Received 29 July 2024; revised 22 September 2024; accepted 14 October 2024. Date of publication 16 October 2024; date of current version 25 October 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3481965

Detection of Zero-Day Attacks in a Software-Defined LEO Constellation Network Using Enhanced Network Metric Predictions

DENNIS AGNEW (Graduate Student Member, IEEE), ASHLEE RICE-BLADYKAS (Graduate Student Member, IEEE), AND JANISE MCNAIR (Senior Member, IEEE)

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

CORRESPONDING AUTHOR: D. AGNEW (e-mail: dennisagnew@ufl.edu)

This work was supported in part by the National Science Foundation under Grant 1738420, and in part by the University of Florida, L3 Harris Corporation—Excellence in Research Fellowship.

ABSTRACT SATCOM is crucial for tactical networks, particularly submarines with sporadic communication requirements. Emerging SATCOM technologies, such as low-earth-orbit (LEO) satellite networks, provide lower latency, greater data reliability, and higher throughput than long-distance geostationary (GEO) satellites. Software-defined networking (SDN) has been introduced to SATCOM networks due to its ability to enhance management while strengthening network control and security. In our previous work, we proposed a SD-LEO constellation for naval submarine communication networks, as well as an extreme gradient boosting (XGBoost) machine-learning (ML) approach for classifying denial-of-service attacks against the constellation. Nevertheless, zero-day attacks have the potential to cause major damage to the SATCOM network, particularly the controller architecture, due to the scarcity of data for training and testing ML models due to their novelty. This study tackles this challenge by employing a predictive queuing analysis of the SD-SATCOM controller design to rapidly generate ML training data for zero- day attack detection. In addition, we redesign our singular controller architecture to a decentralized controller architecture to eliminate singular points of failure. To our knowledge, no prior research has investigated using queuing analysis to predict SD-SATCOM controller architecture network performance for ML training to prevent zero-day attacks. Our queuing analysis accelerates the training of ML models and enhances data adaptability, enabling network operators to defend against zero-day attacks without precollected data. We utilized the CatBoost algorithm to train a multi-output regression model to predict network performance statistics. Our method successfully identified and classified normal, non-attack samples and zero-day cyberattacks with over 94% accuracy, precision, recall, and f1-scores.

INDEX TERMS Software-defined networking (SDN), cybersecurity, LEOs, GEOs, machine learning.

I. INTRODUCTION

SATELLITE communication (SATCOM) is vital for tactical military networks, with satellites often acting as relay stations in space. They receive signals, amplify them, and then retransmit them to ground entry points (GEPs). With the recent development of software-defined networking (SDN), researchers are investigating creative techniques to connect SDN with tactical SATCOM networks [1]. SDN is a networking paradigm that decouples the control plane from the data plane of network forwarding devices. This separation enables the consolidation of control responsibilities under one or more controllers, resulting in improved administration, visibility, and security of the network [2], [3].

In recent years, the private sector has engaged in initiatives for LEO SATCOM networks, including SpaceX's Starlink and Amazon's Kuiper. The Army [4] and the Department of Defense (DoD) [5] have collaborated closely with Starlink and other suppliers to develop Low Earth Orbit (LEO) constellations for military applications. The military forces in Ukraine have utilized Starlink as their primary

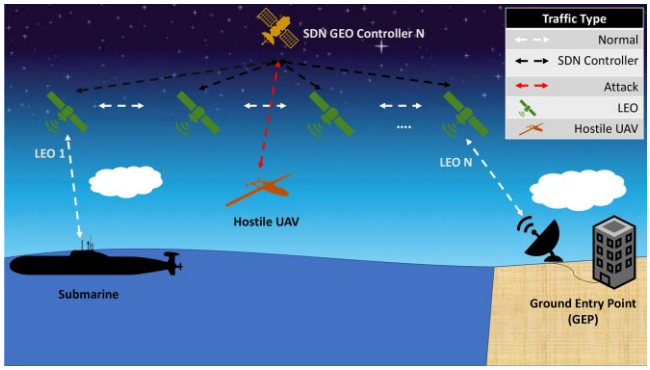


FIGURE 1. Example SD-LEO Constellation for Submarines.

communication infrastructure, with backing from both the U.S. government and Ukraine [6]. LEO constellations offer superior data transfer rates and reduced communication delay compared to conventional geostationary (GEO) satellites [2].

The infrastructure depicted in Figure 1 can be utilized for tactical SD-LEO constellations, serving military entities like submarines. Submarine crews conduct covert operations in hostile environments via discreet information transmission. Therefore, they rarely contact GEPs during long submersion and patrol. In order to establish communication with GEPs, submarines are required to emerge from the depths of the oceans to breach the surface. In order to evade detection by enemy troops, submarines must possess the capability to promptly transmit and receive information, enabling them to swiftly submerge again. Presently, submarine crews commonly communicate by use of GEO satellites positioned at a distance of approximately 36,000km, resulting in a propagation delay of around 250 milliseconds. Future tactical networks might potentially employ LEO constellations positioned at distances of 1500km or less, resulting in a propagation delay of around 30 milliseconds or less [2].

Submarine communication links necessitate increased security measures, in addition to reduced latency. Messages are classified with a high level of security and are frequently time-sensitive, making them attractive targets for malicious entities. A common strategy employed by malicious entities, such as hostile unmanned aerial vehicles (UAVs), involves launching denial-of-service (DoS) attacks [7], [8]. These attacks aim to overwhelm a target by flooding it with excessive network traffic, with the intention of either reducing the available data transfer rate or causing a complete system failure. UAVs offer a risk to SATCOM networks, requiring further investigation, as demonstrated in this study, to tackle this crucial aspect of communication [9]. There is a requirement for network defense models that are capable of identifying and reducing the impact of these attacks. Prior studies [1], [10] have explored the creation of shipboard networks using SDN or proposed SDN-based SATCOM networks for various tactical settings. Additional studies [11] have devised methods to mitigate DoS attacks on ground stations. Nevertheless, these prior studies fail to consider the use of an SD-LEO constellation network for submarines, nor a way for detecting cyber threats in these networks such as DoS attacks.

To address this issue, our prior work [12] presented a relay SATCOM network for submarines on patrol, as well as detection and identification of attack strength framework for DoS attacks. By employing the extreme boosting (XGBoost) machine learning algorithm, our model attained > 97% accuracy, precision, recall, and F1-scores in detecting and classifying different levels of DoS attack intensity. In our study [12], we examined the SD-LEO forwarding plane as a network consisting of multiple-server forwarding queues, specifically following the M/M/C model, where C represents the number of queues. We utilized Simcomponnet, a network traffic simulation program based on the SimPY process-based discrete-event simulation framework, to simulate the network traffic of both normal and varying DoS attack strength classes [13]. During the simulation, we measured queuing theory-based metrics [14] such as the average interarrival time $\overline{(IAT)}$, transmission delay $\overline{(TD)}$, and packet count of packets $\overline{(PC)}$. Using these measurements, we applied the XGBoost ML model to distinguish between regular traffic and DoS attacks. The outcome was an average accuracy of 97% in recognizing and categorizing both regular (non-attack) and attack traffic. The DoS traffic was classified into distinct levels of severity such as 0%, 10%, 50%, and 90% loss of traffic.

One limitation of our prior model is that it requires pre-labeled data for $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$ in order to train the XGBoost algorithm and does not consider controller or GEP security. Since SDN controllers and GEP manage the entire framework, they are prime targets for attackers; nevertheless, ML techniques can be utilized to guard against cyberattacks [15], [16], [17], [18]. Obtaining labeled training data may not always be possible in tactical deployments. Submarine communication is often limited by their communication patterns. Zero-day cyberattacks [19], [20] are new and unknown, therefore pre-trained machine-learning models cannot detect and protect against them.

Past research [19], [21], [22], [23], [24], [25] has conducted queuing analysis on M/M/1 or M/M/1 (single server) models, which do not adequately account for interconnected multi-server satellite constellations (M/M/C). Furthermore, the intrusion detection systems (IDSs) developed in these studies have utilized unsupervised learning and deep learning models that assume the availability of training data and are susceptible to false positives. Our proposed solution is to rapidly and dynamically generate predictive training data to discover future controller architecture cyberattacks. We utilize predictive queuing analysis to calculate the mean interarrival time $\overline{(IAT)}$, mean transmission delay $\overline{(TD)}$, and mean packet count $\overline{(PC)}$ for each controller in the network. By employing our queuing analysis, the network operator can predict the normal functioning of the controller and GEP architecture.

In this study, we employed a Jackson network open (JNO) queueing model [26] to depict the interconnected queues,

where the output of one queue serves as the input for another in a linked manner, symbolizing the cooperative back-and-forth communication between the controllers and GEPs. The JNO model provides a product-form solution for analyzing and evaluating network performance [27]. In order to showcase the precision and efficiency of our queuing analysis in identifying zero-day attacks, we provide a case study with a one-hour simulation of controller and GEP architecture traffic. This simulation includes randomized zero-day cyberattacks within the framework. By employing our advanced predictive queuing analysis, we can accurately identify the occurrence of zero-day cyberattacks with accuracy, precision, recall, and f1-scores over > 94%.

To our knowledge, no prior research has suggested utilizing queuing analysis to predict network performance metrics for an SD-SATCOM controller architecture to improve security against zero-day attacks. Furthermore, there has been no previous research that has shown the efficacy of utilizing queuing analysis predictions in conjunction with a ML model to detect zero-day cyberattacks. Thus, this work presents the following contributions:

- A study is done to examine the queuing behavior of an M/M/C Jackson open queuing SD-LEO management layer for submarines on patrol.
- A novel predictive queuing analysis is proposed for accurately predicting the network performance metrics of average inter-arrival time *($\overline{IAT}$)*, average transmission delay *($\overline{TD}$)*, and average packet loss probability *($\overline{PC}$)* for the SD-LEO management layer.
- A case study shows the prediction's precision, swiftness, and ability to detect zero-day threats. The study successfully detects zero-day cyberattacks with over 94% accuracy, precision, recall, and f1-scores.

A list of acronyms and meanings that are used in the paper is provided in Table 1. The rest of this paper is organized as follows. Section II presents an analysis and examination of previous research and studies. Section III presents an overview of SATCOM in tactical networks, queuing modeling for satellites, and SATCOM cyberattacks. Section IV describes the SD-LEO and management architecture that serves as the foundation for the queuing analysis. Section V describes the methodology of this work. Section VI details our simulation study using MATLAB with the aerospace, mapping, and satellite communications toolboxes. The results of our case study are shown in Section VII. Finally, Section VIII provides the concluding remarks of the study.

TABLE 1. List of acronyms and definitions.

Acronyms	Definitions
SATCOM	Satellite Communications
SDN	Software-Defined Networking
SD-SATCOM	Software Defined SATCOM
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
GEO	Geostationary Orbit
GEP	Ground Entry Point
DoD	Department of Defense
UAV	Unmanned Aerial Vehicle
QBD	Quasi Birth Death
MQL	Mean Queue Length
GS	Ground Station
DTN	Delay Tolerant Network
ML	Machine Learning
Catboost	Categorical Boosting
LightGBM	Light Gradient Boosting
XGBoost	Extreme Gradient Boosting
MAPE	Mean Absolute Percentage Error
DAE	Deep Auto-Encoder
VAE	Variational Auto-encoder
TS-ANN	Two-Stage Artificial Neural Network
λ	Arrival Rate of Packets
μ	Service Rate of Packets
$\overline{IAT}$	Average Interarrival Time of Packets
$\overline{TD}$	Average Transmission Delay of Packets
$\overline{PC}$	Average Packet Count
IDS	Intrusion Detection Systems
JNO	Jackson Network Open Queuing
M/M/1	Single Server Queuing Model
M/M/C	Multi-server Queuing Model
M/G/1	Single General Distribution server
ARP	Address Resolution Protocol
CPU	Central Processing Unit
DDoS	Distributed Denial-of-Service
API	Application Programming Interface
ONOS	Open Network Operating System
SAGIN	Space-Air-Ground-Integration Network
FL	Federated learning
SQM	Signal Quality Monitoring
GNSS	Global Navigation System
HPBW	Half-Power Beam Width
FOV	Field of View
L_{prop}	Path loss
QT	Queuing Theory

II. RELATED WORK

SATCOM networks are vulnerable to various cyberat- tacks, including zero-day attacks that exploit previously unknown vulnerabilities in the network to carry out novel kinds of cyberattacks [19], [20], [28]. ML technologies are particularly efficient methods for combating these dangers [29], [30], [31]. Supervised ML methods, while effective in countering known attacks, are not capable of effectively countering zero-day attacks due to their reliance on training data that accurately represents the attack [32]. Unsupervised learning models have the ability to identify abnormalities in network traffic for the purpose of detecting zero-day attacks. However, it is widely recognized that these models often produce a significant number of incorrect identifications, both in terms of false positives and false negatives [33], [34], [35], [36], [37].

To address this problem, we have proposed implementing a predictive queuing analysis to predict the anticipated network metric values of the SD-SATCOM constellation satellites. The values are subsequently utilized to train the CatBoost-supervised ML model [38] for predicting the expected theoretical values. An attack sample is categorized as anomalous (i.e., zero-day cyberattack) if the obtained values surpass the threshold. To accurately forecast the values, a queuing analysis is conducted using the M/M/C Jackson open network queuing theorem. Prior queuing analysis has employed queuing models that predominantly utilize simplistic M/M/1 or M/G/1 (single server models) queuing methodologies, which do not correctly capture the traffic demands of a dynamic multi-satellite constella- tion comprising several interconnected satellites. Moreover, previous research has predominantly employed computationally demanding combinations of neural networks and unsupervised learning models to identify abnormal patterns of activity and classify them as zero-day attacks which may not be feasible in tactical network deployment due to the limitation of computing resources.

This section presents an examination of current, contemporary techniques that are relevant to this work. To our knowledge, there has been no previous research that has developed a predictive queuing analysis for an SD-SATCOM network to detect zero-day attacks. As a result, we include two literature review subsections in this related work section that perform a comparison examination of our proposed framework and existing methodologies or alternative approaches. Initially, we analyze the current body of research on queuing analysis for SD-SATCOM networks and highlight the distinctions between prior investigations and our own. Next, we analyze prior research that employs zero-day cyberattack defense tactics for satellite networks and highlight the differences between these studies and our own. Finally, we outline the similarities and differences between this study and prior research [19], [21], [22], [23], [24], [25], emphasizing our unique contribution to the state-of-the-art.

A. LITERATURE REVIEW OF QUEUING ANALYSIS FOR SD-LEO

Reference [21] provide an analytical queuing model to assess the performance of SD-SATCOM. The authors integrate and consider the combination of Delay Tolerant Network (DTN) and OpenFlow technologies in their queuing model. Furthermore, the authors construct their queuing model by applying Jackson's theorem to the LEOs, GEOs, and MEOs, treating them as a network of queues. During their examination of queueing, a single GEO satellite is designated as the controller, while the MEO satellite facilitates communication between the GEO and LEO satellites via a store-and-forward DTN mechanism. The authors employ the POX as the controller at the GEO node and utilize OpenvSwitch as the switches at the MEO nodes. In order to verify the accuracy of their queueing analysis, they conduct a simulation using Satellite Tool Kit (STK). The satellite link parameters were set according to the Tr constellation [39]. Furthermore, they utilize Linux Traffic Control and Netem to effectively oversee and control network traffic. Their comparison of their analysis with the simulation results illustrates the validity and precision of their methodology in evaluating SD-SATCOM performance. However, in contrast to our proposed study, their modeling analysis only takes into account a single controller method and neglects to discuss a distributed SD-SATCOM controller design. Furthermore, a solitary GEO controller would consistently be beyond the reach of all MEOs and LEOs in the system, resulting in extra communication delays between LEOs and the GEOs. Having only one controller would create a vulnerability in the architecture, which might have severe consequences, particularly for the secure communication of submarines. Furthermore, the authors of [21] extend their research to determine the duration of time that a file remains in the network as it spreads, while we conduct our queuing study to predict metrics for the detection of zero-day cyberattacks.

Reference [22] presents a model that accounts for time-varying channels in order to aggregate traffic across networks in LEO constellation networks. The authors' queuing model incorporates the variability of realistic satellite channels, which may experience times of extremely poor connectivity due to Land Mobile Satellite (LMS) channels, in both the ground-to-satellite and satellite-to-ground links. Their model comprises three LEO satellites and two ground stations that communicate using time-varying LMS channels. The model utilizes two-dimensional Markov chains for LMS links and M/M/1 queuing to ensure reliable inter-satellite communications. In order to validate their model, the authors employ event-driven simulation implemented in C++ to assess the precision of the model and analyze queuing and end-to-end delays across several scenarios. Furthermore, they utilize Quasi-Birth-Death (QBD) processes as a theoretical framework to evaluate the queuing delay in LEO satellite connections. This framework is then extended to include communication channels within LEO constellations. The model is validated through a comparison with system-level simulations that employ empirical channel statistics. In addition, they assess the distributions of end-to-end delay and analyze the impact of background traffic. Their simulation results confirm the validity of their model and illustrate their ability to appropriately validate the LMS channels in a time-varying LEO constellation. However, in contrast to our suggested framework, their suggested framework focuses on assessing LMS channels from ground-to-satellite and satellite-to-ground, but neglects to take into account a more comprehensive constellation management architecture and synchronization utilizing the cloud. Their suggested methodology accurately predicts the number of packets received at LEO, but it does not take into account other metrics such as average inter-arrival time $(\overline{IAT})$ and transmission delay $(\overline{TD})$ values. In contrast to our work, the focus of the mentioned work is solely on evaluating LMS channels, rather than predicting queuing metrics to detect

zero-day cyberattacks. In addition, their approach does not include the modeling of SDN communication by integrating a controller(s) within the constellation, as it was not within the scope of their research.

Reference [23] presents a time-limited M/G/1 model to account for the intermittent transmission between satellites caused by the fluctuating nature of satellite communica- tion. The authors discuss the challenges related to system modeling and performance analysis in satellite networks, focusing particularly on the acquire-store-forward process of traffic. Traditionally, this process is depicted as a queuing system, with the transmitter serving as the server and the buffer acting as the queue. However, the intermittent nature of satellite connections poses a barrier to adopting traditional vacation laws, as the connection's activity is not influenced by the buffer state. Therefore, the authors extend the queueing model from the usual M/M/1 to M/G/1 to accommodate a more general server queuing distribution. The authors obtain steady-state equations and establish lower limits for performance parameters using transient analysis. The simulation results clearly demonstrate the efficacy of their approach. In order to verify the accuracy of their find- ings, the authors compare the outcomes of their simulations with the theoretical predictions. They provide evidence for the soundness of analysis and the precision of estimating the Mean Queue Length (MQL), Mean Waiting Delay (MWD), and traffic intensity. The substantial resemblance between the analytical and simulation outcomes validates the robustness of their methodology. However, in contrast to our proposed research, their study does not take into account a satellite controller architecture for managing a constellation. Instead, their concentration is on inter-satellite communication. Although the M/G/1 model generates the server distribution, it fails to adequately represent the interconnected nature of an M/M/C Jackson network of satellites. Furthermore, the authors have the capability to anticipate MQL, MWD, and traffic intensity. However, they did not take into account *IAT*, *TD*, and *PC*. Furthermore, the architecture does not consider the synchronization of SDN controllers or the prediction of metrics for defending against zero-day cyberattacks, as these topics were outside the scope of their work.

B. LITERATURE REVIEW OF ZERO-DAY CYBERATTACK DEFENSE FOR SATCOM NETWORKS

Reference [19] focused on enhancing the security of SATCOM communication systems against DoS and zero- day cyberattacks carried out by cyberattackers. A proposed solution to handle these risks is the implementation of a comprehensive deep federated learning (DFL)–based threat detection model. This model aims to proactively identify intrusions in SATCOM networks by utilizing decentralized on-device data. Importantly, the privacy of this data is preserved throughout the process. This approach utilizes a decentralized data-level preprocessing (DLP) system to ensure that the original data is hidden while giving well-processed, statistically altered data for effective threat identification. The proposed model performs federated learning iterations using a novel deep auto-encoder (DAE) structure. It stores local data in safe repositories and only shares the learned weights with the central federated learning server. Federated learning (FL) is a promising solution that addresses the difficulties of classic centralized IDSs and provides unique advantages. FL facilitates the collaborative training of ML models using decentralized data sets. This allows individual participants, such as satellites, ground stations (GS), or end-user devices, to train the Intrusion Detection System (IDS) model locally without compromising the privacy of their sensitive data by sharing it with a central server or data collectors. The main objective is to improve the security and efficiency of a traditional IDS model in a distributed way by utilizing FL, while also addressing issues such as data leakage and the effectiveness of model training.

The DAE model is a neural network that is trained to compress and then reconstruct its inputs, enabling the network to acquire important ideas and correlations among the input data. By exclusively training the DAE model using normal data, it acquires expertise in accurately recreating normal data but faces difficulties in reconstructing atypical data, such as zero-day attacks. The performance of the proposed DFL-IDS model is assessed by conducting evaluations on the UNSW-NB15 and Bot-IoT datasets. The results are then compared to those obtained using the centralized DAE approach. Their DFL-IDS achieves similar detection performance as the centralized DAE, while ensuring data confidentiality and delivering optimal accuracy rates for detecting attacks. However, in contrast to our suggested framework, one limitation of this work is the utilization of the DAE unsupervised learning method to identify the existence of zero-day cyberattacks by assigning anomalous (zero-day attack) labels to data that it finds difficult to reconstruct. As previously stated, unsupervised methods have demonstrated an inclination for generating false positives, which in turn leads to increased investigation and delays for network operators whenever a false positive is detected. In addition, the author's suggested methodology relies on the availability of "normal", nonattack traffic data to train the DAE. However, when it comes to submarine SATCOM communication, the availability of training data for routine occurrences may be scarce due to the uncommon use of the infrastructure. By employing the suggested queuing analysis outlined in this paper, the network operator can generate normal training data for identifying zero-day cyberattacks without making any assumptions about the availability of training data in advance.

Reference [24] investigates the composition of space-air-ground integration networks (SAGIN) and suggests a dedicated collaborative federated learning (FL) framework for SAGIN to identify abnormal traffic caused by cyberattacks, such as zero-day cyberattacks. A specialized traffic detection system is proposed that is specifically designed

to satisfy the unique requirements and characteristics of SAGIN. This approach addresses issues such as the need for manual labeling and feature extraction by enhancing deep learning algorithms and utilizing semi-supervised learn- ing approaches. The developers utilize the Hierarchical Spatiotemporal Feature Learning Network (HAST-NAD) to build their ML model. This model has achieved detection accuracy of over 99% in various public datasets. They further enhance the model by incorporating a one-dimensional convolutional neural network in the input layer. This addition helps extract the spatial features of the data packet sequence. This approach has greater physical significance and requires less computational effort. A one-dimensional convolutional neural network automatic codec is constructed by utilizing the skip connection method within the ladder network. This stage transforms a portion of the initial network space feature extraction into a semi-supervised approach. Auto-encoders facilitate independent training while minimizing cognitive load. Ultimately, the various data packet characteristics are transferred in a certain order to the LSTM neural network following the process of extracting the features. As a result of modifying the HAST-NAD dataset, the authors achieved a detection accuracy, precision, and recall of over 98% for anomalous traffic on the ISCX2012 dataset. However, in contrast to our work, the authors assume that training data is accessible for first labeling certain samples required for their semi-supervised model. The data accessibility of submarine SATCOM networks may be restricted due to the swift deployment and scarce communication. Moreover, the utilization of deep learning techniques requires significant computational resources [40]. Consequently, implementing the authors' proposed model may not be practical for constrained computational devices utilized in tactical communication.

Reference [25] provide a new training feature set that integrates power and Signal Quality Monitoring (SQM) metrics from a single-antenna Global Navigation Satellite System (GNSS) receiver. This feature set is designed to identify spoofing and zero-day cyberattacks. The researchers have created a unique collection of attributes that integrate power and Signal Quality Monitoring (SQM) measurements from a GNSS receiver with a single antenna. The authors provide a Two-Stage Artificial Neural Network (TS-ANN) that utilizes this feature set in combination with multi-correlator finger values to achieve efficient spoof detection. In order to identify zero-day cyberattacks, the authors propose a zero-day attack detector that relies on unsupervised representation learning. This is achieved by utilizing a Variational Autoencoder (VAE) that is trained solely on genuine (nonattack) datasets. By doing so, the detector becomes more proficient in detecting new and previously unseen attack patterns. The authors demonstrate the effectiveness of their proposed approaches by comprehensive experiments on various attack scenarios using publicly available datasets, such as the TEXBAT dataset. Their suggested model yields results on the TEXBAT datasets, showcasing that their TSANN achieves a detection probability (PD) beyond 99% for comparable test datasets. During more complex attack situations, such as the DS-7 attack dataset, the performance may decline to 50.68%. The zero-day detector consistently maintains a detection probability of over 92.5% for zero-day cyberattacks, demonstrating its capacity to effectively detect previously unknown attacks. However, in contrast to our proposed approach, their strategy necessitates and assumes pre-existing training samples to train the unsupervised model, VAE. As previously stated, the availability of this training data may not be practical in situations where submarines on patrol have limited communication capabilities. In addition, the authors intentionally omit any discussion of leveraging SDN for mitigation as it falls outside the paper's scope.

Reference [41] presents the STOP framework to mitigate location spoofing attacks on delivery vehicles employing global positioning system (GPS) satellite networks. Current vehicle inspections, essential for safety, are infrequent and protracted, frequently necessitating extended durations to obtain necessary data when a vehicle is chosen for inspection. The article presents STOP, an advanced vehicle inspection assistance system aimed at enhancing efficiency by providing inspectors with location tracking of cars. Although extensively utilized, GPS is not regarded as entirely secure [42]. A GPS spoofing attack seeks to mislead GPS receivers by transmitting false signals. These are designed to mimic standard GPS signals and can be altered to enable the receiver to determine its location as intended by the attacker. Affordable GPS spoofing devices are readily accessible in the market, enabling an attacker to procure them with ease. STOP is the first system of its kind, incorporating tamper- resistant records that thwart location spoofing assaults. It functions through mobile devices and a central computer, allowing authorities to pre-select vehicles for inspection and obtain requisite data beforehand. The solution enables inspectors to authenticate and endorse the location history of each vehicle, thereby ensuring transparency and security. A prototype was developed on the Android platform and assessed under real-world situations, emphasizing location accuracy, response times, and Bluetooth connection during inspections. The system utilizes the location data from onboard mobile devices to monitor incoming vehicles at inspection sites and employs location verification to digitally authenticate the location chain and inspection data. The evaluation discussion of this prototype yielded insights into the viability of this system and the location retrieval capabilities of Android devices. However, in contrast to our work, the authors' work focuses on protection against GPS spoofing attacks and neglects discussion of their technique's ability to defend against novel, zero-day cyberattacks that aim to spoof GPS systems, since this concern falls outside of the scope of their work. This research describes a GPS spoofing attack security prototype, concentrating mostly on ground tracking devices and minimally on satellite networking security, as the latter falls outside the scope of this study as well.

C. SUMMARY

To our knowledge, there has been no prior research suggesting the use of queueing analysis for predicting the network performance metrics of an SD-SATCOM network in order to improve security against zero-day cyberattacks. Additionally, no previous studies have provided case study data to demonstrate the practicality and precision of detecting these attacks. This section provided a thorough overview of the most recent research in queueing analysis for SATCOM networks and protection methods against zero-day cyberattacks. The majority of the queuing studies primarily focus on the utilization of a common M/M/1 or M/G/1 model or a single controller model. These studies do not take into account the potential application of the analysis to enhance ML-based zero-day cybersecurity detection or the queuing characteristics of a distributed controller architecture. Prior work use queuing analysis to primarily analyze the SD-SATCOM network, LMS time-varying channels, or inter-satellite communication.

Furthermore, we presented a comprehensive summary of the current field of research about safeguarding SATCOM networks against zero-day intrusions. Current research primarily concentrates on the utilization of IDSs through the implementation of distributed federated learning and/or deep learning models. Prior research primarily concentrates on unsupervised or semi-supervised ML algorithms for identifying anomalies in network traffic or ground devices spoofing attack defenses. Nevertheless, unsupervised ML models tend to generate a significant number of incorrect positive results, as evidenced by several studies [33], [34], [35], [36], [37]. Semi-supervised ML models aim to address this issue by selectively labeling a subset of data for training, hence minimizing the occurrence of false positives and enhancing overall performance. Nevertheless, both unsupervised and semi-supervised ML models rely on the presence of training data consisting of regular, non- attack samples to effectively identify anomalies. It is worth noting that deep learning methods are notorious for their high resource requirements [40] which may pose challenges in the context of tactical networks due to resource constraints. In the context of submarine tactical SATCOM communications, the availability of this resource may not be easily accessible and could be restricted. Our study improves upon the current state-of-the-art by developing a predictive queueing analysis that can quickly generate training data for the detection of zero-day cyberattacks in SD-SATCOM networks. This is achieved by dynamically producing nonattack samples. Additionally, we showcase a case study that demonstrates the efficiency and precision of our machine-learning approach in identifying these threats.

III. BACKGROUND

A. SOFTWARE DEFINED NETWORKING FOR LEO CONSTELLATIONS

SDN lets users directly manage data forwarding in network nodes. The networking industry favors SDN because it facilitates network device programming. Stanford University coined the term "SDN" to describe a software protocol that allows servers to direct network switches on packet transmission destinations [43]. The initial SDN standard was OpenFlow. OpenFlow in SDN systems provides a communication protocol for the SDN controller to communicate with network devices like switches and routers, both physical and virtual (hypervisor-based). As depicted in figure 2, SDN can be characterized by three planes:

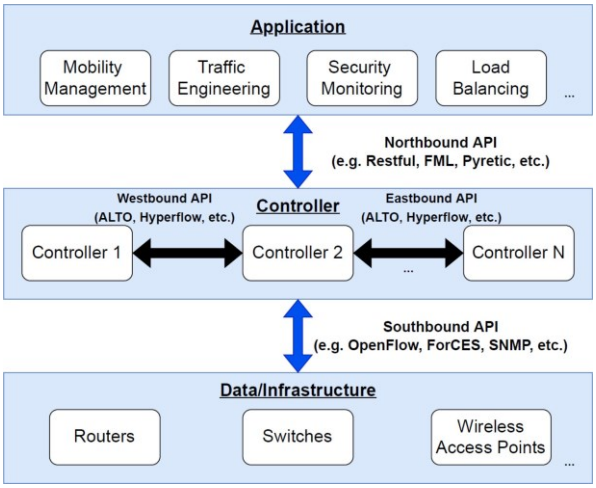


FIGURE 2. General SDN Architecture [54].

1) *Application Plane:* It encompasses SDN applications for the purpose of network management, policy enforcement, and provision of security services.
2) *Control Plane:* This is a centralized control framework that operates the network operating system and offers hardware abstractions to SDN applications in a logical manner. A flow in SDN refers to a sequence of instructions that govern the transmission of packets between a source and a destination. Controllers (SD-GEOs) populate the flow tables of forwarding devices (SD-LEOs) with the flow entries.
3) **Data Plane:** A set of forwarding components employed to transport traffic flows based on instructions received from the control plane.

The infrastructure layer consists of routers, switches, and access points, as depicted in the diagram. The data plane is formed by this layer, which represents the physical network equipment in the network (e.g., relay/forwarding SD-LEOS). Application programming interfaces (APIs) facilitate the transmission of information across different layers of SDN architecture. The controller (e.g., SD- GEO) utilizes southbound APIs such as OpenFlow [43], ForCES [44], PCEP [45], NetConf [46], or IRS [47] to establish communication with the data plane and provide oversight. This capability allows SDN controllers to manage and monitor networks globally, improving cybersecurity by gathering data for ML models to detect attacks. When available, several controllers communicate with each other

via Westbound and Eastbound APIs, such as ALTO [48] or Hyperflow [49]. The highest layer is the application plane. At this level, the network operator can employ functional applications to enhance energy efficiency, con- trol access, manage mobility, and/or ensure security (e.g., ML algorithms). The application layer utilizes northbound APIs, including FML [50], Procera [51], Frenetic [52], and RESTful [53], to establish communication with the control layer. The network operator can utilize these APIs to efficiently communicate the necessary changes to the control layer, thereby empowering the controller to implement the required adjustments to the infrastructure layer.

In contrast to SDN, conventional networks feature forwarding logic managed by forwarding agents. Each forwarding device must be changed to adjust the network. These limits limit network management rules in traditional networks and provide scalability concerns for SD-LEO networks. SDN lets network operators instantly change SD-LEOS forwarding data flows. This makes adapting to traffic and security changes easier.

B. SATCOM CYBERATTACKS

The main goal of attackers is to maliciously disrupt SATCOM networks in order to gain personal advantages. Studies on SATCOM have investigated various risks asso- ciated with cyberattacks aimed at SATCOM networks. Network operators must be cautious in order to reduce the possible impact of various attacks on their SD-SATCOM network. Guo et al. [55] define threats to SATCOM network security as follows:

- *Jamming Attack:* Jamming attacks are classified as active attacks that are designed to interfere with a network node's communication channel. This is accomplished by releasing strong signals or packets, which leads to a decrease in the signal-to-noise ratio (SINR). As a result, it disrupts regular communication with external nodes and causes a loss of availability.
- *Eavesdropping Attack:* Eavesdropping refers to the act of silently intercepting and accessing the self-established information exchange among nodes without disrupting the network. Eavesdroppers have the abil- ity to utilize the gathered data to deduce sensitive information.
- *Spoofing Attack:* Spoofing is an attack that impersonates identity and acquires confidence through authentication. To communicate with the target, the attacker impersonates a reputable satellite, exploiting authentication system flaws. Attacker obtains sensitive information or commits other offenses. Spoofing signals are made to look real, so the recipient tracks them inadvertently. Due to imprecise pseudo-distances and satellite coor- dinates, they mislocate the receiver and increase signal propagation delay.
- *Denial of Service (DoS) Attack:* DoS attacks aim to render network services inaccessible by obstructing legitimate users' access to certain network resources, thereby causing system overloads and hindering the fulfillment of legitimate requests. This is frequently accomplished by flooding a target satellite with false packets in an attempt to exhaust the limited resources on board.

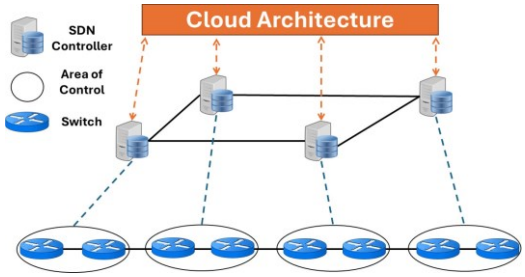


FIGURE 3. Flatly Distributed SDN Controller Architecture.

IV. SD-LEO NETWORK ARCHITECTURE

Figure 1 illustrates the utilization of the SD-LEO constellation by a submarine to communicate with a GEP. The submarine will ascend from the depths of the water and establish a connection with a nearby SD-LEO satellite. The SD-LEOs route traffic through each other to the destination GEP by using the routing logic from the SD-GEO controller within range. The GEP will establish a two-way contact with the submarine until the communication is completed, after which the submarine will submerge again. To achieve nearly worldwide oceanic communication coverage and enhance controller visibility for optimal routing, we have enlarged and enhanced our existing framework based on our past work [12].

Our proposed approach utilizes controllers that operate in a flatly distributed manner, as depicted in Figure 3. Every SD-GEO controller is responsible for overseeing a certain region of the forwarding SD-LEOs. These controllers collaborate to manage the overall network effectively. Every controller transmits information regarding the forwarding SD-LEOs it oversees to the other controller. Section IV-A outlines the procedure for synchronizing modifications through the application of cloud technology, the enhanced SD-GEO controller design, queuing architecture, and cloud integration. Based on our MATLAB simulation, we have found that the ideal number of controllers is four SD-GEO controllers positioned at a distance of 35,768km with each SD-GEO paired to a matching GEP. We discuss these parameters further in Section VI. Each GEP can communicate with each other with the use of a defense-integrated cloud [56].

We note that one potential concern is the network's scalability constraints. While four GEO controllers would enhance the management of the SD-LEO forwarding layer, there may be vendor-specific limitations on the processing power regarding the number of forwarding SD-LEOs that each SD-GEO controller can oversee. To generalize our findings, we simulate our architecture and approach, enabling us to abstract the controller processing rate from

previously published literature, and to test topologies with varying numbers of GEPs connected to varying numbers of controllers. We show that we can temporarily increase the system's overall μ if network demand arriving at a controller, λ, surpasses its processing capabilities

Additionally, we recommend placing 64 SD-LEOS forwarding satellites at a distance of 781km, following a Walker Delta orbital path [57] which will allow for near worldwide communication coverage. This will allow submarines to surface from any major body of water (i.e., oceans, seas, etc) and connect with SD-LEO to communicate with a GEP. Furthermore, depicted in Figure 1 is a hostile UAV that may potentially launch a cyber attack on the SD-GEO controller to disrupt the network's routing and obstruct communication. The malicious UAV could potentially gain unauthorized entry into the system and attempt to initiate cyber attacks such as jamming, eavesdropping, spoofing, DoS, and zero-day attacks. To safeguard against these intrusions, the GEP will systematically gather performance metrics from the SD-GEO satellites and feed them into the cloud architecture that hosts our ML model. The CatBoost ML model will build the normal, nonattack dataset through our analysis and then train on the values to acquire knowledge of the proper values. Subsequently, the model will utilize the gathered information to forecast the anticipated values and subsequently evaluate the collected expected values to identify any abnormal measurement, thereby detecting a zero-day attack. Subsequently, the model will notify the network operator at the GEP to implement countermeasures, such as disconnecting the malicious node from the constellation, in order to safeguard the SD-GEO.

A. CLOUD FUNCTIONALITY

The Department of Defense (DoD) and military organizations have utilized cloud infrastructure for their networking systems [58], [59]. Similarly, under our proposed architecture, the defense-integrated cloud will be a rigorously protected server situated in a military-operated or contracted facility. In the context of SDN, it is referred to as the application plane. The following sums up its primary jobs:

- *Data Aggregation:* Once the data is gathered from the SD-LEOs and SD-GEOs, it will be sent to the GEP. From there, the GEP will transport the data to the cloud infrastructure where it will be stored and also utilized for training and testing the ML model.
- *Cyberattack Detection:* The ML model (CatBoost algorithm), will utilize the data collected from satellites to generate predicted data for our queuing analysis. Its purpose is to determine whether a zero-day attack has taken place. The cloud platform will establish communication with the GEP responsible for the SD-GEO and inform the network operator about the attack.
- *SD-GEO Controllers Synchronization:* Our concept utilizes a flatly distributed design, where each SD-GEO has an equal level of control in the system as shown in Figure 3. They collaborate to ensure the SD-LEOs constellation remains operational. To ensure efficient traffic routing, an SD-GEO must have knowledge of the status of SD-LEOs that are outside of its jurisdiction. Consequently, every SD-GEO is required to exchange status updates with one another. Given the limitations of distance and communication ranges, accomplishing this task in space will be challenging. To facilitate this process, we suggested that the SD-GEOs regularly provide updates on the SD-LEOs they control to each other. To accomplish this, these updates will be uploaded to the cloud using the GEPs. The cloud will store a database of SD-LEO status information received from the SD-GEOs and will only transmit updates to the other SD-GEOs via the GEP if there are any updates available. This will enable the most efficient utilization of network bandwidth by transmitting only updates from the Cloud to SD-GEOs.

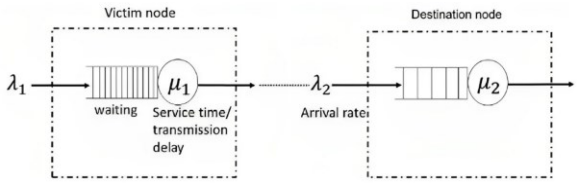


FIGURE 4. Victim Representation of a Cyberattack [64].

V. METHODS

Historically, network monitoring has been used to detect cyberattacks occurring in a network. However, the task of gathering data, categorizing it, and training ML models can be time-consuming because network traffic is unpredictable in nature [60], [61], [62]. In this study, we employ queuing theory (QT) as an alternative approach to minimize the time taken for data collection and training of ML (ML) models. QT facilitates the mathematical analysis and creation of models for SD-SATCOM networks [21], [63]. QT is a mathematical discipline that specifically addresses examining and modeling waiting lines or queues. From a SATCOM networking standpoint, the application of abstraction allows us to categorize forwarding devices (e.g., satellites) as servers, and packets as customers. By abstracting the network and analyzing the server's packet service, the network operator can obtain various performance measures, including the interarrival time (IAT), transmission delay (TD), and packet count (PC) of packets received at a server. These measures are derived by evaluating the queuing of packets at the forwarding device. The metrics described above are influenced by the arrival rate of packets (λ) and the servers' service rate (μ). By employing these measures, a ML model may be trained to identify both normal (non-attack) and attack samples from the network.

A. QUEUEING THEORY MODEL OF DOS CYBERATTACK

Figure 4 illustrates the aftermath of a DoS attack on a server. The victim node is subjected to a significant surge of packets,

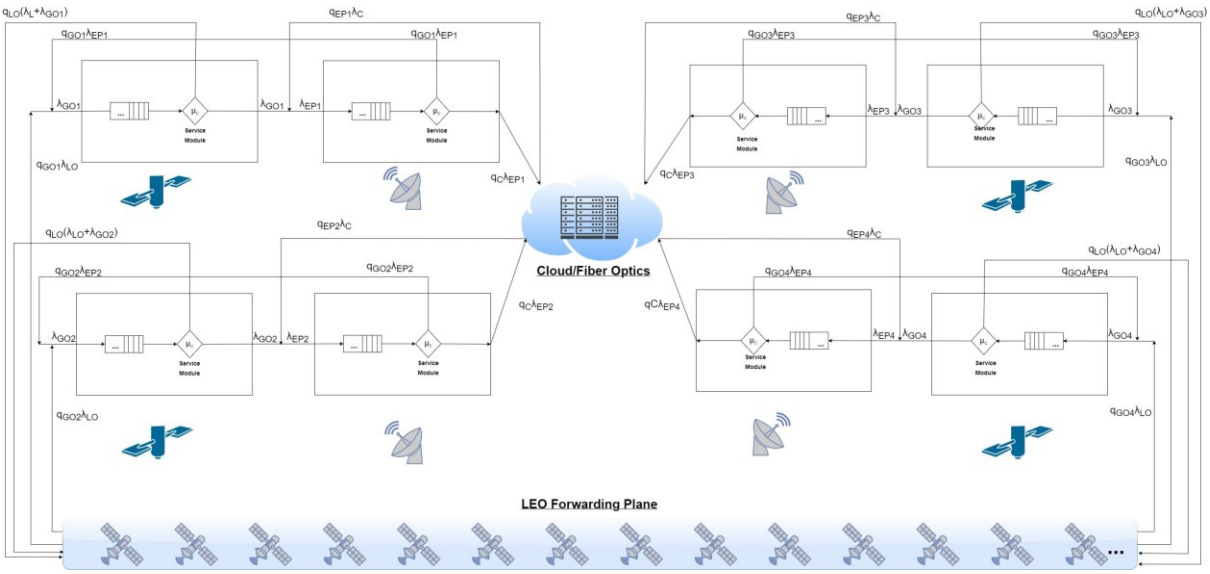


FIGURE 5. JNO M/M/C Queuing Model of SD-LEO Network.

which leads to a cascading effect on the system. As a result, the victim node experiences a higher arrival rate (λ_1), while both the service rate (μ_1) at the victim node and the arrival rate (λ_2) at the destination node decrease. The LEO satellite responsible for relaying data can collect the measurements and transmit them to the GEO controller for analysis by the network operator. Using the acquired measurements, the network operator can deduce the specific type of attack that has taken place and evaluate the magnitude of its impacts on the network. Exclusively available with SDN, the operator can employ the SDN controller to reroute network traffic away from a compromised LEO. This enables the preservation of network connectivity and data transfer speed within the network. To ensure appropriate actions are taken, it is vital to maximize the accuracy of QT measurements when making educated judgments based on QT analysis. Therefore, it is crucial to consider the design of the QT modeling. QT uses Markovian queues. Markovian queues are characterized by their adherence to the Poisson process for arrival rates and exponential distribution for service rates, which enables the property of memorylessness in both arrival and service rates. There are other types of Markovian models, such as M/M/1, M/M/c, M/M/c/K, M/G/1, and M/M/∞ [14]. Therefore, the network operator must take into account the queuing dynamics of SD-Satellites in their network, as it directly impacts the metrics they gather and utilize to detect cyberattacks.

B. QUEUEING THEORY MODEL OF SD-LEO ARCHITECTURE

The controller structure for the SD-LEO model, depicted in Figure 5, consists of a set of Jackson open network M/M/c queues. Jackson networks were chosen because they fit the characteristics and behaviors of the SD-LEO architecture well. Jackson networks are defined by interlinked queues, wherein the output of one queue is transmitted into another queue with equivalent priority. One distinctive advantage of employing the Jackson network model compared to other queuing models is that the employed Jackson networks can be resolved using a product-form solution [26]. The arrival rate of packets at satellite a (λ_a) from satellite b is the product of the sending rate of satellite b and the probability of it transmitting packets to satellite a, expressed as $\lambda_a = \lambda_b \cdot q_a$. This approach provides more tractable solutions for the more complex network interconnections, and better approximates realistic network behavior compared to alternative models [26]. The Jackson model analysis is characterized as "open," where packets exit the queuing system and transition between queues, capturing the interconnected dynamics of SD-SATCOM networks and the corresponding inter-satellite links that are essential for the effective deployment of SDN within the network [65], [66]. Furthermore, statistics from previous military publications and publicly available military standards have been used when available as relevant parameters.

However, it is crucial to recognize the disadvantages and constraints of Jackson open networks, as the analysis fails to accurately predict's the system's network metrics when it is not in a steady state $(0 < \mu < \lambda)$ and is overloaded. Furthermore, Jackson's open network analysis presumes uniform packet priority; hence, priority-based traffic routing would modify the system's behavior and contradict the analysis' network metric predictions. Consequently, these drawbacks impact our assumptions detailed in Section V-B2.

Additionally, our concept assumes that the GEPs will be interconnected via a cloud/server infrastructure. The cloud will only notify the GEPs about changes in topologies or forwarding LEO statuses, if any, in order to restrict the flow of redundant data. If a GEP does not receive any updates from the cloud infrastructure, then both the GEP and the

paired GEO controller can infer that the other aspects of the SD-LEO forwarding plane topologies and statuses are consistent with their current knowledge.

1) CATEGORIZATION OF ARCHITECTURE COMPONENTS

To our knowledge, there has been no previous research that has proposed using queuing analysis to forecast network performance metrics for an SD-LEO controller architecture in order to enhance security against zero-day attacks. The proposed SD-SATCOM network comprises four SD-satellites, each consisting of four SD-GEO controllers, four GEPs, a cloud server, and forwarding SD-LEOs. Since the cloud manages and forwards each controller's global to each other via the GEPs, each controller has a global view of the forwarding SD-LEOs. Due to this global view and control of the networks, cyberattackers target SDN controllers to gain control of the network. Therefore, the objective of this study is to enhance the cyber defense of the controller architecture by employing queueing analysis to minimize the time required for ML training in a scenario of zero-day attacks. Figure 5 illustrates the queuing model of the controller architecture which can be generalized in the following manner:

- *SD-GEO:* The SDN controller is virtualized and located within an SD-GEO as a virtual network function (VNF). The SD-GEO controller is tasked with managing the routing decisions of the forwarding SD-LEO satellites.
- *GEP:* Each SD-GEO is coupled with a ground GEP, which has a corresponding network operator. The GEP is responsible for overseeing the SD-GEO from the ground, collecting network performance statistics, and uploading updates from other SD-GEOs that are received from the cloud to its paired SD-GEO.
- *Cloud:* The data collection, ML model, and global perspective and status of the SD-SATCOM network will be stored in a defense-integrated cloud server. The system will utilize the gathered data to identify cyberattacks and detect previously unknown zero-day cyberattacks by analyzing the regular, non-attack data created through queuing analysis as described in this work. Moreover, it would regularly transmit updates regarding the overall network status of SD-LEOs to each SD-GEO through their paired GEPs, based on any changes that occurred in the network.
- *SD-LEO:* SD-LEO satellites are employed to relay signals between each other and deliver them to the intended recipient. They are the forwarding plane of the network.

2) STATISTICAL ANALYSIS

In this section, we describe our statistical analysis of the SD-

- All packets received within the constellation are regarded as having equal priority.
- Given that the network is in a steady state, the input rate (λ_{in}) is equal to the output rate (λ_{out}) of any component in the architecture, as stated by Burke's theorem [67].

Our equations will have a product form solution since we are utilizing the Jackson network open (JNO) queuing model for our queuing analysis [26]. Figure 5 depicts the queuing model of the controller architecture. In our analysis, we focus on the essential components of the controller architecture one SD-GEO controller, one GEP, and the cloud platform. Let GO_i denote a SD-GEO, EP_i denote a GEP, C denote the cloud, and LO denote the SD-LEO forwarding plane.

The rate at which packets enter an SD-GEO from the SD-LEO forwarding plane is represented as $q_{GO_N}\lambda_{LO}$, where q_{GO_N} denotes the probability that a packet will be sent to the SD-GEO$_N$ controller as a packet-in packet (a packet with an uncertain route path) from the SD-LEO forwarding plane. Thus, the λ_i for each component of the controller architecture can be represented as:

$$\lambda_{GO_i} = q_{GO_i}\lambda_{EP_i} + q_{GO_i}\lambda_{LO} \tag{1}$$

$$\lambda_{EP_i} = q_{EP_i}\lambda_C + \lambda_{GO_i} \tag{2}$$

$$\lambda_C = q_C \left(\sum_{i=1}^{n} \lambda_{EP_i} \right) \tag{3}$$

where in equations (1) and (2), SD-GEO$_N$ receives reply traffic from the SD-LEOs, and the GEP receives updates from the cloud platform. The probability of reply traffic is denoted by q_i. The probability density function for λ_i is defined for $t \geq 0$:

$$f(t) = \lambda_i e^{\lambda_i t} \tag{4}$$

The average interarrival time, $\overline{IAT}$ of packets experienced at each control architecture element is defined:

$$\overline{IAT_i} = \frac{1}{\lambda_i} \tag{5}$$

The service time follows an exponential distribution with parameter μ_i. The probability density function is:

$$g(s) = \mu_i e^{-\mu_i s}, \forall \geq 0 \tag{6}$$

where the average service time, $\overline{T_{st}}$ can be denoted:

$$\overline{T_{st}} = \frac{1}{\mu_i} \tag{7}$$

Using Little's theorem [68], average total waiting time is defined as transmission delays ($\overline{TD}$) and represented as follows:

$$\overline{\quad} = \frac{1}{\quad}$$

LEO architecture. The queuing analysis presented is based on and limited by the following assumptions:

- We assume the SD-LEO constellation is in a steady state and not overloaded. Therefore, the arrival rate of packets $\lambda < \mu$, and $0 < \lambda < \mu$.

$$TD_i = \frac{}{\mu_i - \lambda_i} \tag{8}$$

The normal distribution of network packet arrivals (i.e., non-attacked packets) into each system was decided by the probability of witnessing a number of packet arrivals in a

period from [0, T]. This equation is used to model the traffic volume of the bus:

$$P(n \text{ arrivals in interval } T) = \frac{(\lambda T)^n e^{-\lambda T}}{n!} \quad (9)$$

where T is the IAT, and n represents the number of packets. The average packet count $(\overline{PC})$ can be modeled as the following:

$$PC_i = \lambda_i T_i \quad (10)$$

VI. SD-LEO COVERAGE SIMULATION

To determine the optimal number of SD-GEO controllers and parameters required to ensure the SD-LEO constellation's operation, we conducted a simulation study using MATLAB with the aerospace, mapping, and satellite communications toolboxes. This section outlines the process by which we obtained the parameters required for our constellation and also documents the viability and efficacy of our constellation architecture in providing almost global maritime communication coverage.

A. KEY SATELLITE CONSTELLATION PARAMETERS

1) ANGLE OF INCLINATION

The angle of inclination of a satellite is the tilt of the satellite's orbit plane relative to the Earth's equatorial plane. Inclinations vary based on the mission's requirements and include equatorial orbits (0°), polar orbits (90°), and many other configurations [69].

Inclinations close to 90° are typical for LEO satellites to ensure coverage over the poles and increase Earth coverage. However, inclinations exactly at 90° require additional fuel consumption and correctional maneuvers to correct for orbital perturbations. For this reason, many constellations use inclinations just below 90° like the 86.4 degrees used by the Iridium constellation [70].

2) NUMBER OF PLANES AND SATELLITES PER PLANE

An orbital plane is a 2D plane that contains a satellite orbit path, which may be followed by one or more satellites. In satellite constellations, multiple orbital planes are often used to enhance global coverage and provide redundancy [2]. These planes are separated in such a way that each plane intersects the Earth's equator at a different point, resulting in different values of the right ascension of the ascending node (RAAN). This separation between the planes, combined with the distribution of satellites within each plane, helps to create more complete coverage and minimize the possibility of satellites intersecting.

3) ALTITUDE

Satellite altitude is the height of the satellite's orbit above the Earth's surface. The two levels focused on in this paper are:

- *Low Earth Orbit (LEO):* Typically between 500 km and 2,000 km [2]. These satellites have closer proximity, resulting in higher communications quality, but requiring more satellites for complete coverage.
- *Geostationary Orbit (GEO):* Positioned at approximately 35,786 km above the equator [71]. Satellites in GEO rotate with the Earth, remaining stationary relative to the ground.

4) ANTENNA CHARACTERISTICS

Given that this application is focused on creating a constellation specifically for communication purposes, the antenna characteristics and their effects on the coverage are relevant. While there are many different antenna configurations that can be used onboard satellites, this analysis uses a Gaussian antenna, known for its clear beam pattern and the ability to focus power efficiently within a specified half-power beam width (HPBW). The field response of this Gaussian antenna is given by [72]:

$$f(\text{az}, \text{el}) = \exp\left(-2\log_2\left(\frac{\text{az}}{\text{HPBW}_{\text{az}}}\right)^2\right) \exp\left(-2\log_2\left(\frac{\text{el}}{\text{HPBW}_{\text{el}}}\right)^2\right) \quad (11)$$

where az is the azimuth angle in degrees, el is the elevation angle in degrees, az is the azimuth half-power beamwidth, HPBW_{az} is the azimuth half-power beamwidth, and HPBW_{el} is the elevation half-power beamwidth.

The HPBW plays a critical role in determining the coverage and signal quality of the antenna. A narrower HPBW increases signal strength but narrows the coverage area. In contrast, a wider HPBW broadens coverage while reducing signal strength. Later analysis examines how this relationship affects both the FOV area and the received signal strength, as well as how its importance changes with altitude.

Other configurations, like phased arrays, are often used in practice to improve application-specific performance [73]. Transmitter and receiver power also significantly impact coverage [73]. Higher transmitter power extends the signal reach and reduces the impact of path loss. In some cases, higher receiver sensitivity - which comes with an increase in receiver power - can compensate for lower transmitter power.

These parameters – antenna configuration, HPBW, and transmitter/receiver power levels – are all important in determining the coverage of a given satellite. In this analysis, only the HPBW is examined closely, but others could be modified for future tuning of the constellation parameters.

B. FIELD OF VIEW

For a communications application like this one, a satellite's field of view (FOV) describes the area that a satellite can observe or cover with its antenna [73]. For the LEO satellites in the constellation, their FOV of Earth is of importance, whereas for the GEO satellites, the FOV of the LEO shell is of importance. These relative FOVs are illustrated in Figure

6.

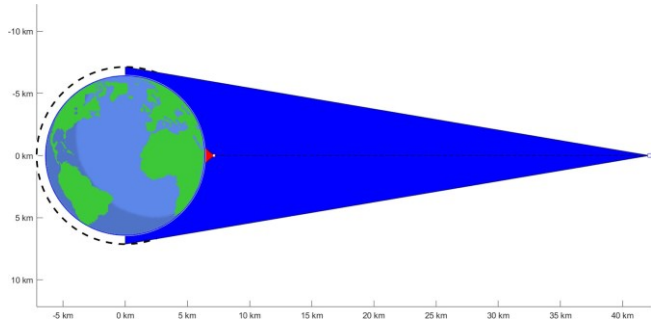


FIGURE 6. LEO and GEO Field of Views.

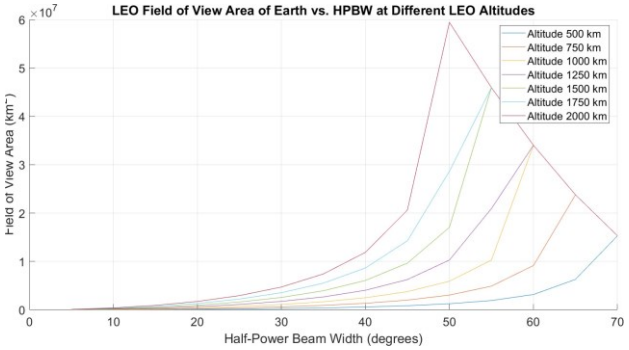


FIGURE 7. LEO Field of View of Earth vs. Half-Power Beam Width and LEO Shell Altitude.

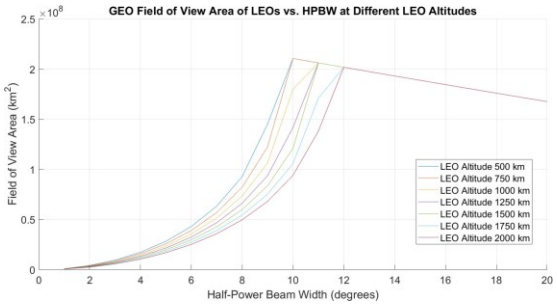


FIGURE 8. GEO Field of View of LEO Shell vs. Half-Power Beam Width and LEO Shell Altitude.

1) CALCULATION

The key parameters influencing the satellite's field of view include:

- *Satellite Altitude:* The height of the satellite above Earth's surface.
- *Antenna HPBW:* The angular spread of the antenna's beam, measured from its central axis to the half-power points, which defines the edge of the primary coverage region.
- *Sphere Radius:* The radius of the spherical model representing the Earth or LEO shell.

Since the curvature of the reference sphere limits the FOV, the central angle [73] - defined as the angle between the nadir and the edge of the beam coverage - is the first step of the calculation and can be obtained from the parameters above. The calculation is derived using trigonometric relationships involving the geometry of the sphere and the satellite's viewing cone and is given as follows [74]:

$$\theta = 90^\circ - \arccos\left[\sin(\alpha) \cdot \frac{h + r}{r}\right] - \alpha \qquad (12)$$

where: θ is the Earth's central angle, defining the outermost boundary of the satellite's FOV, α is the antenna's half- power beam-width, h is the satellite altitude above the Earth's surface, and r is the Earth's radius.

This relationship may be adjusted if the beam-width angle is large enough that the entire sphere's visible region is encompassed by the antenna. This calculated central angle is used to obtain the field of view centered on the satellite's location and is obtained by expanding outward by the central angle.

2) LEO SATELLITES' FIELD OF VIEW OF EARTH

As mentioned previously, the LEO satellite shell is responsible for communications coverage of the submarines in the Earth's oceans. As such, the FOV of LEO satellites with respect to the Earth's surface is considered.

Since the FOV calculation depends on both the altitude and the HPBW of the antenna, these values affect the LEO shell's ability to cover the oceans. The altitude range generally requires a higher HPBW, which can range from 25° to greater than 100° depending on the antenna used.

In general, a higher altitude results in a larger FOV area, as does a larger HPBW. This relationship is shown in Figure 7. It should be noted that an increased FOV area does not necessarily mean that the signal strength is high enough for the communications needed, as will be explored in the next section. Thus, values that maximize the FOV area may or may not maximize the actual coverage area.

3) GEO SATELLITES' FIELD OF VIEW OF LEO SHELL

Since the GEO shell serves as a controller for the LEO shell, the FOV for the GEO constellation is calculated concerning the LEO shell.

Once again, the altitude and the HPBW affect the calculation. However, since the GEO altitude is fixed, the altitude of interest is that of the LEO shell, which will alter the relative distance between the LEO shell (taken as the reference sphere for the FOV calculation) and the GEO shell.

Since this distance is significantly larger than the one between the LEO shell and the Earth's surface, a smaller HPBW is appropriate. Thus, only values up to 20° are tested. The relationship between the LEO shell altitude and HPBW values and the FOV area are shown in Figure 8.

The minimum HPBW that will allow for coverage over the full range of LEO elevation angles - including the polar regions - is given by the following equation:

$$r + h_{GEO} \qquad \pi$$

for LEO is generally considered to be between 500 km and 2000 km above Earth's surface [2]. This lower altitude

$$\theta_{HPBW} = \tan^{-1}\left(\frac{r + h_{LEO}}{r}\right) \cdot \frac{180}{\pi} \tag{13}$$

where h_{LEO} is the LEO satellite altitude above the Earth's surface, h_{GEO} is the GEO satellite altitude above the Earth's surface, and r is the Earth's radius.

C. SIGNAL STRENGTH

The FOV area calculated in the previous section does not encompass any analysis of the communication quality itself, just whether line-of-sight can be established between the transmitting antenna onboard the satellite and the antenna on the receiving satellite or ground station. The received signal strength is one metric used to evaluate the quality of a communications link.

1) SIGNAL STRENGTH CALCULATION

The signal strength is calculated as [75]:

$$SS = P_{tx} + G_{tx} - L_{prop} + G_{rx} - L_{sys} \tag{14}$$

where P_{tx} is the power of the transmitted signal in dBm, G_{tx} is the gain of the satellite transmitter antenna in dBi, G_{rx} is the gain of the receiver antenna in dBi, L_{sys} represents the total system losses in dB, including the inherent losses in the transmitter and receiver circuits, and L_{prop} is the path loss, which is dependent on the propagation model used and environmental factors.

Path loss (L_{prop}) in satellite communications typically accounts for the free-space path loss [75], but may also include atmospheric absorption, and other factors such as rain or gas attenuation depending on the application. In this case, only the free-space path loss is accounted for, given by:

$$L_{prop} = 20\log_{10}(d) + 20\log_{10}(f) + 92.45 \tag{15}$$

where d is the distance between the satellite and the receiver in kilometers and f is the frequency of the transmitted signal in GHz.

2) MAXIMUM RECEIVED SIGNAL STRENGTH

While the signal strength within a satellite's field of view can be assessed in a variety of ways, the maximum signal strength is useful in representing a "best-case" scenario, when a receiver is aligned along the boresight of the satellite's antenna [73].

The maximum signal strength is similarly plotted against the LEO shell altitude and the HPBW. The results are shown in Figure 9, where maximum signal strength is calculated for a receiver aligned along the boresight on Earth's surface.

3) DISTRIBUTION OF SIGNAL STRENGTH THROUGHOUT FOV

As discussed in the sections above, the HPBW affects the coverage shape and the distribution of signal strength through that shape. A larger HPBW generally results in a wider spread of the coverage and a lower maximum signal strength, as shown in the previous section.

This distribution is illustrated in Figure 10, where the signal strength distribution is compared for different HPBW values at the 750km altitude.

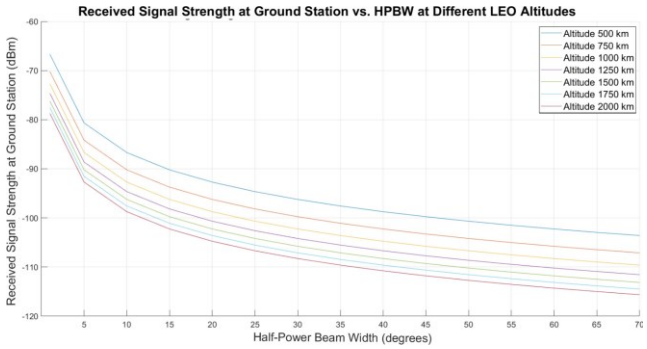


FIGURE 9. Maximum Received Signal Strength at Ground Station vs. Half-Power Beam Width and LEO Shell Altitude.

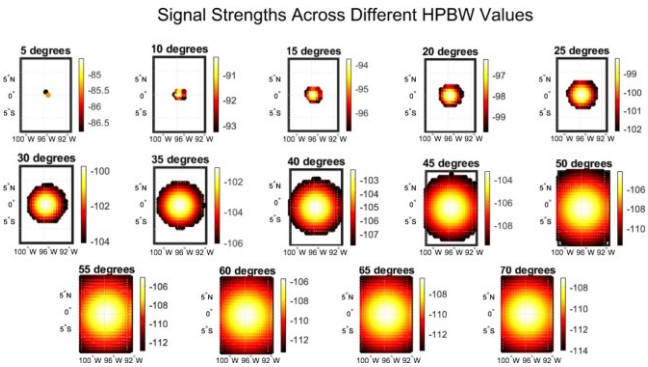


FIGURE 10. Signal Strength Distribution in dBm for Different HPBW values at 750 km Altitude.

The max values in the figure are the same as the ones shown in Figure 9, but the rest of the distribution is a relatively sharp decrease for the low HPBW distributions. For higher HPBW distributions, the max signal strength remains lower but more stable through the center of the distribution.

D. ORBITAL PERIOD

The orbital period is defined as the time it takes for an object to complete one full orbit around another body. For satellites orbiting Earth, the period depends on the altitude and gravitational pull of Earth [76]. In LEO orbits the orbital period is relatively short. Orbits at higher altitudes, however, have longer orbital periods. In GEO, for example, the period matches Earth's rotation (approximately 24 hours), allowing satellites to remain fixed over one location [71].

1) CALCULATION

The equation for the orbital period is given as [76]:

$$T = 2\pi \sqrt{\frac{R3}{\mu}} \tag{16}$$

where T is the orbital period in seconds, R is the semi-major axis, which represents the average orbital radius of the satellite from the center of the planet, and μ is Earth's gravitational parameter. μ is given as follows:

$$\mu = 3.986 \times 10^5 \quad \mathrm{km}^3/\mathrm{s}^2 \tag{17}$$

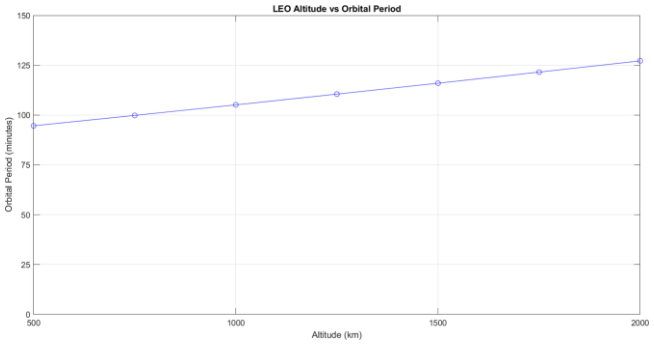


FIGURE 11. Orbital Period vs. LEO Altitude.

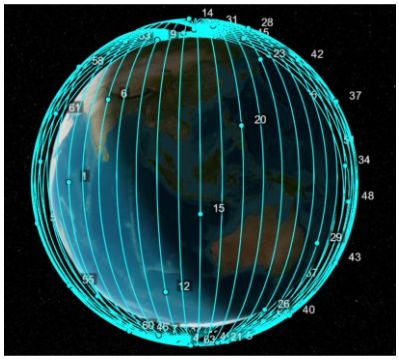


FIGURE 12. 3D Rendering of Simulated Constellation.

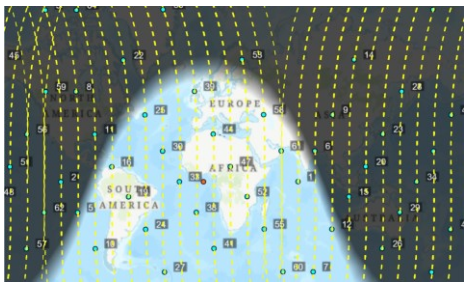


FIGURE 13. 2D Rendering of Simulated Constellation.

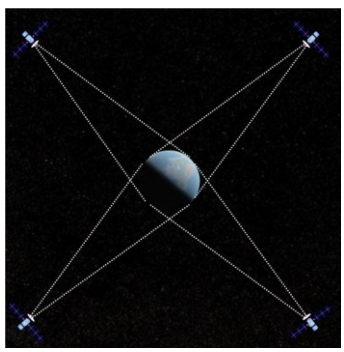


FIGURE 14. GEO Coverage Visualization.

R is calculated as the sum of the radius of the Earth R_{earth} and the altitude of the satellite above the surface h:

$$R = R_{earth} + h \tag{18}$$

2) ORBITAL PERIOD SIMULATIONS

Using these equations, the orbital period for the GEO constellation (with $h = 35768km$), as expected, is 23.93 hours [71]. This is an approximate match to the rotational period of the Earth, allowing it to stay stationary in relation to points on the surface, like ground stations. The orbital period of the LEO altitude range of 500 km to 2000 km was also tested and is shown in Figure 11.

For this application, it was imperative that each point in the orbit received coverage at least hourly. The number of satellites N needed to achieve this can be calculated as:

$$N = \frac{T}{3600}$$

where $\pm \cdot \mathrm{e}$ denotes the ceiling function, rounding up to the nearest integer to ensure complete coverage.

Based on the results shown in Figure 11, all LEO altitudes require at least two satellites per plane to meet this requirement and altitudes over 1750 km require 3 satellites per plane.

E. SELECTED CONSTELLATION

Based on the above simulations, a constellation was selected with a LEO shell altitude of 750 km. At this altitude, based on the orbital period analysis above, only two satellites are needed per plane. Using the results of the signal strength and FOV analyses, the coverage diameter can be estimated as 2300 km. The circumference of the Earth is 40,075 km at the equator [77]. Thus, a minimum of 17 planes are needed to cover all points at the equator. To include the very edge of the coverage area limits, 34 planes were chosen to provide sufficient overlap. A 3D rendering and 2D rendering of this configuration are shown in Figure 12 and Figure 13, respectively.

It can be noted in both the 2D and 3D renderings that, in the polar regions, some areas receive better coverage than others. This is because the inclination angle is not quite 90°, resulting in non-uniformity in the polar coverage but avoiding the perturbations of a perfectly 90° inclination orbit. The GEO shell above it was chosen to have 4 satellites, spaced equidistantly apart along the equator. These provide nearly complete, overlapping coverage of the LEO constellation. This configuration is shown in Figure 14.

This configuration of GEO satellites provides only partial coverage to ground stations on the surface of Earth itself, as shown in Figure 15. A GEO satellite has an absolute maximum coverage area of ± 81 degrees latitude due to the limitations of Earth's horizon line [78] though the functional maximum coverage area is generally below this maximum and was conservatively limited to ±75 degrees latitude for this work. While only 3 GEO satellites are capable of covering much of the Earth's surface, 4 are used in the implementation shown in Figure 15 to provide more

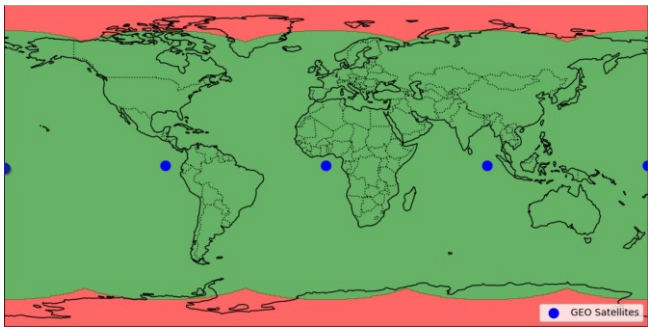


FIGURE 15. GEO Coverage of Earth.

complete coverage of the populated land masses and allow for a range of potential ground stations.

When considering the GEO coverage of the LEO shell, however, it no longer has the same horizon limitation due to the LEO shell's altitude relative to the planetary body. Instead, coverage of the entire range of LEO shell elevation angles - including the polar region - can be achieved by utilizing a large enough HPBW, as given in Equation (13). For the LEO satellite altitude of 750 km used in this constellation, a minimum HPBW of 10.7 degrees would provide complete coverage of all elevation angles. With the combined coverage of the 4 equatorial GEO satellites, GEO shell will have complete coverage of the LEO shell, with only a brief loss of coverage over the poles as the LEO satellites undergo the handover from one GEO to another. Therefore, while only partial coverage is provided from the GEO shell to terrestrial ground stations, the LEO shell is able to ensure that each point on Earth receives service during its 100-minute orbital period and the GEO shell is able to provide complete coverage of the LEO shell, aside from the polar handoff.

F. SCALABILITY

The implementation chosen for this paper aims to minimize the quantity of LEOs required to provide coverage to submarines on patrol, choosing to use only 2 satellites per orbital plane, for a total of 64 satellites. This LEO shell provides service to each point on Earth at least once in its 100 minute orbital period, as shown in Figure 16 (1).(a), but if more frequent coverage or a longer coverage duration is desired, additional satellites can be used. Since the 34 orbital planes already provide complete coverage around the Earth longitudinally, scalability is tested by increasing the number of satellites in each of those orbital planes. Three metrics are examined for these conditions: average connectivity, average reconnection time, and average coverage duration.

Connectivity is the measure of whether a point is covered by one of the LEO satellites at a given point in time. This metric is averaged over the orbital period of the constellation for each point in a grid that spans the surface of the Earth to form the connectivity heat maps shown in Figure 16 (2).(a)-(b). The average across all of the points in the Earth grid is shown in Figure 16 (2).(d). As shown in the heat maps, connectivity is highest at the poles, where the orbital paths come closer as they cross, and lowest at the equator where the paths are the greatest distance apart. There are also two regions of lower connectivity caused by the <90 degrees inclination angle of the constellation. At 2 satellites per orbital plane, average connectivity across all points is still greater than 50%. At 10 satellites, it becomes greater than 99%, becoming 100% at 12 satellites or more per orbital plane.

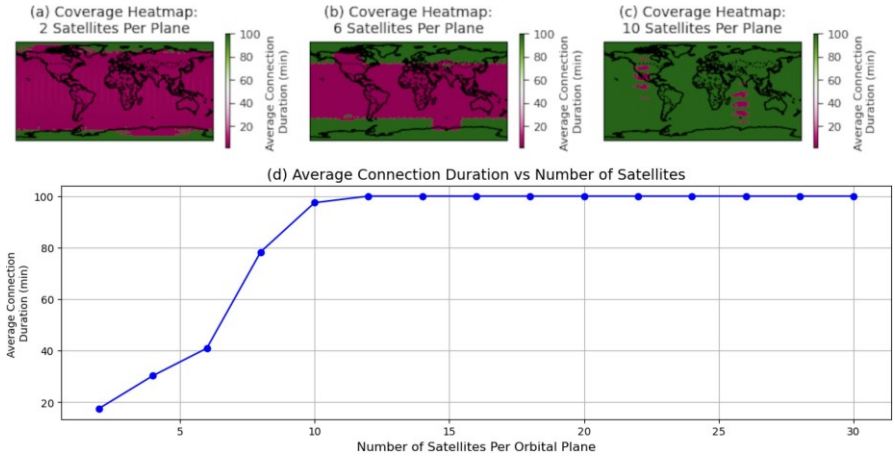
The average reconnection time – how long a point on the Earth grid has to wait before receiving connection again – was also measured for values of satellites per orbital plane ranging from 2 to 30. A similar geometry to that seen in the connectivity simulations was seen, as shown in Figure 16 (3).(a) - (b), with extremes of latitude having the shortest reconnection times whereas the equatorial regions had the longest. The overall distributions of these reconnection times for each number of satellites is shown in Figure 16 (3).(d). At the minimum number of satellites per plane of 2, the mean reconnection time was only 5 minutes, with the highest being 15 minutes. This, too, improved with additional satellites in each orbital plane, decreasing until beginning to plateau at 8 satellites per orbital plane and needing no reconnection time at 12 satellites per orbital plane, signifying uninterrupted coverage.

Finally, the coverage duration was examined to determine how long of a window a submarine (modeled as a grid point on the surface) would have to send a message before losing connectivity. This, too, was better in the polar regions than in the equatorial regions due to the density of coverage, as shown in Figure 16 (1).(a) – (b). Figure 16 (1).(d) shows the overall average of all the points on the Earth grid for each of the satellite numbers, starting at an average of approximately 17 minutes for the minimum 2 satellites per plane and increasing to over 95 minutes of a 100 minute orbital period for 10 satellites. At 12 satellites per orbital period, again, full coverage is achieved.

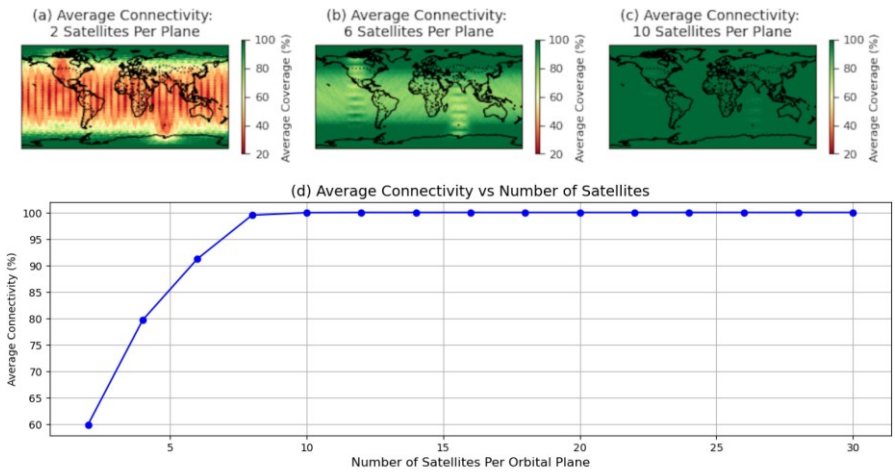
While values up to 30 satellites per orbital plane are tested, near-perfect coverage can be obtained with 10 satellites per orbital plane with full coverage obtained for 12 satellites per orbital plane. Thus, scaling the LEO constellation size beyond this point is likely unnecessary. Since GEO coverage is already complete over the LEO constellation with the exception of the brief loss of coverage due to the handoff time while crossing the poles, the number of satellites in the GEO shell would not need to be adjusted to provide continuous coverage.

VII. CASE STUDY RESULTS

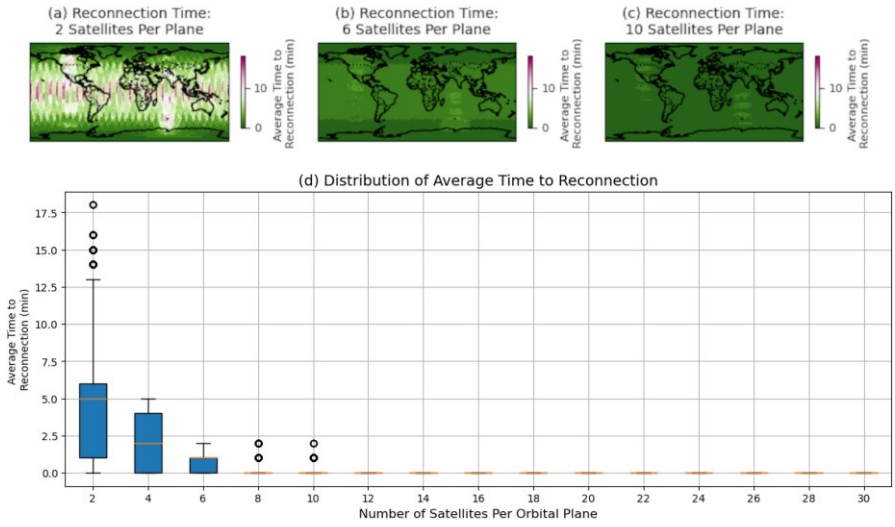
In this section, we present a case study in which we examine a scenario where a network operator possesses a dataset containing one hour of communication between the SD-GEO controller and GEP. The objective of the network operator is to determine whether a zero-day cyberattack has taken place and to assess the accuracy of the prediction analysis in relation to the collected results. In the subsections that



(1)



(2)



(3)

FIGURE 16. Scalability and Coverage (1) Coverage Duration, (2) Average Connectivity, and (3) Reconnection Time.

follow, we will present the outcomes of the queuing analysis simulation and the detection of zero-day cyberattacks using the CatBoost ML algorithm.

A. QUEUING ANALYSIS SIMULATIONS

This research aims to showcase the efficiency, precision, and speed of our predictive queuing analysis in order

Algorithm 1 SD-SATCOM Queuing Network Simulation

1: Create arrays for the $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$
2: Initialize starting λ_{LO} and λ_C
3: Initialize μ for SD-GEO and GEP
4: **for** λ_{LO} and $\lambda_C \leq 550$ packets/sec **do**
5: Initialize SD-GEO and GEP queuing components using SimComponent and SimPy
6: Initialize SD-LEO (λ_{LO}) and the Cloud (λ_C) generators
7: Utilize SimComponent's Randombrancher class function to establish a connection between the SD-GEO and GEO queuing components
8: Connect SD-LEO and Cloud instances to SD-GEO and GEP, respectively, using the Randombrancher class function
9: Assign the probability of connecting branches between components to the simulation parameters specified in Table 2
10: Assign the value of μ to every SD-GEO and GEP service module
11: Perform a 60-second communication simulation using the current values of λ_{LO} and λ_C, as well as the specified parameters
12: Record *IAT*, *TD*, and *PC* over duration of simulation for each SD-GEO and GEP
13: Calculate and Append $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$ for each SD-GEO and GEP for each λ_{LO} and λ_C and parameters in the created arrays
14: Increase λ_{LO} and λ_C values by +50 packets/sec each then repeat the loop until λ_{LO} and $\lambda_C \leq 550$ packets/sec
15: **end for**

to minimize the time required for data collection and training in the network security of the controller architecture for SD-SATCOM network for submarines on patrol. As mentioned in Section VII-A2, we consider a subset of the controller architecture that contains all components: one SD-GEO controller, GEP, and cloud instance. The network traffic of the SD-SATCOM controller architecture was evaluated by utilizing SimComponents [13]. We modified SimComponents to collect average packet inter-arrival times $(\overline{IAT})$, transmission delay $(\overline{TD})$, and packet count $(\overline{PC})$ from the SD-GEO and GEP instances. We modeled the controller architecture as a network of Jackson theorem M/M/C queues as described in Section VII-A2. We modeled our simulation based on the UHF/VHF radio communication commonly utilized in military and SATCOM operations. Algorithm 1 provides the pseudo-code of the simulation. The adjustment of port rate and queue size parameters of the servers, as presented in Table 2, allowed for the attainment of $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$ of received packets. These simulation parameter values were modeled from the military SATCOM wireless communication standards DoD Instruction 8420.02, MIL-STD-188 [79], [80], [81].

For instance, we created models to calculate the probability of a packet successfully reaching the controller, and we also considered the ratio of λ and μ values from related studies [21], [22], [23]. The mean packet size of 1400 bytes for UDP packets used in SATCOM communi- cation has been determined based on prior research [82]. The SD-SATCOM network system design includes unique characteristics such as the number of SD-GEO, GEP, and cloud instances. For the purpose of simulation, specific parameters are selected arbitrarily, including the simulation time and the chance of bidirectional communication. We assigned a probability of 0.50 to the transmission of packets between the different components, which represents the

TABLE 2. Simulation parameters.

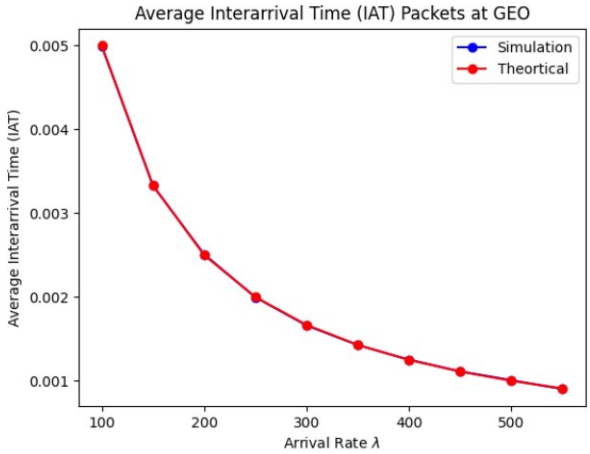
Parameter	Value
Probability packet will be sent: to controller SD-GEO$_N$ from forwarding SD-LEOs, q_{GO_N} back to forwarding SD-LEOs from controller SD-GEO$_N$, q_{LO} from controller SD-GEO$_N$ to the GEP$_N$, q_{EP_N} back from GEP$_N$ to controller SD-GEO$_N$, q_{GO_N} from GEP$_N$ to Cloud, q_C back from Cloud to GEP$_N$, q_{EP_N}	0.50
Average packet size	1400 Bytes
Average SD-Satellite$_i$ service rate, μ_i	1200 packets/sec
Average arrival rate at SD-GEO$_N$ and GEP$_N$	$a_1 = 100$ packets/sec $a_{n+1} = a_n + 50$ packets/sec $a_{n+1} \leq 550$ packets/sec
Number of SD-GEO$_N$	1
Number of GEP$_N$	1
Cloud instance (packet sink)	1
Simulation time for each λ_{LO} and λ_C	60 seconds

bidirectional connection between these devices. All of these numbers are computed based on the information supplied in the military SATCOM standards DoD Instruction 8420.02 and MIL-STD-188. Despite the constraints on vendor and government SATCOM information sharing due to proprietary and security concerns [83], we employed parameters from existing literature and publicly accessible military standards to mimic real-world applications and generalized our results to the best of our knowledge.

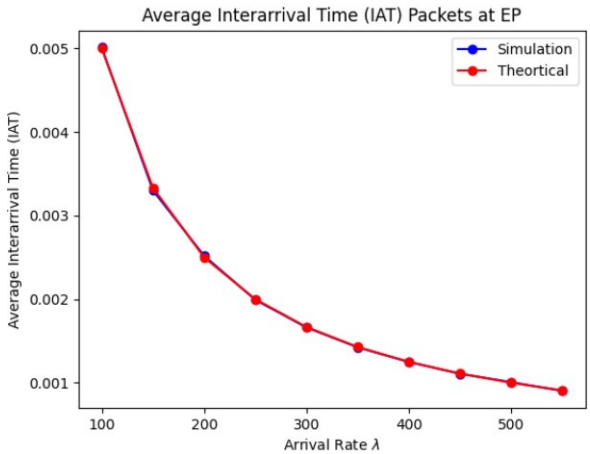
To compare our theoretical results with the simulated results, we performed simulations of the SD-Satellite network's communication for 60 seconds. We varied the combinations of λ_{LO} and λ_C in increasing increments, as indicated in Table 2. Next, we computed the mean values of $\overline{IAT}$, $\overline{TD}$, and $\overline{PC}$ using the simulation data for every combination of λ_{LO} and λ_C. Subsequently, we computed the projected values using the findings from our analysis in Section V and graphically depicted both the theoretical and projected values in Figures 17, 18, and 19 to showcase the precision of our analysis in forecasting the real values. The simulation was conducted on a desktop computer equipped with Microsoft Windows 11 operating system, powered by an Intel 12th Gen Core i7-12700K CPU running at a clock speed of 3.6GHz, and with a total of 16GB RAM. The prediction simulation for a dataset with a duration of one hour has an average CPU utilization of 3.9%. It takes 105 ms to run and make predictions in order to forecast the average inter-arrival time *(IAT)*, average time delay *(TD)*, and average packet count *(PC)* for SD-GEO$_N$ and GEP$_N$. The following subsections will present the results of the simulation for each statistic and provide an analysis of the findings.

1) AVERAGE INTER-ARRIVAL TIME $(\overline{IAT})$

The performance of a network is significantly influenced by the arrival rate of received packets, denoted as λ. If the arrival rate (λ) is more than or equal to the service rate (μ), the network is not in a steady state. In this case, the service module is unable to process packets fast enough to empty

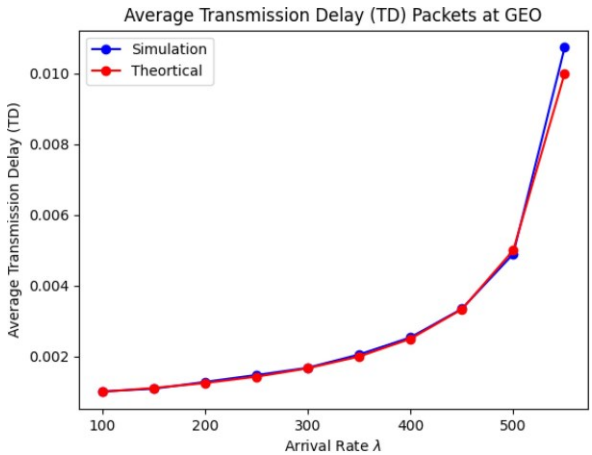


(a)

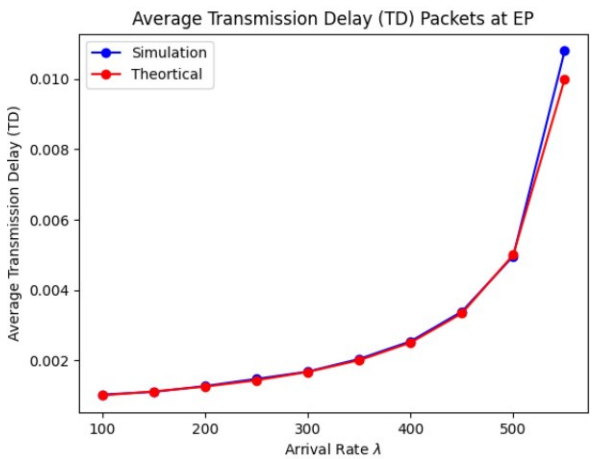


(b)

FIGURE 17. Average interarrival times over increasing $q_{GO1}\lambda_{LO}$ and $q_{EP1}\lambda_C$.



(a)



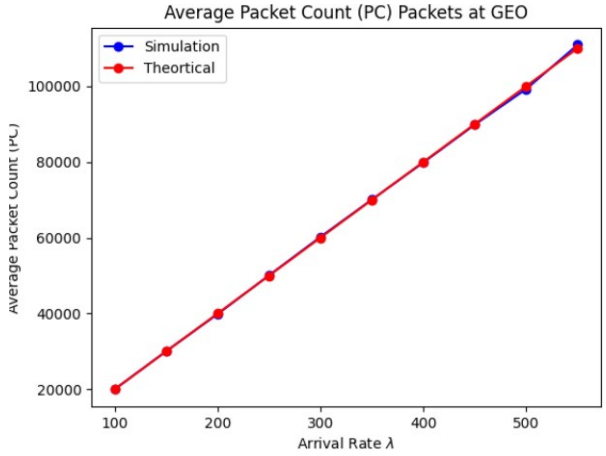
(b)

FIGURE 18. Average transmission delay over increasing $q_{GO1}\lambda_{LO}$ and $q_{EP1}\lambda_C$.

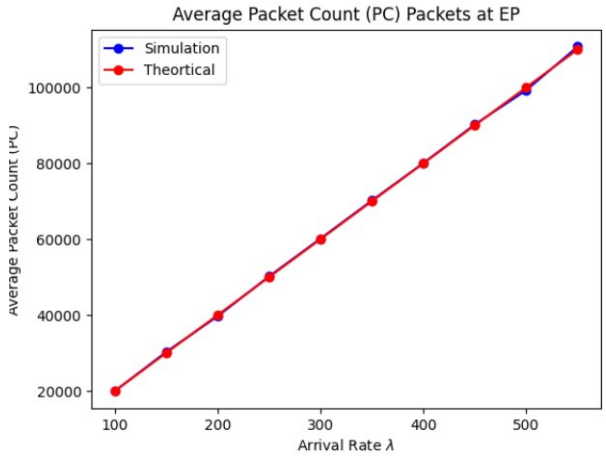
the queues of arriving packets, leading to packet drop due to limited queue space. Attackers may intentionally induce this behavior by introducing malicious packets at a high frequency, denoted as λ_{attack}. Therefore, it is crucial for the network operator to have knowledge of the rate at which packets are arriving at an SD-Satellite. Measuring the $\overline{IAT}$ packets by the network operator is an appropriate method, as these values demonstrate an inverse correlation with the parameter λ. The magnitude of the IAT values represents the speed at which packets are being received by an SD-Satellite. Fluctuations in the average inter-arrival time $(\overline{IAT})$ values may suggest the occurrence of a potential cyberattack, as packets are being received at a higher rate than expected by the SD-Satellite. The average inter-arrival time $\overline{IAT}$ of an SD-Satellite or GEP can be determined using equation (5). In order to determine the anticipated $\overline{IAT}$ at an SD-Satellite in the proposed design, the network operator can utilize the derived expected arrival rate (λ) at each SD-Satellite and GEP, as shown in equations (1) and (2). In order to verify the accuracy of our analysis, we conducted the simulation described earlier and then compared the theoretical findings obtained from our analysis in Section V with the simulation results, which are presented in Figure 17. We present the $\overline{IAT}$ results for SD-GEO$_N$ and GEP$_N$ in Figures 17(a)(b), respectively. As the packet rate (λ) grows, the average inter-arrival time $(\overline{IAT})$ drops because packets flow more rapidly into the SD-GEO and GEP. The simulation accurately aligns with the theoretical values, confirming our equations' validity.

2) AVERAGE TRANSMISSION DELAY $(\overline{TD})$

Transmission delay (TD) refers to the duration required for a packet to be processed by the service module of the packet and forwarded to the next logical destination or hop. The TD can be significantly influenced by the pace at which packets arrive, which is constrained by the limited queue size



(a)



(b)

FIGURE 19. Average packet count over increasing q_{GO1} λ_{LO} and q_{EP1} λ_C.

and processing rate of a server, as shown in equation (8). The magnitude or variation of the $\overline{TD}$ values of a device may suggest the occurrence of a cyberattack, as illustrated in Figure 4. Attackers may attempt to artificially augment the number of packets within a communication node (i.e., SD-GEO and GEP) by injecting supplementary packets into a node's queue, denoted as λ_{attack}. By applying Little's law, the average total waiting time of a packet can be determined as the transmission delay $(\overline{TD})$, as represented by equation (8).

As λ_{LO} and λ_C increase, the number of packets in the system, denoted as L, also increases. Consequently, the processing time for the packets within the service model and their exit time are prolonged. In order to verify our analysis, we conduct the queuing simulation discussed earlier and measure the $\overline{TD}$ as packets exit the SD-GEO and GEP for different values of λ. Figure 18 confirms the accuracy of our analysis, particularly in calculating the correct λ and anticipated $\overline{TD}$ experienced for each SD-GEO and GEP.

3) AVERAGE PACKET COUNT $(\overline{PC})$

The mean number of packets received at SD-GEO or GEP can be represented as $\overline{PC}$. Anomalous $\overline{PC}$ may indicate the presence of malicious activities within the network. Malicious entities have the ability to flood an SD-GEO or cause its neighboring nodes to discard packets, consequently influencing the functioning of a node in the controller architecture. Hence, the network operator must determine the anticipated average $\overline{PC}$ values that are predicted to be received at each SD-GEO and GEP. The equations (10) illustrate the anticipated number of received packets based on our analysis. As we increased the values of λ_{LO} and λ_C, we anticipate a non-exponential increase, in contrast to the other data, because of the linear correlation depicted in equation (10). The simulation values in Figures 19(a)(b) closely align with the theoretical values, confirming the accuracy of our analysis.

B. ML DETECTION RESULTS

We conducted a 1-hour simulation of network traffic for a minimum subset of the satellite controller architecture (consisting of one SD-GEO, GEP, and controller instance) as described in Section VII-A. The traffic consisted of both attack/zero-day and normal traffic, with attack traffic representing 25% of the entire dataset. In order to mimic the unpredictable nature of a zero-day cyberattack, we randomly modify the values of λ, μ, and queue sizes for the SD-GEO and GEO by a range of 50% to 90% during attack scenarios. To conduct testing and verification for the case study, we generated a "key" CSV file including 3600 rows x 6 columns. Each cell in the file was filled with either a 0 or a 1, representing different types of traffic for the SD-GEO and GEP. Specifically, a 0 indicated a normal sample, while a 1 indicated an attack sample. Next, we have produced a CSV file of 3600 rows x 1 column, which includes the λ values for λ_{LO} and λ_C. The initial three columns and final three columns of the key CSV file corresponded to the average inter-arrival time $(\overline{IAT})$, average time duration $(\overline{TD})$, and average processing capacity (PC) for the SD-GEO and GEP, respectively. Each row corresponds to the performance measurements for a duration of one minute. We used the key to create a dataset with 3600 rows x 6 columns. This dataset includes simulated values for both attack and normal (non-attack) samples, resulting in a total of 21,600 samples.

Next, we utilize our predictive queuing analysis to generate the expected nonattack samples by employing the key and λ value CSV files in Python. The data sets are analyzed using the CatBoost algorithm, which is a gradient-boosting algorithm specifically designed for decision trees. In our previous study [12], we utilized the XGBoost algorithm [84] to detect DOS attacks. Nevertheless, this study employs Catboost due to its superior performance, quicker training time, and lower overhead in comparison to other gradient-boosting methods when dealing with datasets including categorical data [85]. In a pragmatic, real-world implementation, the network operator would extract network performance metrics from packet

TABLE 3. Performance of CatBoost ML algorithm for detection of zero-day cyberattacks and average mean absolute percentage error (MAPE) for k-folds (k = 5) cross-validation.

Method	Accuracy	Precision	Recall	F1-score	Model's Total Training and Testing Speed	CPU Usage	MAPE (%)
CatBoost	97.87	96.89	94.52	95.69	8.75s	33.4%	0.002028 ± 0.000059

TABLE 4. Performance of CatBoost ML algorithm for classification of normal traffic vs zero-day cyberattacks.

Class Traffic	Accuracy	Precision	Recall	F1-score
Normal	98.99	98.19	98.99	98.59
Zero-Day	94.52	96.89	94.52	95.69

capture (pcap) files containing categorical data such as packet type, source/destination IP, and port number. Therefore, we improve our previous work by utilizing catboost for this study in anticipation of the requirement for better performance for SATCOM pcap captures.

The CatBoost ML algorithm is combined with the scikit-learn multi-output regressor model and trained as a multi-output regression model using a CSV matrix of lambda values and the corresponding projected values. The CatBoost model essentially learns to predict accurate network performance statistics values based on the lambda values it receives. Next, the predicted values from the CatBoost model are compared to the simulated values. If the absolute difference ($|x_{pred} - x_{sim}|$) between the predicted and simulated values exceeds the threshold of greater than 10%, the simulated values are classified as zero-day cyberattacks.

To evaluate our model in a cloud-based setting, we implement the CatBoost ML method using Google Colaboratory [86]. This platform is designed for experimenting with ML models on high-performance hardware like GPUs and TPUs. We provide the outcomes of the CatBoost ML model in Figure 20, Table 3, and Table 4. We employ the Optuna program [87] to optimize the hyperparameters of our CatBoost model. The optimal hyperparameters are set as follows: a learning rate of 0.0489, an optimal tree depth of 7, a subsample rate of 0.749, a colsample rate by level of 0.999, and a minimum number of data points in each leaf of 34. Table 3 presents the overall accuracy, precision, recall, and F1-score of our model, all of which are above > 94%. This indicates the high effectiveness of our machine-learning model. The CatBoost output was tested with K-fold cross-validation, with k=5, and the average mean absolute percentage error (MAPE) was computed. The low MAPE indicates that the CatBoost ML model can accurately learn the predicted values from our lambda and predicted output, with an average margin of error of 0.002%. This

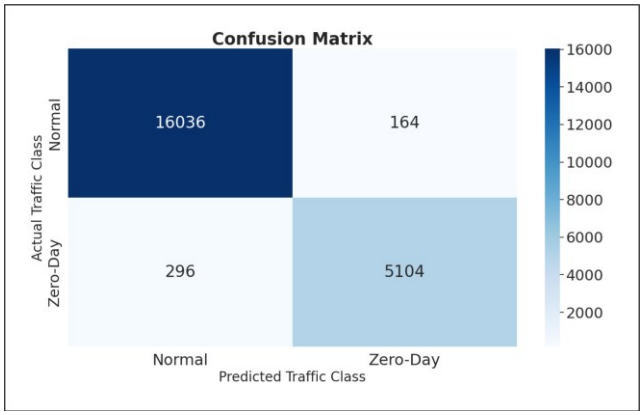


FIGURE 20. Confusion Matrix of CatBoost ML Algorithm for the Different Traffic Classes.

suggests that it can reliably predict the network performance statistics accurately from our provided generated training data from the predictive queuing analysis. We measure the CPU usage (33.4%) and the entire time taken for model training and testing (8.75s), which showcases the system's efficient performance with minimal CPU usage.

The average and standard deviation of the accuracy, precision, recall, and F1-score were calculated for the five folds. The performance of the model in distinguishing normal traffic from zero-day attack samples is seen in Table 4 and Figure 20. The model's performance metrics exceed > 94% for every class, indicating its effectiveness in detecting zero-day cyberattacks. Figure 20 shows that 460 samples out of 21,600 total samples were misclassified (2.13% computed incorrectly). The threshold for the case study was established arbitrarily at a value greater than 10%. However, network operators have the flexibility to determine the most appropriate threshold based on the significance of the communication. They may choose a more sensitive threshold to minimize the occurrence of false negatives, even if it means accepting a higher number of false positives. The effectiveness of our detection method is attributed to the well-documented performance of CatBoost, as well as the predictive queuing analysis and retrieved features from the simulation.

VIII. CONCLUSION AND FUTURE WORK

To our knowledge, no research has investigated using queuing analysis to anticipate SD-SATCOM controller architecture network performance indicators in order to

prevent zero-day attacks. To our knowledge, no prior studies have utilized queueing analysis predictions and a machine-learning model to detect zero-day cyberattacks. Because of its worldwide scope and network control, the SD-GEO controller architecture is vulnerable to attacks, including zero-day attacks. We propose a predictive queuing analysis to assist network operators in swiftly creating ML training for the SD-SATCOM controller architecture by predicting network performance data. To test our findings, we simulated the SD-GEO controller architecture with a modified version of SimComponent, a Python toolkit based on the open-source SimPy framework. We employed MATLAB's aerospace, mapping, and satellite communications toolboxes to analyze SATCOM traffic and develop our queuing architecture. During a one-hour data simulation case study, our Catboost ML model identified and classified attack samples with over 94% accuracy, precision, recall, and f1-scores.

Our future work will focus on integrating our analysis with an intrusion detection system (IDS) and mitigation framework. This integration will enable real-time training on data and allow for swift mitigation utilizing the GEP. In addition, we will enhance the framework to categorize zero-day attacks that may resemble or be similar to recognized cyberattack types. This will provide the network operator with improved guidance on potential strategies to neutralize the attack. We may confront obstacles with real-time data collection, particularly dealing with space projection delays and synchronization issues while using the cloud.

REFERENCES

[1] Q. Zhao, A. J. Brown, J. H. Kim, and M. Gerla, "An integrated software-defined battlefield network testbed for tactical scenario emulation," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2019, pp. 373–378.

[2] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55–61, Apr. 2019.

[3] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "VNF-based service provision in software defined LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6139–6153, Sep. 2021.

[4] A. Walker. "Army's eyes on resilient multi-orbit SATCOM." Nov. 2020. [Online]. Available: https://www.army.mil/article/240491/armys_eyes_on_resilient_multi_orbit_satcom

[5] V. Machi (Space Develop. Agency, Washington, DC, USA). *U.S. Military Places a Bet on LEO for Space Security*. Jun. 2021. [Online]. Available: https://www.sda.mil/us-military-places-a-bet-on-leo-for-space-security/

[6] M. Wall. "1,300 SpaceX Starlink terminals with ukraine's military went offline due to funding shortfall: Report." Nov. 2022. [Online]. Available: https://www.space.com/ukraine-spacex-starlink-terminals-offline-funding-shortfall

[7] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommenda- tions," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100218.

[8] P.-Y. Kong, "A survey of cyberattack countermeasures for unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 148244–148263, 2021.

[9] P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "Satellite-based communications security: A survey of threats, solutions, and research challenges," *Comput. Netw.*, vol. 216, Oct. 2022, Art. no. 109246.

[10] S. Nazari, P. Du, M. Gerla, C. Hoffmann, J. H. Kim, and A. Capone, "Software defined naval network for satellite communications (SDN-SAT)," in *Proc. IEEE Mil. Commun. Conf.*, 2016, pp. 360–366.

[11] M. Usman, M. Qaraqe, M. R. Asghar, and I. Shafique Ansari, "Mitigating distributed denial of service attacks in satellite networks," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 6, 2020, Art. no. e3936.

[12] D. Agnew and J. McNair, "Detection of denial-of-service attacks in a software-defined LEO constellation network," in *Proc. Gov. Microcircuit Appl. Crit. Technol. Conf. (GOMAC Tech)*, 2023, pp. 1–6.

[13] G. Bernstein. "Basic network simulations and beyond in python introduction." 2017. [Online]. Available: https://www.grotto-networking.com/DiscreteEventPython.html

[14] D. Gross, *Fundamentals of Queueing Theory*. Hoboken, NJ, USA: Wiley, 2008.

[15] A. Shoeb and T. Chithralekha, "Resource management of switches and controller during saturation time to avoid DDoS in SDN," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, 2016, pp. 152–157.

[16] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: Threats, mitigations, and future directions," *J. Reliab. Intell. Environ.*, vol. 9, no. 2, pp. 201–239, 2023.

[17] A. Ahmad, E. Harjula, M. Ylianttila, and I. Ahmad, "Evaluation of machine learning techniques for security in SDN," in *Proc. IEEE Globecom Workshops (GC Wkshps*, 2020, pp. 1–6.

[18] J. C. C. Chica, J. C. Imbachi, and J. F. B. Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, pp. 1–23, Jun. 2020.

[19] S. Salim, N. Moustafa, M. Hassanian, D. Ormod, and J. Slay, "Deep federated learning-based threat detection model for extreme satellite communications," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 3853–3867, Feb. 2024.

[20] N. Koroniotis, N. Moustafa, and J. Slay, "A new intelligent satellite deep learning network forensic framework for smart satellite networks," *Comput. Elect. Eng.*, vol. 99, Apr. 2022, Art. no. 107745.

[21] T. Li, H. Zhou, H. Luo, W. Quan, and S. Yu, "Modeling software defined satellite networks using queueing theory," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[22] N. J. H. Marcano, L. Diez, R. A. Calvo, and R. H. Jacobsen, "On the queuing delay of time-varying channels in low earth orbit satellite constellations," *IEEE Access*, vol. 9, pp. 87378–87390, 2021.

[23] Y. Zhu, M. Sheng, J. Li, and R. Liu, "Performance analysis of intermittent satellite links with time-limited queuing model," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2282–2285, Nov. 2018.

[24] H. Xu, S. Han, X. Li, and Z. Han, "Anomaly traffic detection based on communication-efficient federated learning in space-air-ground integration network," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9346–9360, Dec. 2023.

[25] A. Iqbal, M. N. Aman, and B. Sikdar, "Machine and representation learning based GNSS spoofing detectors utilizing feature set from generic GNSS receivers," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 574–583, Feb. 2024.

[26] H. Chen and D. D. Yao, *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, vol. 4. New York, NY, USA: Springer, 2001.

[27] S. P. Meyn and D. Down, "Stability of generalized jackson networks," *Ann. Appl. Probab.*, vol. 4, no. 1, pp. 124–148, 1994.

[28] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, "k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 1, pp. 30–44, Jan./Feb. 2014.

[29] F. Deldar and M. Abadi, "Deep learning for zero-day malware detection and classification: A survey," *ACM Comput. Surv.*, vol. 56, no. 2, pp. 1–37, 2023.

[30] Y. Guo, "A review of machine learning-based zero-day attack detection: Challenges and future directions," *Comput. Commun.*, vol. 198, pp. 175–185, Jan. 2023.

[31] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "Zero-day attack detection: A systematic literature review," *Artif. Intell. Rev.*, vol. 56, no. 10, pp. 10733–10811, 2023.

[32] E. E. Abdallah, W. Eleisah, and A. F. Otoom, "Intrusion detection systems using supervised machine learning techniques: A survey," *Procedia Comput. Sci.*, vol. 201, pp. 205–212, Apr. 2022.

[33] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "Unsupervised algorithms to detect zero-day attacks: Strategy and application," *IEEE Access*, vol. 9, pp. 90603–90615, 2021.

[34] N. S. Arunraj, R. Hable, M. Fernandes, K. Leidl, and M. Heigl, "Comparison of supervised, semi-supervised and unsupervised learning methods in network intrusion detection system (NIDS) application," *Anwendungen und Konzepte der Wirtschaftsinformatik*, vol. 20, no. 6, pp. 10–19, 2017.

[35] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," 2019, *arXiv:1904.10604*.

[36] K. Lee, D. Booth, and P. Alam, "A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms," *Expert Syst. Appl.*, vol. 29, no. 1, pp. 1–16, 2005.

[37] S. Oluwadare and Z. ElSayed, "A survey of unsupervised learning algorithms for zero-day attacks in intrusion detection systems," in *Proc. Int. FLAIRS Conf. Proc.*, vol. 36, 2023, pp. 1–3.

[38] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6639–6649.

[39] F. Long, *Satellite Network Robust QoS-Aware Routing*. Berlin, Germany: Springer, 2014.

[40] Y. Peng, Y. Bao, Y. Chen, C. Wu, and C. Guo, "Optimus: An efficient dynamic resource scheduler for deep learning clusters," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–14.

[41] H. F. Santos, R. L. Claro, L. S. Rocha, and M. L. Pardal, "Stop: A location spoofing resistant vehicle inspection system," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless*, 2020, pp. 100–113.

[42] S. Narain, A. Ranganathan, and G. Noubir, "Security of GPS/INS based on-road location tracking systems," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 587–601.

[43] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[44] E. Haleplidis et al., "Network programmability with ForCES," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1423–1440, 3rd Quart., 2015.

[45] J. P. Vasseur and J. L. Le Roux, "Path computation element (PCE) communication protocol (PCEP)," Internet Eng. Task Force, RFC 5440, 2009.

[46] R. Enns, "NETCONF configuration protocol," Internet Eng. Task Force, RFC 4741, 2006.

[47] G. Huston, "Analyzing the Internet's BGP routing table," *Internet Protoc. J.*, vol. 4, no. 1, pp. 2–15, 2001.

[48] R. Alimi et al., "Application-layer traffic optimization (ALTO) protocol," Internet Eng. Task Force, RFC 7285, 2014.

[49] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterp. Netw.*, vol. 3, 2010, pp. 10–5555.

[50] T. L. Hinrichs, N. S. Gude, M. Casado, J. C. Mitchell, and S. Shenker, "Practical declarative network management," in *Proc. 1st ACM Workshop Res. Enterp. Netw.*, 2009, pp. 1–10.

[51] A. Voellmy, H. Kim, and N. Feamster, "Procera: A language for high-level reactive network control," in *Proc. 1st Workshop Hot Top. Softw. Defin. Netw.*, 2012, pp. 43–48.

[52] N. Foster et al., "Frenetic: A network programming language," *ACM SIGPLAN Not.*, vol. 46, no. 9, pp. 279–291, 2011.

[53] W. Zhou, L. Li, M. Luo, and W. Chou, "REST API design patterns for SDN northbound API," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*. IEEE, 2014, pp. 358–365.

[54] D. Agnew, A. Del Aguila, and J. McNair, "Enhanced network metric prediction for machine learning-based cyber security of a software-defined UAV relay network," *IEEE Access*, vol. 12, pp. 54202–5421, 2024.

[55] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 53–87, 1st Quart., 2021.

[56] S. Cho, S. Hwang, W. Shin, N. Kim, and H. P. In, "Design of military service framework for enabling migration to military SaaS cloud environment," *Electronics*, vol. 10, no. 5, p. 572, 2021.

[57] C.-J. Wang, "Structural properties of a low earth orbit satellite constellation—The walker delta network," in *Proc. IEEE Mil. Commun. Conf.*, vol. 3, 1993, pp. 968–972.

[58] D. A. Powell Jr. and S. Class, "The military applications of cloud computing technologies," M.S. thesis, School Adv. Mil. Stud., United States Army Command Gen. Staff College, Fort Leavenworth, KS, USA, 2013.

[59] F. J. Lebeda, J. J. Zalatoris, and J. B. Scheerer, "Government cloud computing policies: Potential opportunities for advancing military biomedical research," *Mil. Med.*, vol. 183, nos. 11–12, pp. e438–e447, 2018.

[60] Z. Hu et al., "Statistical techniques for detecting cyberattacks on computer networks based on an analysis of abnormal traffic behavior," *Int. J. Comput. Netw. Inf. Secur.*, vol. 12, no. 6, pp. 1–13, 2020.

[61] K. Shaukat et al., "Performance comparison and current challenges of using machine learning techniques in cybersecurity," *Energies*, vol. 13, no. 10, p. 2509, 2020.

[62] D. Beil and A. Theissler, "Cluster-clean-label: An interactive machine learning approach for labeling high-dimensional data," in *Proc. 13th Int. Symp. Vis. Inf. Commun. Interact.*, 2020, pp. 1–8.

[63] S. Sthapit, S. Lakshminarayana, L. He, G. Epiphaniou, and C. Maple, "Reinforcement learning for security-aware computation offloading in satellite networks," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12351–12363, Jul. 2021.

[64] A. Starke et al., "Cross-layered distributed data-driven framework for enhanced smart grid cyber-physical security," *IET Smart Grid*, vol. 5, no. 6, pp. 398-416, 2022. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/stg2.12070

[65] Z. Tang, B. Zhao, W. Yu, Z. Feng, and C. Wu, "Software defined satellite networks: Benefits and challenges," in *Proc. IEEE Comput., Commun. IT Appl. Conf.*, 2014, pp. 127–132.

[66] P. Kumar, S. Bhushan, D. Halder, and A. M. Baswade, "Fybrrlink: Efficient QoS-aware routing in SDN enabled future satellite networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2107–2118, Sep. 2022.

[67] P. J. Burke, "The output process of a stationary M/M/s queueing system," *Ann. Math. Statist.*, vol. 39, no. 4, pp. 1144–1152, 1968.

[68] J. D. Little and S. C. Graves, "Little's law," *Building intuition: Insights from Basic Operations Management Models and Principles*. Boston, MA, USA: Springer, 2008, pp. 81–100.

[69] P. Misra and P. Enge, "Inclination effects on global navigation satellite system (GNSS) performance," *GPS Solut. J.*, vol. 4, no. 4, pp. 49–57, 2001.

[70] C. Fossa, R. Raines, G. Gunsch, and M. Temple, "An overview of the IRIDIUM (R) low earth or-bit (LEO) satellite system," in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, 1998, pp. 152–159.

[71] S. Breiter, I. Wytrzyszczak, and B. Melendo, "Long-term predictability of orbits around the geosynchronous altitude," *Adv. Space Res.*, vol. 35, no. 7, pp. 1313–1317, 2005.

[72] "Phased.GaussianAntennaElement." Accessed: Mar. 10, 2024. [Online]. Available: https://www.mathworks.com/help/phased/ref/phased.gaussianantennaelement-system-object.html#mw_9650afe5-1133-46fa-840a-458ba5624197

[73] W. A. Imbriale, S. S. Gao, and L. Boccia, *Space Antenna Handbook*. Hoboken, NJ, USA: Wiley, 2012.

[74] "Coverage maps for satellite constellation." Accessed: Feb. 24, 2024. [Online]. Available: https://www.mathworks.com/help/map/coverage-maps-for-satellite-constellation.html

[75] X. Dong, Z. Yuan, F. Sun, Q. Zhu, M. Sun, and P. Zhu, "Comparison of simulated and measured power of the earth-space link for satellite-based AIS signals," *Sensors*, vol. 23, no. 15, p. 6740, Jul. 2023.

[76] H. D. Curtis, *Orbital Mechanics for Engineering Students*, 4th ed. Amsterdam, The Netherlands: Elsevier.

[77] N. K. Pavlis, S. A. Holmes, S. C. Kenyon, and J. K. Factor, "The development and evaluation of the earth gravitational model 2008 (EGM2008)," *J. Geophys. Res., Solid Earth*, vol. 117, no. B4, 2012, Art. no. B04406.

[78] D.-H. Jung, H. Nam, J. Choi, and D. J. Love, "Modeling and analysis of Geo satellite networks," *IEEE Trans. Wireless Commun.*, early access, Aug. 28, 2024, doi: 10.1109/TWC.2024.3447229.

[79] *DoD Instruction 8420.02 DoD Satellite Communications*, U.S. Dept. Def., Washington, DC, USA, Nov. 2020.

[80] D. A. Fritz et al., "Military satellite communications: Space-based communications for the global information grid," *Johns Hopkins APL Tech. Dig.*, vol. 27, no. 1, pp. 32–40, 2006.

[81] T. Dzol and M. McMahon, "MIL-STD-188-220A evolution: A model for technical architecture standards development," in *Proc. MILCOM*, vol. 2, 1997, pp. 704–709.
[82] Y.-W. Chong and T.-C. Wan, "Comparative study on hybrid header compression over satellite-wireless networks," *IETE Tech. Rev.*, vol. 30, no. 6, pp. 461–472, 2013.
[83] C. Fleming, M. Reith, and W. Henry, "Securing commercial satellites for military operations: A cybersecurity supply chain framework," in *Proc. Int. Conf. Cyber Warfare Secur.*, vol. 18, no. 1, 2023, pp. 85–92.
[84] T. Chen and C. Guestrin, "XGboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794.
[85] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," 2018, *arXiv:1810.11363*.
[86] E. Bisong and E. Bisong, "Google colaboratory," *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Apress, 2019, pp. 59–64.
[87] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 2623–2631.

DENNIS AGNEW (Graduate Student Member, IEEE) received the B.S. degree in computer engineering from Jackson State University in 2020, and the M.S. degree in electrical and computer engineering from the University of Florida in 2021, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. He has been awarded the Graduate School Preeminence and L3Harris Fellowships with the University of Florida. He is currently a graduate student researcher with the NSF Center for Space, High-Performance, and Resilient Computing. His research interests include smart grids, tactical networks, machine learning, cybersecurity, and software-defined networks.

ASHLEE RICE-BLADYKAS (Graduate Student Member, IEEE) received the first B.S. degree in electrical engineering and the second B.S. degree in psychology from the University of Florida in 2023, where she is currently pursuing the M.S. degree in electrical and computer engineering. She is also a graduate student researcher with the NSF Center for Space, High-Performance, and Resilient Computing. Her research interests include satellite communications, machine learning, tactical networks, and satellite astrodynamics.

JANISE MCNAIR (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical and computer engineering from the University of Texas at Austin and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology. She is currently a Professor with the Department of Electrical and Computer Engineering, University of Florida, where she leads the Wireless and Mobile Systems Laboratory. She is a Faculty Board Member of the Nelms Institute for the Connected World and the NSF Center for Space, High-Performance, and Resilient Computing. Her research is funded by NSF, DoD, government agencies, and industry. Her current research interests include wireless and mobile networking, software- defined networks, network security, the Internet of Things, and smart grid communications security.