Aniyah Hall

Baltimore, MD, aniyahhall1231@gmail.com, 443-608-1082

Objective

I aim to leverage my academic background in computer technology with a focus on health technology and cybersecurity to contribute to developing and implementing secure and innovative solutions in the healthcare industry. I am seeking a challenging position where I can apply my data security, network management, and healthcare technology skills to improve patient data protection and operational efficiency.

Education

Bowie State University, Bowie, MD
14000 Jericho Park Rd, Bowie, MD 20715

- Bachelor of Science in Computer Technology – Health Technology & Cybersecurity
- Education Awards: Dean's List Fall 2021- Fall 2023
- Expected Graduation Date: 05/2025

Delaware State University
1200 N Dupont Hwy, Dover, DE 19901

- Education Awards: Deans List Fall 2020 – Spring 2021
- Attended from August 2020 – May 2021

Technology Summary

Work Experience

Student Researcher, Undergraduate Research Institute (SURI) Program
Bowie State University, Department of Technology & Security – Bowie, MD | October 2024 – December 2024

- Conducted research on securing sensitive data, such as medical information, in simulated Internet of Things (IoT) networks using advanced cryptographic techniques.
- We have designed and implemented simulations of IoT networks to evaluate the effectiveness of various cryptographic methods for data protection.
- Authored and published a comprehensive research paper in IEEE format, presenting findings and contributions on identifying and applying cryptographic methods in IoT security.
- I have gained hands-on experience with IoT protocols, encryption algorithms, and security frameworks, contributing to practical solutions for data integrity and confidentiality in healthcare systems.

Systems: Windows: Windows 10, Windows 11, Ubuntu, Kali

Languages: Python, Java

Applications: Microsoft Word, Microsoft Access, Microsoft Excel

Courses Taken:

Intro to Python programming, Unix Operating Systems, Client Operating Systems, Intro to Database Development, Principles of SercureCoding Using Java, Computer Networking, Advance SecureCoding Using Java, Application of Data Structures, Server Administration I, Foundations of Computer Network & Security, Software & OS Security, Principles & Methods of Intrusion Prevention and Detection, Case Studies in Computer Security and Fundamentals of Cryptography and Application

Undergrad Experience

RELEVANT COURSEWORK & PROJECTS:

Software and Operating Systems

Machine Learning for Biometric Authentication

Technologies Used: Python, Scikit-learn, Random Forests, Convolutional Neural Networks (CNNs), Autoencoders, Support Vector Machines (SVM), AdaBoost, Recurrent Neural Networks (RNNs), Jupyter Notebook, and Kaggle datasets.

- The team project focused on enhancing biometric and behavioral authentication systems using machine learning to improve security and accuracy for sensitive applications in healthcare and finance.
- I developed and trained machine learning models (Random Forest, SVM, CNN, and RNN) to analyze biometric and behavioral data, enabling accurate and adaptive authentication systems.
- Utilized Autoencoders for dimensionality reduction and pattern detection, helping to improve real-time verification without compromising performance.
- She collaborated on implementing Random Forests and k-Nearest Neighbors (k-NN) algorithms to classify behavioral data, such as typing speed and mouse movement patterns, to detect anomalies.
- Created visualizations and tracked model performance metrics using Jupyter Notebook to ensure model reliability and optimize hyperparameters.
- The project successfully demonstrated how machine learning can improve the accuracy and security of authentication systems, providing insights into future applications in industries such as healthcare and finance.

Network Traffic Anomaly Detection
Technologies Used: Python, Scikit-learn, TensorFlow, Pandas, NumPy, Matplotlib, ROC Curve Analysis

- The group project focused on developing a machine-learning model to detect anomalies in network traffic by classifying traffic as normal or anomalous.
- Trained the model on labeled datasets through five training cycles (epochs), achieving 99.89% training accuracy and 99.90% validation accuracy, ensuring the model generalized well to new data.
- We evaluated the model using precision, recall, and F1-score, achieving 100% for all three metrics, indicating the model's excellent performance in identifying anomalies without false positives or missed detections.
- Tracked training and validation loss across epochs to ensure the model was learning efficiently and not overfitting to the training data. The final loss was 0.0562, demonstrating stable performance.
- I visualized model performance with ROC curves and AUC scores to measure the model's ability to distinguish between normal and anomalous traffic, achieving a balanced AUC score of 0.50.
- Optimized batch size to 32 for efficient learning, ensuring fast processing times with each epoch completing in approximately 22–23 seconds.
- Created visualizations with Matplotlib to monitor accuracy and loss trends over epochs, identifying potential overfitting and maintaining high validation accuracy throughout.
- This project successfully demonstrated how machine learning models can enhance network security, providing valuable insights for real-world applications in detecting suspicious or harmful behavior.

Protection of Sensitive Data with Zero Trust Model and Machine Learning Technologies Used: Python, Scikit-learn, Splunk, Wireshark, Snort, Zero Trust Model, User Behavior Analytics (UBA), Natural Language Processing (NLP).

- The team project focused on protecting sensitive data using a Zero Trust model combined with machine learning algorithms to enhance cybersecurity defenses at Bowie State University.
- I contributed by developing machine learning models using Python (Decision Tree, Naive Bayes, KNN, and Logistic Regression) to detect ransomware and suspicious activities within the network.
- Integrated real-time data monitoring tools, including Splunk, Wireshark, and Snort, to analyze network traffic and identify anomalies, improving the detection of potential threats.
- Simulated Zero Trust policies by implementing strict access control and continuous user/device verification following the "never trust, always verify" principle.
- They collaborated on phishing email detection using NLP to identify and classify phishing attempts effectively.
- Used UBA methods to analyze user activity logs, detecting abnormal behaviors and potential insider threats in the system.
- The solution met all project objectives, effectively secured data, and provided actionable insights for improving future threat mitigation strategies.

Cryptography and Applications

Blood Pressure Monitoring System:

Technologies Used: Python, Paho-MQTT, Cryptography Library, Matplotlib, Fernet Encryption, MQTT Broker, Random Data Generation

- The team project focused on developing a secure IoT-based blood pressure (BP) monitoring system using MQTT protocol for real-time data transmission and encryption to protect patient data.
- I implemented MQTT communication using Paho-MQTT to publish encrypted BP data from a simulated device to a subscriber system, ensuring secure data flow.
- We utilized the Cryptography library's Fernet encryption to secure BP readings before transmission and developed decryption logic to simulate real-world healthcare data protection.
- We generated random BP data every two seconds to simulate continuous patient monitoring, providing insights into real-time healthcare solutions.
- Created visualizations with Matplotlib, including line plots of BP over time, size comparison of raw vs. encrypted data, and color-coded scatter plots to identify critical BP readings.
- They collaborated on logging and error-handling features to track MQTT communication and ensure graceful recovery from decryption failures.
- This project successfully demonstrated the balance between data security and performance, providing a foundation for future real-time healthcare solutions with enhanced patient data protection.

Detecting Ransomware Using Machine Learning

Technologies Used: Python, Scikit-learn, Pandas, NumPy, Matplotlib, Jupyter Notebook, Kaggle Dataset

- Team project focused on **detecting ransomware using machine learning algorithms** as part of Bowie State University coursework.
- I **developed and trained machine learning models** (Decision Tree, Naive Bayes, K-Nearest Neighbor, and Logistic Regression) using **Scikit-learn** to classify files as benign or ransomware-infected.
- **Preprocessed the dataset** from Kaggle by cleaning non-numeric data and splitting it into 70% training and 30% testing sets.
- **Evaluated model performance using accuracy and log loss metrics,** with Decision Tree achieving the highest accuracy (92%).
- Used **Matplotlib to plot model performance** for accuracy and log loss comparisons across algorithms.
- Collaborated with the team to **implement the models in Jupyter Notebook**, ensuring smooth execution and troubleshooting any errors.
- The project successfully demonstrated how machine learning models can **enhance ransomware detection**, providing practical insights for future cybersecurity defenses.

RELEVANT COURSEWORK & CLASS PROJECTS:

- Finding methods to measure with use of tools such as Nmap scanner to view open and closed ports and Wireshark to monitor Network Traffic. Nmap was used to map out the network and analyze what can come in and out between host and clients. Wireshark. Wireshark was used to see the traffic of the network and troubleshoot the quality and flow of the network.
- Using local area and long-haul computer communication networks, analysis, design and implementation of network protocols, such as Permissions, Password expire timers, and two-step authentication.
- Recognizing potential threats to confidentiality, integrity and availability, and developing familiarity with current security-related issues in computer science.
- Assessed threats and vulnerabilities to determine the level of risk. This included the use of algorithms, analysis, and basic and advanced data structures. Among the specific data structures

covered are strings, stacks, records, linked lists, trees, graphs, recursion, Sequential and random files, hashing and indexed sequential access methods.

- Used standard computer science algorithms for sorting and searching.

Linux Operating System- Ubuntu, Kali

Technologies used:

- Installed a Linux Operating System of choice in Hyper-V.
- Explored Linux filesystems using relative and absolute pathnames.
- Manage Linux filesystems using Linux commands and setting permissions on files.
- Worked with the Bash Shell by redirecting input, output, and creating a mini shell script.
- Manage Linux processes by creating a process, determine the state of a process, and killing a process.

Secure Coding- Python

Technologies used: Python Compiler IDE

- Encryption Libraries and Protocols: Utilize libraries such as cryptography, PyCryptodome, and SSL/TLS to implement secure data encryption, authentication, and communication.
- Error Handling and Logging: Safely manage errors without exposing sensitive information.

Secure Coding - Using Java

Technologies used: Java Compiler IDE

- Wrote programs that uses inheritance, polymorphism, and arrays.
- Created classes with methods and develop client codes to test those methods.

References

Available Upon Request