Khandaker Abrar Nadib

Email: abrar.nadib@gmail.com | Github: AbrarNad |LinkedIn: abrar-nadib | Web: abrarnadib.github.io

Research Interests

I study how people interpret and disseminate charts and data in interactive, online environments. My work sits at the intersection of **Information Visualization**, **Human–Computer Interaction (HCI)**, and **human-centered data science**. My goal is to improve comprehension of complex data and decision-making through designing better visualization and storytelling systems.

Publications

ReVISit 2: A Full Experiment Life Cycle User Study Framework
Zach Cutler, Jack Wilburn, Hilson Shrestha, Yiren Ding, Brian Bollen, Khandaker Abrar Nadib, Tingying He, Andrew McNutt, Lane Harrison, and Alexander Lex.
IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE VIS 2025, **Best Paper Award**), 2025. **DOI:** 10.48550/arXiv.2508.03876

Interaction Based Credibility Analysis of News on Facebook Using Machine Learning Methodologies
Sadia Sharmin, Sudipa Saha, Tasin Hoque, and Khandaker Abrar Nadib.
In *Proceedings of the 16th International Conference on Signal Image Technology & Internet based Systems (SITIS)*, 2022. **DOI:** 10.1109/SITIS57111.2022.00077

Research Experience

Guardrail Selection in Line Charts for Persuasive Visualizations 2025
Visualization Design Lab & KORE Lab, University of Utah *Salt Lake City, Utah*

- Lead author on a preregistered mixed-design crowd-sourced study of guardrail sampling strategies in persuasive time-series dashboards (COVID-19 cases and stock performance), evaluating trust, performance, and perceived contextual completeness.
- Built an interactive visualization study using the reVISit framework, implementing various guardrail techniques for contextualizing line-chart comparisons and integrated a logging pipeline that captures provenance graphs, interaction events with Supabase/Firebase-backed storage for replay-based analysis of 500+ crowdsourced participants.
- **Tech:** TypeScript, React, Vite, Mantine UI, D3.js, reVISit, Supabase, Firebase.
- **Status:** Under Review in EuroVis 2026.

Ranking Visualizations of Correlation (ReVISit Replication Study) 2024
ReVISit 2 Replication Project *University of Utah*

- Co-designed and implemented a replication of Harrison et al.'s correlation JND study using reVISit's dynamic sequencing and staircase designs.
- Configured OSF preregistration and study materials, and helped manage crowdsourced data collection across multiple visualization conditions (scatterplots, PCPs, hexbins, heatmaps).
- **Status:** Published, also working in a first-authored short paper.

News Credibility Analysis on Facebook using User Interactions 2021 – 2022
Bangladesh University of Engineering and Technology (BUET) *Dhaka, Bangladesh*

- Proposed and evaluated an interaction-based approach for fake news detection on Facebook, using engagement signals rather than language features.
- Trained machine learning models to classify the authenticity of public posts; showed improved performance over content- and NLP-based baselines and language independence.
- **Tech:** Crowdtangle, scikit-learn, pandas, matplotlib.
- **Status:** Published

Education

University of Utah — Salt Lake City, Utah
Doctor of Philosophy in Computer Science — *August 2024 – Present*

Bangladesh University of Engineering and Technology (BUET) — Dhaka, Bangladesh
Bachelor of Science in Computer Science and Engineering — *Feb 2017 – May,2022*

Work Experience

Graduate Research Assistant — January 2025 – Present
University of Utah, Salt Lake City
Visualization Design Lab, KORE Lab

Graduate Research Fellow — August 2024 – December 2025
University of Utah, Salt Lake City
Visualization Design Lab, KORE Lab

Software Engineer — May 2022 – July 2024
Optimizely, Dhaka

Digital Asset Management (DAM) — *November 2022 – July 2024*

- Currently working in the Digital Asset Management (similar to Google Drive) team.
- Implemented Brand Template feature, which lets users create a Template for their brand and define Placeholders that other collaborators can edit. I also implemented Download, Export, Cloning, and Task integration features for Brand Templates.
- DAM Collections are a group of user-defined Assets, including Asset folders. I implemented Searching, Filtering, and Navigation within DAM Collection folders.
- Implemented various asset-specific features like meta information, asset relations, and bulk operations, which enhanced user ability to handle assets.
- Implemented breadcrumbs in the DAM Library to make the navigation more fluid for the users.
- Implemented various user activity tracking for analytics to gain useful insights.
- Made improvements to several backend and UI components in terms of accessibility, performance, and code quality.
- Upgraded and integrated GPT-3.5-turbo model for AI content generation.
- Handled user roles and privileges for various features.
- **Technologies:** Python, Flask, JavaScript, TypeScript, React.js, MySQL, MongoDB, Alembic, Celery, Elasticsearch

Asset Renditions (AR) — *May 2022 – October 2022*

- Worked on implementing and maintaining a feature Asset Rendition. This feature allows users to pre-define "Rendition types", using which whenever users upload a new asset, new "Renditions" of that asset are automatically generated in the background. Example use-case: a user may define two image rendition types- 1. Facebook- 1080*720 crop and Instagram- 720*720 crop. Then whenever the user uploads an image asset, two cropped images will automatically be generated with the given specifications.
- Implemented logging schemes by combining multiple services to enable users and developers to diagnose and debug errors.
- Built three services to generate asset renditions using the given specifications including image and video generators.
- Implemented stateless generators to scale horizontally and integrated asynchronous messaging for decoupling and scaling, for efficiency.
- Integrated the Rendition Service with the local development environment for developers.
- **Technologies:** Python, FastAPI, MySQL, PostgreSQL, Docker, Kubernetes, Message Queue

Projects

Interactive Guardrail-Integrated Line Chart Platform 2024–2025

- Implemented interactive dashboards used as stimuli in the guardrail selection study, including guardrail sampling logic, randomization.
- **Technologies:** TypeScript, React 18, Vite, Mantine UI, Redux Toolkit, D3.js, Vega/Vega-Lite, Arquero, reVISit, Supabase, Firebase.

Correlation JND Replication Experiments (ReVISit Framework) 2024

- Built a staircase-style experiment to measure just-noticeable differences in correlation across multiple visualization types using reVISit's dynamic sequencing.
- **Technologies:** TypeScript, React,D3.js, reVISit, Python (pandas, NumPy, SciPy, statsmodels), firebase.

HappyVis: Visualizing Happiness Around the Globe 2024

- Designed a web-based visualization dashboard for the World Happiness Report dataset, enabling exploration of global happiness scores and contributing factors across countries and years.
- Implemented linked views including an interactive choropleth map, trend line charts, and comparative views to analyze relationships between happiness, GDP per capita, life expectancy, and social support.
- **Technologies:** JavaScript, D3.js.

Awards and Honors

Best Paper Award, IEEE VIS 2025 2025
for "ReVISit 2: A Full Experiment Life Cycle User Study Framework"

Optimizely SPOT Awards (July & October) 2023
Two peer-nominated SPOT awards recognizing problem solving, team contribution, and performance

Board Merit Scholarships (SSC & HSC) 2014, 2016
Education Board Scholarships; ranked 6^{th} (male) in Dhaka Board in HSC

Technical Skills

Research Methods: Online controlled experiments, interviewing, qualitative coding (open-ended response analysis, thematic coding)
Analysis: Mixed-effects models, regression, hypothesis testing; Python (pandas, NumPy, SciPy, statsmodels, matplotlib, scikit-learn, PyTorch
Visualization & Frontend: D3.js, Vega, Vega-Lite, React.js, TypeScript, Mantine UI, Redux Toolkit, Vite
Experiment Platforms & Storage: reVISit, Supabase, Firebase, localforage
Languages: Python, Java, C/C++, SQL, PL/SQL
Databases: MySQL, PostgreSQL, MongoDB, Oracle
Frameworks: Flask, FastAPI, Node.js, Bootstrap
Tools/Software: Git, Docker, Jupyter Notebook, VS Code, PyCharm, IntelliJ, TensorFlow, Playwright, Vitest
Libraries: pandas, NumPy, Arquero (JS), Keras, Matplotlib, SciPy, scikit-learn, PyTorch, OpenCV, OpenGL
Scripting/Markup/Serialization: Bash, TCL, LaTeX, YAML, HTML, JSON