ENVISION THE FUTURE

Interim Progress Report

Mohab Sameh Ibrahim

Adham Hossam

Dania Alaaeldin

SECTION 1: REFINED PROJECT DESCRIPTION

1.1. The problem addressed specifically in this project and its importance:

Current IDS technologies have large potential of improvement, especially in the field of Anomaly-Based IDSs which utilize Machine Learning and Deep Learning approaches in detecting modern-day attacks. Some of the problems that IDSs face in their everyday operation is:

- The fact that attacks are constantly changing.
- IDS must be highly accurate in detecting both known & zero-day attacks.
- Attacks must be detected quickly → Time performance is important.
- Complexity & tradeoffs of implementing Machine Learning & Deep Learning approaches in Anomaly-based IDS.

The importance of this problem lies within the importance of securing systems that are subject to malicious attacks; where such attacks can cause huge financial, regulatory, and reputational losses to affected entities. To elaborate, Dell's Global Data Protection Index reports skyrocketing cyber-attack incidents, where 35% percent of organizations were victims of cyber-attacks in 2019, compared to 28% in 2018.

Therefore, we are aiming to present a Benchmarking Platform for Anomaly-Based Intrusion Detection Systems which aids the development of anomaly-based IDSs to overcome such problems.

1.2. Project scope and expected outcome:

The system provides an all-in-one workbench for developing machine learning and deep learning methods for anomaly-based intrusion detection systems (IDS). The system handles the pipeline in developing such IDSs through the following modules: data input, preprocessing, classification, and metric generation.

The system's user is able to import or create input network data through datasets or live packet capture, preprocess the data as required, utilize various classification methods, and view performance evaluation metrics and comparative analysis that aid the user in decision-making processes and the development of anomaly-based IDSs.

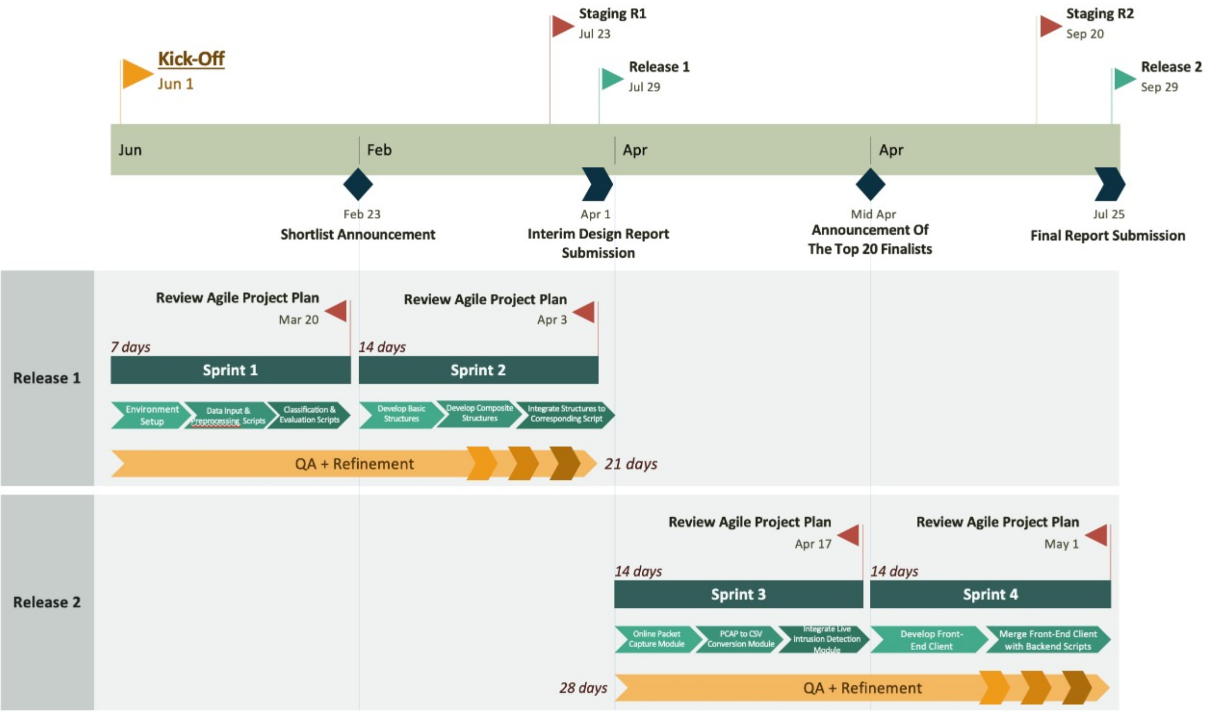
Finally, within the expected outcomes is developing an innovative platform that solves the discussed problems and fulfills its industrial and scientific need. To add, scientific contribution has already been done, were we published a paper which discusses the theoretical aspect of the solution in the 15th **International Conference on Computer Engineering and Systems (ICCES 2020)** published in **IEEE [doi: 10.1109/ICCES51560.2020.9334553].**

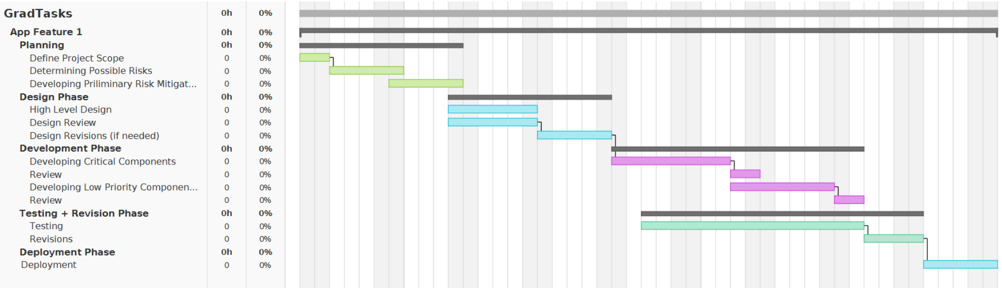
1.3. List and explanation of any changes/adjustments made relative to the abstract report:

Functional and Non-functional requirements have been restructured and adjusted. Moreover, WEKA [doi:10.1109/ANZIIS.1994.396988] is added to current similar solutions; however, this project's scope is directed towards a benchmarking platform directed towards serving the needs of Intrusion Detection Systems' development, unlike WEKA, and offers innovative and tailored aspects to serve such purpose.

SECTION 2: REFINED PROJECT PLAN

2.1. Detailed schedule and milestones (current status progress and issues):


Staging R1
Jul 23
Staging R2
Sep 20
Kick-Off
Jun 1
Release 1
Jul 29
Release 2
Sep 29
Jun
Feb
Apr
Apr
Feb 23
Shortlist Announcement
Apr 1
Interim Design Report Submission
Mid Apr
Announcement Of The Top 20 Finalists
Jul 25
Final Report Submission
Release 1
Review Agile Project Plan
Mar 20
Review Agile Project Plan
Apr 3
7 days
Sprint 1
14 days
Sprint 2
Environment Setup
Data Input & Preprocessing Scripts
Classification & Evaluation Scripts
Develop Basic Structures
Develop Composite Structures
Integrate Structures to Corresponding Script
QA + Refinement
21 days
Release 2
Review Agile Project Plan
Apr 17
Review Agile Project Plan
May 1
14 days
Sprint 3
14 days
Sprint 4
Online Packet Capture Module
PCAP to CSV Conversion Module
Integrate Live Intrusion Detection Module
Develop Front-End Client
Merge Front-End Client with Backend Scripts
28 days
QA + Refinement



GradTasks	**0h**	**0%**
App Feature 1	**0h**	**0%**
Planning	**0h**	**0%**
Define Project Scope	0	0%
Determining Possible Risks	0	0%
Developing Priliminary Risk Mitigat...	0	0%
Design Phase	**0h**	**0%**
High Level Design	0	0%
Design Review	0	0%
Design Revisions (if needed)	0	0%
Development Phase	**0h**	**0%**
Developing Critical Components	0	0%
Review	0	0%
Developing Low Priority Componen...	0	0%
Review	0	0%
Testing + Revision Phase	**0h**	**0%**
Testing	0	0%
Revisions	0	0%
Deployment Phase	**0h**	**0%**
Deployment	0	0%

2.2. Team structure and detailed roles/responsibilities for each member:

Team Member	*Roles*	*Responsibilities*
Mohab Sameh	Team Leader	Insure team productivity, resolve barriers between different technical backgrounds and potential conflict of interest, develop and track plan progress, review requirements' feasibility, communicate with stakeholders, review products and insure quality assurance effectiveness, provide the solution's structural layout, facilitate major design decisions.
Adham Hossam	Team Member	Track solution's version control, develop backend code conforming to specified requirements, unit test granular components, develop, manage, and insure database operation.
Dania Alaaeldin	Team Member	Develop front-end aspects to corresponding to underlying solution's logic, integrate backend code with front-end client, develop automated integration tests, report problematic potential test results, packaging major solution processes and structures as invokable discrete components.

2.3. Contingency and risk mitigation plan:

Risk mitigation plan:

Risk	*Risk Occurrence*	*Risk Impact*	*Action to Take*
Purpose of project is not met due to loss in planning phase.	Low	High	Develop a high-level plan and scope definition to ensure that all team members understand the aim well.
Tasks are not completed on time as agreed on in the project timetable / schedule estimation errors.	Medium	Medium	Make sure that each member acknowledges their tasks and their deadlines and provide constant feedback to one another to guarantee completion on time.
Pressure of running multiple tasks simultaneously and/or change of task durations.	High	Medium	Double check distribution of tasks among team members and project pipeline and study each task well to plan it beforehand.

Contingency Plan:

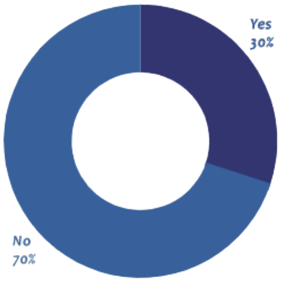
Event	*Trigger*	*Occurrence*	*Impact*	*Solution*
The absence of a team member that is knowledgeable in an area which can cause work delay.	Unexpected circumstances or illness for the member.	High	Medium	Develop a backup of task loads to be distributed evenly among other team members
Hardware malfunction	Overload from system on device, failure of equipment.	Medium	Medium	Ensure that used device meets requirements needed for system, backup system on cloud to guarantee work continuity if event occurs for one member.

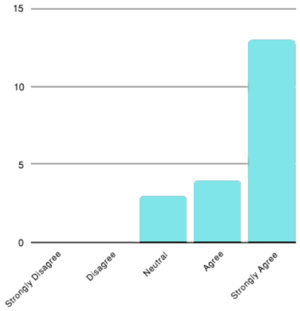
SECTION 3: SYSTEM REQUIREMENTS

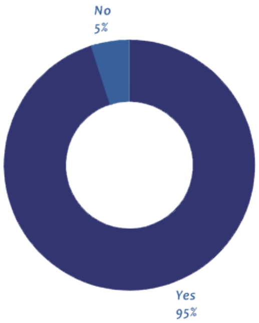
3.1. Requirements elicitation process:

a) *Description of the processes that were actually used for: requirements elicitation, analysis, prioritization, and change management:*

The team held brainstorming sessions to gather general requirements directions and solution approaches. To add, surveys and field-expert suggestions were done where the team gathered input, feedback, and suggestions to further analyze, prioritize the requirements needed by the system's audience.







More high resolution survey results available on the following link: (https://drive.google.com/drive/folders/1kmHuoREJpG8q4eevzqpvVv5ffcNtL7DE?usp=sharing)

b) *List and categorization of system stakeholders, users, and clients:*

- *IT Administrators:* where IT administrators are responsible for the operation of IDSs within the organizational networks, and are considered the most common direct user of such systems.
- *Network Architects/Engineers:* where the architects and engineers are responsible for developing the layout, deployment, and operation plan of IDSs within the organization.
- *Defensive and Testing Security Professionals:* such as security blue teams who continuously tweak and test the organizational Intrusion Detection Systems and look for potential improvement solutions.
- *Researchers*: due to the high interest of anomaly-based IDSs within the scientific community, researchers continuously conduct studies regarding anomaly-based IDSs' machine learning and deep learning pipelines, and will greatly benefit of the system's facilitating functionalities.

- *Governmental Agencies that wish to develop more secure intrusion detection systems that are more effective against zero-day attacks used in cyber warfare*
- *DELL EMC, the main targeted solution stakeholder, where the organization continuously seeks improving their robust security systems.*

Along with general audience who work with information technology and networking who can benefit from the system and understand where to integrate it. To add, such general audience will be able to develop machine learning and deep learning IDS tasks using a simple and intuitive design, which previously needed high knowledge of programming.

c) *Challenges encountered [and lessons learned] during the requirements gathering, analysis, and prioritization phases:*

- Difficulty to gain audience interest: relevant audience and appropriate sharing media must be taken into consideration and relatable, yet interesting, title for the questionnaires/surveys should be carefully chosen.
- Difficulty to retain audience engagement: more relatable and general questions were chosen and taking into consideration the wide variety of possible audience background and experience/knowledge factors.
- Gaining as much possible social reach: the team made sure gain as much user reach as possible through sharing, seeking, and inviting as much user input as possible; while making sure all the input is taken into consideration and carefully analyzed for possible new requirements and unaccounted-for perspectives.

3.2. System requirements list:

a) *Functional requirements:*

An example of functional requirements is illustrated below using the IEEE functional requirements table format

Function Name	View Classifiers Metrics Report
Description	The system shall enable the user to view a report of the performance and accuracy metrics; the report shall be printed within the "Metrics Output" section within the Metrics tab. The printed metrics shall be based on the user-specified classifiers and the user-specified metrics selected by the user from "Select Classifiers" list(the previously performed classification tasks) and "Select Metrics" list(precision, recall, F1-Score, Support, Training Time, Testing Time).
Input	Dataframe dataset, String classifiersSelected[n], String metricsSelected[n]
Output	Metric metricsReturned[n]
Critically	High-Priority
Technical issues	Exceptions might occur if the user tries to preview a metrics report without selecting classifiers or metrics.
Cost and schedule	Follows "Developing High Priority Components" schedule within gantt chart illustrated in refined project plan.
Risks	A risk is present where handling must be performed to prevent the user from previewing a metrics report without selecting classifiers or metrics, which might cause an exception occurrence.
Dependencies with other requirements	Dependent on "Import Dataset" and "Apply Classifier" requirements.
Pre-Condition	No metrics report is printed on the "Metrics Output" section in the "Metrics" tab.
Post-Condition	The user-selected metrics of the user-selected classifiers are printed on the "Metrics Output" section.

Other functional requirements such as Import Dataset, Capture packets, Load Pipeline, Show Dataset Overview, Show Selected Feature Visualization, Apply Preprocessing Technique, Change Classifier Test Options and others are illustrated in IEEE table format in the following file (https://drive.google.com/file/d/1h8CK2gZNjOVsmdw4nhiemnXmXfOcD3j8/view?usp=sharing)

b) *Non-functional requirements:*

- *Security:* The system shall follow common security code convention such as: following data protection, access modifiers, hashing sensitive data, and obfuscating gathered packet data on demand. Also, sensitive database information shall be encrypted with AES-256 encryption, String sanitization shall be performed within any input field within the user interface to avoid possible code injection.

- *Binary compatibility*: The system shall be able to run without recompiling on different environment, thus successfully satisfying binary compatibility.

- *Reliability:* Using Mean Time Between Failures:

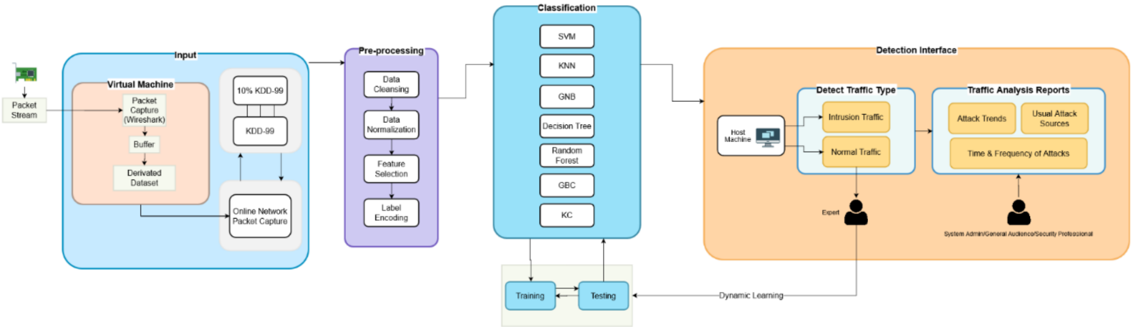
$$MTBF = \frac{Total\ Elapsed\ Time\ -\ Sum\ of\ Downtime}{Number\ of\ Failures}$$

The system shall provide a minimum MTBF of 120 hours. To add, the source code shall follow exception handling code convention; where try and catch methods should be used extensively.

- *Maintainability:* The system shall provide an output interface through the backend server log which describes any occurring issue. To add, resolution of server errors shall not exceed a downtime of 2 hours per event.
- *Portability*: The system shall a web-based interface that can be deployed or accessed through independent portable devices. However, the computation server shall only provide an interface gateway without requiring server portability.
- *Re-usability:* The system shall use component-based programming with object orientation convention, thus achieving reusable components that can be independently tested or deployed with the presence of stubs and drivers.
- *Resource Utilization:* The system shall not saturate the recommended 16GBs of memory during a session. Classification tasks shall saturate CPU resources if needed.

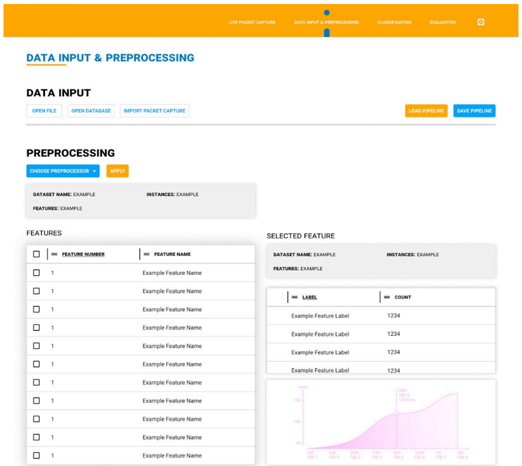
SECTION 4: SYSTEM DESIGN

4.1. High level system architecture, data flows, etc:



More high-res architectural and process diagrams such as Data Flow Diagram, Class Diagram, Architectural Hierarchy, Sequence Diagram, and more available on (https://drive.google.com/drive/folders/1vJTgeHN5iQRTU8BeqskVt5zmJabBuAVX?usp=sharing)

4.2. User interfaces:



DATA INPUT & PREPROCESSING

DATA INPUT

PREPROCESSING

DATASET NAME: EXAMPLE INSTANCES: EXAMPLE

FEATURES: EXAMPLE

FEATURES

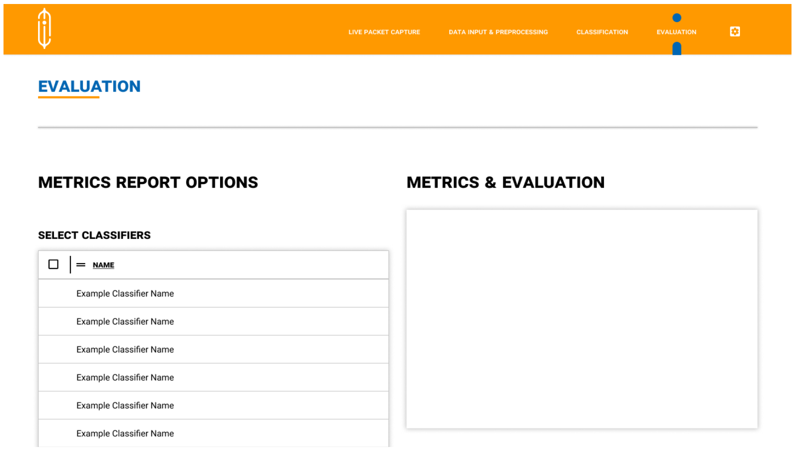
□	FEATURE NUMBER	FEATURE NAME
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name
□	1	Example Feature Name

SELECTED FEATURE

DATASET NAME: EXAMPLE INSTANCES: EXAMPLE

FEATURES: EXAMPLE

LABEL	COUNT
Example Feature Label	1234
Example Feature Label	1234
Example Feature Label	1234
Example Feature Label	1234



EVALUATION

METRICS REPORT OPTIONS

METRICS & EVALUATION

SELECT CLASSIFIERS

□	NAME
	Example Classifier Name
	Example Classifier Name
	Example Classifier Name
	Example Classifier Name
	Example Classifier Name
	Example Classifier Name

Some of the user interface images are illustrated above, more images and higher-resolution versions are available in the following link (https://drive.google.com/drive/folders/17uepgmGOY-paPT3DPbG0GTTEobcl3XjF?usp=sharing)

4.3. Algorithmic components:

The platform supports the ability to import a custom/user-defined algorithm for either preprocessing or classification; however, a wide set of predefined preprocessing algorithms such as: null cleansing, label encoding, and correlation analysis; also, classification algorithms such as: Random Forest, K-Nearest-Neighbors, Support Vector Machines, Long-Short-Term-Memory, Gaussian Naïve Bayes, Decision Trees, Logistic Regression, Gradient Boosting Classifier, Artificial Neural Networks.

4.4. Innovative aspects of the design:

The platform provides innovative aspects compared to similar previous solutions, where the platform provides Intrusion Detection Systems specific functionalities in regard to the pipeline as-a-whole. To add, the platform provides innovative aspects such as direct packet capture to dataset-format conversion to be used within the instantiated pipeline. To elaborate, the platform supports online intrusion detection of network packets, and allows the utilization of specified pipelines to automatically detect the presence of malicious attacks within the captured network traffic.

Moreover, the design focuses on providing an extensible and scalable solution through the use of needed design patterns that address specific logical necessities such as:

- Façade (abstraction of system complexity into the façade node)
- Abstract Factory (instantiation of preprocessors & classifiers within runtime)
- MVC (scalable, flexible, & loosely-coupled web application operative components)
- Singleton (avoiding variable overflows & instance redundancy)
- Strategy Design Pattern (encapsulating the framework's algorithms/classifiers within classes)

SECTION 5: SYSTEM IMPLEMENTATION

5.1. Hardware and software platforms:

Ubuntu 20.04, Windows 10, Visual Studio Code, Streamlit, Apache

5.2. Hardware and software development tools, languages, etc:

Atom, Python 3.7, Bash, HTML, CSS, Javascript, PHP, SQL

5.3. Modules/components acquired from external sources:

The solution utilizes different external modules including; Sci-kit Learn, Keras, Tensorflow, PCAP-to-Dataset conversion scripts, TCPDump.

5.4. Innovative aspects of the implementation

The solution's implementation approach is innovative mainly in the utilization of state-of-the-art technology and frameworks to build the solution. The main framework used for the solution's development is Streamlit, which is an open-source python-based framework for creating machine-learning web applications. Streamlit is a relatively very new technology, and as far as we know (after intensive review) no other Machine learning and Deep learning Intrusion Detection Workbench System has been developed especially making use of the industry-disrupting advantages of Streamlit.

SECTION 6: OTHER RELEVANT ISSUES AND CHALLENGES

6.1. Technical

- System that rely heavily on machine and deep learning tend to be computer intensive, which can cause hardware failure or system crashes due to insufficient resources.
- The high demand of networks and its applications, usually causing high number of send and used packets.
- Anomaly-based IDSs often use GPU-based classification to benefit from the processing power, this can be challenging and cause GPU crashes or overload if the existing GPU did not meet the requirements.
- Various zero-day attacks continue to rise which can be harder to detect.

6.2. Other:

- Some privacy concerns that tend to inspect the system data accompany the implementation of developing such systems, making it a challenge to protect the data.
- Potential legal complications regarding the integration and testing of the IDS within an organization or an existing system.
- Needed policy restrains and considerations to deploy new IDS types with more safety.