| A4  Group 02  Abdalrahman Afifi  Hunter Lavender  Travis Mueller  Zachary Wildasin |
| --- |

# Abstract of Project:

The project revolves around improving the capabilities and security of PICARD, a research platform developed by Dr. Tom Devine of West Virginia University. PICARD focus is on testing and developing semi-supervised, distributed machine learning algorithms. The tool establishes a virtual cluster of machines or containers for data processing. The only way to conduct this research is through the command line. In response, a new web-based graphical user interface (GUI) is being designed, allowing secure remote control, real-time monitoring, and on-the-fly adjustment of computational needs.

Security being an utmost concern, especially given PICARD’s role in running algorithms on datasets, which must be confidential per-user. As an open-source tool, maintaining user’s privacy is a top priority. The project addresses these concerns by developing, with security being at the forefront of our design. The PICARD will be deployed to abide by the West Virginia University’s IT policy to help standardize and elevate security across academic environments.

Key functionalities of the new GUI include the ability to adjust the number of nodes dynamically, the allocation of resources, and real-time hardware usage statistics. Real-time updates on the status of the data and algorithm status, combined with astute visualizations of the results, allowing users to monitor progress, identify significant findings, and form efficient conclusions.

In pursuit of an optimal solution, open-source options are being explored. The project will evaluate solutions through benchmarks, leading to intelligent decision-making for diverse datasets. This approach ensures that PICARD exceeds the requirements for secure, advanced, and scalable machine learning.

[**Abstract of Project: 2**](#_4aecbnq11c3)

[**Vision Statement: 4**](#_5k1m2a4nkz54)

[**Status of Project: 4**](#_e3e994wyaon)

[**Documentation of Testing: 6**](#_248u3fc429vh)

[**System Documentation: 6**](#_pbfk4po140l7)

[Architecture Diagram: 6](#_hhlw5bz2ewju)

[User Interface: 8](#_n8dvdjnfrecd)

[Setup Manual: 12](#_cjcssqhup3ws)

[Node Version Manage (NVM): GitHub Repository 12](#_r4nb5fmrtgui)

[Cloning ThePicard Tool: GitHub 13](#_kyhdx5kbcf55)

[User Manual & Maintenance Manual: 13](#_nnabfeee6efw)

[**Reflections & Instructions for Future Developers: 14**](#_avjzpateflcm)

[**Directory of the Project Archive: 14**](#_kh739d2woi80)

# Vision Statement:

Our capstone project focuses on the technical visualization of The Platform for Intelligent Computer Algorithm Research and Development (PICARD), building upon the foundation established by Group 11 from the previous semester. PICARD is dedicated to conducting the research, development, and performance analysis of semi-supervised machine learning algorithms. Primarily, focusing on the classification within large datasets marked by severe imbalance.

The primary objective of our project is to develop an intuitive full-stack solution for PICARD. This entails the implementation of a web-based graphical user interface (GUI), aimed at providing users with control over real-time resource management capabilities. The focus here resides in optimization of the GUI’s usability, ensuring seamless implementation with the backend infrastructure.

In addition to optimizing the user interface, our project addresses the challenge of system cloning. We strive to establish an efficient method for duplicating PICARD onto different servers. This is crucial for PICARD to remain flexible and scalable, allowing researchers to deploy PICARD in whatever computational environments are necessary.

In conclusion, our technical vision for PICARD revolves around the development of a full-stack solution, efficient system cloning, and comprehensive documentation. Addressing these technical aspects, our capstone aims to elevate PICARD’s capabilities, allowing researchers with less technical background to conduct semi-supervised machine learning within an academic landscape.

# 

# Status of Project:

| Requirement# | Description | Completion Status  yes/No/Partial | Is it demonstrable? | Comment |
| --- | --- | --- | --- | --- |
| 1 | Create Fullstack | yes | yes | GUI/Backend implemented. |
| 2 | Authentication | Partial | yes | Authentication via Google (WVU account) |
| 3 | HTTPs | yes | yes | The framework we use for the GUI utilizes HTTPs. |
| 4 | Insert | yes | no | Functionality to insert new algorithms is possible by uploading them to the tool. |
| 5 | Delete | yes | no | Delete is possible for algorithms affiliated with the user. |
| 6 | Create Jobs through UI | yes | no | Yes, echoes command’s you would use to create said jobs. |

# Documentation of Testing:

Testing of our Graphical User Interface was conducted manually. The way manual user interface is conducted by a tester interacting with the GUI to test the functionality and the overall user’s experience. Benefits to a hands-on experience involves diligent examination of buttons, links, and various forms of interactive components to produce expected results. The navigation between pages is crucial, desiring seamless integration between pages in the application. Also, the tester will ensure data fields input by the user are handled correctly, along with errors.

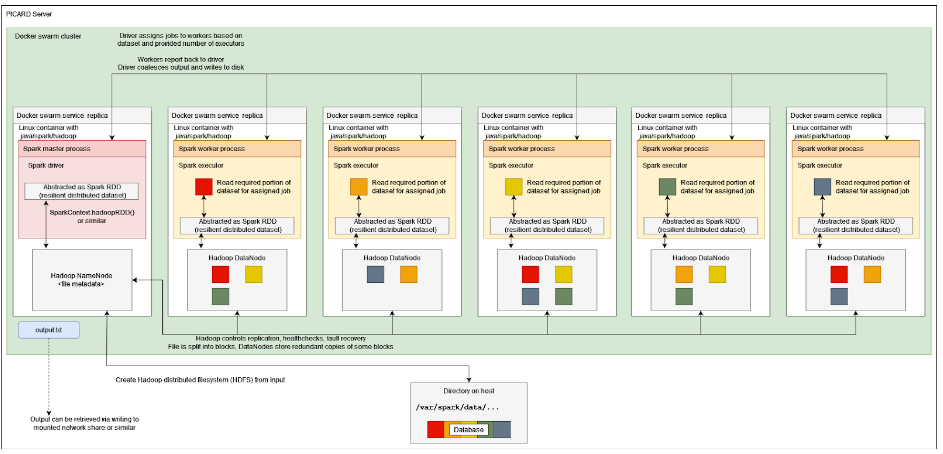
Usability will also be determined by the tester, abiding by the current design principles for UI guidelines. Evaluation following UI design guidelines ensures a clear, logical, and smooth user experience. The tester assesses everything from color schemes to the arrangement of elements to the consistency of design across the entire application. Going through manual testing processes and adhering to UI design principles ensures that each function works effectively and is appealing to the human eye to improve the user's experience using our application.

# System Documentation:

Following section details the inner workings of the system.

## Architecture Diagram:

This is how the algorithms run on docker via server. This is the inner workings of the actual nodes being spun up on the server to run to conduct research.



## User Interface:

## 

## 

## 

## 

## 

## 

## 

## 

## Setup Manual:

### Node Version Manage (NVM): [GitHub Repository](https://github.com/nvm-sh/nvm)

Node Version Manager is a tool installed per-user, per-shell, and works on any POSIX-compliant shell. The most popular platforms are Unix, MacOS, and WindowsWSL.

Installation of the tool is as simple as running the following command in your shell:

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.5/install.sh | bash

After running this command, it is important to restart your shell. Failure to do so may result in NVM remaining unrecognized. Ensure that NVM is configured and working properly so that node versioning matches the packages used to build this tool.

**How to select Node Version along with NPM:**

To use a specific version of Node and its affiliated NPM version, run the following command:

nvm use 20.10.0

This command installs Node version 20.10.0 and the associated NPM version (in this case, 10.2.4).

**If Node and NPM are not already installed, you can use NVM to install them:**

nvm install 20.10.0

Keep in mind that this version exists per user. If you have multiple users on one system, including Sudo and Root, you will need to configure this for each user.

### Cloning ThePicard Tool: [GitHub](https://github.com/hlav22/ThePicard)

After the installation of the Node, NPM, and NVM. Now you can begin installing the tool itself.

**Cloning the Repository:**

git clone <https://github.com/hlav22/ThePicard.git>

**Installing the Dependencies:**

cd into the ./Picard directory.

Run command to install the npm dependencies.

npm install

**Run the application:**

npm start

## User Manual & Maintenance Manual:

**Maintenance:**

To ensure the security and proper functioning of the tool, it is crucial to perform regular updates. This is especially important as individual packages are continually being updated. When using the Dockerfile, spinning up an image should streamline this process. However, it's important to note that certain dependencies might hinder the project's functionality during the build process. The versioning of each package is highly specific, and in case of any issues, it is imperative to inspect these dependencies individually.

**User Manual:**

**The Platform for Intelligent Computer Algorithm Research & Development**

The tool we have designed serves as a front-end interface to facilitate effective usage for users. The installation process for this tool has been detailed above, and it is crucial that you follow these steps to understand how the tool operates on the front-end. The primary goal is to allow users to control parameters and commands effectively, aiding in their construction of parameters along the way.

# Reflections & Instructions for Future Developers:

The PICARD project is an all-purpose research tool with complexities from top to bottom. Initially, understanding the basic functionality of the algorithms and corresponding data may seem overwhelming. However, as one navigates through the swamp of confusion, understanding of the inner workings becomes more comprehensible over time.

For future development, it is highly encouraged to spend an extended period going through documentation from previous semesters. This will ensure your group knows exactly the deliverables of the project and satisfy the requirements of PICARD project. Adaptability for the project is crucial, ensuring whatever bumps in the road you experience along the way, has a contingency plan in place. Having multiple processes in place, if something does not go to plan will save you immense time down the road. In the focal point of the semester, pivoting can be not only frustrating, but also very time consuming. Time is not something you can ever get back, especially during a busy semester. Making sure you have an in-depth understanding of everything will save you a copious amount of time in the long run.

Reflecting on the PICARD project, witnessing a design come to life has been a rewarding experience. Through adversity, to come together as a group and work together in a cohesive unit has allowed us to exceed the initial goals of our project in the first place. Being able to contribute to a multi-group project has been a learning experience for all of us.

# Directory of the Project Archive:

Project archive link via Google Drive, but you must have permission to view this drive.

<https://drive.google.com/drive/folders/1urOFPUDa1dpSDX9ywq1VNaH_FK5mU1eo?usp=drive_link>