Space System Anomaly Detection

Team Name: Space Force

Semester: CIS-497 Spring 2022

Team Members: Chandler Armagost, Parker Arrington, Tristan Clark, Eric Sternberg

Project Client:

Terry Carlson, Chief Information Security Officer (CISO)

U.S. Army Space and Missile Defense Command

Redstone Arsenal, AL, United States

(256)-417-0502

terance.f.carlson.civ@army.mil

TABLE OF CONTENTS

1 PHASE 1:

1.1 BRIEF PROJECT DESCRIPTION:

The USASMDC develops and provides global space and high-altitude communications capabilities for the military and its allies. However, due to the nature of satellites orbiting around the Earth, they are limited by communication windows. Due to this limited time window, as well as ground station operators oftentimes being focused on mission-oriented work, there is little to no time for general anomaly detection. This means that any detection is limited to routine system health checks. Although they are not checked often, ground stations must detect these anomalies because they could potentially be a threat that must be addressed. This creates the need for an effective way to consistently track satellite network and communication anomalies to effectively defend their assets.

1.2 DESCRIBE THE PROBLEM YOUR COMPLETED PROJECT WORK WILL SOLVE:

This system will begin to lay the groundwork for ground operators to have a way to automatically track and monitor network and communication anomalies. The developed system should be a unique idea for the labs in the USASMDC to base a fully-fledged anomaly detection system around.

1.3 IDENTIFY THE STAKEHOLDERS AND HOW THEY WILL BENEFIT FROM AND BE IMPACTED BY YOUR PROJECT:

Stakeholder	***Impact of System***	***Explanation***
USASMDC	High	The USASMDC is the client requesting the new system to allow for ground operators to detect anomalies
Ground operators	High	The ground operators will be the direct users of the system, and this will allow them to detect anomalies efficiently and effectively in the satellite communications

Military	Medium	Although not direct users of the system, they will be affected by the benefit of more accurate communications. They also may be affected if a threat is detected, and they are called into action to retaliate.
US Citizens	Low	While most citizens will not even know the system exists, it will aid with national security

1.4 DESCRIBE THE FUNCTIONAL REQUIREMENTS OF THE PROJECT:

1. The system shall allow for the *ground operator* to receive data from satellites to create a cache of data for real time analysis.
2. The system shall allow for the *ground operator* to start detecting anomalies within a satellite mesh network.
3. The system shall present the *ground operator* with any anomalous data that is present within the system.
4. The system shall allow for the ground operator to restart the system/ continue scanning for anomalous data if a false positive occurs or if no action can be taken at that current time.

1.5 OTHER FUNCTIONAL REQUIREMENTS OUR TEAM DOESN'T BELIEVE WE CAN COMPLETE THIS SEMESTER:

5. The system shall give the *ground operator* possible steps to take so that the anomalous event stops occurring or is being tracked for later interception.
6. The system shall allow the *system administrator* to see a report of all potential anomalous data within the given runtime of the system. This is for review of data by the *system administrator* if needed.

1.6 DESCRIBE THE NON-FUNCTIONAL REQUIREMENTS OF THE PROJECT:

- Security - The system shall give the *system administrator* the ability to set clearance levels within the system to determine if the user should have access to the anomaly detection system.
- System Restraints - must be able to run on resource constrained systems on a linux-like operating system.

- Time - the system should be able to run at near real-time to allow for the best attempt at stopping anomalous events better when critical issues begin to occur.
- Accuracy - the system should be able to correctly identify anomalous events at least 90% of the time.
- Usability- this system should be able to run with little to no experience with the system. Most work will be done in the background and does not require user control.

1.7 TECHNOLOGIES/SKILLS EXPECTED AND REQUIRED.

The technologies needed will be familiarity with machine learning algorithms for the anomaly detection, as well as a way to capture network traffic packets in real-time. Some skills that are expected is an understanding of network traffic and capture files. Another required skill would be a familiarity with python.

1.8 CURRENTLY KNOWN CONSTRAINTS.

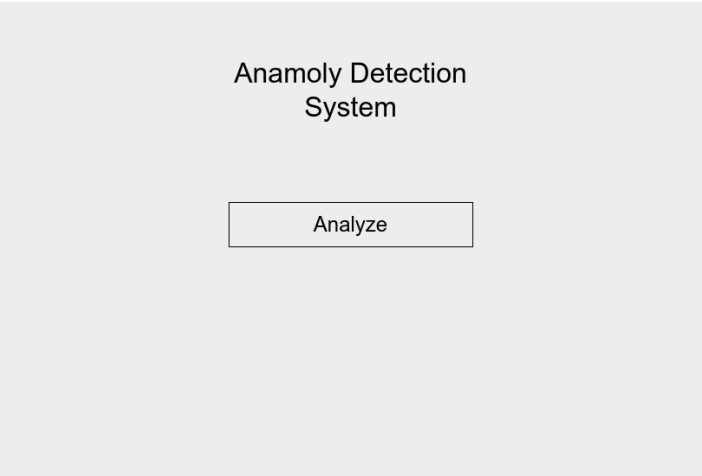
The system must be able to receive and analyze data that the satellite receives. The system must run on the same hardware used by the ground operators that work with the satellites. The completed system must be reviewed and approved by the USASMDC before being sent to the lab for further development.

1.9 USE CASE DIAGRAM.

Use Case Name:	Detect space system network anomalies
Primary Actor:	Ground Operator
Stakeholders and Interests:	USASMDC, Military, Ground Operators, US Citizens
Preconditions:	Operator must have clearance to access system. Satellite must be within access window (even for the mesh system to work).
Success Guarantees (aka Postconditions):	Network anomaly is detected in near real time / system detects that no anomalies are present.

Main Success Scenario (aka Flow of Events)	
Actor Action	**System Response**
1. Use case begins when a communication event or transmission occurs on the satellite mesh network.	
2. Actor receives transmission data from network.	3. System analyses data and reports if any anomalous activity is happening on the network and where it is in the network.
4.	5.
Extensions (aka Alternative Flows)	
Actor identifies false detection and reviews it for legitimacy (return to step 2 after)	

1.10 SYSTEM PROTOTYPE.



2 PHASE 2:

2.1 GENERAL PROJECT FRAMEWORK AND CURRENT RESEARCH:

Due to the nature of this project, the workflow has been slightly adjusted when compared to other projects that are done in the CIS-497 Senior Project. One of the major differences is the need to build a better understanding of the system that our project will be running on. Most, if not all, of the information and data that the USASMDC handles, is considered classified work. This means that our group is not able to readily have access to data from the satellites that this software may be used with in the future. This creates an interesting work environment since the project design has shifted away from the design of anomaly detection systems to a creation of a mock satellite network. The best course of action was provided by our clients. The plan is to create a raspberry pi mesh network and use low-level limited resource operating systems to mimic the satellite network. This is something that is quite new for the project group which means all efforts immediately went into researching how to accomplish these tasks so that we can use this new framework to base our possible anomaly detection software off of.

The research for this project has been primarily focused on the use case of the raspberry pi mesh network. This would include the research into the best operating system for limited resource requirements. Currently, a decision is being made between Tiny Core and Ubuntu Core. The main difference between the two is that there are more resources available for Ubuntu as a whole and that it seems to have more support. Another research topic has been what programming language would best fit the requirements. This leads to the first big decision that must be made for a better understanding of the program from the programmers. It was decided that Python would be best suited for this project. Although this language can be resource intensive, it is better to understand the language and create something that works so that afterward it can be remade in a more machine-level language. Another major research question has been associated with the ability to create the desired mesh network. Currently, this topic has been the most difficult to build a solid understanding of. This is not a common use of Raspberry Pi's which means there is not much documentation on how to make these mesh networks with only raspberry pi's. Most work for the project so far has been done through researching the setup and creation of the system our software will run on. The plan is that once a solution for the mesh network is created, we will move on to the implementation of the anomaly detection software.

2.2 PLANNED FUNCTIONAL REQUIREMENTS AND PLAN OF ACTION FOR EACH:

The system shall allow for the ground operator to start the receiving of data from satellites to begin a cache of data for real-time analysis.

- **Plan of Action:** The plan to complete this functional requirement will be centralized around the cache system and startup. The startup will be completed automatically upon connection to the satellite network to allow for a simple workflow. The cache system

will be implemented through a temp file that will be stored locally on the device. This temp file will be opened at the beginning of each time the system is run. If the temp file does not exist, the software will create the temp file for further operation. Once the file is located and loaded into the software upon startup, This data will be treated as a rolling ticker and only keep a set amount of data. This means that the oldest data will be dumped from the cache and the newest data will be continuously added to the front of the cache. The best way to treat this data will be through a doubly linked list. This allows for a simple model of storing while keeping the system easily traversable for anomaly calculations.

The system shall allow for the ground operator to start detecting anomalies within a satellite mesh network.

- **Plan of Action:** To create a seamless connection to the satellite mesh network and start the software for anomaly detection, the software will be started through a startup runtime script, and then looks for any connections to the system and starts the anomaly detection upon connection. If the system does not find any connections but a connection is currently taking place, there will be a separate bash file that the operator can run to force the software to begin anomaly detection. Also, if the system recognizes a connection but there is no throughput on the network, the software will start a timeout countdown and disconnect the anomaly detection software after a set period. This process will run indefinitely unless force closed by the ground operator.

The system shall present the ground operator with any anomalous data that is present within the system.

- **Plan of Action:** In the event of anomalous activity on the network the ground operator must be notified. The plan behind this is to present the ground operator with an alert that informs them of the anomaly and wherein the network the anomaly is occurring. In doing this it will allow the ground operator or system admin to quickly find the anomaly and assess what is causing it. This anomalous data will be presented through an alert box. When presented with the anomaly the user will be given the option to save this specific data separately or bypass the anomaly and continue with operations.

The system shall allow for the ground operator to restart the system/ continue scanning for anomalous data if a false positive occurs or if no action can be taken at that current time.

- **Plan of Action:** To accomplish our goal of allowing the user to do these tasks, the system must collect the anomalous results, and then display these results to the user in a clear, easy-to-understand method. From this point, the user can then decide whether the data is anomalous or not and whether they need to do anything to solve the anomaly. This

will be done by having the system add each instance of an anomalous result to a list that is output to the user. The user can then decide whether to continue scanning the data if it was a false positive, or if there needs to be something done in response to the anomaly, which would depend on the nature of the anomaly. Anytime the report window is closed, the system will then continue operations of detecting anomalous actions.

2.3 PLANNED NON-FUNCTIONAL REQUIREMENTS AND PLAN OF ACTION FOR EACH:

System Constraints- must be able to run on resource-constrained systems on a Linux-like operating system.

- **Plan of Action:** Due to the nature of satellites and their mission set, network anomaly detection is never the primary focus. With that in mind, the system still needs to be able to detect anomalous events in the network without compromising the mission. Our approach to this is to run a very slimmed-down version of Linux that can emulate the resource restrictions that a satellite system might place on the anomaly detection system. Also, the system will run on a restricted bandwidth connection to help mimic how connections would be on a low bandwidth satellite mesh network.

Time - the system should be able to run in near real-time to allow for the best attempt at stopping anomalous events before critical issues begin to occur.

- **Plan of Action:** The system needs to notify the operator near-real-time of any anomalous events occurring. This means we do not want them to be notified hours later or even the next time the satellite makes a connection. To achieve this, we will need to consider the low bandwidth issue. Because of this, the system must be able to sift through the desired anomaly data within a low bandwidth environment. The best plan of action would include a data mining approach to allow pattern recognition along with the cache system. This will probably need machine learning to recognize these patterns to help train for higher certainty. Our current plan for this is to find a prebuilt anomaly detection algorithm that works with network traffic.

Accuracy - the system should be able to correctly identify anomalous events at least 90% of the time.

- **Plan of Action:** We want our system to be as accurate as possible, however, the system is still prone to be able to falsely detect an anomaly. Our goal stated above is to have the system correctly identify anomalies 90% of the time. Our client cannot give out the exact data, but they directed us towards open-source SAR data that we can use instead. We will use this data to feed the system and have it recognize this data as being an

anomaly. When creating this data set the primary goal is to run as much data as possible through a machine learning algorithm so that our percent correct will be as high as possible. One concern is that our sample data and sample system will not have enough data points for the highest yield possible. The current plan is to get our system to start this machine learning but leave headroom for improvement once real datasets from USASMDC can be implemented for the learning algorithm.

Usability- this system should be able to run with little to no experience with the system. Most work will be done in the background and does not require user control.

- **Plan of Action:** To accomplish this system requirement, the program will mostly have to run itself. This means that the system will run on little to no user input. Our goal is to have the system scanning for anomalous data upon system boot, and then give the ground operator an alert when anomalous data is detected. This allows for user interaction to only be required whenever the user decides whether or not the detected data was truly anomalous or not. The primary goal is to not hinder the workflow of the current system but rather allow for ease of mind that anomalous data is not being passed over the satellite network. Therefore, this means that all plans associated with usability fall into the category of creating little to no user control and input.

3 PHASE 3:

3.1 ASSESSMENT OF FUNCTIONAL REQUIREMENTS

The first functional requirement was the system allowing for the ground operator to start the receiving of data from satellites to begin a cache of data for real-time analysis. This was completed through the collection of packets through the PyShark Python library, which scanned packets and stored them in a cache that held the packets and allowed the system to analyze the data and search for anomalies. The system holds a small amount of data before analyzing it in sections to get the system as close to real-time as possible.

The system shall also allow for the ground operator to start detecting anomalies within a satellite mesh network. This requirement was completed by first obtaining PCAP files from the University of New Brunswick and then feeding the PCAP file to our model so it will understand what to look for. The model was created by training the model and giving it malicious PCAP files and then normal PCAP files so that it can differentiate between malicious data and normal data. We also implemented supervised learning with the intent of getting the model to be as accurate as possible. Unfortunately, even though we used machine learning techniques, the model was not very accurate and would need more training data to increase the accuracy of the system. Once the model is created the python script for real-time anomaly detection can be run on the ground system and it will check 5 second intervals of packet traffic for anomalies.

Another functional requirement is the system shall present the ground operator with any anomalous data that is present within the system. The functional requirement was met by implementing the system to export all the captured anomalous data into an excel spreadsheet. The spreadsheet allows the ground operator to verify if the data is a false positive or not. The results include the entire IP layer of the packet from normal data to metadata as well.

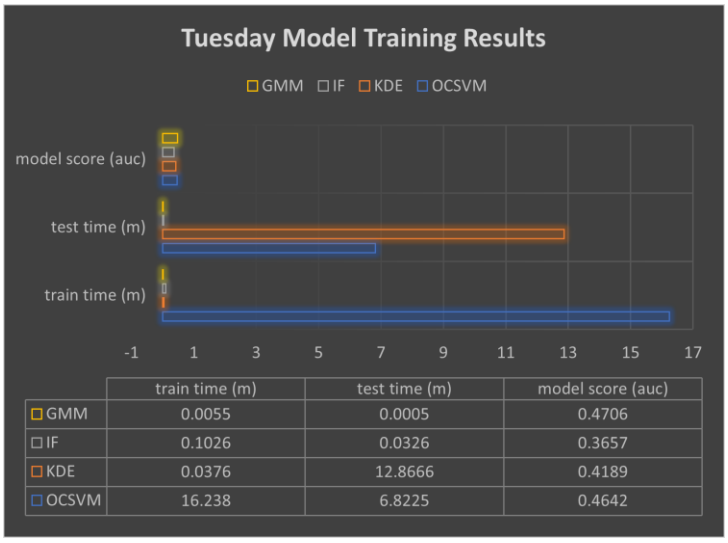
The final functional requirement that we were unable to achieve was having the system allow for the ground operator to restart the system/continue scanning for anomalous data if a false positive occurs or if no action can be taken at that current time. We focused on making the system as accurate as possible in order to detect as many anomalies as we could. The system does, however, continue scanning even when a false positive occurs or if no action can be taken because we set up the system to where it continuously scans for data and does not stop.

3.2 TEST RESULTS

Testing began with determining the best machine learning package that aligns with the requirements of the project. The packages found where NetML and PyOD. Both of these packages use machine learning to determine anomalies within a dataset. Upon further testing and research, it was determined that NetML best fit due to it already being defined

as a network detection machine learning package where PyOD is only a generic machine learning package.

Once NetML was determined to be the best, testing began within NetML to help decide the most accurate model through a built-in testing system in the NetML command line system.



	train time (m)	test time (m)	model score (auc)
GMM	0.0055	0.0005	0.4706
IF	0.1026	0.0326	0.3657
KDE	0.0376	12.8666	0.4189
OCSVM	16.238	6.8225	0.4642

Based off the current test Results, the data is currently inconclusive to help decide the best way to train the models for the highest accuracy. Currently the best model is the Isolation Forest model has the best model score it is just inversely correct. This means that its actual model score is closer to 0.64. These results are still not near the planned .90 or 90 percent accuracy. The future plan for testing is to have more data to test against and allow for more verbose use of the PCAP for testing. The reason this testing was not currently achieved is because current hardware owned by the working team does not support the load, especially in regards to RAM.

With the model being tested. The next step was to test the best way to capture live data and run it against the trained models. This testing was done within python using Pyshark for the live capture. The testing python file is labeled Classification_OD.py within the GitHub. This testing was used to determine downtime on the data capture.

All testing data was obtained from the University of New Brunswick and their Intrusion Detection Evaluation Dataset. The current test were all completed using the Tuesday working hours that included normal data and brute force attacks for anomalies. These anomalies were done by a Kali Linux system.

```
After splitting flows, the number of subflows: 37 and each of them has at least 2 packets.
'_pcap2flows()' ends at 2022-04-13 12:32:35 and takes 0.0011 mins.
'_flow2features()' starts at 2022-04-13 12:32:35
True
'_flow2features()' ends at 2022-04-13 12:32:35 and takes 0.0 mins.
Total Down Time between Intervals: 0.0003522000000000247
```

As seen in the above screenshot from Pycharm, the downtime between 5 second captures is around 0.00035 seconds. This means that there is only 0.00035 seconds of non-capture time for every 5 seconds of live capture. This was determined to be within reason since most if not all connections and traffic within a network would last longer than 0.00035 seconds.

All testing that was completed helped create a working real-time anomaly detection within python using PyShark and NetML. While all testing was done on data that was not from a mesh network. All testing and use cases should directly translate to a mesh network. The main current issue is that all training of models is not creating accurate models to use for the real-time anomaly detection. This will need further testing to create more accurate results.

3.3 LESSONS LEARNED

One major lesson that we have learned is that we tried to do too much with the project and had too broad of a scope for our project instead of focusing on the system itself. For example, at the beginning of the project we started trying to create a model mesh network when it was not necessary. This took a lot of our time when we could have been focused on implementing more training data into the system to make it more accurate and able to detect more anomalies.

The second major lesson that we have learned is time management and collaboration skills. This project required a lot of working together with different team members' schedules and skill sets, which was difficult at first, but once we understood each other's strengths and weaknesses and when the majority of the team was available, we became much more efficient and a more cohesive group.

Similarly, we learned how to communicate with a client. This project was the first assignment that we have had that required communication with a client that made the request for the product, which made the determination of the functional requirements much different. While before we had the ability to interpret the requirements ourselves and had more freedom with it, for this project we ensured that we were following the direction that our client wanted. This required us to meet with the client and communicate with them frequently to both update them on our progress and to determine whether we were going in the right direction.

3.4 Future Direction

If another senior project group were to take over this project, they should focus on making the model more accurate by finding more training data and different anomalies. This would ensure that the system would be able to detect more anomalies and become more accurate in analyzing them. The future group could also determine what types of attacks the anomalies are through supervised learning.

The future group could also try to make the system even faster. While we were able to make the anomaly detection fast and close to real time, there is a delay in which the future group could shorten. This would allow for even faster anomaly detection and alert the ground operators to the anomalies quicker.

Another task that a future group could try to implement would be software security for the system. Due to time constraints, we were unable to protect the model with any security protocols. If a group were to take our existing system and already have the system working and just focus on improvements rather than from scratch, they would be able to improve the code and make it more secure from potential attackers.

A functional requirement that our team decided would be helpful to the ground operator and hoped to accomplish would be giving the ground operators a list of suggested actions to respond to the detected anomalies. This would allow the operators to not only be aware of the intrusion, but they would be able to immediately take defensive actions against the anomalies.

A final functional requirement that would also be very beneficial to the ground operator that we were unable to accomplish would be having the system output a full report of the detected anomalies that were found in each grouping. This would allow the ground operator to see and understand the type of anomaly detected, allowing for a quicker response time. This would inform the operator of any intrusions and cut out any time that would have previously been required for the operator to investigate the anomaly. This time saving measure could be very critical in defending against any intrusion or attacks on the missile defense satellite network.

3.5 Technical Hand-off

The project source code and documentation are currently saved on our GitHub repository. Attached below is the link to it:

https://github.com/C4G-2022/Base

4 WORKS CITED

Brunswick, U. o. (2017). *Intrusion Detection Evaluation Dataset (CIC-IDS2017)*. From https://www.unb.ca/cic/datasets/ids-2017.html

KimiNewt. (2022). *PyPi*. From Pyshark Documentation: https://pypi.org/project/pyshark/

Kun Yang, S. K. (2020). *PyPi*. From NetML Documentation: https://pypi.org/project/netml/

Yue Zhao, Z. N. (2019). *Journal of Machine Learning Research*. From PyOD: A Python Toolbox for Scalable Outlier Detection: https://pyod.readthedocs.io/en/latest/