/TMC.2020.2999852.

[12] N. Tasgetiren *et al.*, "On the distributed software architecture of a data analysis workflow: A case study," in *Concurrency and Computation: Practice and Experience*, John Wiley and Sons Ltd, Apr. 2022. doi: 10.1002/cpe.6522.

[13] J. Roldán, J. Boubeta-Puig, J. Luis Martínez, and G. Ortiz, "Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks," *Expert Syst Appl*, vol. 149, Jul. 2020, doi: 10.1016/j.eswa.2020.113251.

[14] H. Wang, D. Niu, and B. Li, "Distributed Machine Learning with a Serverless Architecture; Distributed Machine Learning with a Serverless Architecture," 2019.

[15] A. Biondi, F. Nesti, G. Cicero, D. Casini, and G. Buttazzo, "A Safe, Secure, and Predictable Software Architecture for Deep Learning in Safety-Critical Systems," *IEEE Embed Syst Lett*, vol. 12, no. 3, pp. 78–82, Sep. 2020, doi: 10.1109/LES.2019.2953253.

[16] Y. Martel *et al.*, "Software Architecture Best Practices for Enterprise Artificial Intelligence." [Online]. Available: https://www.kdnuggets.com/2020/01/odsc-5-skills-every-data-scientist-needs.html

[17] H. Yokoyama, "Machine Learning System Architectural Pattern for Improving Operational Stability," in *Proceedings - 2019 IEEE International Conference on Software Architecture - Companion, ICSA-C 2019*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 267–274. doi: 10.1109/ICSA-C.2019.00055.

[18] C. Pahl, S. Azimi, H. R. Barzegar, and N. El Ioini, "A Quality-driven Machine Learning Governance Architecture for Self-adaptive Edge Clouds."

[19] H. Muccini and K. Vaidhyanathan, "Software architecture for ML-based Systems: What exists and what lies ahead," in *Proceedings - 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI, WAIN 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 121–128. doi: 10.1109/WAIN52551.2021.00026.

[20] L. Rychener, F. Montet, and J. Hennebert, "Architecture Proposal for Machine Learning Based Industrial Process Monitoring," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 648–655. doi: 10.1016/j.procs.2020.03.137.

[21] S. Shukla, M. F. Hassan, L. T. Jung, A. Awang, and M. K. Khan, "A 3-tier architecture for network latency reduction in healthcare internet-of-things using fog computing and machine learning," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 2019, pp. 522–528. doi: 10.1145/3316615.3318222.

[22] S. Catreux, P. F. Driessen, and L. J. Greenstein, "Simulation results for an interference-limited multiple-input multiple-output cellular system," *IEEE Communications Letters*, vol. 4, no. 11, pp. 334–336, Nov. 2000, doi: 10.1109/4234.892193.

[23] J.-M. Alliot, "Using Neural Networks to predict aircraft trajectories Charibde: hybridization of interval methods and evolutionary algorithms for solving difficult optimization problems View project Air Traffic Control View project," 1999. [Online]. Available: https://www.researchgate.net/publication/2487739

[24] S. Chatterjee, R. S. Thakur, R. N. Yadav, L. Gupta, and D. K. Raghuvanshi, "Review of noise removal techniques in ECG signals," *IET Signal Processing*, vol. 14, no. 9. Institution of Engineering and Technology, pp. 569–590, Dec. 01, 2020. doi: 10.1049/iet-spr.2020.0104.

[25] S. Yan, H. Shao, Y. Xiao, B. Liu, and J. Wan, "Hybrid robust convolutional autoencoder for unsupervised anomaly detection of machine tools under noises," *Robot Comput Integr Manuf*, vol. 79, Feb. 2022, doi: 10.1016/j.rcim.2022.102441.

[26] P. Royo, E. Pastor, M. Macias, R. Cuadrado, C. Barrado, and A. Vargas, "An Unmanned Aircraft System to detect a radiological point source using RIMA software architecture," *Remote Sens (Basel)*, vol. 10, no. 11, Nov. 2018, doi: 10.3390/rs10111712.

[27] A. S. Abdalla, A. Yingst, K. Powell, A. Gelonch-Bosch, and V. Marojevic, "Open Source Software Radio Platform for Research on Cellular Networked UAVs: It Works!," *IEEE Communications Magazine*, vol. 60, no. 2, pp. 60–66, Feb. 2022, doi: 10.1109/MCOM.001.2100394.

[28] A. R. Benjamin, ; Dennis O'brien, G. Barnes, B. E. Wilkinson, and W. Volkmann, "Improving Data Acquisition Efficiency: Systematic Accuracy Evaluation of GNSS-Assisted Aerial Triangulation in UAS Operations," 2019, doi: 10.1061/(ASCE)SU.1943.

[29] M. (W. M. Richards, *Software architecture patterns : understanding common architecture patterns and when to use them*.

[30] A. Gajbhiye and K. M. P. D. Shrivastva, "Cloud computing: Need, enabling technology, architecture, advantages and challenges," in *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*,

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.

Institute of Electrical and Electronics Engineers Inc., Nov. 2014, pp. 1–7. doi: 10.1109/CONFLUENCE.2014.6949224.

[31] C. Dong *et al.*, "An optimized optical diffractive deep neural network with OReLU function based on genetic algorithm," *Opt Laser Technol*, vol. 160, May 2023, doi: 10.1016/j.optlastec.2022.109104.

[32] A. Gulz, C. Magnusson, L. Malmborg, H. Eftring, B. Jönsson, and K. Tollmar, *NordiCHI 2008 Building Bridges : proceedings of the 5th Nordic Conference on Human-Computer Interaction : Lund, Sweden, 20-22 October, 2008*.

[33] C. Dong, F. Jiang, S. Chen, and X. Liu, "Continuous Authentication for UAV Delivery Systems under Zero-Trust Security Framework," in *Proceedings - IEEE International Conference on Edge Computing*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 123–132. doi: 10.1109/EDGE55608.2022.00027.

[34] K. Ramezanpour and J. Jagannath, "Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN," *Computer Networks*, vol. 217. Elsevier B.V., Nov. 09, 2022. doi: 10.1016/j.comnet.2022.109358.

[35] IEEE Robotics and Automation Society and Institute of Electrical and Electronics Engineers, *2014 International Conference on Unmanned Aircraft Systems (ICUAS) : conference proceedings : May 27-30, 2014, Wyndham Grand Orlando Resort, Bonnet Creek, Orlando, FL*.

[36] D. L. Van Bossuyt, B. Hale, R. Arlitt, and N. Papakonstantinou, "Zero-Trust for the System Design Lifecycle," *J Comput Inf Sci Eng*, vol. 23, no. 6, Dec. 2023, doi: 10.1115/1.4062597.

[37] N. Mungoli, "Scalable, Distributed AI Frameworks: Leveraging Cloud Computing for Enhanced Deep Learning Performance and Efficiency," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.13738

[38] T. Muhammad, M. T. Munir, M. Z. Munir, and M. W. Zafar, "Integrative Cybersecurity: Merging Zero Trust, Layered Defense, and Global Standards for a Resilient Digital Future," 2017. [Online]. Available: https://orcid.org/0009-0003-1753-3241

[39] Z. Hao *et al.*, "A novel method using LSTM-RNN to generate smart contracts code templates for improved usability," *Multimed Tools Appl*, vol. 82, no. 27, pp. 41669–41699, Nov. 2023, doi: 10.1007/s11042-023-14592-x.

[40] P. U. Chavan, M. Murugan, and P. P. Chavan, "A review on software architecture styles with layered robotic software architecture," in *Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015*, Institute of Electrical and Electronics Engineers Inc., Jul. 2015, pp. 827–831. doi: 10.1109/ICCUBEA.2015.165.

[41] D. Rodrigues *et al.*, "Service-Oriented Architectures for a Flexible and Safe Use of Unmanned Aerial Vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 97–109, Mar. 2017, doi: 10.1109/MITS.2016.2611038.

[42] F. Federici, D. Martintoni, and V. Senni, "A Zero-Trust Architecture for Remote Access in Industrial IoT Infrastructures," *Electronics (Switzerland)*, vol. 12, no. 3, Feb. 2023, doi: 10.3390/electronics12030566.

[43] X. Huang, Y. Liu, L. Huang, S. Stikbakke, and E. Onstein, "BIM-supported drone path planning for building exterior surface inspection," *Comput Ind*, vol. 153, Dec. 2023, doi: 10.1016/j.compind.2023.104019.

[44] S. Ouiazzane, M. Addou, and F. Barramou, "A Zero-Trust Model for Intrusion Detection in Drone Networks." [Online]. Available: www.ijacsa.thesai.org

[45] J. Dunkel, A. Fernández, R. Ortiz, and S. Ossowski, "Event-driven architecture for decision support in traffic management systems," *Expert Syst Appl*, vol. 38, no. 6, pp. 6530–6539, Jun. 2011, doi: 10.1016/j.eswa.2010.11.087.

[46] S. Zhelev and A. Rozeva, "Using microservices and event driven architecture for big data stream processing," in *AIP Conference Proceedings*, American Institute of Physics Inc., Nov. 2019. doi: 10.1063/1.5133587.

[47] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings - 1st International Conference on Peer-to-Peer Computing, P2P 2001*, Institute of Electrical and Electronics Engineers Inc., 2001, pp. 101–102. doi: 10.1109/P2P.2001.990434.

[48] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proceedings - 1st International Conference on Peer-to-Peer Computing, P2P 2001*, Institute of Electrical and Electronics Engineers Inc., 2001, pp. 99–100. doi: 10.1109/P2P.2001.990433.

[49] G. Proietti Mattia and R. Beraldi, "P2PFaaS: A framework for FaaS peer-to-peer scheduling and load balancing in Fog and Edge computing," *SoftwareX*, vol. 21, Feb. 2023, doi: 10.1016/j.softx.2022.101290.

[50] L. Chung, "Client-Server Architecture Clients and Servers Client/Server with File Servers Client/Server with Database Servers Web Client/Server Client/Server Groupware Client/Server with Transaction Processing."

[51] R. M. Adler, "Distributed Coordination Models for Client/Server Computing," *Computer (Long Beach Calif)*, vol. 28, no. 4, pp. 14–22, 1995, doi: 10.1109/2.375173.

[52] V. I. Potapov, O. P. Shafeeva, A. S. Gritsay, V. V. Makarov, O. P. Kuznetsova, and L. K. Kondratukova, "Reliability in the model of an information system with client-server architecture," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Sep. 2019. doi: 10.1088/1742-6596/1260/2/022007.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on August 19,2024 at 03:07:31 UTC from IEEE Xplore. Restrictions apply.