

# Dibyendu Das

Stony Brook University  
New York, USA

🏠 [dibyendu.github.io](https://dibyendu.github.io)  
✉ [dibyendu.das@stonybrook.edu](mailto:dibyendu.das@stonybrook.edu)

## Education

2023 <sub>Fall</sub>	Stony Brook University, NY, USA	Ph.D. in Computer Science
2016 <sub>Aug.</sub>	Indian Institute of Technology (IIT), Kharagpur	M.Tech. in Computer Sc. & Engg.
2012 <sub>Aug.</sub>	Jadavpur University, Kolkata, India	B.E. in Computer Science & Engg.

## Area of Interest

Artificial Intelligence • Stochastic Decision Making • Reinforcement Learning  
[Deep Reinforcement Learning](#) • Meta Reinforcement Learning • Logic

## Employment

2018 <sub>Fall</sub> - present	<b>Graduate Research Assistant</b> at <b>Stony Brook University</b> , NY, USA ■ Working on building a common-sense knowledge-base aided with reinforcement learning based techniques to enable a robotic arm to perform manipulation tasks like grasping, opening doors etc.
2021 <sub>May</sub> - 2021 <sub>Sep.</sub>	<b>Applied Scientist Intern</b> at <b>Amazon AWS</b> , Cupertino, CA, USA ■ Worked on an internal security analysis tool that detects misconfigurations in AWS resources such as S3, EC2, etc. ■ Improved the performance of the analysis tool by adding <i>Scoped Modelling</i> and <i>Differential Analysis</i> ■ Achieved significant time efficiency of the order of ~50% in test runs
2019 <sub>June</sub> - 2019 <sub>Aug.</sub>	<b>Research Intern</b> at <b>CEWIT</b> , Stony Brook University, NY, USA ■ Worked on noise elimination and track prediction of colliding sub-atomic particles inside particle accelerators ■ Achieved significantly high accuracy (~98%) using an Artificial Intelligence technique called <a href="#">Graph Neural Network</a> .
2017 <sub>Aug.</sub> - 2017 <sub>Dec.</sub>	<b>Visiting Research Scientist</b> at <b>INRIA Labs</b> , Nancy, France ■ Worked in the <a href="#">PESTO</a> team ■ Achieved significant success in theoretically verifying the integrity and robustness of security protocols deployed at multi-party cloud infrastructures.
2016 <sub>Aug.</sub> - 2018 <sub>May</sub>	<b>Research Assistant</b> at <b>Indian Institute of Technology (IIT) Kharagpur</b> , India ■ Collaborated with 3 researchers from the Formal Verification Lab ■ Made significant contributions to ongoing research projects which resulted into 2 major publications
2012 <sub>Sept.</sub> - 2014 <sub>July</sub>	<b>Automation Engineer</b> at <b>Idea Device Technologies Pvt. Ltd.</b> , Bangalore, India ■ Developed and maintained a cloud based infrastructure automation product named <a href="#">calm.io</a> (later acquired by <a href="#">Nutanix</a> ) ■ Single-handedly implemented several features (such as, tag-cloud generation, dynamic license generation and verification) to the product.

## Technical Skills

| Legends ⇒ :expert :proficient .familiar

**Prog. Languages** ⇒ :Python :C/C++ :JavaScript[ES6] : $\text{\LaTeX}$  :Bash :OCaml :Prolog :Go

**Developer Tools** ⇒ :Docker :Git :GNU Autotools[Make, Autoconf] :GDB .SVN

**Machine/Deep Learning Frameworks** ⇒ :PyTorch :TensorFlow :ScikitLearn :[Rapids](#) :[GraphNet](#)

**Web Frameworks/Language/Template Engines** ⇒ :Express/NodeJs/jade :Flask/Python/jinja2 :Iris/Go/handlebars

**Frontend Frameworks** ⇒ :ReactJs :ReactNative :jQuery :HTML5 :Bootstrap

**Databases** ⇒ **SQL** (PostgreSQL MySQL SQLite) **NoSQL** (MongoDB Firestore)

**Model Checking Frameworks** ⇒ :Z3 SMT Solver :CoQ Proof Assistant .FramaC .Tamarin

## Projects

### Research Projects

#### TrackML

##### Noise Elimination and [Track Prediction](#) using Machine Learning

■ Worked on a *computational-physics* project, with an objective to eliminate noise and predict tracks of colliding sub-atomic particles inside particle accelerators ■ Achieved exceptionally high accuracy (~98%) in the first phase i.e. noise elimination from the collision data ■ Achieved satisfactory accuracy (~85%) in the second phase i.e. prediction of tracks ■ Used state-of-the-art Artificial Intelligence technique called [Graph Neural Network](#).

#### UnreLyzer

##### A [static Interval analyzer](#) for C like **unreliable** programs

■ Designed a novel mathematical framework to incorporate reliability of outcome within a certain confidence bound, while executing programs in highly unreliable environments ■ Developed a [tool](#) as the proof of concept, that models environmental unreliabilities in form of operational semantics of programs and predicts sharp confidence bounds on the outcomes ■ This type of analysis is extremely useful in real time control systems working under extreme conditions.

### Web Applications

#### UT3

##### An online multi-player and single-player implementation of [Ultimate Tic-Tac-Toe](#)

Stack → [Express](#) + [React](#) + [NoSQL](#)    ML Framework → [Tensorflowjs](#)    [[Source Code](#)]

■ A single-page real-time application built using web-sockets, to play the [ultimate form of tic-tac-toe](#) where  $81 \times 81$  cells are divided into  $9 \times 9$  grids. When playing in multi-player mode, 2 players can take turns to win individual grids and thereby win the entire board ■ This game can also be played in single-player mode against an agent trained using the [AlphaZero](#) algorithm.

#### AccuRacist

##### A [geometry-based game](#) for testing the precision of unaided eyes

Language → [PHP](#) & Database → [MongoDB](#)    [[Source Code](#)]

■ This app starts off by showing the user some geometrical figures which are significantly distorted ■ It asks the user to make those figures as accurate as possible and then it calculates a score based on the accuracy & time taken to solve each puzzle ■ It stores the average score for each user in a remote database and also shows some interesting distribution over global scores.

#### Dev-Assist

##### A [GitHub bot](#) that does a list of useful things – starting from code linting to automating issue management, scheduled notifications and many more

Language → [NodeJs](#)    Framework → [Probot](#)    Database → [MongoDB](#)    [[Source Code](#)]

■ A GitHub App built with Probot framework, that does a list of useful things such as: i) **Code Linting** (Css, Javascript & Go), ii) **Code Splitting**, iii) **Automatic Reminder on Issues and Pull Requests**, iv) **No Merging while Work in progress** v) **Locking of Closed Issues & PRs** & vi) **Archiving of Inactive Issues & PRs** ■ For more details (along with screen-shots) about all its features, please see the [README](#) file.

### Mobile Application

#### Packet 2FA

##### [Two Factor Authentication App](#) for Network Packet Monitoring

Framework → [ReactNative](#)    Language → [C](#) + [Javascript](#)    Database → [SQLite3](#)

■ This prototype app acts like a **Two Factor Authentication** tool ■ It integrates with a real time network packet monitoring system installed at the user's location and alerts the user by sending notifications to the phone for any **potentially suspicious activity** over the network ■ It also verifies the user's authenticity before setting up any communication.