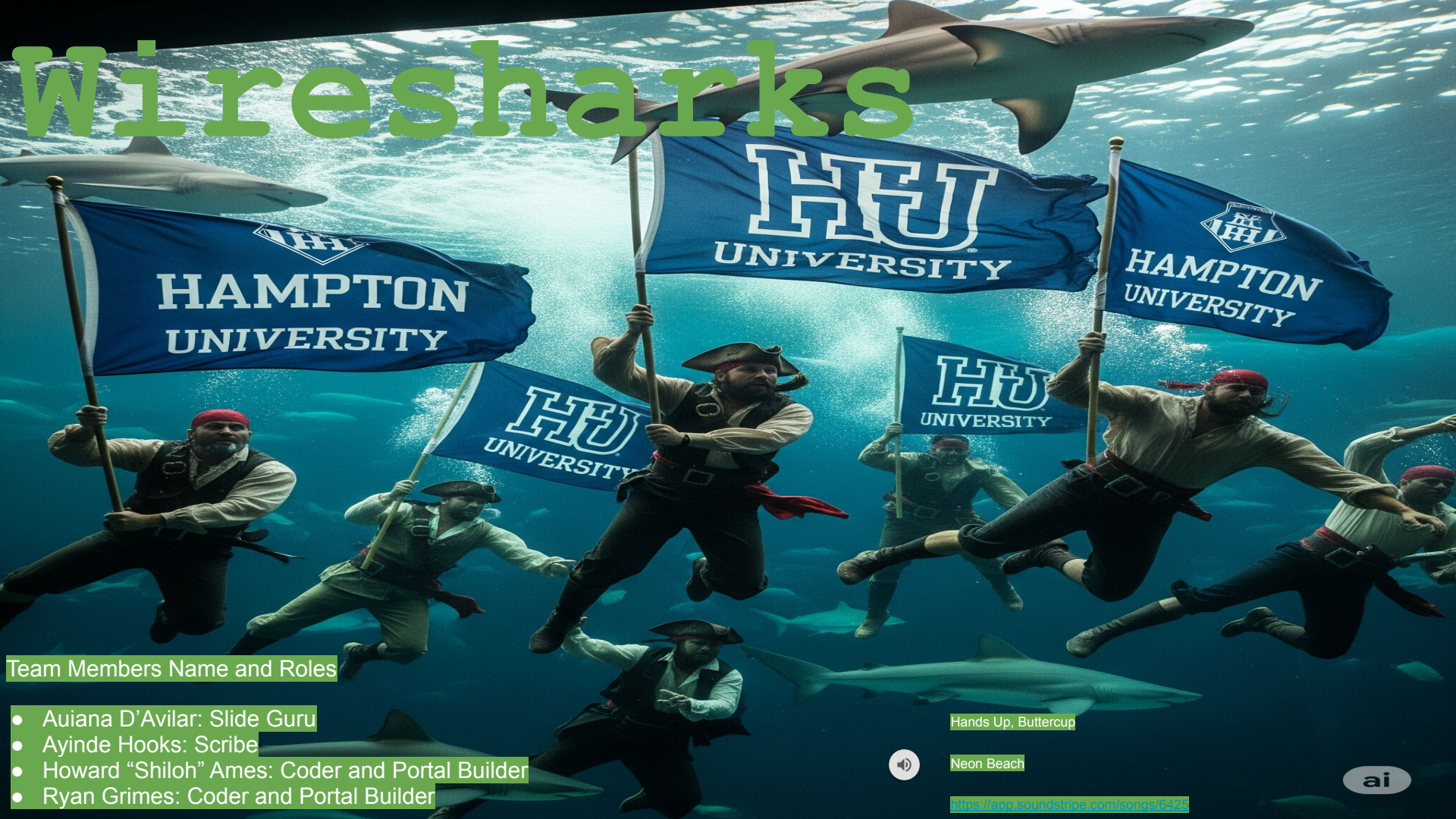


Wiresharks



Team Members Name and Roles

- Auiana D'Avilar: Slide Guru
- Ayinde Hooks: Scribe
- Howard "Shiloh" Ames: Coder and Portal Builder
- Ryan Grimes: Coder and Portal Builder

Hands Up, Buttercup

Neon Beach

<https://app.soundstripe.com/songs/R425>



Meet the Sharks

Ayinde Hooks



Coder and
Portal Builder

Ryan Grimes



Coder and Portal
Builder

Howard "Shiloh" Ames



Scribe

Auiana D'Avilar



Slide Guru

Why Are We Doing This?



Goal:

Evaluate how easy it is to reproduce existing papers from ICSE with Code Repos Artifacts Selection .

Key Questions We Ask:

- Did the repo include all necessary code?
- Was the dataset public and accessible?
- Were instructions up to date?
- Any issues with dependencies or hardware?
- How close were our results to theirs?

Score Card

Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
<p>1: Unavailable/Impossible to find.</p> <p>2: Paywalled/Very hard to access.</p> <p>3: Available, but via obscure link.</p> <p>4: Open-access, direct link.</p> <p>5: Open-access, easily searchable.</p>	<p>1: No code/Private repository.</p> <p>2: Code available, but major parts missing/broken.</p> <p>3: Code available, incomplete/needs big fixes.</p> <p>4: Code available, mostly complete, minor issues.</p> <p>5: Code fully available, complete.</p>	<p>1: dataset missing.</p> <p>2: Mentioned, but completely inaccessible.</p> <p>3: Available, but very hard to find/access.</p> <p>4: Available, but metadata is poor/incomplete</p> <p>5: Fully accessible with complete metadata.</p>	<p>1: No info/Impossible to meet.</p> <p>2: Vague/Requires rare hardware.</p> <p>3: Specific, but hard to meet.</p> <p>4: Specific, but common hardware.</p> <p>5: Clear, common, and flexible hardware.</p>	<p>1: No info/Mandatory custom GPU.</p> <p>2: GPU required, vague/high-end specs.</p> <p>3: GPU optional, specific specs.</p> <p>4: GPU optional, common specs.</p> <p>5: No GPU required (CPU-only).</p>	<p>1: None or misleading.</p> <p>2: Very poor</p> <p>3: Basic or needs much interpretation.</p> <p>4: Clear but few details.</p> <p>5: Comprehensive & clear.</p>	<p>1: Cannot be run due to critical issues or missing parts.</p> <p>2: Can't run without major problems; needs expert help or significant workarounds.</p> <p>3: Can be run with some effort; requires troubleshooting or minor fixes.</p> <p>4: Runs well with minimal effort; minor adjustments might be needed.</p> <p>5: Runs perfectly by simply following the instructions; no issues.</p>	<p>1: Cannot be run due to critical issues or missing parts.</p> <p>2: Can't run without major problems; needs expert help or significant workarounds.</p> <p>3: Can be run with some effort; requires troubleshooting or minor fixes.</p> <p>4: Runs well with minimal effort; minor adjustments might be needed.</p> <p>5: Runs perfectly by simply following the instructions; no issues.</p>	<p>1 (Impossible): Cannot be run due to critical issues or missing parts.</p> <p>2 (Very Difficult): Can't run without major problems; needs expert help or significant workarounds.</p> <p>3 (Doable): Can be run with some effort; requires troubleshooting or minor fixes.</p> <p>4 (Mostly Smooth): Runs well with minimal effort; minor adjustments might be needed.</p> <p>5 (Plug and Play): Runs perfectly by simply following the instructions; no issues.</p>

Project 1: Business Flow Tampering (BFT) Detector

What it does:

Automatically finds ways to bypass website paywalls and ads.

Our Experience:

- Code was available, but instructions were very outdated.
- Major Python version and software conflicts prevented it from running.
- Hardware issues (Docker/virtualization) were a dead end.

Reproducibility Rating: 2/5 (Very Difficult)

Couldn't run the project to compare our results.

BTF Detector Scorecard

	Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
BTF Detector	3) The paper was accessible, but the exact method wasn't clear, suggesting it wasn't a direct, easy find.	3) Code was on GitHub and cloned, but installation failed repeatedly due to very outdated instructions for modern Python.	2) Datasets were mentioned but had no clear links or instructions, so we couldn't find or access them.	3) Specific OS and Python versions were required, and a crucial virtualization issue on the host made setting up a compatible environment very hard.	1) No GPU requirements were stated in the documentation, making it unclear if one was needed or not.	2) Instructions were outdated and caused extensive troubleshooting, forcing us to go far beyond the provided guide.	2) Setting it up was extremely challenging due to constant Python conflicts and an unfixable hardware barrier (BIOS virtualization).	1) We couldn't get the project to run at all, so we couldn't execute any experiments or verify the paper's claimed results.	2 (Very Difficult): Can't run without major problems; needs expert help or significant workarounds.

The background of the slide is a surreal underwater scene. Several pirates, dressed in traditional pirate attire like hats and vests, are swimming and holding up blue and white flags that feature the Hampton University logo (a stylized 'HU'). They are surrounded by numerous sharks of various sizes, creating a sense of being in a dangerous, hostile environment. The lighting is dim and blue, typical of an underwater setting.

Project 2: Fairify

What it does:

Formally checks AI models for fairness (ensures similar people get similar outcomes).

Our Experience:

- Paper, Code, and Data were all available.
- Python version conflicts (needed older TensorFlow) required complex setup.
- **Critical Problem:** Script ran for hours, but NEVER produced any output files.

Reproducibility Rating: 3/5 (Doable)

We could run the project, but without results, we couldn't verify its claims.

Fairify Scorecard

	Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
Fairify	4) The research paper (2212.06140 v2.pdf) was easily found and accessed, and its GitHub repo was clearly referenced.	4) The project's code was publicly available on GitHub, and we could clone it. All necessary parts were there.	5) All AI models and testing data, including German Credit (GC), were provided within the GitHub package and were accessible.	3) It required a specific older Python version (3.9) not standard on our modern Kali system, needing special pyenv setup.	5) No GPU was strictly needed; the tool successfully ran using only the CPU, even with TensorFlow warnings.	2) Instructions were insufficient for common setup issues (like Python version fixes, pyenv setup, and missing output folders), requiring much external troubleshooting.	2) Setting it up was very difficult due to Python version conflicts, pyenv setup issues, and the persistent problem of results not being saved.	1) Even after the script ran for hours, no result files were generated. We could not get the project to produce verifiable output, making true reproduction impossible.	3 (Doable): Can be run with some effort; requires troubleshooting or minor fixes..

The background of the slide is a surreal underwater scene. Several large sharks are swimming in the water. In the center, a group of pirates, dressed in traditional pirate attire like hats and vests, are holding up blue and white flags. The flags feature the letters 'HU' and the words 'HAMPTON UNIVERSITY'. The scene is lit with a blue, watery glow, creating a mysterious and somewhat ominous atmosphere.

Project 3: Bad Snakes

What it does:

Evaluates tools designed to find malicious Python packages on PyPI.

Our Experience:

- Paper and Code were open-access and easily found.
- Python version and dependency conflicts also caused setup issues.
- **Major Problem:** The actual original datasets used in the paper were NOT available due to privacy.

Reproducibility Rating: 2/5 (Very Difficult)

Missing crucial data prevented us from verifying their quantitative results.

Bad Snakes Scorecard

	Paper Availability	Availability of Code and Software	Availability of Datasets	Computer Requirements	GPU Requirements	Documentation Quality	Ease of Setup	Reproducibility of Results	Rating
Bad Snakes	5) The research paper was free, open-access, and easily available for download.	5) All necessary code was free, open-source, provided in the GitHub repository, and was successfully cloned.	1) The crucial original datasets were explicitly not provided and were inaccessible due to licensing/privacy.	2) Required specific Python versions (3.8/3.9) that conflicted with newer systems, and faced unresolvable hardware issues for Docker.	1) The documentation provided no information about GPU requirements, making it unclear if one was needed.	2) Instructions were outdated for modern Python and system environments, leading to extensive troubleshooting beyond the guide.	2) Setting up was very difficult due to Python conflicts, missing dependencies, and an unfixable hardware barrier (Docker virtualization).	1) Setting up was very difficult due to Python conflicts, missing dependencies, and an unfixable hardware barrier (Docker virtualization).	2 (Very Difficult) - "Can't run without major problems; needs expert help or significant workarounds."

A surreal underwater scene featuring several large sharks swimming around a group of pirates. The pirates are dressed in traditional pirate attire, including hats and bandanas, and are holding various flags. Some of the flags are blue with white text that reads "HAMPTON UNIVERSITY" and "HU". The scene is set in a deep blue ocean with sunlight filtering down from the surface, creating a dramatic and somewhat chaotic atmosphere. The pirates appear to be in a state of panic or urgency, as they are surrounded by the large, menacing sharks.

Common Challenges

- **Outdated Instructions:** Frequent source of setup failures.
- **Python Versioning:** Newer Python often breaks older project dependencies.
- **Missing Datasets:** A significant barrier to full reproduction.
- **Environment Specificity:** Unexpected hardware or system settings blocking progress.
- **Lack of Clear Output:** Scripts running but no results appearing.

The background of the slide is a surreal underwater scene. Several large sharks are swimming in the water. In the center, a group of pirates, including Jack Sparrow, are swimming and holding blue flags with the 'HT' logo of Hampton University. The scene is lit with a blue, ethereal glow.

Technology Used

Programming Languages: Python, JavaScript (for web tools)

Core Libraries/Frameworks: TensorFlow/Keras, Numpy, Z3, Poetry, Bandit, scikit-image (attempted)

Automation/Virtualization: Google Puppeteer, Docker (attempted for isolation), pyenv (for Python version management)

Environment: Apple Terminal, Kali Linux (WSL), VS Code, Github, and Docker

The background of the slide is a surreal, dark blue underwater scene. Several large, realistic-looking sharks are swimming in the water. In the center, a group of pirates, dressed in traditional pirate attire like hats and bandanas, are swimming and holding up several flags. The flags are blue and white, featuring the Hampton University logo (a stylized 'HU') and the text 'HAMPTON UNIVERSITY'. The scene is lit with a cool, blue light, creating a mysterious and somewhat ominous atmosphere.

Key Takeaways

Reproducing scientific software can be extremely challenging.

For better reproducibility, projects need:

- Clear, up-to-date documentation.
- Accessible datasets (or clear alternatives).
- Robust environment setup instructions (e.g., using Docker or explicit versioning).